A Broadcom Company

technologies

Reference Guide
for Db2 12 for z/OS

Instant Access to Essential Information

Broadcom Inc.
1320 Ridder Park Drive
San Jose, California 95131
United States
+1 408.433.8000

ca.com/db2

Reference Guide for Db2 12 for z/OS

The *CA Reference Guide for Db2 12* for z/OS is an easy-to-use booklet that gives you instant access to information, such as SQL- and Db2-related system commands, utility syntax, catalog tables, SQL return codes, system limits, and much more.

See *IBM DB2 Command Reference*, *IBM DB2 SQL Reference*, *IBM DB2 Reference Summary,* other IBM references, and the various CA User Guides for complete descriptions of commands and syntax.

Corrections and Comments

If you have any corrections you would like to report or if you have any comments about our *Reference Guide*, please send us an email at db2refguide@ca.com.

Notation Conventions

Convention	Description
UPPER CASE characters	Must be entered as shown.
Lower case characters	User-specified variable.
( ) parentheses	Must be entered where shown
|	OR condition
{ }	Enclose optional parameters. Select only 1.
{{ }}	Enclose optional parameters. Select one or more.
[ ]	Enclose required parameters. Select only 1.
[[ ]]	Enclose required parameters. Select one or more.
■	Syntax example
Parameter not enclosed in any of the above	Required parameter
Indentation	For readability only.
, after parameter	Parameter(s) can be repeated. A comma is required between repeated parameters.
, before parameter	The parameter requires a comma.
: after parameter	Repeatable parameter(s). The colon is required between the repeated parameters.
; semicolon	Must be entered where shown.
underline	Default parameter or abbreviation for a command.
italics	Subordinate block.

NOTICES

The term “Broadcom” refers to Broadcom Inc. and/or its subsidiaries. For more information, please visit www.broadcom.com.

Broadcom reserves the right to make changes without further notice to any products or data herein to improve reliability, function, or design. Information furnished by Broadcom is believed to be accurate and reliable. However, Broadcom does not assume any liability arising out of the application or use of this information, nor the application or use of any product or circuit described herein, neither does it convey any license under its patent rights nor the rights of others.

The information in this publication could include typographical errors or technical inaccuracies, and the authors assume no responsibility for its accuracy or completeness. The statements and opinions expressed in this book are those of the authors and are not necessarily those of CA.

Any reference in this publication to third-party products and websites is provided for convenience only and shall not serve as the authors’ endorsement of such products or websites. Your use of such products, websites, any information regarding such products or any materials provided with such products or on such websites shall be at your own risk.

To the extent permitted by applicable law, the content of this book is provided “AS IS” without warranty of any kind, including, without limitation, any implied warranties of merchantability, fitness for a particular purpose, or non-infringement. In no event will the authors or CA be liable for any loss or damage, direct or indirect, arising from or related to the use of this book, including, without limitation, lost profits, lost investment, business interruption, goodwill or lost data, even if expressly advised in advance of the possibility of such damages. Neither the content of this book nor any software product referenced herein serves as a substitute for your compliance with any laws (including but not limited to any act, statute, regulation, rule, directive, standard, policy, administrative order, executive order, and so on (collectively, “Laws”) referenced herein or otherwise. You should consult with competent legal counsel regarding any such Laws.

How can you establish a solid foundation for successful Db2 for z/OS management?

Enterprises are storing and managing ever-increasing amounts of mission essential data residing on IBM® Db2™ for z/OS™ that must be accessible around the clock. Downtime or delayed processing due to performance bottlenecks can cause more than a loss of productivity—it can mean losing customers. In today's business environment, organizations must ensure optimal performance for their databases and applications.

Enterprises need database management tools for Db2 for z/OS that help resolve issues quickly, before they become problems that drag down system performance and decrease productivity.

Database Management for DB2 for z/OS

Broadcom Database Management for Db2 for z/OS solutions provide a powerful and flexible set of tools designed to ensure optimal database and SQL performance, efficient administration, and reliable backup and recovery of Db2 databases, systems, and apps. With these interoperable solution suites, your enterprise can also improve service levels, data availability, and application responsiveness— helping to reduce your costs.

Database Management for Db2 for z/OS functions are bundled into the following suites:

- *Administration Suite for Db2 for z/OS*
- *Performance Suite for Db2 for z/OS*
- *Recovery Suite for Db2 for z/OS*
- *Utility Suite for Db2 for z/OS*

Dependability you can trust – Db2 for z/OS management made simplier

Our database management solutions are flexible and give you choice without giving up control. You can use full automation where appropriate, powerful manual control when desired, or expert software-guided assistance. You can help reduce the cost and risk associated with Db2 version upgrades, streamline workflows, use available skill sets better, make more precise updates, and improve responsiveness across complex database and application environments.

Administration Suite for Db2 for z/OS

***CA RC/Query®*:** Query, analyze, maintain, and manage your Db2 subsystem.

***CA RC/Update™*:** Automate labor-intensive tasks related to Db2 objects and data.

***CA RC/Migrator™ for Db2 for z/OS*:** Automate administrative tasks to increase database availability and decrease human error.

***CA RC/Compare™ for Db2 for z/OS*:** Compare and synchronize Db2 schemas.

***CA RC/Extract™ for Db2 for z/OS*:** Extract, archive, and distribute referentially related Db2 data.

***CA RC/Secure™ for Db2 for z/OS*:** Manage privileges, monitor authority, and perform impact analysis for Db2.

Performance Suite for Db2 for z/OS

CA Bind Analyzer™ for Db2 for z/OS: Reduce resource contention and accelerate application development.

***CA Detector® for Db2 for z/OS*:** Optimize performance by detecting, diagnosing, and evaluating Db2 performance problems.

***CA Subsystem Analyzer for Db2 for z/OS*:** Manage Db2 subsystem performance at the object level.

***CA Plan Analyzer® for Db2 for z/OS*:** Provide efficient and easy-to-understand Db2 plan analysis.

***CA SYSVIEW® Performance Management Option for Db2 for z/OS*:** Monitor Db2 subsystems and Db2 applications outside of the z/OS environment.

Recovery Suite for Db2 for z/OS

***CA Log Analyzer™ for Db2 for z/OS*:** Audit Db2 data changes, migrate change, and back out errant updates.

***CA Recovery Analyzer™ for Db2 for z/OS*:** Streamline and automate complex Db2 recovery tasks.

***CA Fast Recover™ for Db2 for z/OS*:** High-performance recovery for Db2 table spaces and indexes.

***CA Merge/Modify™ for Db2 for z/OS*:** Create full Db2 image copies by merging log data and image copies.

***CA Quick Copy for Db2 for z/OS*:** Create fast, consistent, and accurate Db2 image copies.

Utility Suite for Db2 for z/OS

CA Database Analyzer™ for Db2 for z/OS: Automate Db2 object maintenance, gather information, and take corrective action.

***CA Fast Check for Db2 for z/OS*:** Monitor Db2 database consistency with in-depth integrity checking.

***CA Fast Index® for Db2 for z/OS*:** Build Db2 indexes while minimizing I/O.

***CA Fast Load® for Db2 for z/OS*:** Enable high-speed loading of large amounts of Db2 data.

***CA Fast Unload® for Db2 for z/OS*:** Enable high-speed unloading of large amounts of Db2 data.

***CA Rapid Reorg® for Db2 for z/OS*:** Reorganize and automate Db2 space management, online and offline.

Also includes: CA Log Analyzer™ for Db2 for z/OS, CA Recovery Analyzer™ for Db2 for z/OS, CA Fast Recover™ for Db2 for z/OS, CA Quick Copy for Db2 for z/OS, and CA Merge/Modify™ for Db2 for z/OS

Additional Key Management Functionality

CA Report Facility: easy to use intuitive interface to access and reporting business data.

To learn more about our market-leading technology solutions, and how they can help you achieve real savings in your Db2 for z/OS environment, please visit us today: broadcom.com/db2

CA Database Management for DB2 for z/OS **1**

1. SQL 1-1

Syntax Elements 1-1
Queries 1-5
Statements 1-8
Functions 1-52
Durations 1-76
Reserved Words 1-77
Data Types 1-79
Predicate Processing Summary 1-80

2. UTILITIES 2-1

On-line DB2 2-1
Stand-Alone Utilities 2-17
Authorities Required 2-20

3. COMMANDS 3-1

DB2 Commands 3-1
DSN Commands 3-8
CICS Attachment Facility 3-14
TSO Commands 3-14
z/OS IRLM Commands 3-15
IMS Commands 3-16
ADMTPROC Commands 3-16

4. SYSTEM TABLES 4-1

Catalog Tables 4-1
Directory Tables 4-227
Other System Tables 4-232

5. USER TABLES 5-1

PLAN_TABLE 5-1
DSN_COLDIST_TABLE 5-8
DSN_DETCOST_TABLE 5-9
DSN_FILTER_TABLE 5-13
DSN_FUNCTION_TABLE 5-14
DSN_KEYTGTDIST_TABLE 5-15
DSN_PGRANGE_TABLE 5-16
DSN_PGROUP_TABLE 5-17
DSN_PREDICAT_TABLE 5-19
DSN_PREDICAT_SELECTIVITY_TABLE 5-21
DSN_PTASK_TABLE 5-22
DSN_QUERY_TABLE 5-23
DSN_QUERYINFO_TABLE 5-24
DSN_SORT_TABLE 5-25
DSN_SORTKEY_TABLE 5-26
DSN_STATEMENT_CACHE_TABLE 5-27
DSN_STATEMNT_TABLE 5-31
DSN_STRUCT_TABLE 5-33
DSN_STAT_FEEDBACK_TABLE 5-34
DSN_VIEWREF_TABLE 5-35

6. MISCELLANEOUS SYSTEM INFORMATION 6-1

SQLCA 6-1
SQLDA 6-4
SQL Limits 6-8
DB2 Limits 6-10
RLST Table 6-13
Security Authorizations 6-16
DSNZPARM Parameter Information 6-22
IFCIDS 6-34

7. RETURN CODES 7-1

SQL Codes 7-1
SQL State / SQL Code Cross-Reference 7-36
SQLSTATE Class Codes 7-41
CAF Return Codes 7-42
Resource Type Codes 7-43

Index

1. SQL

Syntax Elements

The SQL syntax elements include: expressions, predicates, and search conditions.

Expressions

expression:

{+ | –} [*function-invocation* | (*expression*) | *constant* | *column-name* | *variable* | *special-register* | *scalar-fullselect* | *time-zone-specific-expression* | *labeled-duration* | *case-expression* | *cast-specification* | *XMLCAST-specification* | *array-element-specification* | *array-constructor* | *OLAP specification* | *row-change-expression* | *sequence-reference*]

Use an operator to separate multiple occurrences of these expression elements.

operator:

[CONCAT | || | / | * | + | –]

Note: CONCAT is the same as ||.

special-register:

[CURRENT APPLICATION COMPATIBILITY |
CURRENT APPLICATION ENCODING SCHEME |
CURRENT CLIENT_ACCTNG |
CURRENT CLIENT_APPLNAME |
CURRENT CLIENT_CORR_TOKEN
CURRENT CLIENT_USERID |
CURRENT CLIENT_WRKSTNNAME |]
[CURRENT DATE | CURRENT_DATE] |
CURRENT DEBUG MODE |
CURRENT DECFLOAT ROUNDING MODE |
CURRENT DEGREE |
CURRENT EXPLAIN MODE |
CURRENT GET_ACCEL_ARCHIVE |
[CURRENT {LOCALE} LC_CTYPE | CURRENT_LC_CTYPE] |
CURRENT MAINTAINED {TABLE} TYPES {FOR OPTIMIZATION}
| CURRENT MEMBER |
CURRENT OPTIMIZATION HINT |
CURRENT PACKAGE PATH |
CURRENT PACKAGESET |
[CURRENT PATH | CURRENT_PATH] |
CURRENT PRECISION | CURRENT QUERY ACCELERATION |
CURRENT REFRESH AGE |
CURRENT ROUTINE VERSION |
CURRENT RULES |
[CURRENT SCHEMA | CURRENT_SCHEMA] |
CURRENT SERVER |
CURRENT SQLID |
[CURRENT TIME | CURRENT_TIME] |
[CURRENT TIMESTAMP |
CURRENT_TIMESTAMP] { (6) |
(*integer*) } {WITHOUT TIME ZONE | WITH TIME ZONE} |
CURRENT TIME ZONE |
SESSION TIME ZONE |
ENCRYPTION PASSWORD |
[SESSION_USER | USER] |

CURRENT TEMPORAL SYSTEM_TIME |
CURRENT TEMPORAL BUSINESS_TIME]

The forms with the underscore follow the SQL standard.

time-zone-specific-expression:

[*function-invocation* | (*expression*) | *constant* |
column-name | *variable* | *special-register* |
scalar-fullselect | *case-expression* | *cast-specification*]
AT [LOCAL | TIME ZONE [' {+|-} *th:tm* ' | *function-invocation*
|
(*expression*) | *constant* | *column-name* | *variable* |
special-register | *scalar-fullselect* | *case-expression* |
cast-specification]]

labeled-duration:

[*function-invocation* | (*expression*) | *constant* | *column-name*
| *variable*] [YEAR{S} | MONTH{S} | DAY{S} | HOUR{S} |
MINUTE{S} | SECOND{S} | MICROSECOND{S}]

case-expression:

CASE [*searched-when-clause* | *simple-when-clause*]
{ ELSE NULL|ELSE *result-expression*} END

searched-when-clause:
[[WHEN *search-condition* THEN [*result-expression*|NULL]]]

simple-when-clause:
expression [[WHEN *expression* THEN [*result-expression*|
NULL]]]

cast-specification:

CAST ([*expression*|NULL | *parameter-marker*] AS *data-type*)

data-type:
[*built-in-type* | *distinct-type-name* | *array-type*]
see CREATE TABLE

xmlcast-specification:

XMLCAST ([*expression* | NULL | *parameter-marker*]
AS *data-type*)

array-element specification:

array-expression [*array-index*]

array-constructor:

ARRAY [{*fullselect* | [[*element-expression* | NULL,]]}]

OLAP-specification:

[*ordered-OLAP-specification* | *numbering-specification* |
aggregation-specification]

ordered-OLAP-specification:

[RANK | DENSE_RANK] ( )
OVER ({*window-partition-clause*} *window-order-clause*)

window-partition-clause:
PARTITION BY [[*partitioning-expression,*]]

window-order-clause:
ORDER BY [[*sort-key-expression*
{ASC {NULLS LAST} | ASC NULLS FIRST |
DESC {NULLS FIRST} | DESC NULLS LAST},]]

numbering-specification:

ROW_NUMBER ( )
OVER ({*window-partition-clause*} *{window-order-clause}*)

window-partition-clause:
PARTITION BY [[*partitioning-expression,*]]

window-order-clause:
ORDER BY [[*sort-key-expression*
{ASC {NULLS LAST} | ASC NULLS FIRST |
DESC {NULLS FIRST} | DESC NULLS LAST},]]

aggregation-specification:
aggregate-function OVER ({*window-partition-clause*}
{RANGE BETWEEN UNBOUNDED PRECEDING AND
UNBOUNDED FOLLOWING | *window-order-clause*
{RANGE BETWEEN UNBOUNDED PRECEDING AND CURRENT
ROW | *window-aggregation-group-clause*}})

aggregate-function:
[AVG|CORRELATION|COUNT|COUNT_BIG| COVARIANCE|
MAX | MIN | STDDEV | SUM | VARIANCE]

window-partition-clause:
PARTITION BY [[*partitioning-expression,*]]

window-order-clause:
ORDER BY [[*sort-key-expression*
{ASC {NULLS LAST} | ASC NULLS FIRST |
DESC {NULLS FIRST} | DESC NULLS LAST},]]

window-aggregation-group-clause:
[ROWS|RANGE] [*group-start*|*group-between*|*group-end*]

group-start:
[UNBOUNDED PRECEDING |
unsigned-constant PRECEDING | CURRENT ROW]

group-between:
BETWEEN *group-bound-1* AND *group-bound-2*

group-bound-1:
[UNBOUNDED PRECEDING |
unsigned-constant [PRECEDING | FOLLOWING]
|
CURRENT ROW]

group-bound-2:
[UNBOUNDED FOLLOWING |
unsigned-constant [PRECEDING | FOLLOWING]
|
CURRENT ROW]

group-end:
[UNBOUNDED FOLLOWING |
unsigned-constant FOLLOWING]

row-change-expression:
ROW CHANGE [TIMESTAMP|TOKEN] FOR *table-designator*

sequence-reference:
[NEXT|PREVIOUS] VALUE FOR *sequence name*

Predicates

row-value-expression:
([[expression,]])

basic-predicate:
[*expression* [= | <> | > | < | >= | <=] *expression* |
row-value-expression [= | <> | > | < | >= | <=] *row-value-expression*]

quantified-predicate:
[*expression* [= | <> | > | < | >= | <=] [SOME | ANY | ALL]
(*fullselect1*) | *row-value-expression* = [SOME | ANY]
(fullselect2) | *row-value-expression* <> ALL (fullselect2)]

ARRAY_EXISTS predicate:

ARRAY_EXISTS (*array-expression, array-index)*

BETWEEN-predicate:

expression {NOT} BETWEEN *expression* AND *expression*

DISTINCT-predicate:

[*expression* IS {NOT} DISTINCT FROM *expression* |
row-value-expression IS {NOT}
DISTINCT FROM *row-value-expression*]

EXISTS-predicate:

{NOT} EXISTS (*fullselect*)

IN-predicate:

[*expression1* {NOT} IN [(*fullselect1*) | ([[*expression2*,]])] |
row-value-expression {NOT} IN (*fullselect2*)]

LIKE-predicate:

match-expression {NOT}
LIKE *pattern-expression* {ESCAPE *escape-expression*}

pattern-expression:

[*constant* | *special-register* | *host-variable* |
scalar-function {(*constant* | *special-register* |
host-variable)} |
array-element | *CAST* [*constant* | *special-register* |
host-variable | *scalar-function* {(*constant* |
special-register | *host-variable*)} | *array-element*] |
pattern-expression1 CONCAT pattern-expression2]

escape-expression:

a single character in the form of
[*constant*| *host-variable* |
scalar-function {(*constant* | *host-variable*)} |
CAST [*constant* | *host-variable* |
scalar-function {(*constant* | *host-variable*)}]

NULL-*predicate*:

expression IS {NOT} NULL

XMLEXISTS-*predicate*:

XMLEXISTS (*xquery-expression-constant*
{ PASSING {BY REF} [[*xquery-argument,*]]})

xquery-argument:

[*xquery-context-item-expression* |
xquery-context-item-expression AS identifier]

Search Conditions

search-condition:

{NOT} [*predicate* {SELECTIVITY *numeric-constant*} |
(*search-condition*)] {{ [AND|OR] {NOT} [*predicate* |
(*search-condition*)] }}

Queries

The SQL queries include: subselect, fullselect, select-statement, and embedded select-INTO-statement

Subselect

subselect:

select-clause from-clause {*where-clause*} {*group-by-clause*} {*having-clause*} {*order-by-clause*} {*offset-clause*} {*fetch-clause*}

select-clause:

SELECT {ALL|DISTINCT}
[* | [[*expression* { {AS} *new-column-name*} |
unpacked-row | *table-name.** | *view-name.** |
*correlation-name.**,]]

unpacked-row:
*UNPACK function-invocation.** AS ([[*field-name data-type*,]])

from-clause:

FROM [[*table-reference*,]]

table-reference:

[*single-table-reference*] |[*single-view-reference*] |
[*nested-table-expression*] |[*data-change-table-reference*] |
[*table-function-reference*] |[*table-locator-reference*] |
[*xmltable-expression*] |[*collection-derived-table*] |
[*joined-table*]

single-table-reference:

table-name {{ *period-specification* }} {*correlation-clause*}

period-specification:

FOR [SYSTEM_TIME |BUSINESS_TIME]
[AS OF *value* | FROM *value1* TO *value2* |
BETWEEN *value1* AND *value2*]
Note: AS OF TIMESTAMP can be specified instead of FOR SYSTEM_TIME AS OF. Do not specify SYSTEM_TIME and BUSINESS_TIME more than once per table.

correlation-clause:

{AS} *correlation-name* {([[*new-column-name*,]])}

single-view-reference:

view-name {{ *period-specification* }} {*correlation-clause*}

period-specification:

FOR [SYSTEM_TIME |BUSINESS_TIME]
[AS OF *value* | FROM *value1* TO *value2* |
BETWEEN *value1* AND *value2*]
Note: AS OF TIMESTAMP can be specified instead of FOR SYSTEM_TIME AS OF. Do not specify SYSTEM_TIME and BUSINESS_TIME more than once per table.

correlation-clause:

{AS} *correlation-name* {([[*new-column-name*,]])}

nested-table-expression:

{TABLE} (*fullselect*) {*correlation-clause}*

correlation-clause:

{AS} *correlation-name* {([[*new-column-name*,]])}

data-change-table-reference:

[FINAL TABLE (*INSERT statement*) |
[FINAL | OLD] TABLE (*searched UPDATE statement*) |
OLD TABLE (*searched DELETE statement*) |
FINAL TABLE (*MERGE statement*)] {*correlation-clause*}

correlation-clause:

{AS} *correlation-name* {([[*new-column-name*,]])}

table-function-reference:

TABLE (*function-name* ({[[*expression* |
TABLE *transition-table-name*,]]})
{*table-UDF-cardinality-clause*})
{*correlation-clause* | *typed-correlation-clause*}

table-UDF-cardinality-clause:

[CARDINALITY *integer-constant* |
CARDINALITY MULTIPLIER *numeric-constant*]

correlation-clause:

{AS} *correlation-name* {([[*new-column-name*,]])}

typed-correlation-clause:

{AS} *correlation-name* ([[*column-name data-type,]]*)

Note: This clause is required for generic table functions. Do not specify it for any other table functions.

data-type:

see CREATE TABLE

table-locator-reference:

TABLE (*table-locator-variable* LIKE *table-name*)
{*correlation-name*}

xmltable-expression:

xmltable-function {*correlation clause*}

correlation-clause:

{AS} *correlation-name* {([[*new-column-name*,]])}

collection-derived-table:

UNNEST ([[[*ordinary-array-expression*,]] |
associative-array-expression]) {WITH ORDINALITY}
{*correlation-clause*}

correlation-clause:

{AS} *correlation-name* {([[*new-column-name*,]])}

joined-table:

[*table-reference* {INNER|[LEFT|RIGHT|FULL] {OUTER}}
JOIN *table-reference* ON *join-condition* |
table-reference CROSS JOIN *table-reference* | (*joined-table*)]

join-condition:

for INNER, LEFT OUTER and RIGHT OUTER:
search-condition
for FULL OUTER:
[[*full-join-expression = full-join-expression* AND]]
Multiple occurrences of these full join conditions may be separated by the AND connector.
full-join-expression:
[[*column-name* | *cast-function*] |
COALESCE ([*column-name* | *cast-function*]
[[[,*column-name* | ,*cast-function*]]])]

where-clause:

WHERE *search-condition*

group-by-clause:

GROUP BY
[[*grouping-expression* | *grouping-sets* | *super-groups*,]]

having-clause:

HAVING *search-condition*

order-by-clause:

ORDER BY [[[*sort-key* {ASC|DESC},]] | INPUT SEQUENCE |
ORDER OF *table-designator*]

sort-key:

[*column-name* | *integer* | *sort-key-expression*]

offset-clause:

OFFSET offset-row-count ROW{S}

fetch-clause:

FETCH [FIRST | NEXT] {1|*integer*} ROW{S} ONLY

Fullselect

fullselect:

[*subselect* | (*fullselect*) | *values-clause*] {{ [UNION |EXCEPT|
INTERSECT] {DISTINCT | ALL} [*subselect* | (*fullselect*)] }}
{*order-by-clause*} {*offset-clause*} {*fetch-clause*}

values-clause:

VALUES [sequence-reference | ([[sequence-reference,]])]

SELECT Statement

select-statement:

{WITH [[*common-table-expression*,]]} *fullselect*
{*update-clause*} {*read-only-clause*} {*optimize-clause*}
{*isolation-clause*} {*queryno-clause*} {SKIP LOCKED DATA}

common-table-expression:

table-identifier {([[*column-name*,]])} AS (*fullselect*)

update-clause:

FOR UPDATE {OF [[*column-name,*]]}

read-only-clause:

FOR [FETCH|READ] ONLY

optimize-clause:

OPTIMIZE FOR *integer* ROW{S}

isolation-clause:

WITH [RR {*lock-clause*} | RS {*lock-clause*} | CS | UR]

lock-clause:

USE AND KEEP [EXCLUSIVE | UPDATE | SHARE] LOCKS

***queryno-clause*:**

QUERYNO *integer*

Embedded SELECT INTO Statement

select-INTO-statement:

{WITH [[*common-table-expression*,]] *select-clause*
INTO [[[*target-variable*,]] | *array-variable*[*array-index*]]

from-clause {*where-clause*} {*group-by-clause*} {*having-clause*}
{*order-by-clause*} {*isolation-clause*} {SKIP LOCKED DATA}
{*query-no-clause*} {*offset-clause*} {*fetch-clause}*

Statements

The following SQL statements are available:

ALLOCATE CURSOR

ALLOCATE *cursor-name*
CURSOR FOR RESULT SET *rs-locator-variable*

ALTER DATABASE

ALTER DATABASE *dbname*
[[BUFFERPOOL *bpname* |
INDEXBP *bpname* |
STOGROUP *stogroup-name* |
CCSID *ccsid-value*]]

ALTER FUNCTION (external scalar or external table)

ALTER [FUNCTION function-name {({{ *parameter-type*,}})} |
SPECIFIC FUNCTION specific-name] [[*option-list*]]

parameter-type:
data-type {AS LOCATOR}

data-type:
[*built-in-type* | *distinct-type-name*] see CREATE TABLE

option-list:
EXTERNAL NAME [*external program name*|*identifier*] |
LANGUAGE [ASSEMBLE|C|COBOL|JAVA|PLI] |
PARAMETER STYLE [SQL|JAVA] |
[NOT DETERMINISTIC|DETERMINISTIC] |
[RETURNS NULL ON NULL INPUT|CALLED ON NULL INPUT] |
[MODIFIES SQL DATA | READS SQL DATA |
CONTAINS SQL | NO SQL] |
[NO EXTERNAL ACTION | EXTERNAL ACTION] |
[PACKAGE PATH *package path* | NO PACKAGE PATH] |
[NO SCRATCHPAD | SCRATCHPAD *length*] |
[NO FINAL CALL | FINAL CALL] |
[ALLOW PARALLEL | DISALLOW PARALLEL] |
[NO DBINFO | DBINFO] |
CARDINALITY *integer* |
[NO COLLID | COLLID *collection-id*] |
WLM ENVIRONMENT [*name* | (*name*,*)] |
ASUTIME [NO LIMIT | LIMIT integer] |
STAY RESIDENT [NO | YES] |
PROGRAM TYPE [SUB | MAIN] |
SECURITY [DB2 | USER | DEFINER] |
[STOP AFTER SYSTEM DEFAULT FAILURES |
STOP AFTER *integer* FAILURES |
CONTINUE AFTER FAILURE] |
RUN OPTIONS *run-time-options* |
[INHERIT SPECIAL REGISTERS|DEFAULT SPECIAL REGISTERS]|
STATIC DISPATCH | [SECURED | NOT SECURED]

Synonyms included for clauses are:
VARIANT for NOT DETERMINISTIC
NOT VARIANT for DETERMINISTIC
NOT NULL CALL for RETURNS NULL ON NULL
INPUT NULL CALL for CALLED ON NULL INPUT

external-java-routine-name:
{ *jar-name:* } *method-name* {*method-signature*}

jar-name:
{schema-name.} jar-id

method-name:
{{ package-id [. | /] }} [. | !] method-id

method-signature:
{({{ java-datatype,}})}

ALTER FUNCTION (compiled SQL scalar)

ALTER *function-designator*
[{ ALTER }
{ ACTIVE VERSION | ALL VERSIONS | VERSION *routine-version-id* }
[[*option-list*]] |
REPLACE {ACTIVE VERSION | VERSION *routine-version-id* }
routine-specification |
ADD VERSION *routine-version-id routine specification* |
ACTIVATE VERSION *routine-version-ID* |
REGENERATE {ACTIVE VERSION |
VERSION *routine-version-id*} {USING APPLICATION COMPATIBILITY *application-value*} |
DROP-VERSION *routine-version-id*]

function-designator:
[FUNCTION *function-name* {({{ *data-type*,}})} |
SPECIFIC FUNCTION *specific-name*]

routine-specification:
({{ *parameter-name* | *data-type*,}} RETURNS *data-type2*
{{ *option-list* }} *SQL-routine-body*

data-type, data-type2:
[*built-in-type* |*distinct-type-name* | a*rray-type-name*]
see CREATE TABLE

SQL-routine-body:
SQL-control-statement

SQL-routine-body:
SQL-control-statement

options-list:
{NOT DETERMINISTIC | DETERMINISTIC} |
{EXTERNAL ACTION | NO EXTERNAL ACTION} |
{READS SQL DATA|CONTAINS SQL|MODIFIES SQL DATA} |
{CALLED ON NULL INPUT |
RETURNS NULL ON NULL INPUT} |
{[ALLOW | DISALLOW] PARALLEL} |
{STATIC DISPATCH} |
{[DISALLOW|ALLOW|DISABLE] DEBUG MODE} |
{QUALIFIER schema-name} |
{PACKAGE OWNER authorization-name } |
{ASUTIME NO LIMIT|ASUTIME LIMIT integer} |
{INHERIT SPECIAL REGISTERS|DEFAULT SPECIAL REGISTERS}|
{WLM ENVIRONMENT FOR DEBUG MODE name} |
{CURRENT DATA NO|CURRENT DATA YES} |
{DEGREE 1|DEGREE ANY} |
{CONCURRENT ACCESS RESOLUTION
[USE CURRENTLY COMMITTED|WAIT FOR OUTCOME]} |
{DYNAMICRULES RUN |
DYNAMICRULES [BIND|DEFINEBIND|DEFINERUN|
INVOKEBIND| INVOKERUN]} |
{APPLICATION ENCODING SCHEME [ASCII | EBCDIC |
UNICODE]} |
{WITHOUT EXPLAIN | WITH EXPLAIN} |
{WITHOUT IMMEDIATE WRITE |WITH IMMEDIATE WRITE} |
{ISOLATION LEVEL CS | ISOLATION LEVEL [RS | RR | UR]} |
{OPTHINT '' |OPTHINT string-constant} |
{SQL PATH [[schema-name | SYSTEM PATH |
SESSION USER | USER,]]} |
{QUERY ACCELERATION [NONE | ENABLE |
ENABLE WITH FAILBACK | ELIGIBLE | ALL]} |
{GET_ACCEL_ARCHIVE [NO | YES]} |
{REOPT NONE |REOPT [ALWAYS | ONCE]} |

{VALIDATE RUN |VALIDATE BIND} |
{ROUNDING DEC_ROUND_[CEILING | DOWN | FLOOR |
HALF_DOWN | HALF_EVEN | HALF_UP | UP]} |
{DATE FORMAT [ISO|EUR|USA|JIS|LOCAL]} |
{DECIMAL ([15|31|15,s|31,s])} |
{FOR UPDATE CLAUSE REQUIRED |
FOR UPDATE CLAUSE OPTIONAL]} |
{TIME FORMAT [ISO|EUR|USA|JIS|LOCAL]} |
[SECURED | NOT SECURED] |
{BUSINESS_TIME SENSITIVE YES|NO} |
{SYSTEM_TIME SENSITIVE YES|NO} |
{ARCHIVE SENSITIVE YES|NO } |
{APPLCOMPAT compatibility-level} |
{CONCENTRATE STATEMENTS [OFF | WITH LITERALS]}

ALTER FUNCTION (inlined SQL scalar)

ALTER [FUNCTION *function-name* {({{ *parameter-type*,}})} |
SPECIFIC FUNCTION *specific-name*]
[[*option-list*]]

parameter-type:
data-type

data-type:
[*built-in-type* | *distinct-type-name*] see CREATE TABLE

option-list:
[NOT DETERMINISTIC | DETERMINISTIC] |
[EXTERNAL ACTION | NO EXTERNAL ACTION] |
[CONTAINS SQL | READS SQL DATA] |
STATIC DISPATCH |
CALLED ON NULL INPUT |
[NOT SECURED|SECURED]

ALTER FUNCTION - SQL table:

ALTER *function-designator* RESTRICT [[*option-list*]]

function-designator:
FUNCTION *function-name* {({{ *parameter-type*,}})} |
SPECIFIC FUNCTION *specific-name*]

parameter-type:
data-type

data-type:
[built-in-type | distinct-type-name]
See CREATE TABLE.

options-list:
{NOT DETERMINISTIC | DETERMINISTIC} |
{EXTERNAL ACTION | NO EXTERNAL ACTION} |
{READS SQL DATA | CONTAINS SQL} |
{CALLED ON NULL INPUT} |
{INHERIT SPECIAL REGISTERS} |
{STATIC DISPATCH} |
{CARDINALITY integer} |
{SECURED | NOT SECURED }

ALTER INDEX

ALTER INDEX index-name { REGENERATE }
{USING APPLICATION COMPATIBILITY applcompat-value}
[[BUFFERPOOL bpname | CLOSE [YES|NO] | COPY [NO|YES] |
PIECESIZE integer [K|M|G] | *using-specification* |
free-specification | *gbpcache-specification* |
[CLUSTER | NOT CLUSTER] | [NOT PADDED | PADDED] |
COMPRESS [NO | YES] |
ADD [COLUMN (column-name {ASC|DESC|RANDOM}) |
INCLUDE COLUMN (column-name)}]]]
{{ ALTER *partition-element* {{ *using-specification* |
free-specification | *gbcache-specification* |
DSSIZE integer G}},}}

using-specification, free-specification, gbpcache-specification:
see CREATE INDEX

partition-element:
PARTITION integer {ENDING {AT} ([[*constant*|MAXVALUE |
MINVALUE,]]) {INCLUSIVE} }

ALTER MASK

ALTER MASK *mask-name* [ENABLE|DISABLE|REGENERATE
{USING APPLICATION COMPATIBILITY *applcompat-value*}]

ALTER PERMISSION

ALTER PERMISSION *permission-name*
[ENABLE | DISABLE | REGENERATE
{USING APPLICATION COMPATIBILITY applcompat-value}]

ALTER PROCEDURE - external:

ALTER PROCEDURE procedure-name [[*option-list*]]

option-list:

DYNAMIC RESULT SETS *integer* |
EXTERNAL NAME ['*string*' | *identifier*] |
LANGUAGE [ASSEMBLE | C | COBOL | JAVA | PLI | REXX] |
PARAMETER STYLE [SQL | GENERAL | GENERAL WITH NULLS
| JAVA] |
[NOT DETERMINISTIC | DETERMINISTIC] |
[PACKAGE PATH *package path* | NO PACKAGE PATH] |
[MODIFIES SQL DATA | READS SQL DATA | CONTAINS SQL |
NO SQL] |
[NO DBINFO | DBINFO] |
[NO COLLID | COLLID collection-id] |
WLM ENVIRONMENT [*name* | (*name*, *)] |
ASUTIME [NO LIMIT | LIMIT integer] |
STAY RESIDENT [NO | YES]
PROGRAM TYPE [SUB | MAIN] |
SECURITY [DB2 | USER | DEFINER] |
RUN OPTIONS run-time options |
COMMIT ON RETURN [NO | YES] |
[INHERIT | DEFAULT] SPECIAL REGISTERS |
CALLED ON NULL INPUT |
[STOP AFTER SYSTEM DEFAULT FAILURES |
STOP AFTER integer FAILURES |
CONTINUE AFTER FAILURE] |
[DISALLOW | ALLOW | DISABLE] DEBUG MODE

Synonyms included for clauses are:

RESULT SET *for* DYNAMIC RESULT SETS
RESULT SETS *for* DYNAMIC RESULT SETS
VARIANT *for* NOT DETERMINISTIC
NOT VARIANT *for* DETERMINISTIC
STANDARD CALL *for* DB2SQL
SIMPLE CALL *for* GENERAL
SIMPLE CALL WITH NULLS *for* GENERAL WITH NULLS
NULL CALL for CALLED ON NULL INPUT

ALTER PROCEDURE - SQL-external:

ALTER PROCEDURE *procedure-name* [[*options-list*]]

option-list:

DYNAMIC RESULT SETS *integer* |
EXTERNAL NAME ['*string*' | *identifier*] |
[NOT DETERMINISTIC | DETERMINISTIC] |
[MODIFIES SQL DATA | READS SQL DATA | CONTAINS SQL] |
[NO COLLID | COLLID collection-id] |
WLM ENVIRONMENT [name | (name, *) |
ASUTIME [NO LIMIT | LIMIT integer] |
STAY RESIDENT [NO | YES] |
PROGRAM TYPE [SUB | MAIN] |
SECURITY [DB2 | USER | DEFINER] |
RUN OPTIONS run-time options |
COMMIT ON RETURN [NO | YES] |
[INHERIT | DEFAULT] SPECIAL REGISTERS |
[STOP AFTER SYSTEM DEFAULT FAILURES |
STOP AFTER integer FAILURES |
CONTINUE AFTER FAILURE]

ALTER PROCEDURE - SQL-native:

PROCEDURE *procedure-name*
[{ALTER} {ACTIVE VERSION |ALL VERSIONS |
VERSION *routine-version-id*} [[*option-list*]] |
REPLACE [ACTIVE VERSION |VERSION *routine-version-id*]
routine-specification |
ADD VERSION *routine-version-id routine-specification* |
ACTIVATE VERSION *routine-version-id* |
REGENERATE {ACTIVE VERSION|VERSION *routine-version-id*}
{USING APPLICATION COMPATIBILITY
applcompat-value} |
DROP VERSION routine-version-id]

routine-specification:

{({{ *parameter-declaration*,}})} {{ *option-list* }}
SQL-routine-body

parameter-declaration:
{IN|OUT|INOUT} parameter-name *data-type*

data-type:
[*built-in-type* |*distinct-type-name* | a*rray-type-name*]
see CREATE TABLE

option-list:

{NOT DETERMINISTIC | DETERMINISTIC} |
{ MODIFIES SQL DATA | READS SQL DATA | CONTAINS SQL} |
{CALLED ON NULL INPUT} |
{DYNAMIC RESULT SETS integer} |
{[DISALLOW | ALLOW | DISABLE] DEBUG MODE} |
{PARAMETER CCSID [ASCII | EBCDIC | UNICODE]} |
{QUALIFIER schema-name} |
{PACKAGE OWNER authorization-name} |
{ASUTIME NO LIMIT | ASUTIME LIMIT integer} |

{COMMIT ON RETURN NO | COMMIT ON RETURN YES |
AUTONOMOUS} |
{INHERIT SPECIAL REGISTERS |
DEFAULT SPECIAL REGISTERS} |
{WLM ENVIRONMENT FOR DEBUG MODE name } |
{[DEFER | NODEFER] PREPARE} |
{CURRENT DATA NO | CURRENT DATA YES} |
{DEGREE 1 | DEGREE ANY} |
{CONCURRENT ACCESS RESOLUTION [USE CURRENTLY
COMMITTED | WAIT FOR OUTCOME]} |
{DYNAMICRULES RUN |
DYNAMICRULES [BIND | DEFINEBIND | DEFINERUN |
INVOKEBIND | INVOKERUN]} |
{APPLICATION ENCODING SCHEME
[ASCII | EBCDIC | UNICODE]} |
{WITHOUT EXPLAIN |WITH EXPLAIN} |
{WITHOUT IMMEDIATE WRITE | WITH IMMEDIATE WRITE} |
{ISOLATION LEVEL CS | ISOLATION LEVEL [RS | RR | UR]} |
{WITHOUT KEEP DYNAMIC | WITH KEEP DYNAMIC} |
{OPTHINT '' | OPTHINT 'string-constant'} |
{SQL PATH [[schema-name | SYSTEM PATH |
SESSION USER | USER,]]} |
{RELEASE AT COMMIT | RELEASE AT DEALLOCATE} |
{QUERY ACCELERATION [NONE | ENABLE {WITH FAILBACK} |
ELIGIBLE | ALL]}|
{GET_ACCEL_ARCHIVE [NO | YES]} |
{REOPT NONE | REOPT [ALWAYS | ONCE]} |
{VALIDATE RUN | VALIDATE BIND} |
{ROUNDING DEC_ROUND_[CEILING | DOWN | FLOOR |
HALF_DOWN | HALF_EVEN | HALF_UP | UP]} |
{DATE FORMAT [ISO | EUR | USA | JIS | LOCAL]} |
{DECIMAL[(15) | (31) | (15,*s*) | (31,*s*)]} |
{FOR UPDATE CLAUSE REQUIRED |
FOR UPDATE CLAUSE OPTIONAL} |
{TIME FORMAT [ISO | EUR | USA | JIS | LOCAL]} |
{BUSINESS_TIME SENSITIVE YES |
BUSINESS_TIME SENSITIVE NO} |
{SYSTEM_TIME SENSITIVE YES |
SYSTEM_TIME SENSITIVE NO} |
{ARCHIVE SENSITIVE YES | ARCHIVE SENSITIVE NO} |
{APPLCOMPAT *level*} |
{CONCENTRATE STATEMENTS [OFF | WITH LITERALS]}

SQL-routine-body:
SQL-control statement

ALTER SEQUENCE

ALTER SEQUENCE sequence-name
[[RESTART {WITH numeric-constant} |
INCREMENT BY numeric-constant |
[NO MINVALUE | MINVALUE numeric-constant] |
[NO MAXVALUE | MAXVALUE numeric-constant] |
[NO CACHE | CACHE numeric-constant] |
[NO CYCLE | CYCLE] |
[NO ORDER | ORDER]]]

ALTER STOGROUP

ALTER STOGROUP stogroup-name
[[ADD VOLUMES ([[[volume-id,]] | [['*',]]]) |
REMOVE VOLUMES ([[[volume-id,]] | [['*',]]])]]
{ DATACLAS dc-name } { MGMTCLAS mc-name }
{ STORCLAS sc-name }

ALTER TABLE

ALTER TABLE table-name
[[ADD {COLUMN} *column-definition* |
ALTER {COLUMN} *column-alteration* |
RENAME COLUMN
    source-column-name TO target-column-name |
DROP {COLUMN} column-name RESTRICT |
ADD PERIOD {FOR} period-definition |
ADD [*unique-constraint* | *referential-constraint* |
    check-constraint] |
DROP [PRIMARY KEY | [UNIQUE | FOREIGN KEY | CHECK |
    CONSTRAINT] constraint-name] |
ADD PARTITION BY *partitioning-clause* |
ADD PARTITION { *partition-clause* } |
ALTER PARTITION integer *partition-clause* |
ROTATE PARTITION [FIRST | integer] TO LAST
    rotate-partition-clause |
ADD ORGANIZE BY HASH organization clause |
ALTER ORGANIZATION
    SET HASH SPACE integer [K | M | G] |
DROP ORGANIZATION |
ADD {SYSTEM} VERSIONING USE HISTORY TABLE
    history-table-name {*extra-row-option*} |
DROP {SYSTEM} VERSIONING |
ADD {{MATERIALIZED} QUERY}
    materialized-query-definition |
ALTER {MATERIALIZED}
    QUERY *materialized-query-definition* |
DROP {MATERIALIZED} QUERY |
DATA CAPTURE [NONE | CHANGES] |
[VOLATILE | NOT VOLATILE] {CARDINALITY} |
[ADD CLONE clone-table-name | DROP CLONE] |
ADD RESTRICT ON DROP | DROP RESTRICT ON DROP |
[ACTIVATE | DEACTIVATE] ROW ACCESS CONTROL |
[ACTIVATE | DEACTIVATE] COLUMN ACCESS CONTROL |
APPEND [NO|YES] |
AUDIT [NONE | CHANGES | ALL] |
VALIDPROC [program-name | NULL] |
[ENABLE ARCHIVE USE archive-table-name |
    DISABLE ARCHIVE]]]

column-definition:
column-name *data-type* {*default-clause*} { NOT NULL}
{*column-constraint*} {*generated-clause*} {IMPLICITLY HIDDEN}
{AS SECURITY LABEL}
{FIELDPROC program-name {({{ constant,}})}}
{INLINE LENGTH integer}

data-type:
[built-in-type | distinct-type-name]
see CREATE TABLE.

XML-type-modifier:
XMLSCHEMA [[*XML-schema-specification*
    {ELEMENT *element-name*},]]

XML-schema-specification:
[ID registered-XML-schema-name |
    [URL target-namespace | NO NAMESPACE] |
        {LOCATION schema-location}]

default-clause:
{WITH} DEFAULT {constant | [SESSION_USER | USER] |
CURRENT SQLID | NULL | *cast-function-name* ([*constant* |
    SESSION_USER | USER] CURRENT SQLID | NULL])}

generated-clause:
[GENERATED {ALWAYS | BY DEFAULT} {*as-identity-clause* |
as-row-change-timestamp-clause} |
GENERATED {ALWAYS}
[*as-row-transaction-start-id-clause* |
as-row-transaction-timpstamp-clause |
as-generated-expression-clause]]
as-identity clause:
AS IDENTITY {({{ START WITH numeric constant |
INCREMENT BY [*1* | numeric constant] | [NO
CACHE | CACHE 20 | CACHE integer] | [NOCYCLE |
CYCLE] [NO MAXVALUE | MAXVALUE numeric
constant] | [NO MINVALUE | MINVALUE numeric
constant] |
[ORDER | NO ORDER] }},)}
as-row-change-timestamp-clause:
FOR EACH ROW ON UPDATE AS ROW CHANGE
TIMESTAMP
as-row-transaction-timestamp-clause:
AS ROW [[BEGIN | START] | END]
as-row-transaction-start-id-clause:
AS TRANSACTION START ID
as-generated-expression-clause:
AS ([DATA CHANGE OPERATION | *special-register* |
session-variable])
special-register:
[CURRENT CLIENT_ACCTING |
CURRENT CLIENT_APPLNAME |
CURRENT CLIENT_CORR_TOKEN |
CURRENT CLIENT_USERID |
CURRENT CLIENT_WRKSTNNAME |
CURRENT SERVER |
CURRENT SQLID |
SESSION_USER]
session-variable:
[SYSIBM.PACKAGE_NAME |
SYSIBM.PACKAGE_SCHEMA |
SYSIBM.PACKAGE_VERSION]
column-constraint:
[references-clause | check-constraint]

column-alteration:
column-name
[SET [DATA TYPE altered-data-type
{INLINE LENGTH integer} | default clause |
INLINE LENGTH integer |
GENERATED {ALWAYS | BY DEFAULT}
{*identity-alteration* | *as-transaction-timestamp-clause* |
as-transaction-id-clause}] | DROP DEFAULT]
identity-alteration:
[[RESTART {WITH numeric-constant} |
SET INCREMENT BY numeric-constant |
SET [NO MINVALUE | MINVALUE numeric constant] |
SET [NO MAXVALUE | MAXVALUE numeric-constant] |
SET [NO CACHE | CACHE integer-constant] |
SET [NO CYCLE | CYCLE] | SET [NO ORDER | ORDER]]]

unique-constraint:
{CONSTRAINT constraint-name} [PRIMARY KEY | UNIQUE]
([[column-name,]] {, BUSINESS_TIME WITHOUT OVERLAPS})

referential-constraint:
{CONSTRAINT constraint-name} FOREIGN KEY
([[column-name {PERIOD BUSINESS_TIME},]]) *references-clause*

references-clause:
REFERENCES table-name
{([[column-name {PERIOD BUSINESS_TIME},]])}
{ON DELETE [RESTRICT | CASCADE | SET NULL | NO ACTION]} {ENFORCED | NOT ENFORCED} {ENABLE QUERY OPTIMIZATION}

check-constraint:
{CONSTRAINT constraint-name} CHECK (*check-condition*)

partitioning clause:
{RANGE} ([[*partition-expression*,]]) ([[*partition-element*,]])

partition-expression:
column-name {NULLS LAST} {ASC|DESC}

partition element:
ENDING {AT} ([[*constant* | MAXVALUE | MINVALUE,]]) {INCLUSIVE} {HASH SPACE *integer* [K | M | G]}

partition-clause:
[ENDING {AT} ([[*constant* | MAXVALUE | MINVALUE,]]) {INCLUSIVE} |HASH SPACE *integer* [K | M | G]]

partition-rotation:
ENDING {AT} ([[*constant* | MAXVALUE | MINVALUE,]]) {INCLUSIVE} RESET

extra-row-option:
ON DELETE ADD EXTRA ROW

materialized-query-definition:
(full-select) *refreshable-table-options*

refreshable-table-options:
DATA INITIALLY DEFERRED REFRESH DEFERRED
{{ [MAINTAINED BY SYSTEM |
MAINTAINED BY USER] |
[ENABLE QUERY OPTIMIZATION |
DISABLE QUERY OPTIMIZATION] }}

materialized- query-table-alteration:
SET [[MAINTAINED BY [SYSTEM | USER] |
[ENABLE|DISABLE] QUERY OPTIMIZATION]]

period-definition:
[SYSTEM_TIME (begin-column-name , end-column-name) |
BUSINESS_TIME (begin-column-name, end-column-name)
{EXCLUSIVE | INCLUSIVE}]

organization-clause:
UNIQUE ([[*columnname*,]]) {HASH SPACE 64 M |
HASH SPACE integer [K | M | G]}

ALTER TABLESPACE

ALTER TABLESPACE {dbname.} tsname
[[BUFFERPOOL bpname | CCSID ccsid-value |
CLOSE [YES | NO] | COMPRESS [YES | NO] |
DROP PENDING CHANGES | DSSIZE integer G |
LOCKMAX [SYSTEM | integer] | LOCKSIZE [ANY |
TABLESPACE | TABLE | PAGE | ROW | LOB] |
{NOT} LOGGED | MAXROWS integer |
MAXPARTITIONS integer | MEMBER CLUSTER [YES|NO] |
SEGSIZE integer | TRACKMOD [YES|NO] |
using-block | *free-block* | *gbpcache-block*]]
{{ ALTER PARTITION integer [[*using-block* | *free-block* |
gbpcache-block | COMPRESS [YES|NO] |
TRACKMOD [YES|NO]]] }}

using-block:

[[USING [VCAT catalog-name | STOGROUP stogroup-name]
| PRIQTY integer | SECQTY integer | ERASE [YES | NO]]]

free-block:

[[FREEPAGE integer | {PCTFREE 5 | PCTFREE [*smallint* |
{5 | *smallint}* FOR UPDATE *smallint*]

gbpcache-block:

GBPCACHE [CHANGED | ALL | SYSTEM | NONE]

ALTER TRIGGER (advanced)

ALTER TRIGGER trigger-name
[{ALTER} {ACTIVE VERSION | VERSION trigger-version-id]
[[*option-list*]] |
REPLACE {ACTIVE VERSION | VERSION trigger-version-id}
trigger-specification |
ADD VERSION trigger-version-id *trigger specification* |
REGENERATE {ACTIVE VERSION |
VERSION trigger-version-id}
USING APPLICATION COMPATIBILITY
applcompat-value} |
DROP VERSION trigger-version-id]

trigger-specification:

trigger-activation-time trigger-event ON
[*table-name* | *view-name*]
{REFERENCING [[OLD {ROW} {AS} *correlation-name* |
NEW {ROW} {AS} *correlation-name* |
OLD_TABLE {AS} *table-identifier* |
NEW_TABLE {AS} *table-identifier* }}
trigger-granularity {{ *option-list* }} *triggered-action*

trigger-activation-time:

[{NO CASCADE} BEFORE | AFTER | INSTEAD OF]

trigger-event:

[INSERT | DELETE | UPDATE {OF [[*column-name*,]]}]

trigger-granularity:

{FOR EACH STATEMENT | FOR EACH ROW}

option-list:

{[DISALLOW | ALLOW | DISABLE] DEBUG MODE} |
{QUALIFIER *schema-name*} |
{ASUTIME [NO LIMIT | LIMIT integer]} |
{WLM ENVIRONMENT FOR DEBUG MODE *name*} |
{CURRENT DATA [YES | NO]} |
{CONCURRENT ACCES RESOLUTION
[USE CURRENTLY COMMITTED | WAIT FOR OUTCOME]} |
{DYNAMICRULES [RUN | BIND]} |
{APPLICATION ENCODING SCHEME
[ASCII | EBCDIC | UNICODE]} |
{[WITHOUT | WITH] EXPLAIN} |
{[WITHOUT | WITH] IMMEDIATE WRITE} |
{ISOLAITON LEVEL [CS | RS | RR | UR]} |
{OPTHINT ['' | *string-constant*]} |
{SQL PATH [[schema-name | SYSTEM PATH |
SESSION USER | USER,]]} |
{RELEASE AT [COMMIT | DEALLOCATE]} |
{ROUNDING DEC_ROUND_[CEILING | DOWN | FLOOR |
HALF_DOWN | HALF_EVEN | HALF_UP | UP]} |
{DATE FORMAT [ISO | EUR | USA | JIS | LOCAL]} |
{DECIMAL ([15 | 31 | 15,s | 31,s])} |
{TIME FORMAT [ISO | EUR | USA | JIS| LOCAL]} |
{FOR UPDATE CLAUSE [REQUIRED | OPTIONAL]} |
{{NOT} SECURED} |
{BUSINES_TIME SENSITIVE [YES | NO]} |
{SYSTEM_TIME SENSITIVE [YES | NO]} |
{ARCHIVE SENSITIVE [YES | NO]} |
{APPLCOMPAT *level*} |
{CONCENTRATE STATEMENTS [OFF | WITH LITERALS]}

triggered-action:

{WHEN (*search-condition*)} *SQL-trigger-body*

SQL-trigger-body:

[*SQL-control-statement* | *triggered-SQL-statement*]

ALTER TRIGGER (basic)

ALTER TRIGGER trigger-name [SECURED | NOT SECURED]

ALTER TRUSTED CONTEXT

ALTER TRUSTED CONTEXT *context-name*
[[ALTER [[SYSTEM AUTHID *authorization-name* |
[NO DEFAULT ROLE | DEFAULT ROLE *role-name*
{WITHOUT ROLE AS OBJECT OWNER |
WITH ROLE AS OBJECT OWNER}] |
[ENABLE|DISABLE] |
[NO DEFAULT SECURITY LABEL |
DEFAULT SECURITY LABEL *seclabel-name*] |
ATTRIBUTES ([[[ADDRESS *address-value* |
ENCRYPTION *encryption-value* |
SERVAUTH *servauth-value*] |
JOBNAME *jobname-value,]]*) |
ADD ATTRIBUTES ([[[ADDRESS *address-value* |
SERVAUTH *servauth-value*] |
JOBNAME *jobname-value,*]]) |
DROP ATTRIBUTES ([[[ADDRESS {*address-value*} |
SERVAUTH {*servauth-value*}] |
JOBNAME {*jobname-value*},]]) |
user-clause]]

user-clause:

[ADD USE FOR [[*authorization-name* {*use-options*} |
EXTERNAL SECURITY PROFILE profile-name {use-options} |
PUBLIC {WITHOUT AUTHENTICATION |
WITH AUTHENTICATION},]] |
REPLACE USE FOR [[*authorization-name* {*use-options*} |
EXTERNAL SECURITY PROFILE profile-name |
PUBLIC {WITHOUT AUTHENTICATION |
WITH AUTHENTICATION},]] |
DROP USE FOR [[*authorization-name* |
EXTERNAL SECURITY PROFILE profile-name | PUBLIC,]]]

use-options:
{ROLE *role-name*}
{SECURITY LABEL *seclabel-name*}
{WITHOUT AUTHENTICATION |WITH AUTHENTICATION}

ALTER VIEW

ALTER VIEW view-name REGENERATE
{USING APPLICATION COMPATIBILITY *applcompat-value*}

ASSOCIATE LOCATORS

ASSOCIATE {RESULT SET} LOCATOR{S}
([[rs-locator-variable,]]) WITH PROCEDURE [procedure-name | host-variable]

BEGIN DECLARE SECTION

Marks beginning of host variable declaration section.

CALL

CALL [procedure-name | host-variable]
{({{ expression | NULL | TABLE transition-table-name ,}}) |
USING DESCRIPTOR descriptor-name}

CLOSE cursor-name

COMMENT

COMMENT ON
[[alias-designator |
COLUMN [table-name | view-name].column-name |
function-designator {ACTIVE VERSION |
VERSION routine-version-id} |
INDEX index-name |
PACKAGE *collection-id.pkg-name* {{VERSION} version-id} |
PLAN *plan-name* |
PROCEDURE procedure-name
{ACTIVE VERSION|VERSION routine-version-id} |
ROLE *role-name* | SEQUENCE *sequence-name* |
TABLE [table-name | view-name] |
TRIGGER *trigger-name* {ACTIVE VERSION |
VERSION trigger-version-id} |
TRUSTED CONTEXT context-name |
TYPE distinct-type-name | MASK mask-name |
PERMISSION permission-name |
VARIABLE variable-name] |
multiple-column-list] IS string-constant

alias-designator:

{PUBLIC} ALIAS *alias-name* {FOR TABLE | FOR SEQUENCE}

multiple-column-list:

[*table-name* | *view-name*] ([[*column-name* IS *string-constant*,]])

function-designator:
[FUNCTION function-name {({{ *parameter-type*,}})} |
SPECIFIC FUNCTION *specific-name*]
parameter type:
data-type {AS LOCATOR}
data-type:
see CREATE TABLE

COMMIT

COMMIT {WORK}

CONNECT

CONNECT {TO [location-name | host-variable]
{authorization} | RESET | authorization}

authorization:
USER host-variable USING host-variable

CREATE ALIAS

CREATE {PUBLIC} ALIAS [table-alias | sequence-alias]

table-alias:
alias-name FOR {TABLE} [*table-name* | *view-name* |
alias-name2]

sequence-alias:
alias-name FOR SEQUENCE *sequence-name*

CREATE AUXILIARY TABLE

CREATE [AUXILIARY | AUX] TABLE aux-table-name
IN {dbname.} tsname
STORES table-name {APPEND NO | APPEND YES}
COLUMN *column-name* {PART integer}

CREATE DATABASE

CREATE DATABASE dbname {{ STOGROUP
{SYSDEFLT | stogroup-name} | BUFFERPOOL bpname |
INDEXBP bpname | AS WORKFILE {FOR member-name} |
CCSID [ASCII|EBCDIC|UNICODE] }}

CREATE FUNCTION

There are six types of functions: compiled SQL scalar, external scalar, external table, sourced, inlined SQL scalar, and SQL table.

CREATE FUNCTION (compiled SQL scalar)

CREATE FUNCTION *function-name* ({{ *parameter-declaration*,}})
[*function-definition* | WRAPPED *obfuscated-statement-text*]

parameter-declaration:
parameter-name [*data-type* | TABLE LIKE [*table-name* |
view-name] AS LOCATOR]

data-type:
[built-in-type | distinct-type-name | array-type-name]
see CREATE TABLE

function-definition:
RETURNS data-type2 {VERSION V1 |
VERSION routine-version-id}
{*option-list*} *SQL-routine-body*

SQL-routine-body:
SQL-control-statement

option-list:

{LANGUAGE SQL}
{SPECIFIC specific-name}
{NOT DETERMINISTIC | DETERMINISTIC}
{EXTERNAL ACTION | NO EXTERNAL ACTION}
{READS SQL DATA | CONTAINS SQL | MODIFIES SQL DATA}
{CALLED ON NULL INPUT | RETURNS NULL ON NULL INPUT}
{STATIC DISPATCH}
{[ALLOW | DISALLOW] PARALLEL}
{[DISALLOW | ALLOW | DISABLE] DEBUG MODE}
{PARAMETER CCSID [ASCII | EBCDIC | UNICODE]}
{QUALIFIER *schema-name*}
{PACKAGE OWNER *authorization-name*}
{ASUTIME NO LIMIT | ASUTIME LIMIT *integer*}
{INHERIT SPECIAL REGISTERS | DEFAULT SPECIAL REGISTERS}
{WLM ENVIRONMENT FOR DEBUG MODE *name*}
{CURRENT DATA NO | CURRENT DATA YES}
{DEGREE 1 | DEGREE ANY}
{CONCURRENT ACCESS RESOLUTION
[USE CURRENTLY COMMITED | WAIT FOR OUTCOME]}
{DYNAMICRULES RUN | DYNAMICRULES [BIND | DEFINEBIND |
DEFINERUN | INVOKEBIND | INVOKERUN]}
{APPLICATION ENCODING SCHEME [ASCII | EBCDIC | UNICODE]}
{WITHOUT EXPLAIN | WITH EXPLAIN}
{WITHOUT IMMEDIATE WRITE | WITH IMMEDIATE WRITE}
{ISOLATION LEVEL CS | ISOLATION LEVEL [RS | RR | UR]}
{OPTHINT '' | OPTHINT *string-constant*}
{QUERY ACCELERATION [NONE|ENABLE {WITH FAILBACK}|
ELIGIBLE | ALL]}
{GET_ACCEL_ARCHIVE [NO | YES]}
{SQL PATH [[schema-name | SYSTEM PATH | SESSION USER | USER,]]}
{REOPT NONE | REOPT [ALWAYS | ONCE]}
{VALIDATE RUN | VALIDATE BIND}
{ROUNDING DEC_ROUND_[CEILING | DOWN | FLOOR |
HALF_DOWN | HALF_EVEN | HALF_UP | UP]}
{DATE FORMATE [ISO | EUR | USA | JIS | LOCAL]}
{DECIMAL ([15{,s} | 31{,s}])}
{FOR UPDATE CLAUSE REQUIRED |
FOR UPDATE CLAUSE OPTIONAL}
{TIME FORMAT [ISO | EUR | USA | JIS | LOCAL]}
{NOT SECURED | SECURED}
{BUSINESS_TIME SENSITIVE YES |
BUSINESS_TIME SENSITIVE NO}
{SYSTEM_TIME SENSITIVE YES |
SYSTEM_TIME SENSITIVE NO}
{ARCHIVE SENSITIVE YES | ARCHIVE SENSITIVE NO}
{APPLCOMPAT *level*}
{CONCENTRATE STATEMENTS OFF |
CONCENTRATE STATEMENTS WITH LITERALS}

CREATE FUNCTION (External scalar)

CREATE FUNCTION function-name ({{ *parameter-declaration*,}}) RETURNS [data-type2 {AS LOCATOR} | data-type3 CAST FROM data-type4 {AS LOCATOR}] *option-list*

parameter-declaration:

{parameter-name} [data-type {AS LOCATOR} | TABLE LIKE [table-name | view-name] AS LOCATOR]

data-type:

[built-in-type| distinct-type-name]
see CREATE TABLE

option-list:

{SPECIFIC specific-name}
{PARAMETER [[CCSID [ASCII | EBCDIC | UNICODE] |
VARCHAR [NULTERM | STRUCTURE]]]}
EXTERNAL {NAME [external-program-name | identifier]
LANGUAGE [ASSEMBLE | C | COBOL | PLI | JAVA]
PARAMETER STYLE [SQL | JAVA]
{NOT DETERMINISTIC | DETERMINISTIC}
{FENCED}
{RETURNS NULL ON NULL INPUT | CALLED ON NULL INPUT}
{READS SQL DATA | NO SQL | MODIFIES SQL DATA |
CONTAINS SQL}
{EXTERNAL ACTION | NO EXTERNAL ACTION}
{NO SCRATCHPAD | SCRATCHPAD {100 | length}
{NO PACKAGE PATH | PACKAGE PATH package-path}
{NO FINAL CALL | FINAL CALL}
{ALLOW PARALLEL | DISALLOW PARALLEL}
{NO DBINFO | DBINFO}
{NO COLLID | COLLID collection-id}
{WLM ENVIRONMENT [name | (name)}
{ASUTIME [NO LIMIT | LIMIT integer]}
{STAY RESIDENT [NO | YES]}
{PROGRAM TYPE [SUB | MAIN]}
{SECURITY [DB2 | USER | DEFINER]}
{STOP AFTER SYSTEM DEFAULT FAILURES |
STOP AFTER integer FAILURES |
CONTINUE AFTER FAILURE}
{INHERIT | DEFAULT] SPECIAL REGISTERS}
{STATIC DISPATCH}
{NOT SECURED | SECURED}
{RUN OPTIONS run-time-options}

Note: DISALLOW PARALLEL *becomes default if any of the following are specified:* NOT DETERMINISTIC, EXTERNAL ACTION, FINAL CALL, MODIFIES SQL, SCRATCHPAD.

external-java-routine-name:

{ *jar-name:*} *method-name* {*method-signature*}

jar-name:

{schema-name.} jar-id

method-name:

{{ package-id [.|/] }} class-id [.|!] method-id

method-signature:

{({{ java-datatype,}})}

See ALTER FUNCTION *for synonyms included for clauses.*

CREATE FUNCTION (External table)

CREATE FUNCTION function-name ({{ *parameter-declaration*,}}) RETURNS [TABLE ([[column-name data-type {AS LOCATOR},]]) |
GENERIC TABLE] *option-list*

option-list:

{SPECIFIC specific-name}
{PARAMETER [CCSID [ASCII | EBCDIC | UNICODE] |
VARCHAR [NULTERM | STRUCTURE]]}
EXTERNAL {NAME ['string' | identifier]}
LANGUAGE [ASSEMBLE | C | COBOL | PLI]
PARAMETER STYLE SQL
{NOT DETERMINISTIC | DETERMINISTIC}
{FENCED}
{[RETURNS NULL | CALLED] ON NULL INPUT}
{READS SQL DATA | CONTAINS SQL | NO SQL}
{EXTERNAL ACTION | NO EXTERNAL ACTION}
{NO PACKAGE PATH | PACKAGE PATH package-path}
{NO SCRATCHPAD | SCRATCHPAD {100 | length}}
{NO FINAL CALL | FINAL CALL}
DISALLOW PARALLEL
{NO DBINFO | DBINFO}
{CARDINALITY integer}
{NO COLLID | COLLID collection-id}
{WLM ENVIRONMENT [name | (name)]}
{ASUTIME [NO LIMIT | LIMIT integer]}
{STAY RESIDENT [NO | YES]}
{PROGRAM TYPE [SUB | MAIN]}
{SECURITY [DB2 | USER | DEFINER]}
{RUN OPTIONS run-time-options }
{INHERIT | DEFAULT] SPECIAL REGISTERS}
{STATIC DISPATCH}
{STOP AFTER SYSTEM DEFAULT FAILURES |
STOP AFTER integer FAILURES |
CONTINUE AFTER FAILURE}
{NOT SECURED | SECURED}

parameter-declaration:

{parameter-name} [parameter-type]

parameter-type:

see ALTER FUNCTION

data-type:

[built-in-type | distinct-type-name]
see CREATE TABLE

See ALTER FUNCTION *for synonyms included for clauses.*

CREATE FUNCTION (sourced)

CREATE FUNCTION function-name ({{ *parameter-declaration*,}}) RETURNS data-type2 {AS LOCATOR}
{SPECIFIC specific-name}
{PARAMETER CCSID [ASCII | EBCDIC | UNICODE]}
{SOURCE [function-name {([[parameter-type,]])} |
SPECIFIC specific-name]}

parameter-declaration:

{parameter-name} *parameter-type*

parameter-type:

see ALTER FUNCTION

data-type:

[*built-in-type | distinct-type-name*]
*s*ee CREATE TABLE

CREATE FUNCTION (inline SQL scalar)

CREATE FUNCTION *function-name*
({{ parameter-declaration,}})
[*function-definition* | WRAPPED *obfuscated-statement-text*]

parameter-declaration:
parameter-name *data-type*

data-type:
[built-in-type | distinct-type-name]
see CREATE TABLE

fuction-definition:
RETURNS data-type2 {LANGUAGE SQL} option-list
RETURN statement

option-list:
Note: These options can be specified once, in any order.
{SPECIFIC *specific-name*}
{PARAMETER CCSID [ASCII | EBCDIC | UNICODE}
{NOT DETERMINISTIC | DETERMINISTIC}
{EXTERNAL ACTION | NO EXTERNAL ACTION}
{READS SQL DATA|CONTAINS SQL }
{CALLED ON NULL INPUT}
{STATIC DISPATCH}
{NOT SECURED | SECURED}

CREATE FUNCTION (SQL table)

CREATE FUNCTION function-name
({{ parameter-declaration,}})
RETURNS TABLE ([[column-name data-type2,]])
option-list SQL-routine-body

parameter-declaration:
parameter-name *parameter-type*
parameter-type:
[data-type1 | TABLE LIKE [table-name | view-name]
AS LOCATOR]

data-type1, data-type2:
[built-in-type | distinct-type-name]
see CREATE TABLE

option-list:
{LANGUAGE SQL}
{SPECIFIC specific-name}
{NOT DETERMINISTIC | DETERMINISTIC}
{EXTERNAL ACTION | NO EXTERNAL ACTION}
{READS SQL DATA | CONTAINS SQL}
{CALLED ON NULL INPUT}
{INHERIT SPECIAL REGISTERS}
{STATIC DISPATCH}
{CARDINALITY integer}
{PARAMETER CCSID [ASCII | EBCDIC | UNICODE]}
{NOT SECURED | SECURED}

SQL-routine-body:
[RETURN statement |
BEGIN ATOMIC RETURN statement END]

CREATE GLOBAL TEMPORARY TABLE

CREATE GLOBAL TEMPORARY TABLE table-name
[([[*column-definition,*]]) | LIKE [table-name | view-name]]
{CCSID [ASCII | EBCDIC | UNICODE]}

column-definition:
column-name *data-type* {NOT NULL}

data-type:
[built-in-type | distinct-type-name]
see CREATE TABLE

CREATE INDEX

CREATE {UNIQUE {WHERE NOT NULL}}
INDEX index-name ON
[table-name ([[[column-name | key-expression]
{ASC | DESC | RANDOM},]]
{,BUSINESS_TIME [WITHOUT | WITH] OVERLAPS}) |
aux-table-name]
other-options

other-options:

{XML-index specification}
{INCLUDE ([[column-name,]])}
{{ {NOT CLUSTER | CLUSTER} |
PARTITIONED |
{NOT PADDED | PADDED} |
{*using-specification*} |
{*free-specification*} |
{*gbpcache-specification*} |
{DEFINE [YES | NO]} |
{COMPRESS [YES|NO]} |
{[INCLUDE | EXCLUDE] NULL KEYS} }}
{PARTITION BY {RANGE}
([[partition element {{ *using-specification* } |
{ *free-specification* } |
{ *gbpcache-specification* } |
DSSIZE integer G }},]])}
{{ BUFFERPOOL bpname | {CLOSE [YES | NO]} |
{DEFER [NO | YES]} | PIECESIZE integer [K | M | G] |
{COPY [NO|YES]}

XML-index specification:

GENERATE KEY{S} USING XMLPATTERN *XML_pattern_clause*
AS *SQL-data-type*

XML-pattern-clause:

{*prolog*} *pattern-expression*

prolog:

[[declare namespace *NCName* = *StringLiteral* ; |
declare default element namespace *StringLiteral* ;]]

pattern-expression:

{{ [/ | //] [*forward-axis* [*element-name* | * |
nsprefix:* | *:*NCName*] | .] }}
{[/ | //] [@*attribute-name* | attribute::*attribute-name* |
@* | attribute::* | *forward-axis* text( ) |
function-step]}

forward-axis
{child:: | descendant:: | self:: | descendant-or-self::}

function-step:
[fn::upper-case(*) | fn::exists ([element-name | * |
nsprefix:* | *:NCNName | child::element-name |
child::* | child::nsprefix:* | child::*:NCNAME |
@attribute-name | attribute::attribute-name |
@* | attribute::*])]

SQL-data-type:

SQL [VARCHAR (*integer*) | DECFLOAT{(34)} | DATE |
TIMESTAMP {(12)}]

using-specification:

USING [VCAT catalog-name | STOGROUP stogroup-name
{{ {PRIQTY [12 | integer]} {SECQTY integer}
{ERASE [NO|YES]} }}]

free-specification:

[[{FREEPAGE [0 | integer]} | {PCTFREE [10 | integer]}]]

gbpcache-specification:
{GBPCACHE [CHANGED | ALL | NONE]}
partition element:
PARTITION integer {ENDING {AT} ([[constant|MAXVALUE | MINVALUE,]]) {INCLUSIVE}

CREATE MASK

CREATE MASK *mask-name* ON *table-name*
{{AS} correlation-name} FOR COLUMN column-name
RETURN case-expression {DISABLE | ENABLE}

CREATE PERMISSION

CREATE PERMISSION *permission-name* ON *table-name*
{{AS} correlation-name}
FOR ROWS WHERE *search-condition*
ENFORCED FOR ALL ACCESS {DISABLE | ENABLE}

CREATE PROCEDURE (external)

CREATE PROCEDURE *procedure-name*
{({{ *parameter-declaration*,}})}
option-list
parameter-declaration:
{IN | OUT | INOUT} {parameter-name} *parameter-type*
parameter-type:
[data-type {AS LOCATOR} |
TABLE LIKE [table-name | view-name] AS LOCATOR]
data-type:
[built-in-type | distinct-type-name]
see CREATE TABLE
option-list:
{[DYNAMIC RESULT SETS [0 | integer]}
{PARAMETER [[CCSID [ASCII | EBCDIC | UNICODE] |
VARCHAR [NULTERM | STRUCTURE]]]
EXTERNAL {NAME ['string' | identifier]}
LANGUAGE [ASSEMBLE | C | COBOL | PLI | JAVA | REXX]
{MODIFIES SQL DATA | READS SQL DATA |
CONTAINS SQL | NO SQL}
{PARAMETER STYLE [SQL | GENERAL | JAVA |
GENERAL WITH NULLS]}
{NOT DETERMINISTIC | DETERMINISTIC}
{NO PACKAGE PATH | PACKAGE PATH package-path}
{FENCED} {NO DBINFO | DBINFO}
{NO COLLID | COLLID collection-id}
{WLM ENVIRONMENT [name | (name,*)]}
{ASUTIME [NO LIMIT|LIMIT integer]}
{STAY RESIDENT [NO|YES]}
{PROGRAM TYPE [SUB|MAIN]}
{SECURITY [DB2|USER|DEFINER]}
{STOP AFTER SYSTEM DEFAULT FAILURES |
STOP AFTER integer FAILURES |
CONTINUE AFTER FAILURE}
{RUN OPTIONS run-time-options}
{COMMIT ON RETURN [NO|YES] }
{[INHERIT | DEFAULT] SPECIAL REGISTERS }
{CALLED ON NULL INPUT}
{DISALLOW DEBUGMODE | ALLOW DEBUG MODE |
DISABLE DEBUG MODE}
See ALTER PROCEDURE *and* ALTER FUNCTION *for 'Synonyms included for' clauses.*
external-java-routine-name:
{ *jar-name:*} *method-name* {*method-signature*}

jar-name:
{schema-name.} jar-id}
method-name:
{package-id [.|/]} class-id [.|!] method-id
method-signature:
{({{ java-datatype,}})}

CREATE PROCEDURE (SQL external)CREATE PROCEDURE
procedure-name
{({{ parameter-declaration,}})}
option-list
SQL-routine-body

parameter-declaration:
{IN | OUT | INOUT} parameter-name *parameter-type*
parameter-type:
[built-in-datatype |
TABLE LIKE table-name AS LOCATOR]
built-in-type:
see CREATE TABLE.

option-list:
Note: These options can be specified once, in any order.
LANGUAGE SQL {FENCED}
{EXTERNAL NAME ['string' | identifier]}
{DYNAMIC RESULT SETS [0|integer]}
{PARAMETER CCSID [ASCII | EBCDIC | UNICODE]}
{PARAMETER VARCHAR [NULTERM | STRUCTURE]}
{NOT DETERMINISTIC | DETERMINISTIC}
{CALLED ON NULL INPUT }
{MODIFIES SQL DATA | CONTAINS SQL | READS SQL DATA}
{NO DBINFO }
{NO COLLID | COLLID collection-id}
{WLM ENVIRONMENT [name | (name,*)]}
{ASUTIME [NO LIMIT | LIMIT integer]}
{STAY RESIDENT [NO | YES]}
{PROGRAM TYPE [MAIN | SUB]}
{SECURITY [DB2 | USER | DEFINER]}
{RUN OPTIONS run-time-options}
{COMMIT ON RETURN [NO|YES]}
{[INHERIT | DEFAULT] SPECIAL REGISTERS}
{STOP AFTER SYSTEM DEFAULT FAILURES |
STOP AFTER integer FAILURES |
CONTINUE AFTER FAILURE}
See ALTER PROCEDURE *and* ALTER FUNCTION *for 'Synonyms included for' clauses.*

CREATE PROCEDURE (SQL native)
CREATE PROCEDURE procedure-name
{({{ *parameter-declaration,*}})}
[{VERSION [V1|routine-version-id]}
{option list} SQL routine-body |
WRAPPED *obfuscated-statement-text*]

parameter-declaration:
{IN | OUT | INOUT} parameter-name *parameter-type*
parameter-type:
[*data-type* | TABLE LIKE [table-name | view-name]
AS LOCATOR]
data-type:
[built-in-type | distinct-type-name | array-type-name]
see CREATE TABLE

option-list:
Note: These options can be specified once, in any order.

{LANGUAGE SQL} {NOT DETERMINISTIC | DETERMINISTIC}
{CONTAINS SQL | READS SQL DATA | MODIFIES SQL DATA}
{CALLED ON NULL INPUT }
{DYNAMIC RESULT SETS [0 | integer]}
{[DISALLOW | ALLOW | DISABLE] DEBUG MODE}
{PARAMETER CCSID [ASCII | EBCDIC | UNICODE]}
{QUALIFIER schema-name}
{PACKAGE OWNER authorization-name }
{ASUTIME [NO LIMIT | LIMIT integer]} |
{COMMIT ON RETURN [NO | YES] | AUTONOMOUS}
{[INHERIT | DEFAULT] SPECIAL REGISTERS}
{WLM ENVIRONMENT FOR DEBUG MODE name}
{[DEFER | NODEFER] PREPARE}
{CURRENT DATA [NO | YES]}
{DEGREE [1|ANY]}
{CONCURRENT ACCESS RESOLUTION [USE CURRENTLY COMMITTED | WAIT FOR OUTCOME]}
{DYNAMICRULES [RUN|BIND|DEFINEBIND|DEFINERUN | INVOKEBIND | INVOKERUN]}
{APPLICATION ENCODING SCHEME [ASCII |EBCDIC | UNICODE]}
{[WITHOUT | WITH] EXPLAIN}
{[WITHOUT | WITH] IMMEDIATE WRITE}
{ISOLATION LEVEL [CS|RS|RR|UR]}
{[WITHOUT | WITH] KEEP DYNAMIC}
{OPTHINT [’’ | *’string-constant*’]}
{SQL PATH [[*schema-name* | SYSTEM PATH | USER | SESSION USER,]]}
{QUERY ACCELERATION [NONE| ENABLE {WITH FAILBACK} |ELIGIBLE | ALL]}
{GET_ACCEL_ARCHIVE [NO | YES]}
{RELEASE AT [COMMIT | DEALLOCATE]}
{REOPT [NONE | ALWAYS | ONCE]}
{VALIDATE [RUN | BIND]}
{ROUNDING DEC_ROUND_[CEILING | DOWN | FLOOR | HALF_DOWN | HALF_EVEN | HALF_UP | ROUND_UP]}
{DATE FORMAT [ISO | EUR | USA | JIS | LOCAL]} |
{DECIMAL [(15) | (31) | (15,*s*) | (31,*s*)]}
{FOR UPDATE CLAUSE [REQUIRED | OPTIONAL]}
{TIME FORMAT [ISO | EUR | USA | JIS | LOCAL]}
{BUSINESS_TIME SENSITIVE [YES|NO]}
{SYSTEM_TIME SENSITIVE [YES|NO]}
{ARCHIVE SENSITIVE [YES|NO]}
{APPLCOMPAT compatibility-level}
{CONCENTRATE STATEMENTS [OFF | WITH LITERALS]}

SQL-routine-body:

{*SQL-control-statement*}
{ALTER DATABASE *statement*}
{ALTER FUNCTION *statement* (external scalar, external table, sourced, SQL scalar, or SQL table)}
{ALTER INDEX *statement*}
{ALTER PROCEDURE statement (external, SQL – external, or SQL – native}
{ALTER SEQUENCE *statement*}
{ALTER STOGROUP *statement*}
{ALTER TABLE *statement*}
{ALTER TABLESPACE *statement*}
{ALTER TRUSTED CONTEXT *statement*}
{ALTER VIEW *statement*}
{COMMENT *statement*}

{COMMIT *statement*}
{CONNECT *statement*}
{CREATE ALIAS *statement*}
{CREATE DATABASE *statement*}
{CREATE FUNCTION *statement* (external scalar,
external table, or sourced}
{CREATE GLOBAL TEMPORARY TABLE *statement*}
{CREATE INDEX *statement*}
{CREATE PROCEDURE *statement*}
{CREATE ROLE *statement*}
{CREATE SEQUENCE *statement*}
{CREATE STOGROUP *statement*}
{CREATE SYNONYM *statement*}
{CREATE TABLE *statement*}
{CREATE TABLESPACE *statement*}
{CREATE TRUSTED CONTEXT *statement*}
{CREATE TYPE *statement*}
{CREATE VIEW *statement*}
{DECLARE GLOBAL TEMPORARY TABLE *statement*}
{DELETE *statement*}
{DROP *statement*}
{EXCHANGE *statement*}
{EXECUTE IMMEDIATE *statement*}
{GRANT *statement*}
{INSERT *statement*}
{LABEL *statement*}
{LOCK TABLE *statement*}
{MERGE *statement*}
{REFRESH TABLE *statement*}
{RELEASE *statement*}
{RELEASE SAVEPOINT *statement*}
{RENAME *statement*}
{REVOKE *statement*}
{ROLLBACK *statement*}
{SAVEPOINT *statement*}
{SELECT INTO *statement*}
{SET CONNECTION *statement*}
{SET *special-register statement*}
{TRUNCATE *statement*}
{UPDATE *statement*}
{VALUES INTO *statement*}

Note: The following statements are not permitted in an *SQL-routine-body:* ALTER FUNCTION (SQL scalar) or ALTER PROCEDURE (SQL native) with an ADD VERSION or REPLACE clause.

CREATE ROLE

CREATE ROLE role-name

CREATE SEQUENCE

CREATE SEQUENCE sequence-name
{{ AS [INTEGER | data-type] |
START WITH numeric-constant |
[INCREMENT BY [1 | numeric-constant]] |
[NO MINVALUE | MINVALUE numeric-constant] |
[NO MAXVALUE | MAXVALUE numeric-constant] |
[NO CYCLE | CYCLE] |
[CACHE 20 | NO CACHE | CACHE integer-constant] |
[NO ORDER | ORDER] }}

data-type:
[built-in-type | distinct-type-name]
built-in-type:
[[SMALLINT | [INTEGER | INT] | BIGINT] |
[DECIMAL | DEC | NUMERIC] {(5,0) | (integer {, integer})}]

CREATE STOGROUP

CREATE STOGROUP *stogroup-name*
{VOLUMES ([[[*volume-id,*]] | [['*',]]])}
VCAT *catalog-name*
{DATACLAS *dc-name* } {MGMTCLAS *mc-name* }
{STORCLAS *sc-name* }

Note: Do not specify the same volume-id more than once.

CREATE SYNONYM (deprecated)

CREATE SYNONYM synonym
FOR auth-id.[table-name | view-name]

CREATE TABLE

CREATE TABLE table-name
[([*column-definition* | *period-definition* |
unique-constraint | *referential-constraint* |
check-constraint],) |
LIKE [table-name | view-name] { copy-options} |
as-result-table {copy-options} |
materialized-query-definition]
{ [IN [{dbname.} tsname | DATABASE dbname] }
{partitioning-clause}
{organization-clause}
{EDITPROC program-name
{ [WITH|WITHOUT] ROW ATTRIBUTES } }
{VALIDPROC program-name }
{AUDIT [NONE|CHANGES|ALL]} {OBID integer}
{DATA CAPTURE [NONE|CHANGES]}
{WITH RESTRICT ON DROP}
{CCSID [ASCII | EBCDIC | UNICODE]
{ {NOT} VOLATILE {CARDINALITY} }
{LOGGED|NOT LOGGED}
{COMPRESS [NO|YES]}
{APPEND [NO|YES] }
{DSSIZE integer G}
{BUFFERPOOL bpname}
{MEMBER CLUSTER}
{TRACKMOD [YES|NO]}
{PAGENUM [ABSOLUTE|RELATIVE]}

column-definition:
column-name *data-type*
{ NOT NULL } {*generated-column-definition*}
{column-constraint}
{ {WITH} DEFAULT {constant | USER | SESSION_USER |
CURRENT SQLID | NULL |
cast-function-name ([constant | [USER |
SESSION_USER] | CURRENT SQLID | NULL])} }|
{FIELDPROC program-name {(constant,)}
{AS SECURITY LABEL} {IMPLICITLY HIDDEN }
{INLINE LENGTH integer }

data-type:
built-in-type | *distinct-type-name*

built-in-type:

[SMALLINT | [INTEGER | INT] | BIGINT] |
[DECIMAL | DEC | NUMERIC {(integer{ ,integer }) |
[FLOAT{(53) | (integer)} | REAL | DOUBLE{ PRECISION }] |
DECFLOAT {16 | 34} |
[[CHARACTER | CHAR] {((1)| (integer)} |
[[CHARACTER | CHAR] VARYING | VARCHAR](integer)]
{FOR [SBCS | MIXED | BIT] DATA}]| |
[[CHARACTER | CHAR] LARGE OBJECT | CLOB)]]
{(1M) | (integer{K|M|G}) {FOR [SBCS|MIXED] DATA]}] |
[GRAPHIC {(1) | (integer)} | VARGRAPHIC (integer) |
DBCLOB {(1M) | (integer{K | M | G})}] |
[BINARY {(1)|(integer)}| [BINARY VARYING|VARBINARY]
{(integer)}|
[BINARY LARGE OBJECT|BLOB] {(1)|(integer{K|M|G })}] |
[DATE|TIME|TIMESTAMP {(6) | (integer)}
{[WITHOUT| WITH] TIME ZONE] |
ROWID | XML (*XML-type-modifier*)]

XML-type-modifier:

XMLSCHEMA [[*XML-schema-specification*
{ELEMENT element-name} ,]]

XML-schema-specification:

[ID registered-XML-schema-name |
[URL target-namespace | NO NAMESPACE]
{LOCATION schema-location }]

generated-column-definition:

[GENERATED {ALWAYS|BY DEFAULT}
{*as-identity-clause* | *as-row-change-timestamp-clause* } |
GENERATED {ALWAYS}
{*as-row-transaction-timestamp-clause* |
as-row-transaction-start-id-clause}]

as-identity-clause: see ALTER TABLE

as-row-change-timestamp-clause:

FOR EACH ROW ON UPDATE AS ROW CHANGE TIMESTAMP

as-row-transaction-timestamp-clause:

AS ROW ON [BEGIN|END]

as-row-transaction-start-id-clause:

AS TRANSACTION START ID

column-constraint:

{CONSTRAINT constraint-name] [PRIMARY KEY|UNIQUE |
references-clause | CHECK(check-condition)]

period-definition:

{PERIOD
{[[SYSTEM_TIME] (begin-column-name , end-column-name)] |
[[BUSINESS_TIME] (begin-column-name , end-column-name
[EXCLUSIVE | INCLUSIVE])]}}

unique-constraint:

{CONSTRAINT constraint-name}
[PRIMARY KEY | UNIQUE] ([[column-name,]]
{,BUSINESS_TIME WITHOUT OVERLAPS})

referential-constraint:

{CONSTRAINT constraint-name}
FOREIGN KEY ([[column-name,] {,PERIOD BUSINESS_TIME}])
references-clause

references-clause:
REFERENCES table-name {(column-name,)}
{ON DELETE [RESTRICT|CASCADE|SET NULL| NO ACTION]|
{ENFORCED | NOT ENFORCED}
{ENABLE QUERY OPTIMIZATION}

check-constraint:
{CONSTRAINT constraint-name}
CHECK (*check-condition*)

as-result-table:
{(column-name,)} AS (fullselect) WITH NO DATA

copy-options:
[EXCLUDING|INCLUDING] IDENTITY
{COLUMN ATRIBUTES} |
[EXCLUDING|INCLUDING]
ROW CHANGE TIMESTAMP { COLUMN ATTRIBUTES } |
{ [EXCLUDING|INCLUDING] { COLUMN } DEFAULTS } |
EXCLUDING XML TYPE MODIFIERS]

partitioning-clause: see ALTER TABLE

organization-clause:
ORGANIZE BY HASH UNIQUE (column-name ,)
{ HASH SPACE [64 M | integer [K|M|G] }

materialized-query-definition: see ALTER TABLE

refreshable-table-options:
DATA INITIALLY DEFERRED REFRESH DEFERRED
{ MAINTAINED BY [SYSTEM|USER] }
{ [ENABLE|DISABLE] QUERY OPTIMIZATION }

CREATE TABLESPACE

CREATE {LARGE|LOB} TABLESPACE tsname
{IN [DSNDB04 | dbname]}
{*using-block*} {*free-block*} {*gbpcache-block*}
{ {DEFINE [YES|NO] } {LOGGED|NOT LOGGED}
{TRACKMOD [YES|NO]} {DSSIZE *integer* G}
{*partition-by-growth-specification*|
partition-by-range-specification|
SEGSIZE *integer*}
{BUFFERPOOL *bpname*}
{CCSID [ASCII|EBCDIC|UNICODE]} {CLOSE [YES|NO]}
{COMPRESS [NO|YES]}
{LOCKSIZE [ANY|TABLESPACE|TABLE|PAGE|ROW|LOB]}
{INSERT ALGORITHM [0|level]}
{LOCKMAX [SYSTEM|*integer*] {MAXROWS *integer*}
{ SEGSIZE *integer* }
{PAGENUM [ABSOLUTE|RELATIVE]}

using-block:
USING [VCAT *catalog-name* | STOGROUP *stogroup-name*
{PRIQTY *integer* } {SECQTY *integer* } {ERASE [NO|YES]}]

free-block:
{FREEPAGE [0|*integer*]}
{PCTFREE [5 |smallint | smallint FOR UPDATE smallint]}

gbpcache-block:
GBPCACHE [CHANGED|ALL|SYSTEM|NONE]

partition-by-growth-specification:
MAXPARTITIONS *integer* {MEMBER CLUSTER}
{NUMPARTS *integer*}

partition-by-range-specification:

{NUMPARTS *integer* {MEMBER CLUSTER}
{ ({{ PARTITION *integer* {*using-block*}
{*free-block*} {*gbcache-block*} {COMPRESS [NO|YES}
{TRACKMOD [YES|NO]} ,}})}
{MEMBER CLUSTER}

CREATE TRIGGER (advanced)

CREATE TRIGGER *trigger-name*
{[VERSION [V1 | trigger_version_id]
[NO CASCADE BEFORE|AFTER|INSTEAD OF]
[INSERT|DELETE|UPDATE {OF *column-name*,}]
ON [*able-name*|*view-name*]
{REFERENCING [OLD {AS} *correlation-name* |
NEW {AS} *correlation-name* |
OLD_TABLE {AS} *identifier* |
NEW_TABLE {AS} *identifier*]
[FOR EACH ROW | FOR EACH STATEMENT]
{NOT SECURED | SECURED} {*trigger-action*}}
[WRAPPED *obfuscated-statement-text*]

trigger-action:

{WHEN (*search-condition*)} [SQL-trigger-body]

SQL-trigger-body

[*triggered-SQL-statement* |
BEGIN ATOMIC [[triggered-SQL-statement;]] END]

CREATE TRIGGER (basic)

CREATE TRIGGER *trigger-name*
{[NO CASCADE BEFORE | AFTER | INSTEAD OF]
[INSERT|DELETE|UPDATE {OF *column-name*,}]
ON [*able-name*|*view-name*]
{REFERENCING [OLD {AS} *correlation-name* |
NEW {AS} *correlation-name* | OLD_TABLE {AS} *identifier*
|
NEW_TABLE {AS} *identifier*]
[FOR EACH ROW | FOR EACH STATEMENT]
MODE DB2SQL {NOT SECURED|SECURED} {*trigger-action*}}
[WRAPPED *obfuscated-statement-text*]

trigger-action:

{WHEN (search-condition)} [SQL-trigger-body]

SQL-trigger-body

[*triggered-SQL-statement* |
BEGIN ATOMIC [[triggered-SQL-statement;]] END]

CREATE TRUSTED CONTEXT

CREATE TRUSTED CONTEXT *context-name*
BASED UPON CONNECTION USING SYSTEM AUTHID *auth-name*
{NO DEFAULT ROLE | DEFAULT ROLE *role-name*} {WITHOUT ROLE AS OBJECT OWNER | WITH ROLE AS OBJECT OWNER AND QUALIFIER} {ENABLE|DISABLE}
{NO DEFAULT SECURITY LABEL |
DEFAULT SECURITY LABEL *seclabel-name*}
ATTRIBUTES ([[ADDRESS *address-value* , |
ENCRYPTION *encryption-value* |
SERVAUTH *servauth-value* , |
JOBNAME *jobname-value*]])
{WITH USE FOR [[*auth-name* {*user-options*} |
EXTERNAL SECURITY PROFILE *profile-name* |
PUBLIC [WITHOUT|WITH] AUTHENTICATION,]] }

user-options:
{ROLE *role-name*}
{SECURITY LABEL *seclabel-name*}
{[WITHOUT|WITH] AUTHENTICATION}

built-in-type:
[SMALLINT|[INTEGER|INT]|BIGINT]|
[DEC|NUMERIC |DECIMAL] {(5,0) | (*integer* {, *integer*})] |
[FLOAT {(53)|*integer*)}|REAL|DOUBLE {PRECISION }] |
DECFLOAT (34)|(16) |
[[CHARACTER|CHAR] {(1)|(*integer*)} |
[[[CHARACTER|CHAR] VARYING] | VARCHAR] (*integer*)]
{CCSID [EBCDIC|ASCII|UNICODE]}] |
{ FOR [SBCS|MIXED|BIT] DATA] } |
[[CHARACTER|CHAR] LARGE OBJECT | CLOB]
{(1M) | (*integer* {K|M|G})
{CCSID [EBCDIC|ASCII|UNICODE]} |
{FOR [SBCS|MIXED] DATA}
{GRAPHIC [(1)|] |VARGRAPHIC (*integer*) |
DBCLOB (1M | (integer [K|M|G)}
[BINARY {(1)|*integer*} | [BINARY VARYING|VARBINARY]
(*integer*) | [BINARY LARGE OBJECT|BLOB]
{ (1M)|(*integer* {K|M|G}) } |
[DATE|TIME| TIMESTAMP {(6)|(*integer*) }
{[WITHOUT|WITH} TIME ZONE}

data-type2:
[INTEGER|INT] | [VARCHAR] (*integer*)]
{CCSID [EBCDIC|ASCII|UNICODE]}] |
{ FOR [SBCS|MIXED] DATA | BIT] }
[[CHARACTER|CHAR] VARYING]

CREATE TYPE (distinct)

CREATE TYPE *distinct-type-name* AS *source-data-type*
{INLINE LENGTH *integer*}

Note: INLINE LENGTH is only valid for a LOB source-data-type.

source-data-type:
[SMALLINT|[INTEGER|INT]|BIGINT]|
[DECIMAL|DEC|NUMERIC] {(5,0) | (*integer* {, *integer* })] |
[FLOAT {(53)|*integer*)}|REAL|DOUBLE {PRECISION }] |
DECFLOAT (34)|(16) |
[[CHARACTER|CHAR] {(1)|(*integer*)} |
[[[CHARACTER|CHAR] VARYING] | VARCHAR] (*integer*)]
{CCSID [EBCDIC|ASCII|UNICODE]}] |
{ FOR [SBCS|MIXED] DATA|BIT] }
[[CHARACTER|CHAR] LARGE OBJECT | CLOB]
{(1M) | (*integer* {K|M|G}) {FOR [SBCS|MIXED] DATA}
{CCSID [EBCDIC|ASCII|UNICODE]} |
[GRAPHIC {(*integer*)} | VARGRAPHIC (*integer*) |
DBCLOB [(1M) | (*integer* [K|M|G]) |
{CCSID [EBCDIC|ASCII|UNICODE]} |
[BINARY {(1)|*integer*} |
[BINARY VARYING|VARBINARY] (*integer*) |
[BINARY LARGE OBJECT|BLOB] { (1M)|(*integer* {K|M|G}) }|
[DATE|TIME| TIMESTAMP {(6)|(*integer*) }
{[WITHOUT|WITH} TIME ZONE} | ROWID

CREATE VARIABLE

CREATE VARIABLE *variable-name data-type*
{DEFAULT [NULL|*constant*|*special-register*]

data-type:
built-in-type | *array-type-name*
see CREATE TABLE

CREATE VIEW

CREATE VIEW *view-name* {(*column-name*,)}
AS {WITH *common-table-expression*,} *fullselect*
{WITH [CASCADED|LOCAL] CHECK OPTION}

DECLARE CURSOR

DECLARE *cursor-name*
{NO SCROLL|
{ASENSITIVE|INSENSITIVE |
SENSITIVE {DYNAMIC|STATIC} } SCROLL}
CURSOR
{ [WITHOUT|WITH] HOLD }
{ [WITHOUT RETURN|WITH RETURN {TO [CALLER|CLIENT}
{ [WITHOUT|WITH] ROWSET POSITIONING } |
FOR [*select-statement* | *statement-name*]

DECLARE GLOBAL TEMPORARY TABLE

DECLARE GLOBAL TEMPORARY TABLE *table-name*
[(*column-definition*,) | [LIKE [*table-name*|*view-name*] |
as-result-table] *copy-options*]
{CCSID [ASCII|EBCDIC|UNICODE] }
{ ON COMMIT [DELETE ROWS|PRESERVE ROWS|
DROP TABLE]
{LOGGED | NOT LOGGED [ON ROLLBACK DELETE ROWS|
ON ROLLBACK PRESERVE ROWS]}

column-definition: see CREATE TABLE.

as-result-table:

AS (fullselect) WITH NO DATA

copy-options:

{[EXCLUDING | INCLUDING] IDENTITY {COLUMN ATTRIBUTES}
{ [EXCLUDING { COLUMN } | INCLUDING {COLUMN} | USING TYPE }] DEFAULTS }

identity-options:

AS IDENTITY {({{ START WITH [1|*numeric-constant*] |
INCREMENT BY [1 | *numeric-constant*] |
[NO MINVALUE | MINVALUE *numeric-constant*] |
[NO MAXVALUE | MAXVALUE *numeric-constant*] |
[NOCYCLE | CYCLE] |
[NO CACHE|CACHE 20|CACHE integer-constant] }},)}

DECLARE STATEMENT

DECLARE *statement-name*, STATEMENT

DECLARE TABLE

DECLARE [*table-name*|*view-name*]
TABLE *column-name*
[*built-in-type* | *distinct-type-name*] |
{NOT NULL|NOT NULL WITH DEFAULT}

built-in-type:

see CREATE TABLE.

DECLARE VARIABLE

DECLARE *host-variable*,
VARIABLE { CCSID *integer-constant* |
{CCSID [EBCDIC|ASCII|UNICODE] }
{FOR[SBCS DATA|MIXED DATA|BIT DATA] }}

DELETE (searched)

DELETE FROM [*table-name*|*view-name*] {*period-clause*}
{*correlation-name*} {*include-column*}
SET {*assignment-clause*}
{WHERE *search-condition*} [*fetch-clause*]
{*isolation clause*} | SKIP LOCKED DATA]}
{QUERYNO *integer*}

Notes: When specifying period-clause, do not specify fetch-clause. Do not duplicate the same clause.

period-clause:

FOR PORTION OF BUSINESS_TIME
{FROM value1 to value2 | BETWEEN value1 and value2}

include-column:

INCLUDE (*column-name* {data type} ,)

data-type:

{built-in-type} | [*distinct-type*] (see CREATE TABLE)

assignment-clause:

[[*column-name* = [*expression*|NULL] |
(*column-name*,) = ([[[*expression*|NULL]] | *row-fullselect*])

isolation-clause:

[WITH [RR|CS|RS]

DELETE (cursor-positioned)

DELETE FROM [*table-name*|*view-name*]
{*correlation-name*} WHERE CURRENT OF *cursor-name*
{ FOR ROW [*host-variable*|*integer-constant*] OF ROWSET}

DESCRIBE CURSOR

DESCRIBE CURSOR [*cursor-name*|*host-variable*]
INTO *descriptor-name*

DESCRIBE INPUT

DESCRIBE INPUT *statement-name* INTO *descriptor-name*

DESCRIBE OUTPUT

DESCRIBE {OUTPUT} *statement-name*
INTO *descriptor-name* {USING [NAMES|LABELS|ANY|BOTH]}

DESCRIBE PROCEDURE

DESCRIBE PROCEDURE [*procedure-name*|*host-variable*]
INTO *descriptor-name*

DESCRIBE TABLE

DESCRIBE TABLE *host-variable* INTO *descriptor-name*
{USING [NAMES|LABELS|ANY|BOTH]}

DROP

DROP {*alias-designator*} |
[DATABASE *database-name* |
[FUNCTION *function-name* {(parameter-type,)} {RESTRICT} |
SPECIFIC FUNCTION *specific name* {RESTRICT}] |
INDEX *index-name* | MASK *mask-name* |
PACKAGE *collection-id.package-name* {{VERSION} *version-id*}
|
PERMISSION *permission-name* |
PROCEDURE *procedure-name* {RESTRICT} |
ROLE *role-name* |SEQUENCE *sequence-name* {RESTRICT} |
STOGROUP *stogroup-name* | SYNONYM *synonym-name* |
TABLE [*table-name*|*alias-name*] |
TABLESPACE {*database-name.*} *tablespacename* |
TRIGGER *trigger-name*| TRUSTED CONTEXT *context-name*|
TYPE *type-name* {RESTRICT} |
VARIABLE variable-name {RESTRICT} |
VIEW [*view-name*|*alias-name*]]]

alias-designator:
[PUBLIC] {ALIAS *alias-name*} [FOR TABLE|FOR SEQUENCE]

parameter-type:
data-type {AS LOCATOR}

data type:
built-in-type | *distinct-type-name* | *array-type-name*
see CREATE TABLE

END DECLARE SECTION

EXCHANGE
EXCHANGE DATA BETWEEN TABLE table-name1 AND table-name2

EXECUTE
EXECUTE *statement-name*
{USING [*variable*, | *array-variable*[*array-index*] |
DESCRIPTOR *descriptor-name* | *source-row-data*]}

source-row-data:
[USING [[*host-variable-array* | *host-variable*,| DESCRIPTOR *descriptor-name*]]
{FOR [*host-variable*|*integer-constant*] ROWS }

EXECUTE IMMEDIATE
EXECUTE IMMEDIATE [*host-variable*|*string-expression*]

EXPLAIN
EXPLAIN [PLAN|ALL] {SET QUERYNO=*integer*}
FOR *explainable-sql-statement* |
STMTCACHE [ALL|STMTID [*id-host-variable*|*integer-constant*]|
STMTTOKEN [*token-host-variable*|*string-constant*] |
PACKAGE *package-scope specifications*]
STABILIZED DYNAMIC QUERY STMTID
[*id-host-variable*|*integer-constant*]
COPY 'CURRENT'| COPY 'INVALID'

package-scope-specification:
COLLECTION *collection-name* PACKAGE *package-name*
{VERSION *version-name*} {COPY *copy-id*}

FETCH
FETCH {INSENSITIVE|SENSITIVE} {WITH CONTINUE}
fetch-orientation {*FROM*} *cursor-name*
{*single-row-fetch*|*multiple-row-fetch*}

fetch-orientation:
[BEFORE|AFTER|*row-positioned*|*rowset-positioned*]
row-positioned:
{ NEXT|PRIOR|FIRST|LAST|CURRENT {CONTINUE} |
ABSOLUTE [*host-variable*|*integer-constant*] |
RELATIVE [*host-variable*|*integer-constant*] }

rowset-positioned:
[NEXT ROWSET | PRIOR ROWSET | FIRST ROWSET |
LAST ROWSET | CURRENT ROWSET |
ROWSET STARTING AT [ABSOLUTE|RELATIVE]
[*host-variable*|*integer-constant*]]

single-row-fetch:
INTO [*target-variable*, | *array-variable*[*array-index*]|
DESCRIPTOR *descriptor-name*]
target-variable:
global-variable-name
host-variable-name
SQL-parameter-name
SQL-variable-name
transition-variable-name

multiple-row-fetch:
FOR [*host-variable*|*integer-constant*] ROWS
{INTO [*host-variable-array*,|DESCRIPTOR *descriptor-name*]

FREE LOCATOR

FREE LOCATOR [[*host-variable*,]]

GET DIAGNOSTICS

GET {CURRENT|STACKED} DIAGNOSTICS
[*statement-information* | *condition-information* | *combined-information*]

statement-information:
[[*variable1* = [*statement-information-item-name*], |
DB2_GET_DIAGNOSTICS_DIAGNOSTICS |
DB2_SQL_NESTING_LEVEL]]
statement-information-item-name:
[[DB2_LAST_ROW | DB2_NUMBER_PARAMETER_MARKERS
| DB2_NUMBER_RESULT_SETS | DB2_NUMBER_ROWS |
DB2_RETURN_STATUS | DB2_SQL_ATTR_CURSOR_HOLD |
DB2_SQL_ATTR_CURSOR_ROWSET |
DB2_SQL_ATTR_CURSOR_SCROLLABLE |
DB2_SQL_ATTR_CURSOR_SENSITIVITY |
DB2_SQL_ATTR_CURSOR_TYPE | MORE | NUMBER |
ROW_COUNT,]]

condition-information:
CONDITION [*variable2* | *integer*]
[*variable3* = [[*condition-information-item-name* |
connection-information-item-name ,]]]
condition-information-item-name:
[CATALOG_NAME | CONDITION_NUMBER |
CURSOR_NAME | DB2_ERROR_CODE1 |
DB2_ERROR_CODE2 | DB2_ERROR_CODE3 |
DB2_ERROR_CODE4 | DB2_INTERNAL_ERROR_POINTER |
DB2_LINE_NUMBER | DB2_MESSAGE_ID |
DB2_MODULE_DETECTING_ERROR |
DB2_ORDINAL_TOKEN_*n* | DB2_REASON_CODE |
DB2_RETURNED_SQLCODE | DB2_ROW_NUMBER |
DB2_SQLERRD_SET | DB2_SQLERRD1 | DB2_SQLERRD2 |
DB2_SQLERRD3 | DB2_SQLERRD4 | DB2_SQLERRD5 |
DB2_SQLERRD6 | DB2_TOKEN_COUNT | MESSAGE_TEXT |
RETURNED_SQLSTATE | SERVER_NAME]
connection-information-item-name:
[DB2_AUTHENTICATION_TYPE |DB2_AUTHORIZATION_ID|
DB2_CONNECTION_STATE | DB2_CONNECTION_STATUS |
DB2_ENCRYPTION_TYPE |
DB2_SERVER_CLASS_NAME |
DB2_PRODUCT_ID]

combined-information:
variable4 = ALL [STATEMENT | {{ [CONDITION | CONNECTION]
{ *variable5* | *integer* } ,}}]

GRANT (authorization)
GRANT *authorization-specification*
TO [*auth-name* | ROLE *role-name* | PUBLIC,]
WITH GRANT OPTION

GRANT (collection privileges)
GRANT [CREATE|PACKADM] [ON|IN] COLLECTION
[*collection-id*,| *] TO [*auth-name* | ROLE *role-name* |
PUBLIC,] {WITH GRANT OPTION}

GRANT (database privileges)
GRANT [[DBADM | DBCTRL | DBMAINT | CREATETAB |
CREATETS | DISPLAYDB | DROP | IMAGCOPY | LOAD |
RECOVERDB | REORG | REPAIR | STARTDB | STATS |
STOPDB,]] ON DATABASE *dbname*, TO [*auth-name* | ROLE
role-name | PUBLIC,]
{WITH GRANT OPTION}

GRANT (function or procedure privileges)
GRANT EXECUTE ON [FUNCTION
[*function name* {(*parameter-type*,)}, | *] |
SPECIFIC FUNCTION *specific-name*, |
PROCEDURE [*procedure-name*, | *****]
TO [*auth-name* | ROLE *role-name* | PUBLIC,] {WITH GRANT
OPTION}

parameter-type:
data-type {AS LOCATOR}

data-type:
[built-in-type] | distinct-type-name | array-type-name
See CREATE TABLE

GRANT (package privileges)
GRANT [ALL|[BIND|COPY| [EXECUTE | RUN] ,]]
ON PACKAGE [*collection-id*.[* | *package-name*] ,]
TO [*auth-name* | ROLE role-name | PUBLIC,]
{WITH GRANT OPTION}

GRANT (plan privileges)
GRANT [BIND|EXECUTE,] ON PLAN *plan-name*,
TO [*auth-name* | ROLE *role-name* | PUBLIC,]
{WITH GRANT OPTION}

GRANT (schema privileges)
GRANT [ALTERIN|CREATEIN|DROPIN,]
ON SCHEMA [*schema-name*, | *]
TO [*auth-name* | ROLE *role-name* | PUBLIC,]
{WITH GRANT OPTION}

GRANT (sequence privileges)
GRANT [[ALTER|USAGE ,]] ON SEQUENCE *sequence-name*,
TO [*auth-name* | ROLE *role-name* | PUBLIC,]
{WITH GRANT OPTION}

GRANT (system privileges)

GRANT [ACCESSCTRL | ARCHIVE | BINDADD | BINDAGENT| BSDS | CREATEALIAS | CREATEDBA | CREATEDBC | CREATESG | CREATETMTAB | CREATE_SECURE_OBJECT | DATAACCESS | DBADM {[WITH|WITHOUT] ACCESSCTRL} {[WITH|WITHOUT] DATACCESS} | DEBUGSESSION | DISPLAY | EXPLAIN | MONITOR1 | MONITOR2 | RECOVER | SQLADM | STOPALL | STOSPACE | SYSADM | SYSCTRL | SYSOPR | TRACE,] TO [*auth-name* | ROLE *role-name* | PUBLIC,] {WITH GRANT OPTION}

GRANT (table or view privileges)

GRANT [ALL {PRIVILEGES} | [ALTER | DELETE | INDEX | INSERT | REFERENCES {(*column-name*,)} | SELECT | TRIGGER | UNLOAD | UPDATE{(*column-name*,)}]
ON {TABLE} [*table-name*|*view-name*,]
TO [*auth-name* | ROLE *role-name* | PUBLIC,]
{WITH GRANT OPTION}

GRANT (type or JAR file privileges)

GRANT USAGE ON [TYPE *type-name*, | JAR *jar-name*,]
TO [*auth-name* | ROLE *role-name* | PUBLIC ,]
{WITH GRANT OPTION}

GRANT (use privileges)

GRANT USE OF [BUFFERPOOL *bpname*, |
ALL BUFFERPOOLS | STOGROUP *stogroup-name*, |
TABLESPACE [{*dbname.*}*tsname*,]]
TO [*auth-name* | ROLE *role-name* | PUBLIC,]
{WITH GRANT OPTION}

GRANT (variable privileges)

GRANT [ALL {PRIVILEGES} | [READ | WRITE,]]
ON VARIABLE *variable-name*
TO [*auth-name* | ROLE *role-name* | PUBLIC ,]
{WITH GRANT OPTION}

HOLD LOCATOR

HOLD LOCATOR [[*host-variable*,]]

INCLUDE

INCLUDE [SQLCA | SQLDA | *member-name*]

INSERT

INSERT INTO [*table-name*|*view-name*] { (*column-name*,) }
{*include-column*} { OVERRIDING USER VALUE }
[VALUES [*expression* | DEFAULT | NULL, |
([*expression* | DEFAULT | NULL ,])] |
{WITH *common-table-expression*, } *fullselect*
{isolation clause} { QUERYNO integer } |
{ *multiple-row-insert* }]

include-column:

INCLUDE (*column-name* | *data-type* ,)
data-type:
built-in-type | distinct-type (see CREATE TABLE)

isolation-clause:

{WITH [RR | RS | CS] }

multiple-row-insert:

VALUES [*expression* | *host-variable-array* | NULL | DEFAULT]
| (*expression* | *host-variable-array* | NULL | DEFAULT,)
{FOR [*host-variable* | *integer-constant*] ROWS}
{ ATOMIC | NOT ATOMIC CONTINUE ON SQLEXCEPTION }

LABEL

LABEL ON [[TABLE [*table-name*|*view-name*] |
ALIAS *alias-name* | COLUMN [*table-name.column-name* |
view-name.column-name]]
IS *string-constant* |
[*table-name* | *view-name*] (*column-name* IS *string-constant*,
)]

LOCK TABLE

LOCK TABLE *table-name* {PARTITION *integer*}
IN [SHARE|EXCLUSIVE] MODE

MERGE

MERGE INTO [*table-name*|*view-name*] | *correlation-clause* |
{*include-columns*} USING *table-reference* | *source-values*
ON *search-condition*
[WHEN | *matching-condition* |
THEN *modification-operation*|*signal-statement*
[ELSE IGNORE]]
{NOT ATOMIC CONTINUE ON SQLEXCEPTION}
{QUERYNO *integer*}

correlation-clause:

{{AS} *correlation-name,column-name*}

include-columns:

INCLUDE (*column-name* | *data-type*,)
data-type:
built-in-type | *distinct-type* (see CREATE TABLE)

source-values:

(VALUES [*values-single-row*|*values-multiple-row*])
{AS} correlation-name (column-name ,)

values-single-row:

[[expression | NULL] | ([[expression | NULL]] ,)]

values-multiple-row:

[[expression | host-variable-array | NULL] |
([[expression | host-variable-array | NULL]] ,)]
FOR [host-variable | integer-constant] ROWS

matching-condition

[NOT] {MATCHED} [AND *search-condition*]

modification-operation:

[UPDATE SET *assignment-clause* | DELETE | *insert-operation*]

assignment-clause:

[[*column-name* = [*expression*|DEFAULT|NULL] |
(*column-name*,) = ([[[*expression*|DEFAULT|NULL]] |
row-fullselect])

insert-operation:

INSERT {(*column-name* ,)}
VALUES [*expression*|DEFAULT |NULL] |
([[*expression* | DEFAULT | NULL ,]])

OPEN

OPEN cursor-name {USING [*variable*, | *array-variable*[*array-index*] |
DESCRIPTOR descriptor-name]}

PREPARE

PREPARE *statement-name* {INTO *descriptor-name*
{ USING [NAMES|LABELS|ANY|BOTH]}}
[FROM [*string-expression* |
{ATTRIBUTES *attr-host-variable*} FROM *variable*]

attr-host-variable must be a valid attribute-string. string-expression is supported for PLI only.

attribute-string:

[[[ASENSITIVE | INSENSITIVE | SENSITIVE { DYNAMIC }
STATIC] |
[NO SCROLL|SCROLL] | [*holdability*] | [*returnability*] |
[*rowset-positioning*] |{offset-clause} {fetch-first-clause }
{ [read-only-clause | update-clause] }
{ optimize-clause } { isolation-clause |
FOR [MULTIPLE ROWS | SINGLE ROW] |
[ATOMIC | NOT ATOMIC CONTINUE ON SQLEXCEPTION] |
concurrent-access-resolution |
[WITHOUT | WITH] EXTENDED INDICATORS |
CONCENTRATE STATEMENTS [OFF | WITH LITERALS]]]

holdability:

[WITHOUT|WITH] HOLD]

returnability:

[WITHOUT | WITH] RETURN { TO CALLER | TO CLIENT}

rowset-positioning:

[WITHOUT | WITH] ROWSET POSITIONING

concurrent-access-resolution:

[SKIP LOCKED DATA | USE CURRENTLY COMMITTED |
WAIT FOR OUTCOME]

REFRESH TABLE

REFRESH TABLE *table-name* {QUERYNO *integer*}

RELEASE (connection)

RELEASE [*location-name*|*host-variable*| CURRENT | ALL
{SQL}]

RELEASE SAVEPOINT

RELEASE {TO} SAVEPOINT *savepoint-name*

RENAME

RENAME [{TABLE} *source-table-name* TO *new-table-identifier* |
INDEX *source-index-name* TO *new-index-identifier*]

REVOKE (authorization privileges)

REVOKE *authorization-specification*
FROM [*auth-name* | ROLE *role-name* | PUBLIC,] |
{BY [[[*auth-name* | ROLE *role-name* ,]] | ALL]}
{ [INCLUDING | NOT INCLUDING] DEPENDENT PRIVILEGES}
{RESTRICT}

REVOKE (collection privileges)

REVOKE [CREATE | PACKADM] [IN|ON]
COLLECTION [*collection-id*, | *]
FROM [[*auth-name* | ROLE *role-name* | PUBLIC,]]
{BY [[[*auth-name* | ROLE *role-name* ,]] | ALL]}
{ [INCLUDING | NOT INCLUDING] DEPENDENT PRIVILEGES}

REVOKE (database privileges)

REVOKE [DBADM | DBCTRL | DBMAINT | CREATETAB |
CREATETS | DISPLAYDB | DROP | IMAGCOPY | LOAD |
RECOVERDB | REORG | REPAIR | STARTDB | STATS |
STOPDB | UNLOAD,]
ON DATABASE *dbname*,
FROM [[*auth-name* | ROLE *role-name* | PUBLIC,]]
{BY [[[*auth-name* | ROLE *role-name*,]] | ALL]}
{ [INCLUDING | NOT INCLUDING] DEPENDENT PRIVILEGES]

REVOKE (function or procedure privileges)

REVOKE EXECUTE ON
[FUNCTION [*function-name*{(*parameter-type*,)}, | *] |
SPECIFIC FUNCTION *specific-name*, |
PROCEDURE [*procedure-name*, | *]]
FROM [*auth-name* | ROLE *role-name* | PUBLIC,]
BY [[*auth-name* | ROLE *role-name* ,]| ALL]
{ [INCLUDING | NOT INCLUDING] DEPENDENT PRIVILEGES }
{RESTRICT}

parameter-type:

data-type {AS LOCATOR}

data-type:

[built-in-type] | distinct-type-name | array-type-name
See CREATE TABLE

REVOKE (package privileges)

REVOKE [ALL | [BIND | COPY | [EXECUTE | RUN ,]] ON
[PACKAGE] [*collection-id*.[* | *package-name*] ,]
FROM [*auth-name* | ROLE *role-name* | PUBLIC ,]
{BY { *auth-name*, | ROLE *role-name* | ALL }}
{ [INCLUDING|NOT INCLUDING] DEPENDENT PRIVILEGES }

REVOKE (plan privileges)

REVOKE [BIND|EXECUTE ,] ON PLAN *plan-name*,
FROM [*auth-name* | ROLE *role-name* | PUBLIC ,]
{BY {*auth-name*, | ROLE *role-name* | ALL}}
{ [INCLUDING|NOT INCLUDING] DEPENDENT PRIVILEGES }

REVOKE (schema privileges)

REVOKE [ALTERIN|CREATEIN|DROPIN ,]
ON SCHEMA [*schema-name*, | *]
FROM [*auth-name* | ROLE *role-name* | PUBLIC ,]
BY { [[*auth-name* | ROLE *role-name* ,]] | ALL }
{ [INCLUDING|NOT INCLUDING] DEPENDENT PRIVILEGES }

REVOKE (sequence privileges):

REVOKE [[ALTER | USAGE,]] ON SEQUENCE *sequence-name*,
FROM [*auth-name* | ROLE *role-name* | PUBLIC ,]
{BY [[*auth-name* | ROLE *role-name* ,]] | ALL}
{ [INCLUDING | NOT INCLUDING] DEPENDENT PRIVILIGES }

REVOKE (system privileges):

REVOKE [[ACCESSCTRL | ARCHIVE | BINDADD | BINDAGENT
|
BSDS | CREATEALIAS | CREATEDBA | CREATEDBC |
CREATESG | CREATETMTAB | CREATE_SECURE_OBJECT |
DATAACCESS |
DBADM | DEBUGSESSION | DISPLAY | EXPLAIN | MONITOR1
|
MONITOR2 | RECOVER | SQLADM | STOPALL |STOSPACE |
SYSADM | SYSCTRL | SYSOPR | TRACE],
FROM [*auth-name* | ROLE *role-name* | PUBLIC ,]
{BY [[*auth-name* | ROLE *role-name* ,] | ALL]}
{ [INCLUDING | NOT INCLUDING] DEPENDENT PRIVILIGES }

REVOKE (table or view privileges)

[REVOKE ALL {PRIVILEGES} | [ALTER | DELETE | INDEX | INSERT |
SELECT | REFERENCES | UNLOAD| UPDATE | TRIGGER ,]]
ON {TABLE} [*table-name*|*view-name* ,]
FROM [*auth-name* | ROLE *role-name* | PUBLIC ,]
{BY [[*auth-name* | ROLE *role-name* ,] | ALL]}
{ [INCLUDING | NOT INCLUDING] DEPENDENT PRIVILIGES }

REVOKE (type or JAR file privileges):

REVOKE USAGE ON [TYPE [[*type-name* ,]] | JAR [[*jar-name* ,]]]
FROM [[*auth-name* | ROLE *role-name* | PUBLIC ,]]
{BY [[[*auth-name* | ROLE *role-name* ,]] | ALL] }
{ [INCLUDING|NOT INCLUDING] DEPENDENT PRIVILEGES}
{ RESTRICT }

REVOKE (use privileges)

REVOKE USE OF [BUFFERPOOL *bpname*, | ALL BUFFERPOOLS |
STOGROUP *stogroup-name*, | TABLESPACE [[{*dbname*.}*tsname* ,]]]
FROM [[*auth-name* | ROLE *role-name* | PUBLIC ,]]
{BY [[[*auth-name* | ROLE *role-name* ,]] | ALL]}
{ [INCLUDING|NOT INCLUDING] DEPENDENT PRIVILEGES }

REVOKE (variable privileges)

REVOKE [ALL {PRIVILEGES} | [READ | WRITE,]]
ON VARIABLE *variable-name*
FROM [*auth-name*|ROLE *role-name* |PUBLIC ,]
{BY [[*auth-name* | ROLE *role-name* ,] | ALL]} {RESTRICT}

ROLLBACK

ROLLBACK {WORK} {TO SAVEPOINT {*savepoint-name*}}

SAVEPOINT

SAVEPOINT *savepoint-name* {UNIQUE}
ON ROLLBACK RETAIN CURSORS { ON ROLLBACK RETAIN LOCKS }

SELECT

WITH [*common-table-expression*], *fullselect*
[*update-clause* | *read-only-clause* | *optimize-clause* | *isolation-clause* | *queryno-clause* | SKIP LOCKED DATA]

SELECT INTO

WITH [*common-table-expression*], *select-clause* INTO
[target-variable, |array-variable[array-index]] |
from-clause | [where-clause] | [group-by-clause] |
[having clause] | [order-by-clause] |
[isolation-clause | [SKIP LOCKED DATA] |
[QUERYNO integer] | [offset-clause]
[FETCH-FIRST [1|ROW|ROWS] ONLY]

target-variable:

global-variable-name
host-variable-name
SQL-parameter-name
SQL-variable-name
transition-variable-name

SET assignment-statement

SET *assignment-clause*

assignment-clause:

array-variable-name [*array-index*] = [*expression*|NULL] | *target-variable*|CURRENT [PACKAGESET|PACKAGE PATH|SERVER] |
[target-variable = [expression|NULL|DEFAULT] | (target-variable) = [expression|NULL|DEFAULT] | row-subselect | VALUES [expression|NULL|DEFAULT] |
(expression|NULL|DEFAULT)

target-variable:

global-variable-name
host-variable-name
SQL-parameter-name
SQL-variable-name
transition-variable-name

SET CONNECTION

SET CONNECTION [*location-name*|*host-variable*]

SET CURRENT APPLICATION COMPATIBILITY

SET CURRENT APPLICATION COMPATIBILITY {=} *string-constant* | *variable*

SET CURRENT APPLICATION ENCODING SCHEME

SET CURRENT {APPLICATION } ENCODING SCHEME {=} [*string-constant* | *host-variable*]

SET CURRENT DEBUG MODE

SET CURRENT DEBUG MODE {=}
[*host-variable*|DISALLOW|ALLOW|DISABLE]

SET CURRENT DECFLOAT ROUNDING MODE

SET CURRENT DECFLOAT ROUNDING MODE {=}
[ROUND_CEILING | ROUND_DOWN | ROUND_FLOOR | ROUND_HALF_DOWN | ROUND_HALF_EVEN | ROUND_HALF_UP |
ROUND_UP | *string-constant* | *host-variable*]

SET CURRENT DEGREE

SET CURRENT DEGREE=[*string-constant*|*host-variable*]

SET CURRENT EXPLAIN MODE

SET CURRENT EXPLAIN MODE {=} [NO|YES|EXPLAIN|*host-variable*]

SET CURRENT GET_ACCEL_ARCHIVE

SET CURRENT GET_ACCEL_ARCHIVE {=} [NO|YES|*host-variable*]

SET CURRENT LOCALE LC_CTYPE

SET [CURRENT { LOCALE } LC_CTYPE | CURRENT_LC_CTYPE] {=} [*string-constant* | *host-variable*]

SET CURRENT MAINTAINED TABLE TYPES FOR OPTIMIZATION

SET CURRENT MAINTAINED {TABLE} TYPES {FOR OPTIMIZATION} {=} [ALL |NONE |SYSTEM | SESSION USER | USER | *host-variable*]

SET CURRENT OPTIMIZATION HINT

SET CURRENT OPTIMIZATION HINT=[*string-constant* | *host-variable*]

SET CURRENT PACKAGE PATH

SET CURRENT PACKAGE PATH {=} [{ SESSION USER | USER | CURRENT PACKAGE PATH | CURRENT PATH ,}
{{ collection-id | host-variable | string-constant ,}} ,]

SET CURRENT PACKAGESET

SET CURRENT PACKAGESET= [SESSION_USER | USER | *string-constant* | host-variable]

SET CURRENT PRECISION

SET CURRENT PRECISION = [*string-constant* | *host-variable*]

SET CURRENT QUERY ACCELERATION

SET CURRENT QUERY ACCELERATION {=} [NONE |ENABLE| ENABLE WITH FAILBACK |ELIGIBLE |ALL |*host-variable*]

SET CURRENT REFRESH AGE

SET CURRENT REFRESH AGE {=} [*numeric-constant* | ANY | *host-variable*]

SET CURRENT ROUTINE VERSION

SET CURRENT ROUTINE VERSION {=} [*routine-version-id* | *host-variable* | *string-constant*]

SET CURRENT RULES

SET CURRENT RULES=[*string-constant* | *host-variable*]

SET CURRENT SQLID

SET CURRENT SQLID=[SESSION_USER | USER | *string-constant* | *host-variable*]

SET CURRENT TEMPORAL BUSINESS_TIME

SET CURRENT TEMPORAL BUSINESS_TIME {=} [NULL|*expression*]

SET CURRENT TEMPORAL SYSTEM_TIME

SET CURRENT TEMPORAL SYSTEM_TIME {=} [NULL|*expression*]

SET ENCRYPTION PASSWORD

SET ENCRYPTION PASSWORD {=} [*password-variable* | *password-string-constant*] {WITH HINT {=} [*hint-variable* | *hint-string-constant*] }

SET PATH

SET {CURRENT} PATH {=} [[*schema-name* | SYSTEM PATH | USER | SESSION_USER | {CURRENT} PATH | CURRENT PACKAGE PATH | *host-variable* | *string-constant* ,]]

SET SCHEMA

SET [{CURRENT} SCHEMA | CURRENT_SCHEMA] {=} [*schema-name* | USER | SESSION_USER | *host-variable* | DEFAULT | *string-constant*]

SET SESSION TIME ZONE

SET SESSION TIME ZONE {=} [*string-constant* | *variable*]

SIGNAL

{ *label* } SIGNAL [SQLSTATE {VALUE} [*sqlstate-string-constant* | *SQL-variable-name* | *SQL-parameter-name*] | *SQL-condition-name*] {*signal-information*}

signal-information:

[SET MESSAGE_TEXT = *diagnostic-string-expression* | *(diagnostic-string-expression*)]

TRANSFER OWNERSHIP

TRANSFER OWNERSHIP OF | *object* | TO | *new-owner* | REVOKE PRIVILEGES

object:

DATABASE *database-name* | INDEX *index-name* | STOGROUP *stogroup-name* | TABLE *table-name* | TABLESPACE {*dbname.tablespace-name*} | VIEW *view-name*

new-owner:

ROLE *role-name* | USER *auth-name* | SESSION_USER

TRUNCATE

TRUNCATE {TABLE} *table name* {[DROP|REUSE] STORAGE}
[IGNORE DELETE TRIGGERS |
RESTRICT WHEN DELETE TRIGGERS] {IMMEDIATE}

UPDATE

(searched)
UPDATE [*table-name*|*view-name*] {*period-clause*}
{*correlation-name*} {*include-column*}
SET *assignment-clause* {WHERE search-condition}
[{*isolation-clause*} | {SKIP LOCKED DATA}]
{QUERYNO *integer*}

cursor-positioned
UPDATE [*table-name*|*view-name*] {*correlation-name*}
SET *assignment-clause* WHERE CURRENT OF *cursor-name*
{ FOR ROW [*host-variable*|*integer-constant*] OF ROWSET }

period-clause:
FOR PORTION OF BUSINESS_TIME
{FROM value1 to value2 | BETWEEN value1 and value2}

include-column:
INCLUDE ([[column-name | data-type ,]])

data-type:
built-in-type | distinct-type (see CREATE TABLE)

assignment clause:
[column-name=[expression | DEFAULT | NULL] |
(column-name,) = ([[[expression| DEFAULT | NULL ,]] |
row fullselect | UNPACK-function-invocation])]

isolation-clause:
WITH [RR|CS|RS]}

VALUES

VALUES [expression | (*expression*,)]

VALUES INTO

VALUES [*expression* | NULL | ([[*expression* | NULL ,]])]
INTO [[*target-variable*,] |
array-variable [array-index]]

target-variable:
global-variable-name
host-variable-name
SQL-parameter-name
SQL-variable-name
transition-variable-name

WHENEVER

WHENEVER [NOT FOUND | SQLERROR | SQLWARNING]
[CONTINUE | [GOTO | GO TO] {:} *host-label*]

Naming Definitions

alias-name:
Qualified or unqualified name designating an alias when preceded by the keyword ALIAS. Otherwise, it designates a table or view and can be local or remote. Rules for naming are same as for *table-name* or *view-name* below. A fully qualified *alias-name* can refer to a server that is not the current server, but the table or view of that alias must be local to that server.

auth-name:
Short identifier designating a set of privileges. It can also designate a user or group of users or a role.

aux-table-name:
Qualified or unqualified name designating an auxiliary table. For auxiliary table name rules see table-name.

bpname:
Identifies a buffer pool. The 4KB buffer pools are named BP0, BP1, BP2-49. The 8K buffer pools are BP8K and BP8K1 through 9, The 16K buffer pools are BP16K, BP16K1-9. The 32K buffer pools are named BP32K and BP32K1 through BP32K9.

built-in-data-type:
Qualified or unqualified name that identifies an IBM-supplied data type.

cast-function-name:
A name that matches the name of the distinct type for the column. Can only be specified for columns defined as distinct type.

catalog-name:
Short identifier designating an ICF catalog.

check-condition:
Statement used to specify table check constraints.

clone-table-name:
A qualified or unqualified name that designates the name of a clone table.

collection-id:
Long identifier identifying a package collection.

column-name:
Qualified or unqualified name designating a column of a table or view. Unqualified, it is a long identifier; qualified, it is a qualifier followed by a period and a long identifier. Qualifier is a table, view, correlation name, synonym, or alias.

constant:
A constant or "literal" specifies a value. There are two types of constants; string constants which are either character or graphic, and numeric constants which are integer, floating-point, or decimal. All constants have the attribute NOT NULL.

constraint-name:
Short identifier designating check or referential constraint on a table.

context_name:
An unqualified SQL identifier that designates a trusted context.

correlation-name:
Long identifier designating a table, view, or individual rows of a table or view.

cursor-name:
Long identifier designating an SQL cursor.

data type:
Every column of a DB2 table has a data type, defining a "set" of data the column will contain and the length of the data field. See Data Types in this section for additional information.

dbname:
Short identifier designating a data base.

distinct-type-name:
Qualified or unqualified name that designates a distinct data type

descriptor-name:
Host-identifier designating an SQLDA; it may be preceded by a colon.

explainable-sql-statement:
Any static or dynamic SQL statement except for a *statement-string*.

external-program-name:
A name that specifies the program that runs when the function is invoked or the procedure name is specified in a CALL statement.

function-name:
Qualified or unqualified name that designates a user-defined function, a cast-function generated when a distinct type was created, or a built-in function.

host-label:
Identifies a host language statement.

host-variable:
Sequence of tokens designating a host variable which includes at least one host-identifier.

index-name:
Qualified or unqualified name of an index. Unqualified, it is a long identifier; in an SQL statement, it is implicitly qualified by the authorization ID of that statement. Qualified, it is a short identifier followed by a period and a long identifier.

large:
A partitioned table space that contains more than 64GB of compressed or uncompressed data.

location-name:
Location identifier designating a data base management system.

mask-name:
A qualified or unqualified name that designates a mask.

member-name:
Name of a member of the partitioned data set.

package-id:
Short identifier designating a package.

package-name:
Name of object created during the bind of a package. Name consists of *location-name*, *collection-id*, *package-id*, and *version-id* separated by periods. Version ID allows multiple versions of a package to have the same first three qualifiers in the name.

parameter-marker:
A question marker (?) that appears where a host variable could appear if the statement were a static SQL statement.

parameter-name:
An SQL identifier that designates a parameter in a procedure or function.

password:
Using DB2 storage groups, password is the control or master integrated catalog password. It protects the data sets and is passed to VSAM when the data sets are used by DB2. Using User Defined data sets, password is defined by you, using VSAM access method services. It is passed to VSAM when the data sets are used by DB2.

permission name:
A qualified or unqualified name that designates a permission.

plan-name:
Short identifier designating an application plan.

procedure-name:
Name designating a user-written program that can be invoked through an SQL CALL statement, i.e. a stored procedure.

profile-name:
An SQL identifier that corresponds to a RACF profile name.

program-name:
Short identifier designating an exit routine or a stored procedure.

result-expression:
Specifies an expression that follows the THEN and ELSE keywords of the CASE expression. ALL result expressions must be compatible.

role-name:
An unqualified SQL identifier that designates a role.

routine-version-id:
An SQL identifier of up to 124 bytes that designates a version of a routine.

rs-loc-var:
Name designating a 4-byte variable used by DB2 to uniquely identify a query result set returned by a stored procedure.

savepoint-name:
An unqualified SQL identifier that designates a savepoint

schema-name:
A schema name or authorization id used to qualify an object. Objects qualified with a schema name are distinct types, stored procedures, triggers, user-defined functions, built-in data-types and built-in functions.

sequence:
A sequence is a stored object that simply generates a sequence of numbers in a monotonically ascending (or descending) order.

server-name:
AN SQL identifier that designates an application server.

specific-name:
Qualified or unqualified name that designates a unique name for a user-defined function.

SQL-condition-name:
An SQL identifier that designates a condition in an SQL function or an SQL procedure.

SQL-label:
An SQL identifier that designates a label in an SQL function or an SQL procedure.

SQL-parameter-name:
An SQL identifier that designates a parameter in the SQL body of an SQL function or an SQL procedure.

SQL-variable-name:
An SQL identifier that designates a variable in an SQL routine body.

statement-name:
Long identifier designating a prepared SQL statement.

statement-string:
The statement string must be one of the following SQL statements: ALTER, INSERT, COMMENT ON, LABEL ON, COMMIT, LOCK TABLE, CREATE, REVOKE, DELETE, ROLLBACK, DROP, SET CURRENT DEGREE, EXPLAIN, SET CURRENT SQLID, GRANT, and UPDATE.

source-table-name:
Qualified or unqualified identifier of an existing table on the server to be renamed.

stogroup-name:
Short identifier designating a storage group.

string constant:
A string constant specifies a varying-length character string. It is represented by a sequence of characters that starts and ends with a string delimiter, either an apostrophe (‘) or a quotation mark (“). The number of bytes between the delimiter must not be greater than 254. An X followed by a sequence of characters that starts and ends with a string delimiter can be used to specify characters that do not have a keyboard representation in hexadecimal notation.

synonym-name:
Long identifier designating a synonym when preceded by the keyword SYNONYM. Otherwise, it designates a local table or local view and can be used wherever the name of a table or view can be used in an SQL statement. A qualified name is never interpreted as a synonym.

table-locator-variable:
A host variable with a table locator type.

table-name:
Qualified or unqualified name that designates a table. Period required between any of the parts. Fully qualified, it has 3 parts; 1st is a *location-name* designating the DBMS at which the table is stored, 2nd is the authorization ID of the owner of the table, 3rd is a long identifier. A 2-part table-name is implicitly qualified by the *location-name* of the current server. A 1-part or unqualified *table-name* is a long identifier with 2 implicit qualifiers; 1st implicit qualifier is the *location-name* of the current server, 2nd is the authorization ID. Authorization ID is owner of plan or package for static statements, or the CURRENT SQLID for dynamic statements.

target-name:
New unqualified name for a table.

trigger-name:
A qualified or unqualified name that designates a trigger.

tsname:

Short identifier designating a table space of an identified data base. If a data base is not identified, a *table-space-name* designates a table space of data base DSNDB04.

version-id:

An identifier up to 64 bytes long assigned to a package when the package is created.

view-name:

Qualified or unqualified name that designates a view. Period required between any of the parts. Fully qualified, it has 3 parts; 1st is a *location-name* designating the DBMS at which the view is stored, 2nd is the authorization ID of the owner of the view, 3rd is a long identifier. A 2-part *view-name* is implicitly qualified by the *location-name* of the current server. A 1-part or unqualified *view-name* is a long identifier with 2 implicit qualifiers: 1st implicit qualifier is the *location-name* of the current server, 2nd is the authorization ID. Authorization ID is owner of plan or package for static statements, or the CURRENT SQLID for dynamic statements.

volume-id:

Label used to direct the placement of objects on specific mass storage devices.

XML-value-expression:

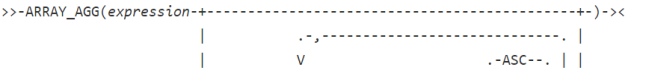
A value of the transient XML data type. The transient XML data type only exists during query processing. There is no persistent data of this type and it is not an external data type that can be declared in application program.

Functions

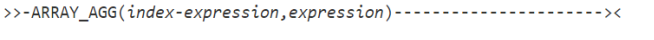
SQL functions include: aggregate, scalar, table, and MQ series table functions.

Aggregate Functions

ARRAY_AGG



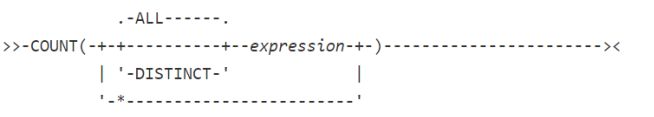
or



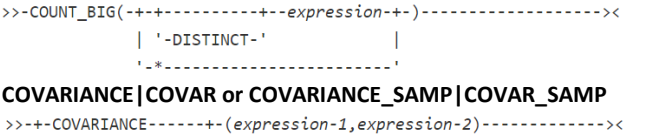
AVG

[CORRELATION|CORR]

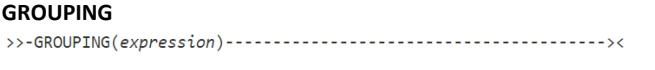
COUNT



COUNT_BIG

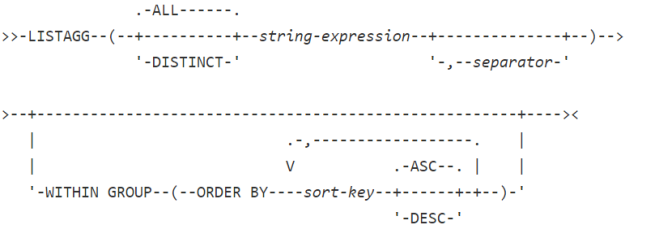


COVARIANCE|COVAR or COVARIANCE_SAMP|COVAR_SAMP

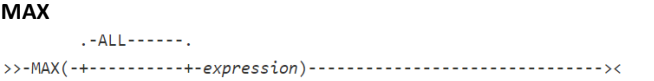


GROUPING

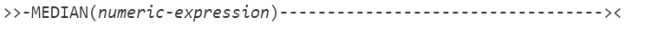
LISTAGG



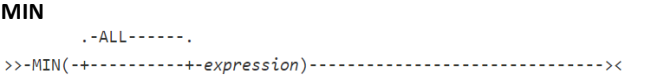
MAX



MEDIAN

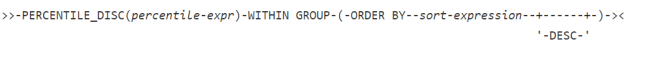


MIN



PERCENTILE_CONT

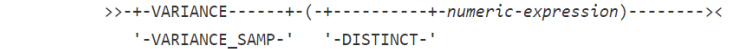
PERCENTILE_DISC



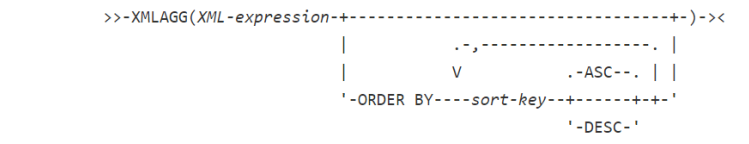
[STDDEV|STDDEV_POP] and STDDEV_SAMP

SUM

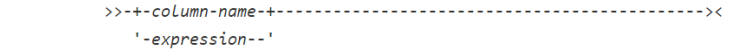
[VARIANCE|VAR|VAR_POP] or [VARIANCE_SAMP|VAR_SAMP]



XMLAGG



sort-key



Scalar Functions

[ABS|ABSVAL] (*numeric-expression*)

- ABS(BONUS) → 1000.00
  (in BONUS format, returning the absolute value of –1000.00.)

ACOS (*numeric-expression*)

- ACOS (ANGLE) → +0.1572969586431309E+00
  (in double precision floating-point format, returning the arcosine of ANGLE, value .98765432. Valid numeric values for expression are –1 to 1.)

ADD_MONTHS (*date-expression,numeric-expression*)

- ADD_MONTHS (HIREDATE,3) → '2016-11-30')
  (*in date format, returning the date plus 3 months; when date-expression is '2016-08-31', and the day is last day of the month, the last day of the resulting month will be returned.*)

ARRAY_DELETE (*array-expression*
[*,array-index1*[*,array-index2*]])

- ARRAY_DELETE(*value*, '*index-value*', '*index-value2*')

ARRAY_FIRST (*array-expression*)

- ARRAY_FIRST(*list*)

ARRAY_LAST (*array-expression*)

- ARRAY_LAST(*list*)

ARRAY_NEXT (*array-expression,array-index*)

- ARRAY_NEXT(*list,'index-value'*)

ARRAY_PRIOR (*array-expression,array-index*)

- ARRAY_PRIOR(*list,'index-value'*)

ASCII (*string-expression*)

- ASCII ('C') → 67
  (*in integer format, returns the leftmost character of the argument, value 67*)

ASCII_CHR (*expression*)

- ASCII_CHR (88) → x
  (*returns a built-in data type of BIGINT, INTEGER, or*

SMALLINT that has the ASCII code value that is specified by the argument
of the argument)

[ASCII_STR | ASCIISTR] *(string-expression)*

- ASCII_STR (FIRST_NAME) → 'TEDDY"
  (*result of the function is a varying-length character string in the system ASCII CCSID*)

ASIN (*numeric-expression*)

- ASIN (ANGLE) → +0.1413499368151766E+01
  (in double precision floating-point format, returning the arcosine of ANGLE, value .98765432. Valid numeric values for expression are –1 to 1.)

ATAN (*numeric-expression*)

- ATAN (ANGLE) → +0.7791870626446206E+00
  (in double precision floating-point format, returning the arc tangent of ANGLE, value .98765432.)

ATANH (*numeric-expression*)

- ATANH (ANGLE) → +0.2540702142243786E+01
  (in double precision floating-point format, returning the hyperbolic arc target in radians of ANGLE value .98765432. Valid numeric values for expression are –1 to 1.)

ATAN2 (*numeric-expression1, numeric-expression2*)

- ATAN2 (XCOORD, YCOORD) →
  +0.110714871779409E+01
  (in double precision floating-point format, returning the arc tangent in radians of XCOORD, value2 and YCOORD, value 4.)

BIGINT (*numeric or string expression*)

- BIGINT (SALARY) → 74588
  (in big integer representation, returns a value of 74588 for a salary of 74588.95)

BINARY (*string-expression* {, *integer*})

- BINARY ('HBK', 3) → BX'C8C2D2
  (*in binary format, returning a value of BX 'C8C2D2' integer must be between 1 and 255*)

[BITAND|BITANDNOT|BITOR|BITXOR]
(*expression,expression2*)

- BITAND (10101010'',''11110000) → '10100000'
  (Performs a bitwise AND operation.)
- BITANDNOT (10101010'',''11110000) → '00001010'
  (Performs a bitwise NOT AND operation)
- BITOR (10101010'',''11110000) → '11111010'
  (Performs a bitwise OR operation
- BITXOR (10101010'',''11110000) → '01011010'
  (Performs a bitwise exclusive OR operation)

BITNOT
(*expression*)
BITXNOT (10101010'') → '01010101'
(Flips the bits operation

BLOB
(*expression* {,*integer*})
- BLOB(LASTNAME) → 'E2D4C9E3C8'
  (in BLOB format returning the BLOB representation of LASTNAME data type VARCHAR(15) value 'SMITH'.)

CARDINALITY (*array-expression*)
- SET xxx = CARDINALITY(*value*)

CCSID_ENCODING (*expression)*
- CCSID_ENCODING (500) → EBCDIC
  (in CHAR(8) format, returning the encoding scheme for CCSID 500, the value EBCDIC.)

[CEIL | CEILING] (*expression*)
- CEIL(MIN(SALARY)/12) → 1279.00000000
  (in DECIMAL(15,8) format, returning the smallest integer value that is greater than or equal to MIN (SALARY)/12, data type DECIMAL (15,15-7) VALUE 1278.33333333.)

CHAR (*integer-expression*)
- CHAR(EDLEVEL) → '14bbbb'
  (in CHAR (6) format, returning the character string representation of EDLEVEL data type SMALLINT, value 14. In CHAR (11) format for INTEGER data type.)

CHAR (*decimal-expression* { ,*decimal-character* })
- CHAR(SALARY,',' → '0017750,00'
  (in CHAR(2+p) Format, where p is the precision of the decimal expression, returning the character representation of SALARY DEC(9,2) value 17750.00 substituting',' for the default period decimal marker.)

CHAR (*floating-point-expression*)
- CHAR(XCOORD) → '2.0E0'
  (in CHAR(24) format, returning the character representation of XCOORD, data type DOUBLE, value 2.)

CHAR (*decimal-floating-point-expression*) DECFLOAT

CHAR (*character-expression* { ,*integer* {,CODEUNITS16 | ,CODEUNITS32 | ,OCTETS })
- CHAR(FIRSTNAME, 8) → 'PHILIP'
  (in CHAR(8) format, returning a fixed length character string representation of FIRSTNME data type VARCHAR(12) value 'PHILIP'. For FIRSTNAME with length greater than 8 a warning message is returned.)

CHAR (*graphic-expression* { ,*integer* {, CODEUNITS16 | ,CODEUNITS32 } })
- CHAR(PICKS) →
  (returns a fixed length character representation of a graphic string)

CHAR (*datetime-expression* {,ISO | USA | EUR | JIS | LOCAL]})
- CHAR(HIREDATE,USA) → '08/09/2002'
  (in CHAR(10) format)

CHAR (*rowid-expression*)
- CHAR(EMP_ROWID) -> *40 byte ids*
  '*(in CHAR(40) format, returning bit data that does not have an associated CCSID.)*

CHAR9 (*decimal-expression{ ,decimal-character}*

CHARACTER_LENGTH (*character-expression,* [CODEUNITS16 | CODEUNITS32 | OCTETS])

- CHARACTER_LENGTH(NAME,CODEUNITS32)
  FROM T1 WHERE NAME = 'Jurgen' -> 6
- CHARACTER_LENGTH(NAME,CODEUNITS16)
  FROM T1 WHERE NAME = 'Jurgen' -> 6
- CHARACTER_LENGTH(NAME,OCTETS)
  FROM T1 WHERE NAME = 'Jurgen' -> 7
  (each in VARCHAR(128) format, encoded in Unicode UTF-8.)

CHARACTER_LENGTH (*graphic-expression,* [CODEUNITS16 | CODEUNITS32])

CLOB (*character-expression* { ,*integer* { [CODEUNITS16 | CODEUNITS32 | OCTETS] })

- CLOB(LASTNAME) -> 'SMITH'
  (in CLOB format returning the CLOB representation of LASTNAME, data type VARCHAR (15), value 'SMITH'.)

CLOB (*graphic-expression,{integer* [CODEUNITS16 | CODEUNITS32]})

[COALESCE | VALUE] (*expression,* [[*expression* ,]])

- COALESCE (ZIP,'00000-0000') -> '00000-0000'
  (in ZIP's format, returning the first non-null value in the list. COALESCE conforms to standard SQL.)

COLLATION_KEY (*string-expression, collation-name* {, integer})

- COLLATION_KEY () ->
  (returns a varying-length binary string that represents the collation key of the string-expression in the specified collation-name)

COMPARE_DECFLOAT (decfloat-*expression1* , decfloat-*expression2)*

- COMPARE_DECFLOAT (DECFLOAT(2,17), DECFLOAT(2,17)
  -> 0
  (returns a SMALLINT value that indicates whether the two arguments are equal or unordered, or whether one argument is greater than the other)

CONCAT (string-*expression1, string-expression2*)

- CONCAT (LASTNAME, FIRSTNAME) -> 'SMITHPHILIP'
  (in CHAR format returning the link of two strings into one string expression for LASTNAME, data type VARCHAR(15), value SMITH' and FIRSTNAME, data type VARCHAR(12), value 'PHILIP'.)

CONTAINS (column-name, search-argument { , { QUERYLANGUAGE = value} {RESULTLIMIT = value} {SYNONYM = [OFF | ON]} } , {*string-constant}*)

- CONTAINS (RESUME , 'SQL') -> 1
  (in large integer format returning 1 if yes or 0 if no)

COS (numeric-*expression*)

- COS (ANGLE) -> +0.5506493986967134E+00
  (in double precision floating-point format, returning the cosine of ANGLE, VALUE .9876532.)

COSH (numeric-*expression*)

- COSH (ANGLE) -> +0.1528689204821580E+01
  (in double precision floating-point format, returning the hyperbolic cosine in radians of ANGLE, VALUE .9876532.)

DATE (*expression*)

- DATE('2002059') -> 2002-02-28
  (in DATE format, returns the date)

DAY (*expression*)

- DAY(CURRENT DATE) → 8
  (in INTEGER format, if today is '2002-12-08'.)

DAYOFMONTH (*expression*)

- DAYOFMONTH(HIREDATE) -> 8
  (in INTEGER format, returning the day part of the HIREDATE, value '2002-12-08.)

DAYOFWEEK (*expression*)

- DAYOFWEEK(HIREDATE) -> 1
  (in INTEGER format, returning a value from 1 to 7 representing the day of the week where 1 is Sunday and 7 is Saturday for the value of HIREDATE, value '2002-12-08'.)

DAYOFWEEK_ISO (*expression*)

- DAYOFWEEK_ISO (HIREDATE) -> 7
  (in INTEGER format, returning a value from 1 to 7 representing the day of the week where 1 is Monday and 7 is Sunday for the value of HIREDATE, value '2002-12-08'.)

DAYOFYEAR (*expression*)

- DAYOFYEAR(HIREDATE) -> 342
  (in INTEGER format, returning a value from 1 to 366 where 1 is January 1 for the value of HIREDATE, value '2002-12-08'.)

DAYS (*expression*)

- DAYS(HIREDATE) → 731192
  (in INTEGER format, the number of days for HIREDATE value '2002-12-08' since 12/31/0000)

DBCLOB ([*character-expression | graphic-expression*] { ,integer} { , CODEUNITS16 | , CODEUNITS32 })

- DBCLOB(GRAPHIC-DATA) → âjâoârâdâaânârâeânâe.
  (in DBCLOB format returning the DBCLOB representation of GRAPHIC_DATA datatype VARGRAPHIC(50) value 'JORDANRENE'.)

DECFLOAT (*numeric-expression* | *string-expression* {,34|,16})

- DECFLOAT( ) →
  (result of the function is a DECFLOAT with the implicitly or explicitly specified number of digits of precision)

DECFLOAT_FORMAT (*string-expression* {,*format-string*})

- DECFLOAT( ) →
  (result is a DECFLOAT(34). Result can be null.)

DECFLOAT_SORTKEY (*decfloat-expression*)

- DECFLOAT_SORTKEY( ) →
  (result is a fixed length binary string with a length attribute of 9 if decfloat-expression is a DECFLOAT(16) value or 17 if decfloat-expression is a DECFLOAT(34) value)

[DEC | DECIMAL] [(*numeric-expression* {,*precision-integer* {,*scale-integer* } }) | (*string-expression*] {,*precision-integer* {,*scale-integer* {, *decimal-character* } } })]

- DECIMAL(EDUCLVL,3,0) → 019.
  (in DECIMAL(3,0) format, with a precision of 3 [total # of digits] and a scale of 0 [digits to right of decimal])
- DECIMAL(SALARY,9,2,'.') → 0023019.50
  (in CHAR(09) format, with a precision of 9 [total # of digits] and a scale of 2[digits to right of decimal] using a decimal-character of '.' to delimit the decimal digits from the whole number.)

DECODE (*expression1, expression2, result-expression* [*else-expression*])

- DECODE →

[DECRYPT_BINARY| DECRYPT_BIT|DECRYPT_CHAR| DECRYPT_DB|] (*encrypted-expression {,password-string-expression* | ,DEFAULT*}{,ccsid-constant}*)

- DECRYPT_CHAR(SSN,'Joe456') → '289-46-8823'.
  (where SSN contains the encrypted value from SET ENCRYPTION PASSWORD ='Joe456' INSERT INTO EMP(SSN) VALUES ENCRYPT_TDES '289-46-8823';).

DEGREES (*numeric-expression*)

- DEGREES(ANGLE) → 5658842415386325E+02
  (in double precision floating-point format returning the number of degrees expressed in radians for ANGLE, value .98765432.)

DIFFERENCE (*expression-1, expression-2*)

- DIFFERENCE (SOUNDEX('Katy') , SOUNDEX('Katie'))) → 4
  (returns a value from 0 to 4 that represents the difference between the sounds of two strings based on applying the SOUNDEX function to the strings)

DIGITS (*numeric-expression*)

- DIGITS(SALARY) → 005245000
  (in CHAR(9) format, if the data type of SALARY is DECIMAL(9,2) and the value is $52,450)

[DOUBLE | DOUBLE_PRECISION | FLOAT] ([*numeric-expression* | *string-expression*])

- DOUBLE((SALARY+COMM) /12) → +0.1597529166600000E+04
  (returns the double precision floating-point format representation of (SALARY +COMM/12), value 1597.5291666.)
- DOUBLE((BONUS) → +0.2500000000000000E+03
  (returns the double precision floating-point format representation of char(08) column BONUS*, value '00000250'.)*

DSN_XMLVALIDATE *([string-expression* | *xml-expression], [schema-name-string* | *target-namespace-uri-string, schema-location-string])*

- DSN_XMLVALIDATE ( ) → XML value
  (the result of DSN_XMLVALIDATE is a varying length binary value of up to. 50MB.The result has no meaning outside of providing input to the XMLPARSE function.)

EBCDIC_CHR (*expression*)

- EBCDIC_CHR(88) → *i*
  (returns the character that has the EBCDIC code value that is specified by the argument)

EBCDIC_STR (*string-expression*)

- EBCDIC_STR (FIRST_NAME) 'TEDDY'→
  (returns a value of a built-in character or graphic string that is not a LOB)

[ENCRYPT_TDES|ENCRYPT] (*to-be-encrypted-string {, password-string-expression {,hint-string-expression}*)

- INSERT INTO EMP(SSN) VALUES
  ENCRYPT ('289-46-8823','Atlantic','Ocean')
  → 31.62 approx.
  (where the approximate value 31.62 is contained in a VARBINARY or VARCHAR FOR BIT DATA.)

EXP (*numeric-expression*)

- EXP(BASE_DATA) → +0.2890703256906984E+03
  (in double precision floating point returns the exponential function of BASE_DATA data type DECIMAL (6,5) value 5.66667.)

EXTRACT ({YEAR|MONTH|DAY) FROM
(date-expression|timestamp-expression)

- EXTRACT(YEAR FROM HIRE_DATE) → 1987
  (returns the year portion of the date, the result is 1987, if HIRE_DATE is 1987-06-21)

EXTRACT ({HOUR | MINUTE | SECOND*)* FROM *(time-expression|timestamp-expression*)

- EXTRACT(HOUR FROM START_TIME) → 11
  (returns the hour portion of the START_TIME, the result is 11 if the time is 11:30:22)

EXTRACT ({HOUR | MINUTE | SECOND |
TIMEZONE_HOUR | TIMEZONE_MINUTE) FROM
({*date-expression | time-expression | timestamp-expression*})

- EXTRACT(TIMEZONE_HOUR FROM
  START_TIME) → 11
  (returns the hour portion of the START_TIME adjusted for time zone, the result is 11 if the time is 11:30:22)

FLOAT (*numeric-expression*)

- FLOAT(BONUS)/2 → +0.5000000000000E+03
  (in FLOAT format)

FLOOR (*numeric-expression*)

- FLOOR (MIN(SALARY+COMM) / 12) → 1380.0000000
  (in DECIMAL (15,7) format, returning the largest integer value less than or equal to MIN (SALARY + COMM) / 12), data type DECIMAL (15,15-8) VALUE 1380.5833333.)

GENERATE_UNIQUE| GENERATE_UNIQUE_BINARY ()

- GENERATE_UNIQUE() → *unique-value*
  (in a bit data CHAR(13) format, that is unique for all executions of the function. Theresult will include the internal form of the Universal Time, Coordinated (UTC.)

GETHINT (*encrypted-data*)

- GETHINT(SSN) → 'Ocean'
  (where SSN contains the encrypted value from INSERT INTO EMP(SSN) VALUES ENCRYPT('289-46-8823','Atlantic', 'Ocean');)

GETVARIABLE (*string-constant*
{, default-value | , CAST(NULL AS VARCHAR(1)) })

- GETVARIABLE('SYSIBM.PLANNAME') → 'INQUP'
  (where the active plan being executed is 'INQUP'.)

GRAPHIC ([*character-expression* |
graphic-expression]{,*integer*} { , [CODEUNITS16 | CODEUNITS32] })

- GRAPHIC(LASTNAME) → âSâMâIâTâH
  (Returning a fixed length graphic string representation of LASTNAME data type VARCHAR(15), value 'SMITH'.)

HASH_CRC32, HASH_MD5, HASH_SHA1, and HASH_SHA256 *(expression)*

HEX (*expression*)

- HEX(DEPT) → 'C5F1F1'
  (in CHAR(6) format, if the data type of DEPT is CHAR(3) and the value is 'E11')

HOUR (*expression*)

- HOUR(081500.) → 8
  (in INTEGER format, returns the hour portion)

IDENTITY_VAL_LOCAL ( )

- IDENTITY_VAL_LOCAL( ) → 00000000000000000000000000000001.
  (in DECIMAL (31, 0) format, returning the most recently-assigned value for an identity column)

IFNULL (*expression, expression*)

- IFNULL (BONUS,0) → .00
  (in DECIMAL (9,2) format, returning the first non-null value in a list of only two expressions. BONUS data type DECIMAL(9,2,), value is NULL.)

INSERT (*source-string, start, length,*
insert-string { ,[CODEUNITS16 | CODEUNITS32 | OCTETS])

- CHAR (INSERT(WORKDEPT, 2,0,'-' ||
  JOB || '-'),20) → 'E-OPERATORS-11'
  (in CHAR(20) format returning a fixed length character representation of the value for WORKDEPT, value 'E11' with the expression in JOB, value 'OPERATOR' inserted beginning at position 2 deleting 0 bytes.

[INTEGER | INT] (*numeric-expression | string-expression*)

- INTEGER(AVG(SALARY)) → 44083
  (in INTEGER format, after truncating the calculated average salary)

JULIAN_DAY (*expression*)

- JULIAN_DAY(HIREDATE) → 2452617
  (in INTEGER format returning the number of days from January1, 4713 B.C. to the date specified.)

LAST_DAY (*expression*)

- LAST_DAY (HIREDATE) → 2002-12-31
  (in DATE format, returns a date that represents the last day of the month for '2002-12-08')

[LCASE | LOWER] (*expression* { , *locale-name*} { , integer})

- LCASE (LASTNAME) → lee
  (in VARCHAR (15) FORMAT returning the value of LASTNAME, value 'LEE' in lower case.)

LEFT (*string-expression, length {* ,[CODEUNITS16 | CODEUNITS32 | OCTETS]) })

LEFT (*graphc-expression, length {*,[CODEUNITS16 | CODEUNITS32] })

LEFT (*binary-expression, length*)

- LEFT (WORKDEPT, 1) →E
  (in CHAR (1) format returning the leftmost characters from WORKDEPT, value 'E11' for a length of 1.)

LENGTH (*expression*)

- LENGTH(LASTNAME) → 7
  (in INTEGER format, if the data type of LASTNAME is VARCHAR(20) and the value is 'Johnson'; note that if the data type is fixed-length CHAR(20), the result would be 20!)

[LOG | LN] (*numeric-expression*)

- LOG(BASE_DATA) →
  +0.1734601643623227E+01
  (in double precision floating-point format returning the natural logarithm of BASE_DATA, value 5.6667.)

LOCATE (*search-string, source-string {,start* { ,[CODEUNITS16 | CODEUNITS32]})

- LOCATE('M', LASTNAME, 4) → 5
  (in INTEGER format returns the starting position of the first occurrence of search string in LASTNAME, value 'YAMAMOTO' beginning search at position 4.)

LOCATE_IN_STRING (*source-string, search-string* {,*start,* {,*instance*} {,[CODEUNITS16|CODEUNITS32|OCTETS] }})▪

LOCATE_IN_STRING ('M', LASTNAME, 4)
(in integer or smallint format returns the starting position of the value 'M')

LOG10 (*numeric-expression*)

- LOG10(BASE_DATA) →
  +0.7533279221258787E+00
  (in double precision floating-point format returning the base 10 logarithm of BASE_DATA, data type DECIMAL (6,5), value 5.66667.)

LOWER (*string-expression* {, *locale-name*}{ , *integer*})

- LOWER (PLANT_NAME) → 'williamsfield'
  (returns a value for 'WILLIAMSFIELD' in which all the characters have been converted to lowercase characters)

LPAD (*string-expression*, *integer* {, *pad*})

- LPAD (LASTNAME,5,'Z') →
  'ZZZZZSMITH'
  (returns a string that is padded on the left, with the pad value of 'Z' for 5 places)

LTRIM (*string-expression*,{*trim-expression*})

- LTRIM (FIRSTNME) → 'SYBIL'
  (in VARCHAR format returning FIRSTNAME, value 'bbbSYBIL' without the leading blanks. Synonymous with STRIP (FIRSTNME, L).)

MAX (*expression* { , [[*expression* ,]] })

- MAX('A', 'C', 'X') → 'X'
  (returns the maximum value from a set of values)

MAX_CARDINALITY (*array-expression*)

- SET parameter=MAX_CARDINALITY(value) ;
  (maximum number of elements an array can contain)

MICROSECOND (*expression*)

- MICROSECOND(TIMESTAMP *('*2002-03-05-10.30.30.999998')) → 999998
  (in INTEGER format)

MIDNIGHT_SECONDS (*expression*)

- MIDNIGHT_SECONDS (CURRENT TIME) → 43121
  (in INTEGER format returning a value from 0 to 86400 representing the number of seconds from midnight to the time specified.)

MIN (*expression* {, {*expression* ,} })

- MIN('A', 'C', 'X') → 'A'
  (returns the minimum value from a set of values)

MINUTE (*expression*)

- MINUTE(081500.) → 15
  (in INTEGER format)

MOD (*numeric-expression-1, numeric-expression-2*)

- MOD(5,2) → 1
  *(in INTEGER format returning the remainder of the calculation 5-(5/2)*2.)*

MONTH (*expression*)

- MONTH(INSURCOV) → 12
  (in INTEGER format, if the date of INSURCOV is '2002-12-08')

MONTHS_BETWEEN (*expression-1,expression-2*)

- MONTHS_BETWEEN (RETIRE_DT,HIRE_DT) → 53.3225806…
  (returns the number of months difference between the two dates, if RETIRE_DT is '2009-10-15' and HIRE_DT is '2005-05-05')

MQREAD ({*receive-service* {,*service-policy*} })

- MQREAD() → *mq-message*
  (Retrieve the message at the beginning of the queue, specified by the default publishing service (DB2.DEFAULT.PUBLISHER), using the default policy.).
- MQREAD('MQSERVICE1', 'MQPOLICY2',) → *mq-message*
  (Read the message at the beginning of the queue specified by the service 'MQSERVICE1', using the policy 'MQPOLICY2').
  (The schema is DB2MQ1C or DB2MQ2C).

MQREADCLOB ({*receive-service* {,*service-policy*} })

- MQREADCLOB() → *mq-message*
  (Retrieve the message at the beginning of the queue, specified by the default publishing service (DB2.DEFAULT.PUBLISHER), using the default policy. The message is returned as a CLOB).

MQRECEIVE ({*receive-service* {,*service-policy* {,*correl-id*} } })

- MQRECEIVE() → *mq-message*
  (Retrieve the message from the beginning of the queue specified by the default service (DB2.DEFAULT.SERVICE), using the default policy (DB2.DEFAULT.POLICY), without specifying any correlation id.)

(Correl-ID can only be specified if a receive service and service policy are specified.)

MQRECEIVECLOB ({*receive-service* {,*service-policy* {,*correl-id*} } })

- MQRECEIVECLOB() → *mq-message*

  (Retrieve the message from the beginning of the queue specified by the default service (DB2.DEFAULT.SERVICE), using the default policy (DB2.DEFAULT.POLICY without specifying any correlation id. The message is returned as a CLOB).)

  (Correl-ID can only be specified if a receive service and service policy are specified.)

MQSEND ({*send-service* , {*service-policy* ,} } *msg-data* {,*correl-id*})

- MQSEND('MQ Test message') → '1'

  (Send the string 'MQ Test message' to the default service (DB2.DEFAULT.SERVICE), using the default policy (DB2.DEFAULT. POLICY), and no correlation identifier.)

  (Correl-ID can only be specified if a send service and service policy are specified.)

MULTIPLY_ALT (*exact-numeric-expression-1, exact-numeric-expression-2*)

- MULTIPLY_ALT(SALARY, 1.05) → 44100.00

  (in DECIMAL(11,2) format, the result of multiplying salary 42000.00)

NEXT_DAY (*expression, string-expression*)

- NEXT_DAY(CURRENT DATE, 'MON') → 2005-04-25-00.00.00.000000

  (returns a timestamp that represents the first weekday, named by 'MON' , that is later than the CURRENT DATE of '2005-04-19')

NORMALIZE_DECFLOAT (*decfloat-expression*)

- NORMALIZE_DECFLOAT (DECFLOAT(-120)) → 1.2E+2

  (returns a DECFLOAT value that is the result of the argument set to its simplest form.)

NORMALIZE_STRING (*unicode-string*, [NFC | NFD | NFKC | NKKD] {, integer})

- NORMALIZE_STRING →

  (returns a value of a built-in character string or graphic string data type that is either Unicode UTF-8 or Unicode UTF-16, and is not a LOB. Takes a Unicode string argument and returns a normalized string.)

NULLIF (*expression, expression*)

- NULLIF(500-50, 450) → NULL

  (Returns a NULL if the arguments are equal. If the two argument values are not equal, the value of the first argument is returned.)

NVL (*expression[, expression]*)

- Synonym for the COALESCE function

OVERLAY (*source-string, insert-string , start* { *, length* } ,
[CODEUNITS16 | CODEUNITS32 | OCTETS])

- OVERLAY →
  (returns a string where a substring of length, beginning at start has been deleted from source-string, and where insert-string has been inserted into source-string beginning at start.)

PACK (CCSID 1208 | CCSID DEFAULT [,*expression*])

- SELECT PACK(CCSID …)
  (result is a binary string that contains a data type array and a packed representation of each non-null expression argument.)

SET :udf_result =

PACK(CCSID 1208, 'Alina', DATE('1977-08-01'),DOUBLE(0.5));

POSITION ({*search-string, source-string,*
[CODEUNITS32 | CODEUNITS16 | OCTETS]

- POSITION('T', 'MY DATABASE', OCTETS)→ 6
  (In integer format returning the starting position of the first occurrence of the search-string 'T' within source-string 'MY DATABASE'.)

POSSTR (*source-string, search-string*)

- POSSTR(LASTNAME, 'M')→ 3
  (in INTEGER format returning the starting position of the first occurrence of search-string, 'M' within source-string, LASTNAME value 'YAMAMOTO'.)

POWER (*numeric-expression1*, *numeric-expression2*)

- POWER (INT1, INT2) → 16
  (in INTEGER format returning the value of INT1, value 2, raised to the power of INT2, value 4.)

QUANTIZE (*expression-1*,*expression-2*)

- QUANTIZE () →
  (returns a DECFLOAT value that is equal in value (except for any rounding) and sign to numeric-expression and which has an exponent set to equal to the exponent of exp-expression)

QUARTER (*expression*)

- QUARTER (HIREDATE) → 2
  (in INTEGER format returning a number 1 to 4 representing the quarter of the year the date HIREDATE, value '1972-06-19' occurs.)

RADIANS *(numeric-expression)*

- RADIANS (ANGLE) →
  +0.1704389412796149E-01
  (in double precision floating-point format returning the number of radians of ANGLE value .9765432 expressed in degrees.)

RAISE_ERROR *(sqlstate, diagnostic_string)*

- select empo,
  case when sex = 'm' then 'male'
  when sex = 'f' then 'female'
  else RAISE_ERROR ('71001','no gender')
  → 'SQLCODE = -438, ERROR:
  APPLICATION RAISED ERROR
  WITH DIAGNOSTIC TEXT: 'no gender' *(in CHAR or VARCHAR format)*

RAND ({*numeric-expression}*)

- RAND( ) → +0.3763304037863064E+00
  (in double precession floating-point format returning a random number between 0 and 1. If the optional seed value is used it must range from 0 to 2147483646.)

REAL ([*numeric-expression* | *string-expression*])

- Real(EMPNO) → +0.300000000E+03
  (in single precision floating-point returning the value of EMPNO, data type CHAR(6) value'000300'.)

REPEAT (*expression*, *integer*)

- REPEAT (FIRSTNAME, 2) → PHILIPPHILIP
  (in VARCHAR format returning the value of FIRSTNAME, value 'PHILIP' repeated 2 times.)

REPLACE (*source-string, search-string{,replace-string}*)

- REPLACE (JOB,'REP', 'SALES') → FIELDSALES
  (in VARCHAR format returning the value for JOB with value 'FIELDREP' having replaced all occurrences of 'REP' with 'SALES'. If 'REP' were not found the value of JOB would be returned unchanged.)

RID (*table-designator*)

- RID (EMP) →
  (returns the RID(s) of rows that match the where clause; the result of the function is BIGINT)

RIGHT (*string-expression, integer*
{, [CODESUNIT32 | CODEUNITS16 | OCTETS]})

- RIGHT (WORKDEPT,2) → 21
  (in CHAR format returning the 2 right most characters of WORKDEPT, data type CHAR(3) value 'E21'.)

ROUND (*numeric-expression1{*,0|,*numeric-expression2*)

- ROUND (MIN(SALARY)/12,2 →1278.33000000
  (in DECIMAL (15,8) format, returning MIN(SALARY) /12, value 1278.33333333 data type, DEC (15,15-7) rounded to 2 places right of the decimal.)

ROUND_TIMESTAMP (*expression* {', DD' | ,*format-string*})

- ROUND_TIMESTAMP(CURRENT TIMESTAMP) → '2002-12-09-00.00.00.000000'
  (returns a timestamp, rounded to the unit specified by the format-string. If format-string is not specified, the current timestamp of '2002-12-08-14.28.49.179227 is rounded to the nearest day, as if 'DD' is specified for format-string.

ROWID (*expression*)

- SELECT EMPNO FROM EMP WHERE EMP_ROWID = ROWID (X'4000000038000104000 F02010000000000021C') → '000300'
  (casts the value of the input expression as a rowid)

RPAD (*string-expression ,integer* {, *pad*})
RPAD (LASTNAME,5,'X') → 'SMITHXXXXX'
(returns a string that is padded on the right, with pad value or blanks, 5 times

RTRIM (*string-expression{,trim-expression*)
- RTRIM(JOB) →'CLERK'
(in VARCHAR format returning JOB, value 'CLERK bbb' without the trailing blanks. Synonymous with STRIP (JOB,T).)

SCORE (*column-name , search-argument* {,QUERYLANGUAGE = value} {,RESULTLIMIT = value} {,SYNONYM = [OFF|ON]})
- SCORE(RESUME, 'programmer AND (SQL)') → 32
(in FLOAT format)

SECOND (*expression* {, *integer-constant*})
- SECOND('2005-04-19-11.31.32.696688') → 32
(in INTEGER format)

SIGN (*numeric-expression*)
- SIGN(BONUS) → -1.00
(in DECIMAL format returning an indicator of the sign for BONUS, value –1000.00. Indicator values are: -1 for <0, 0 for 0, and 1 for > 0.)

SIN (*numeric-expression*)
- SIN(ANGLE) → +0.8347366289524786E+00
(in double precision floating-point format returning the sine of ANGLE, value .98765432.)

SINH (*numeric-expression*)
- SINH(ANGLE) → +0.1156239890739822E+01
(in double precision floating-point format returning the hyperbolic sine of ANGLE, value .98765432.)

SMALLINT ([*numeric-expression* | *string-expression*])
- SMALLINT(BONUS/12) → 33
(in SMALLINT format returning a small integer representation of BONUS/12, value 33.33333333, using truncation.)

SOUNDEX *(expression)*
- SOUNDEX ('Katy') → K300
(returns a 4 character code that represents the sound of the words in the argument)

[SOAPHTTPNC | SOAPHTTPNV] *(endpoint_url, soap_action, soap_input*)
- SOAPHTNCC ( ) →
(returns a CLOB representation of XML data that results from a SOAP request to the web service specified by the first argument

[SOAPHTTPC | SOAPHTTPV] (*endpoint_url, soap_action, soap_body)*
---Deprecated in V10
- SOAPHTTPC ( ) →
(returns a CLOB representation of XML data that results from a SOAP request to the web service specified by the first argument

SPACE (*numeric-expression)*

- SPACE(LENGTH(LASTNAME)) → ‘ ’
  (in VARCHAR format returning a SBCS string of blanks for the length of LASTNAME, value ‘SMITH’.)

SQRT (*numeric-expression)*

- SQRT(EDLEVEL) →
  +0.3741657386773941E+01
  (in double precision floating-point format returning the square root of EDLEVEL, value 14.)

STRIP (*string-expression* {, [BOTH | B | LEADING | L | TRAILING | T]}{,*trim-constant*})

- STRIP(REMARKS, L, ‘*’) → ‘NEW PROD’
  *(in a string expression, returns the value ‘NEW PROD’ from a value of ‘**NEW PROD’ in the REMARKS column. If the second argument is omitted, the strip-character is removed from the beginning and the end, if third argument is omitted, a blank is assumed.)*

SUBSTR (*string-expression*, *start* {,*length*})

- SUBSTR(DESCRIP,1,6) → ‘FURNIS’
  (in CHAR(6) format, if DESCRIP contains the value ‘FURNISHINGS’)

SUBSTRING (*character-expression*, *start* {,*length*}
{ , [CODESUNIT32 | CODEUNITS16 | OCTETS]})

| (*graphic-expression*, *start* {,*length*}
{ , [CODESUNIT32 | CODEUNITS16]})

| (*binary-expression*, *start* {,*length*}

- SUBSTR(DESCRIP,1,6,OCTETS) → ‘FURNIS’
  (in CHAR(6) format, if DESCRIP contains the value ‘FURNISHINGS’)

TAN (*numeric-expression*)

- TAN(ANGLE) → +0.151591308540088E+01
  (in double precision floating-point format returning the target of ANGLE, value .98765432.)

TANH (*numeric-expression*)

- TANH(ANGLE) → +0.7563603426340487E+00
  (in double precision floating-point format returning the hyperbolic target of ANGLE, value .98765432.)

TIME (*expression*)

- TIME(‘23.48.12’) + 2 HOURS → 01.48.12
  (in TIME format)

TIMESTAMP (*expression-1* {,*expression-2*})

- TIMESTAMP(HIREDATE,‘08.15.00’) →
  2002-09-12-08.15.00.000000
  (in TIMESTAMP format)

TIMESTAMPADD (*interval,number,expression*)

- TIMESTAMPADD (8, 10, PROC_TIME) →
  (returns the result of adding the specified number (10) of the specified interval (8=HOURS) to the specified timestamp, PROC_TIME

TIMESTAMPDIFF (*numeric-expression, string-expression*)

- TIMESTAMPDIFF (64, (CAST (CURRENT_TIMESTAMP)– CAST(HIRE_DT AS TIMESTAMP)) →
  (returns an estimated number of intervals of the type defined by the first argument, based on the difference between two timestamps)

TIMESTAMP_FORMAT (*string-expression, format-string* { , 6 | , *precision-constant*})

- TIMESTAMP_FORMAT('2002-12-08 16:10:10', 'YYYY-MM-DD HH24:MI:SS' → 2002-12-08-16.10.10.000000
  (returns a valid TIMESTAMP, in the only valid string format of 'YYYY-MM-DD HH24:MM:SS')

TIMESTAMP_ISO (*expression*)

- **TIMESTAMP_ISO (DATE('1988-09-27'))** → '1988-09-27.00.00.00.00000'
  (returns a timestamp value that is based on a date, a time, or a timestamp argument. If the argument is a date, TIMESTAMP_ISO inserts a value of zero for the time and microseconds parts of the timestamp. If the argument is a time, TIMESTAMP_ISO inserts the value of CURRENT DATE for the date part of the timestamp and a value of zero for the microseconds part of the timestamp

TIMESTAMP_TZ (*expression-1{ ,expression-2}*)

- **TIMESTAMP_TZ_ ('**2008-02-29.00.00.00.00000**' , -3.00))** → '2008-02-29.00.00.00.00000-03.00'
  (returns a timestamp with time zone value that is based on a timestamp argument.

TO_CHAR

- Character to VARCHAR *(character-expression)*
- Timestamp to VARCHAR *(timestamp-expression , format-string)*
- Decimal floating-point to VARCHAR *(decimal-floating-point-expression { , format-string)*

SEE VARCHAR_FORMAT

TO_DATE (*string-expression , format-string* {,6 | , *precision-constant*)

- SEE TIMESTAMP_FORMAT

TO_NUMBER|DECFLOAT_FORMAT (string-expression {, format-string})

- SEE DECFLOAT_FORMAT

TOTALORDER (*decfloat-expression1,decfloat-expression2*)

- TOTALORDER ( ) → -1
  (returns a SMALLINT value of -1, 0, or 1 that indicates the comparison order of two arguments)

TRANSLATE (*string-expression* { ,*to-string*{ ,*from-string* {,' ' | ,*pad*} } })

- TRANSLATE(JOB, 'OUT', FIELD) → 'OUT REP'
  (in CHAR format returning the value of JOB, value 'FIELDREP' translating each occurrence of 'FIELD' to 'OUT' using the default pad character of space .)

TRIM ({[BOTH|B|LEADING|L|TRAILING|T]{ *trim-constant}* FROM} *string-expression*)

TRIM_ARRAY TRIM_ARRAY|ARRAY_TRIM (*array-expression,numeric-expression*)

[TRUNCATE | TRUNC] (*numeric-expression1{*,0 | *,numeric-expression2}*)

- TRUNC(MIN(SALARY/12) ,3) → 1278.333
  (in DECIMAL format returning the value of MIN (SALARY/12) value 1278.33333333 truncated at 3 digits right of the decimal.)

TRUNC_TIMESTAMP (*expression)*
{'DD' | *,format-string*}

- TRUNC_TIMESTAMP
  (CURRENT TIMESTAMP, 'YEAR') →
  2002-01-01-00.00.00.000000
  (returns a timestamp, value from '2002-12-08-16.10.10.012000' truncated to the YEAR unit specified.)

[UCASE | UPPER] (*expression* {,*locale-name-string*} {,*integer*})

- UCASE(LASTNAME) → 'LEE'
  (in VARCHAR format returning the value of LASTNAME value 'lee' in uppercase.)

UNICODE *(string-expression)*

- UNICODE ('ABC') → 65
  (returns the Unicode UTF-16 code value of the leftmost character of the argument as an integer.)

UNICODE_STR (*string-expression* {,UTF8 | ,UTF16})

- UNICODE_STR ('ABC') → 'ABC'
  (returns a string in Unicode UTF-8 or UTF-16, depending on the specified option, that represents a Unicode encoding of the input string.)

UNPACK (e*xpression*)

- SELECT UNPACK(myUDF(C1)).* AS (Name VARCHAR(40) CCSID UNICODE,
  DOB DATE,
  Score DOUBLE)

UPPER (*string-expression* { , *locale-name-string*} { , *integer*})

- UPPER ('dallas') → 'DALLAS'
  (returns a string in which all the characters have been converted to uppercase characters.)

[VALUE | COALESCE] (*expression* [[, *expression*]])

- VALUE(ZIP,'00000-0000') → '00000-0000'
  (in ZIP's format, returning the first non-null value in the list)

VARBINARY (*string-expression* {, *integer*})

- VARBINARY('KBH-93',3) → BX'D2C2C8''
  (returning a varying-length binary string with length attribute of 3 and actual length 3)

VARCHAR

Integer to Varchar

(*integer-expression)*

- VARCHAR(EDLEVEL) → '14'
  (in VARCHAR format returning the character string representation of EDEVEL data type SMALLINT, value 14.)

Decimal to Varchar

(*decimal-expression {,decimal-character})*

- VARCHAR(SALARY,' , ') → '17750,35'
  (in VARCHAR format returning the character representation of SALARY DEC(9,2) value 17750.00 substituting ' , ' for the default period decimal marker.)

Floating-point to Varchar

(*floating-point-expression)*

- VARCHAR(XCOORD) → '2.0E0'
  (in VARCHAR format the character representation of XCOORD data type DOUBLE value2.)

Decimal floating-point to Varchar

(decimal-*floating-point-expression)*

- VARCHAR(NEGNUM) '-1234.56'
  (in VARCHAR format the character representation of NEGNUM data type (VARCHAR(254).)

Character to Varchar

(*character-expression* { ,*integer*{, CODESUNIT32 | CODEUNITS16 | OCTETS } })

- VARCHAR(JOB,5) → 'CLERK'
  (in VARCHAR format returning the value of JOB, data type CHAR(8), value 'CLERK'. For JOB with length greater than 5 a warning message is returned.)

Graphic to Varchar

(*graphic-expression* {,*integer* {,CODESUNIT32|CODEUNITS16}})

- VARCHAR(NAME,3) 'Jür'
  (in VARCHAR format returning the value of NAME, data type GRAPHIC(3), value "Jür' – x'004A00FC0072''. For JOB with length greater than 5, a warning message is returned.

(*graphic-expression* { ,*integer* {,CODESUNIT32|CODEUNITS16}})

- VARCHAR(JOB,5) → 'CLERK'
  (in VARCHAR format returning the value of JOB, data type CHAR(8), value 'CLERK'. For JOB with length greater than 5 a warning message is returned.)

Datatime to Varchar

(datetime-expression)

- VARCHAR(HIREDATE) → '2000-07-07'
  (in install or precompiler option DATE FORMAT returning a varying-length string representation of HIREDATE data type DATE, value '2000-07-07'.)

Row ID to Varchar

(*row-ID-expression)*

- VARCHAR(EMP_ROWID) → '.'
  (in VARCHAR format returning a bit string that does not have an associated CCSID.)

VARCHAR_BIT_FORMAT (*expression {,format-string}*)

- VARCHAR__BIT_FORMAT ('123-abc',xxx-xxx')_→ '123ABC'
  (returns a varying-length bit data string)

VARCHAR9 (*decimal-expression, decimal-character*)

- VARCHAR9 ('000.1','.1') → '0.1'
  (returns a varying length character string of a decimal in the format indicated by decimal-character

VARCHAR_FORMAT

Character to Varchar

(*character-expression*)

- VARCHAR_FORMAT(JOB) 'CLERK'
  (in VARCHAR format returning the value of JOB, data type CHAR(8), value 'CLERK'.)

Timestamp to Varchar

(*timestamp-expression,format-string*)

- VARCHAR_FORMAT (CURRENT TIME, 'YYYY-MM-DD HH24:MI:SS') → '2002-12-08 15:39:53'
  (returns a character representation of a timestamp in the format indicated by format-string from the current timestamp value of '2002-12-08-15.39.53.012310'.)

Decimal floating-point to Varchar

(decimal-*floating-point-expression {,format-string})*

- VARCHAR(NEGNUM,'9999.99') '-1234.56'
  (in VARCHAR format the character representation of NEGNUM data type VARCHAR(254).)

VARGRAPHIC ([*char-expression* | *graphic-expression*] {,*integer* {, [CODESUNIT32 | CODEUNITS16] } })

- HEX(VARGRAPHIC('SCHOOL')) → '42A2428342842964293'
  (in CHAR(24) format, after translating the CHAR(6) string to its VARGRAPHIC(6) (12-byte) equivalent, then each byte to its two-byte hexadecimal representation)

VERIFY_GROUP_FOR_USER ([SESSION_USER | USER] { ,group-name-expression})

- VERIFY_GROUP_FOR_USER(SESSION_USER, 'MGR') → 1
  (in LARGE INTEGER format returning a value from 1 if yes or 0 if no)

VERIFY_ROLE_FOR_USER ([SESSION_USER | USER] [[, *role-name-expression*]])

- VERIFY_ROLE_FOR_USER(SESSION_USER, 'MGR') → 1
  (in LARGE INTEGER format returning a value from 1 if yes or 0 if no)

VERIFY_TRUSTED_CONTEXT_ROLE_FOR_USER

([SESSION_USER | USER] [[, *role-name-expression*]])

- VERIFY_TRUSTED_CONTEXT_ROLE_FOR_USER(SESSION_USER, 'MGR') → 1
  (in LARGE INTEGER format returning a value from 1 if yes or 0 if no)

WEEK (*expression)*

- WEEK(HIREDATE) → 50
  (in INTEGER format returning a value from 1 to 54 representing the week of the year for HIREDATE data type DATE, value '2002-12-08'.)

WEEK_ISO (*expression*)

- WEEK_ISO(HIREDATE) → 49
  (in INTEGER format returning a value from 1 to 53 representing the week of the year for '2002-12-08'. The week starts with Monday and includes 7 days. Week 1 is the first week of the year to contain a Thursday, which is equivalent to the first week containing January 4.

WRAP (*object-definition-string*

- SELECT WRAP('CREATE FUNCTION salary(wage DECFLOAT)
  RETURNS DECFLOAT
  RETURN wage * 40 * 52')
  FROM SYSIBM.SYSDUMMY1
  →
  CREATE FUNCTION salary(wage DECFLOAT) WRAPPED DSN12015
  ablGWmdiWmtyTmduTmJqTmtaUmtCUmZqUmdiXodK
  3idaWmdaWmdaWmZG1mIaG
  icaGy31TyStm_qGbe3sDxdxjtC8ymVGLpMXnuL8lkmNu
  RhhZ6qYJ2YYdXGaa

XMLATTRIBUTES

(attribute-value-expression {AS attribute-name} ,)

- XMLATTRIBUTES ( ) → XML value
  (result is an XML sequence that contains an XQuery attribute node for each non-null attribute-value-expression argument.)

XMLCOMMENT *(string-expression*)

- XMLCOMMENT ( ) → XML value
  (returns a value of a built-in character or graphic string that is not a LOB and is not bit data. The contents of the comment node are the value of the input string expression mapped to Unicode (UTF-8.)

XMLCONCAT (*XML-expression*{{ *, XML-expression*}})

- XLMCONCAT (
  XMLELEMENT (NAME "First", e.fname),
  XMLELEMENT (NAME "last", e.lname))
  → '<First>John</First><last>Smith</last>'
  (in CHAR format returning the concatenated values delimited by the XML tags.)

XMLDOCUMENT ({XML-expression,})

- XMLDOCUMENT ( ) → XML value
  (returns an XML value with a single document node and zero or more nodes as its children. The content of the generated XML document node is specified by a list of expressions.)

XMLELEMENT

(NAME *element-name* { ,*xmlnamespace-declaration* }
{ ,*xmlattributes-function* }
{{ ,*element-content-expression* }} { OPTION [EMPTY ON NULL | NULL ON NULL | XMLBINARY { USING } [BASE64 | HEX]})

- XLMELEMENT (NAME "First", e.fname),
  → '<First>John</First>'
  (in CHAR format.)
- XLMELEMENT (NAME "Emp", XMLATTRIBUTES (e.id)),
  → '<Emp ID="1001"></Emp>'
  (in CHAR format.)

XMLFOREST ({ *xmlnamespace-function ,* } [[*element-content-expression* { AS *element-name* ,]] { OPTION [EMPTY ON NULL | NULL ON NULL | XMLBINARY {USING} [BASE64 | HEX]] })

- XLMFOREST (e.hire, e.dept AS "department")
  → '<HIRE>2000-05-24</HIRE>
  <department>Shipping</deparment>'
  (in CHAR format.)

XMLMODIFY (*xquery-update-constant* { , [[*xquery-variable-expression* AS *identifier* ,]])

- XMLMODIFY ( ) → XML value
  (modifiesXML argument an XML value.)

XMLNAMESPACES ({DEFAULT | NO DEFAULT } [[*namespace-uri* AS *namespace-prefix* ,]])

- XML2CLOB(XMLFOREST(XLMNAMESPACES (DEFAULT 'HTTP://HR.ORG','HTTP://FED.GOV' AS "D"), JOB AS "D:JOB")) → <D:JOB xmins="HTTP://HR.ORG" xmins:D="HTTP://FED.GOV">OPERATOR</ D:JOB>XML associations
  (default namespace prefix associated with 'http://hr.org', "d:"is replacement prefix for 'http://fed.gov').

XMLPARSE (DOCUMENT [*string-expression* | *XML-host-variable*] {STRIP WHITESPACE | PRESERVE WHITESPACE})

- XMLPARSE ( ) → XML value
  (parses the argument as an XML document and returns an XML value.)

XMLPI (NAME *pi-name* { , *string-expression*})

- XMLPI ( ) → XML value
  (returns a value of a built-in character or graphic string that is not a LOB and is not bit data. The resulting string will be converted to UTF-8 and parsed to check for conformance to the content of XML processing instruction.)

XMLQUERY (*xquery-expression-constant* {PASSING { BY REF} [[xquery-context-item-expression | xquery-variable-expression AS identifier ,]]} {RETURNING SEQUENCE {BY REF }}{EMPTY ON EMPTY})

- XMLQUERY ( ) → XML value
  (returns an XML value from the evaluation of an XPath expression using specified input arguments, a context item, and XPath variable.)

[XMLSERIALIZE|XML2CLOB] ({CONTENT} XML-expression AS data-type {[VERSION '1.0' | EXCLUDING XMLDECLARATION | INCLUDING XMLDECLARATION]})

- XMLSERIALIZE ( ) → XML value
  (returns a serialized XML value of the specified data type that is generated from the XML-expression argument.)

XMLTABLE ({*xmlnamespaces* ,} *row-xquery-expression-constant* {PASSING { BY REF} [[*xquery-context-item-expression* | *xquery-variable-expression* AS *identifier* ,]]} {COLUMNS {*column-name data-type* {*default-clause* | *PATH column-xquery-expression-constant*}} *column-name* FOR ORDINALITY})

XMLTEXT (*string-expression*)

- XMLTEXT ( ) → XML value
  (returns a value of a built-in character or graphic string that is not bit data. Any character in the resulting string must be a valid XML 1.0 character when it is converted to UTF-8.)

XMLXSROBJECTID *(xml-value-expression)*

XSLTRANSFORM *(xml-document, xsl-stylesheet, xsl-parameters)*

YEAR (*expression*)
- YEAR(BIRTHDATE) → 1954
  (in INTEGER format)

Table Functions

ADMIN_TASK_LIST ( )

ADMIN_TASK_OUTPUT (*task-name*, *num-invocations*)

ADMIN_TASK_STATUS ({*max-history*})

MQ Series Table Functions

MQREADALL ({*receive-service* {,*service-policy*} }
{ { ,}*num-rows* })
- MQREADALL() → *mq-message-table*
  (Returns a table containing the messages in the queue specified by the default publishing service (DB2.DEFAULT.SERVICE),using the default policy(DB2.DEFAULT.POLICY) Does not clear the messages in the queue like a receive.)

MQREADALLCLOB ({*receive-service*
{, *service-policy*} } { { ,}*num-rows* })
- MQREADALLCLOB() → *mq-message-table*
  (Returns a table containing the messages in the queue specified by the default publishing service (DB2.DEFAULT.SERVICE), using the default policy (DB2.DEFAULT. POLICY). Messages are contained in CLOB format. Does not clear the messages in the queue like a receive.)

MQRECEIVEALL ({*receive-service* {, *service-policy*} }
{,*correl-id*} { {,}*num-rows*})
- MQRECEIVEALL() → *mq-message-table*
  (Returns a table containing the messages in the queue specified by the default publishing service (DB2.DEFAULT.SERVICE), using the default policy (DB2.DEFAULT. POLICY). Clears all the messages in the queue after the receive.)

MQRECEIVEALLCLOB ({*receive-service {, service-policy*} }
{,*correl-id*} { {,}*num-rows*})
- MQRECEIVEALLCLOB() → *mq-message-table*
  (Returns a table containing the messages in the queue specified by the default publishing service (DB2.DEFAULT.SERVICE), using the default policy (DB2.DEFAULT. POLICY). Messages are contained in CLOB format. Clears the MQ message queue.)

MQRECEIVEALLXML ({receive-service { ,service-policy} {,correl-id} } { {,}num-rows})

- MQRECEIVEALLXML() → mq-message-table
  (Returns a table containing the messages in the queue specified by the default publishing service (DB2.DEFAULT.SERVICE), using the default policy (DB2.DEFAULT.POLICY). Messages are in data type DB2XML.XMLVARCHAR. Clears the messages from the queue.)

Durations

The SQL durations include computed and labeled:

Computed Durations

Date Durations

- CURRENT DATE – BIRTHDATE → 00410508.
  (*in DEC(8,0) format, if born 41 years, 5 months, and 8 days ago*)

Time Durations

- END_TIME – START_TIME → 083000.
  (*in DEC(6,0) format, if eight-and-a-half hours have elapsed*)
- CURRENT TIMEZONE → –060000.
  (*the special register CURRENT TIMEZONE is a time duration in DEC(6,0) format, and the value above is what it would be set to if 6 hours behind GMT*)

Timestamp Durations

- CURRENT TIMESTAMP – CURRENT TIMEZONE → 20020602151500.000000
  (*in DEC(14+s,s) (where s is the number of fractional seconds) format, at 9:30 a.m. on June 2, 2002, if current time zone is 6 hours behind GMT*)

Labeled Durations

[*function-invocation* | (*expression*) | *constant* | *column-name* | *variable*] [YEAR{S} | MONTH{S} | DAY{S} | HOUR{S} | MINUTE{S} | SECOND{S} | MICROSECOND{S}]

- ORDER_DATE + 90 DAYS → '2002-08-31'
  (in DATE format, for orders placed on June 2, 2002)
- END_TIME – 8 HOURS → '09.00.00'
  (in TIME format, if end time is 5 p.m.)
- CURRENT TIMESTAMP + 1 MICROSECOND →
  one microsecond from *now (in TIMESTAMP format)*

Reserved Words

SQL Standard reserved words in alphabetical order:

ADD
AFTER
ALL
ALLOCATE
ALLOW
ALTER
AND
ANY
ARRAY
ARRAY_EXISTS
AS
ASENSITIVE
ASSOCIATE
ASUTIME
AT
AUDIT
AUX
AUXILIARY
BEFORE
BEGIN
BETWEEN
BUFFERPOOL
BY
CALL
CAPTURE
CASCADED
CASE
CAST
CCSID
CHAR
CHARACTER
CHECK
CLONE
CLOSE
CLUSTER
COLLECTION
COLLID
COLUMN
COMMENT
COMMIT
CONCAT
CONDITION
CONNECT
CONNECTION
CONSTRAINT
CONTAINS
CONTENT
CONTINUE
CREATE
CURRENT
CURRENT_DATE
CURRENT_LC_
CTYPE
CURRENT_PATH
CURRENT_
SCHEMA
CURRENT_TIME
CURRENT_
TIMESTAMP
CURSOR
DATA
DATABASE
DAY
DAYS
DBINFO
DECLARE
DEFAULT
DELETE
DESCRIPTOR
DETERMINISTIC
DISABLE
DISALLOW
DISTINCT
DO
DOCUMENT
DOUBLE
DROP
DSSIZE
DYNAMIC
EDITPROC
ELSE
ELSEIF
ENCODING
ENCRYPTION
END
ENDING
END-EXEC
(COBOL only)
ERASE
ESCAPE
EXCEPT
EXCEPTION
EXECUTE
EXISTS
EXIT
EXPLAIN
EXTERNAL
FENCED
FETCH
FIELDPROC
FINAL
FIRST
FOR
FREE
FROM
FULL
FUNCTION
GENERATED
GET
GLOBAL
GO
GOTO
GRANT
GROUP
HANDLER
HAVING
HOLD
HOUR
HOURS
IF
IMMEDIATE
IN
INCLUSIVE
INDEX
INHERIT
INNER
INOUT
INSENSITIVE
INSERT
INTERSECT
INTO
IS
ISOBID
ITERATE
JAR
JOIN
KEEP
KEY
LABEL
LANGUAGE
LAST
LC_CTYPE
LEAVE
LEFT
LIKE
LIMIT1
LOCAL
LOCALE
LOCATOR
LOCATORS
LOCK
LOCKMAX
LOCKSIZE
LONG
LOOP
MAINTAINED
MATERIALIZED
MICROSECOND
MICROSECONDS
MINUTE
MINUTES
MODIFIES
MONTH
MONTHS
NEXT
NEXTVAL
NO
NONE
NOT
NULL
NULLS
NUMPARTS
OBID
OF
OFFSET1
OLD
ON
OPEN
OPTIMIZATION
OPTIMIZE
OR
ORDER
ORGANIZATION
OUT
OUTER
PACKAGE
PARAMETER
PART
PARTITION
PARTITIONED
PARTITIONING
PATH
PERIOD
PIECESIZE
PLAN
PRECISION
PREPARE
PREVVAL
PRIOR
PRIQTY
PRIVILEGES
PROCEDURE
PROGRAM
PSID
PUBLIC
QUERY
QUERYNO
READS
REFERENCES
REFRESH
RELEASE
RENAME
REPEAT
RESTRICT
RESULT
RESULT_SET_
LOCATOR
RETURN
RETURNS
REVOKE
RIGHT
ROLE
ROLLBACK
ROUND_CEILING
ROUND_DOWN

ROUND_FLOOR
ROUND_HALF_
DOWN
ROUND_HALF_
EVEN
ROUND_HALF_
UP
ROUND_UP
ROW
ROWSET
RUN
SAVEPOINT
SCHEMA
SCRATCHPAD
SECOND
SECONDS
SECQTY
SECURITY
SELECT
SENSITIVE
SEQUENCE
SESSION_USER
SET
SIGNAL
SIMPLE
SOME
SOURCE
SPECIFIC
STANDARD
STATEMENT
STATIC
STATEMENT
STAY
STOGROUP
STORES
STYLE
SUMMARY
SYNONYM
SYSDATESYSTEM
SYSFUN
SYSIBM
SYSPROC
SYSTEM
SYSTIMESTAMP
TABLE
TABLESPACE
THEN
TO
TRIGGER
TRUNCATE
TYPE
UNDO
UNION
UNIQUE
UNTIL
UPDATE
USER
USING
VALIDPROC
VALUE
VALUESVARIABLE
VARIANT
VCAT
VERSIONING
VIEW
VOLATILE
VOLUMES
WHEN
WHENEVER
WHERE
WHILE
WITH
WLM
XMLCAST
XMLEXISTS
XLMLNAME
SPACES
YEAR
YEARS
ZONE

Data Types

The following table shows the data type byte counts:

Data Type	Byte Count (add 1 if nulls are allowed)
INTEGER or INT	4
SMALLINT	2
BIGINT	8
REAL DOUBLE or FLOAT	4 (if n from 1 to 21) 8 (if n from 22 to 53)
DECIMAL or NUMERIC(x,y)	INTEGER (x/2)+1 (x is the precision)
DECFLOAT(16)	9
DECFLOAT(34)	17
CHAR(n)	n (max 255)
CLOB	6
Inline CLOB	6 + inline byte count
DBCLOB	6
Inline DBCLOB	6 + (inline char count * 2)
GRAPHIC(n)	2n
Inline DBCLOB	6 + (inline char count * 2)
VARCHAR(n)	N + 2 (1 <= n <= 32704)
VARGRAPHIC(n)	2n + 2
LONG VARCHAR	2 x (INTEGER(INTEGER(m-i-k)/j)/2) where: m = max row size i = sum of all col byte counts which are not LONG j = num of LONG columns in table k = num of LONG columns that allow nulls
BLOB	6
Inline BLOB	6 + inline byte count
DATE	4
TIME	3
TIMESTAMP(*p*) WITHOUT TIME ZONE	INTEGER ((*p*+1)/2) + 7 (p is the precision)
TIMESTAMP(*p*) WITH TIME ZONE	INTEGER ((*p*+1)/2) + 9 (p is the precision)
LONG VARGRAPHIC	(see LONG VARCHAR above)
BINARY(n)	n (max 255)
VARBINARY	n + 2 (max 32704)
XML	(max 4000) UTF-8
ROWID	17
DISTINCT TYPE	The length of the source data type upon which the distinct type was based.

Predicate Processing Summary

The following table lists many of the simple predicates and tells whether those predicates are Indexable or Stage 1.

Note: A description of the terms is provided after the table.

Predicate Type	Indexable	Stage 1
COL = value	Y	Y
COL = noncol expr	Y	Y
COL IS NULL	Y	Y
COL op value	Y	Y
COL op noncol expr	Y	Y
COL BETWEEN value1 AND value2	Y	Y
COL BETWEEN noncol expr1 AND noncol expr2	Y	Y
COL BETWEEN expression1 AND expression2	Y	Y
COL LIKE 'pattern'	Y	Y
COL IN (list)	Y	Y
COL IS NOT NULL	Y	Y
COL LIKE host variable	Y	Y
T1.COL = T2 col expr	Y	Y
T1.COL op T2 col expr	Y	Y
COL=(noncor subq)	Y	Y
COL = ANY (noncor subq)	Y	Y
COL op ALL (noncor subq)	Y	Y
COL IN (noncor subq)	Y	Y
(COL1,...COLn) IN (noncor subq)	Y	Y
COL = ANY (cor subq)	Y	Y
COL IN (cor subq)	Y	Y
COL IS NOT DISTINCT FROM value	Y	Y
COL IS NOT DISTINCT FROM noncol expr	Y	Y
T1.COL1 IS NOT DISTINCT FROM T2 col expr	Y	Y
COL IS NOT DISTINCT FROM (noncor subq)	Y	Y
XMLEXISTS	Y	N
COL <> value	N	Y
COL <> noncol expr	N	Y
COL NOT BETWEEN value1 AND value2	N	Y
COL NOT BETWEEN noncol exp1 AND noncol	N	Y
COL NOT IN (list)	N	Y
COL NOT LIKE ' char'	N	Y
COL LIKE '%char'	N	Y
COL LIKE '_char'	N	Y
T1.COL <> T2 col expr	N	Y
COL <> (noncor subq)	N	Y
COL IS DISTINCT FROM value	N	Y
COL IS DISTINCT FROM noncol expr	N	Y
T1.COL1 IS DISTINCT FROM T2 col expr	N	Y
COL IS DISTINCT FROM (noncor subq)	N	Y
value BETWEEN COL1 AND COL2	N	N
COL BETWEEN COL1 AND COL2	N	N

Predicate Type	Indexable	Stage 1
value NOT BETWEEN COL1 AND COL2	N	N
T1.COL1 = T1.COL2	N	N
T1.COL1 op T1.COL2	N	N
T1.COL1 <> T1.COL2	N	N
COL = ALL (noncor subq)	N	N
COL <> ANY (noncor subq)	N	N
COL <> ALL (noncor subq)	N	N
COL NOT IN (noncor subq)	N	N
(COL1,...COLn) NOT IN (noncor subq)	N	N
COL = (cor subq)	N	N
COL = ALL (cor subq)	N	N
COL op (cor subq)	N	N
COL op ANY (cor subq)	N	N
COL op ALL (cor subq)	N	N
COL <> (cor subq)	N	N
COL <> ANY (cor subq)	N	N
COL <> ALL (cor subq)	N	N
(COL1,...COLn) IN (cor subq)	N	N
COL NOT IN (cor subq)	N	N
(COL1,...COLn) NOT IN (cor subq)	N	N
T1.COL1 IS DISTINCT FROM T2.COL2	N	N
T1.COL1 IS NOT DISTINCT FROM T2.COL2	N	N
COL IS NOT DISTINCT FROM (cor subq)	N	N
EXISTS (subq)	N	N
NOT EXISTS (subq)	N	N
expression = value	N	N
expression <> value	N	N
expression op value	N	N
expression op (subq)	N	N
NOT XMLEXISTS	N	N

The following terms are used in the previous table:

- *subq*
  A correlated or noncorrelated subquery
- *noncor subq*
  A non-correlated subquery
- *cor subq*
  A correlated subquery
- *Op*
  any of the operators >, >=, <, <=, ¬>, ¬<
- *Value*
  A constant, host variable, or special register
- *Pattern*
  Any character string that does not start with the special characters for percent (%) or underscore (_).
- *Char*
  Any character string that does not include the special characters for percent (%) or underscore (_).

- *Expression*
  Any expression that contains arithmetic operators, scalar functions, aggregate functions, concatenation operators, columns, constants, host variables, special registers, or date or time expressions.
- *noncol expr*
  A non-column expression, which is any expression that does not contain a column. That expression can contain arithmetic operators, scalar functions, concatenation operators, constants, host variables, special registers, or date or time expressions.
- *Tn col expr*
  An expression that contains a column in table Tn. The expression might be only that column.
- *Predicate*
  A predicate of any type

2. UTILITIES

On-line DB2

DSNU UTILITY

{(*utility-name*) INDSN(*dataset-name*[(*member*)])}
[CONTROL (NONE|*control-option*)]
[DB2I (NO|YES} [DISCDSN(*dataset-name*)]
[COPYDSN(*dataset*) [COPYDSN2(*dataset*)]]
[RCPYDSN1(*dataset*) [RCPYDSN2(*dataset*)]]
[RECDSN(*dataset*)]
[PUNCHDSN(*dataset*)] {EDIT (NO|SPF|TSO)}
{RESTART (NO|CURRENT|PHASE|PREVIEW)}
{SUBMIT (NO|YES|PROMPT)}
{SYSTEM (DSN|*subsystem-name*|*group-attach*)}
[UID(*utility-id*)]
{UNIT (SYSDA|*unit-name*)} [VOLUME(*volser*)] [LIB(*dataset*)]

BACKUP SYSTEM

[FULL|DATA ONLY]
[ALTERNATE_CP(*copy_pool*) |
DBBSG(*stogroup*) | LGBSG(*stogroup*)]
[ESTABLISH FCINCREMENTAL| END FCINCREMENTAL]
[FORCE] |
[DUMP *dumpclass-spec* |FORCE |
DUMPONLY [TOKEN(x'*byte-string*') | [*dumpclass-spec*]]

dumpclass-spec:

DUMPCLASS(dc1 dc2 dc3 dc4 dc5,)

CATENFM

Note: With the introduction of single-phase migration in DB2 12, the CATENFM utility is not needed for the migration process and is removed.

CATMAINT

UPDATE LEVEL (*catalog-level*)

[SCHEMA SWITCH (*schema-name,schema-name*)|
OWNER FROM(*owner,owner,*) TO ROLE |
VCAT SWITCH(*vcat,vcat*) | UTILX {BASIC|EXTENDED}

CHECK DATA

tablespace-spec [PART *integer*] [CLONE]
[INCLUDE XML TABLESPACES ALL | *xml-spec*]
[SHRLEVEL REFERENCE | CHANGE]
drain-spec SCOPE {PENDING| AUXONLY| ALL| REFONLY|
XMLSCHEMAONLY}
{AUXERROR REPORT|INVALIDATE}
{LOBERROR REPORT|INVALIDATE}
{XMLERROR REPORT|INVALIDATE}
FOR EXCEPTION IN *table-name1* USE *table-name2*
{DELETE [NO|YES {LOG YES|NO}} {EXCEPTIONS 0|*integer*}
{ERRDDN SYSERR|*ddname*} |
{WORKDDN SYSUT1 , SYSUT2 | *ddname1,ddname2* |
ddname1 [, SYSUT2] | [SYSUT1,*ddname2*]} |
[PUNCHDDN SYSPUNCH|*ddname*]
[SORTDEVT *device-type*]| [SORTNUM *integer*]

tablespace-spec:

TABLESPACE [*database-name.*]*tablespace-name*

xml-spec:

{(*tablespace-spec*|*xml-column-spec*)}

xml-column-spec:
TABLE [*schema-name*] *table-name*
XMLCOLUMN *column-name*

drain-spec:
[DRAIN_WAIT *integer*] [RETRY *integer*] [RETRY_DELAY *integer*]

CHECK INDEX
{LIST *listdef-name* | (*index-name* [PART *integer*] |
(ALL) TABLESPACE [*dbname.*]*tsname* [PART *integer*] [CLONE]
[SHRLEVEL [REFERENCE|CHANGE]
{DRAIN_WAIT *integer* | IRLMRWT *value* }
{RETRY *integer* | UTIMOUT} [RETRY_DELAY *integer*]
[SORTDEVT *device-type*] [SORTNUM *integer*]
{PARALLEL 0 | *num-subtasks*}

CHECK LOB
lob-tablespace-spec [SHRLEVEL REFERENCE|CHANGE]
drain-spec [EXCEPTIONS 0 | integer**]**
[PUNCHDDN SYSPUNCH | ddname]
[SORTDEVT *device-type*] [SORTNUM *integer*]

lob-tablespace-spec:
TABLESPACE [*database.*]*tablespace* [CLONE]

drain-spec:
[DRAIN_WAIT *integer*] [RETRY *integer*] [RETRY_DELAY *integer*]

COPY
{*copy-spec* | *concurrent-spec* | *filterddn-spec*}
[CLONE]
[SHRLEVEL REFERENCE|CHANGE]
[SCOPE [ALL|PENDING]]

copy-spec:
[LIST *listdef-name* | *dataset-spec*] |
[FULL NO|YES|*changelimit-spec*]
[*tablespace-spec* | *indexname-spec*]
[FULL NO|YES|*changelimit-spec*]
[DSNUM ALL|*integer*]
dataset-spec
[PARALLEL [number-objects] TAPEUNITS[number]]
[CHECKPAGE] [SYSTEMPAGES YES|NO]
[FLASHCOPY NO|YES|CONSISTENT]
[FCCOPYDDN(*ddname*)]FULL

concurrent-spec:
[LIST *listdef-name* | *dataset-spec*] CONCURRENT |
[*tablespace-spec* | *indexname-spec*]
[DSNUM ALL|*integer*]
dataset-spec

filterddn-spec:
[LIST *listdef-name* | *dataset-spec*] |
FILTERDDN(*ddname*) CONCURRENT
[*tablespace-spec* | *indexname-spec*]
[DSNUM ALL|*integer*]

dataset-spec
COPYDDN (ddname2|ddname1[,ddname2]) |
[RECOVERYDDN (ddname4|ddname3[,ddname4)]]
[RECOVERYDDN (ddname4|ddname3[,ddname4])]]

changelimit-spec
CHANGELIMIT [(ANY|(*percent1*[,*percent2*])]
[REPORTONLY]

tablespace-spec:
TABLESPACE [*database-name.*]*tablespace-name*

indexname-spec:
INDEXSPACE [*database.*]*tablespace* |
INDEX [*creator.*]*index-name*

COPYTOCOPY
{LIST listdef-name *from-copy-spec dataset-spec* |
{*ts-num-spec* | *index-name-spec*} *from-copy-spec dataset-spec*}}
[CLONE]
TABLESPACE [*database.*] *tablespace-name*
DSNUM ALL | *integer*

ts-num-spec:
TABLESPACE [*database.*] *tablespace-name*
DSNUM ALL | *integer*

index-name-spec:
INDEXSPACE [*database.*] *indexspace* |
INDEX [*creator.*] *index-name* DSNUM ALL | *integer*

from-copy-spec:
{FROMLASTCOPY | FROMLASTFULLCOPY |
FROMLASTINCRCOPY |FROMCOPY *dsn* [FROMVOLUME
{CATALOG | *volser* {FROMSEQNO n]]}

dataset-spec
COPYDDN (ddname2 | ddname1[,ddname2]) |
[RECOVERYDDN (ddname4 | ddname3[,ddname4)]
[RECOVERYDDN (ddname4 | ddname3[,ddname4)]]

DIAGNOSE [TYPE(integer,)]
[ALLDUMPS (x'*abend-code*',] [NODUMPS (x'*abend-code*',)
[*display statement*] [*wait statement*] [*abend statement*]

display statement:
DISPLAY {OBD [*db.*] *tsname* [ALL | TABLES | INDEXES] |
SYSUTIL | MEPL | AVAILABLE |
DBET {DATABASE *database* | TABLESPACE {*db.*] *tsname* |
INDEX *index-name* [CLONE] }

wait statement:
WAIT {MESSAGE *message-id* [INSTANCE *integer*] |
TRACEID [*integer* | x'*trace-id*'] [INSTANCE *integer*]}

abend-statement:
WAIT {MESSAGE *message-id* [INSTANCE *integer*] |
TRACEID [*integer* | x'*trace-id*'] [INSTANCE *integer*]}

EXEC SQL {*declare-cursor-spec* | *non-select dynamic SQL statement*}
ENDEXEC

non-select dynamic SQL statements:
[ALTER | COMMENT ON | COMMIT | CREATE | DELETE |
DROP | EXPLAIN | GRANT | INSERT | LABEL ON |
LOCK TABLE | RENAME | REVOKE | ROLLBACK |
SET CURRENT DECFLOAT ROUNDING MODE |
SET CURRENT DEGREE |
SET CURRENT LOCALE LC_CTYPE |
SET CURRENT OPTIMIZATION HINT | SET PATH |
SET CURRENT PRECISION | SET CURRENT RULES |
SET CURRENT SQLID | UPDATE]

LISTDEF *list-name list-options*

list-options:

{INCLUDE | EXCLUDE} [*type-spec*]}
{LIST reference-list | *initial-object-spec* }
[CLONED YES|NO]
{DEFINED YES|NO|ALL} {RI} [BASE|LOB|XML|ALL]
[HISTORY|ARCHIVE] [BASIC NO|YES] EXTENDED NO|YES]

type-spec:

{TABLESPACES | INDEXSPACES [COPY NO|YES] }

initial-object-spec:

{DATABASE database |*tablespace-spec* | *indexspace-spec* |
table-spec | *index-spec* }
[PARTLEVEL (integer |integer1:integer2,)]

tablespace-spec:
TABLESPACE [dbname.] tablespace-name

indexspace-spec:
INDEXSPACE [dbname.] index-space-name

table-spec:
TABLE [creator-id.] table-name

index-spec:
INDEX [creator-id.] index-name]

LOAD [DATA]

{INDDN SYSREC | INDDN *ddname* | INCURSOR *cursor-name*}
[PREFORMAT] [COPYDICTIONARY 1 | *integer*]
[PRESORTED NO|YES]
[PARALLEL(*num sub-tasks*)] [ROWFORMAT BRF|RRF]
[RBALRSN_CONVERSION NONE | BASIC | EXTENDED]
[*resume-spec*] [*flashcopy-spec*] [KEEPDICTIONARY]
[REUSE] [LOG YES|NO[NOCOPYPEND]] *workddn-spec*
[SORTKEYS 0 |NO | *integer*] *format-spec* [FLOAT S390|IEEE]
[EBCDIC|ASCII|UNICODE] [CCSID *integer*] [NOSUBS]
[ENFORCE CONSTRAINTS|NO] [ERRDDN SYSERR | *ddname*]
[MAPDDN SYSMAP | *mapddn*] [DISCARDDN SYSDISC |
ddname]
[DISCARDS 0|*integer*] [SORTDEVT *device-type*]
[SORTNUM *integer*]
[CONTINUEIF(*start* [:*end*]) = X'*byte-string* | '*character-string*']
IGNORE(WHEN) | [DECFLOAT_ROUNDMODE
{ROUND_CEILING|
ROUND_DOWN | ROUND_FLOOR |
ROUND_HALF_DOWN | ROUND_HALF EVEN |
ROUND_HALF_UP | ROUND_UP}] |
override-spec | [INDEXDEFER NONE | NPI | ALL
[NONUNIQUE]]
[IMPLICIT_TZ '*timezone-string*']
INTO-TABLE-spec

resume-spec:

[RESUME NO [SHRLEVEL NONE] [REPLACE] [*copy-spec*]
[*statistics-spec*] | RESUME YES [SHRLEVEL NONE|CHANGE]]

labeled-duration-expression

[CURRENT_DATE |CURRENT_TIMESTAMP
{WITH TIME ZONE}]
[*constant* | YEAR | YEARS | MONTH |MONTHS |
DAY | DAYS |
HOUR | HOURS | MINUTE | MINUTES |
SECOND | SECONDS |
MICROSECOND | MICROSECONDS]

workddn-spec:

[WORKDDN (SYSUT1,SORTOUT) |
WORKDDN [*ddname,ddname2*) | (*ddname1*[,SYSUT1]) |
(*,ddname2*)]

copy-spec:

[COPYDDN (SYSCOPY | (*ddname1*[,*ddname2*])
RECOVERYDDN(*ddname3*[,*ddname4*]

flashcopy-spec:

[FLASHCOPY NO|YES|CONSISTENT [FCCOPYDDN(*ddname*)]]

statistics-spec:

STATISTICS [*stat-table-spec*] [*stat-index-spec*]
[REPORT NO|YES]
[UPDATE ALL|ACCESSPATH|SPACE|NONE]
[INVALIDATECACHE NO |YES]
[HISTORY ALL|ACCESSPATH|SPACE|NONE]
[FORCEROLLUP YES|NO]

stat-table-spec
TABLE [(ALL)] [SAMPLE *integer*] | [USE PROFILE]
TABLE (*table-name*)
[SAMPLE *integer*] [COLUMN ALL|(*column-name,*)]]
[COLGROUP(*column-name*) [*colgroup-stats-spec*]]
[USE PROFILE]

colgroup-stats-spec:
[FREQVAL COUNT *integer* [MOST|BOTH|LEAST]
[HISTOGRAM NUMQUANTILES 100|*integer*]

stat-index-spec
[INDEX(ALL) [*correlation-stats-spec*] |
(*index-name*) *correlation-stat-spec*]

correlation-stats-spec:
[KEYCARD [FREQVAL NUMCOLS1 COUNT10] |
[FREQVAL NUMCOLS *integer* COUNT *integer*] |
[HISTOGRAM [NUMCOLS 1 NUMQUANTILES 100] |
[NUMCOLS *integer* [NUMQUANTILES 100|*integer*]]

format-spec:

[FORMAT UNLOAD| SQL/DS| SPANNED YES|NO|
DELIMITED [COLDEL','|*coldel*] [CHARDEL ' " '|*chardel*]
[DECPT '.' |*decpt*]

into-table-spec:

INTO TABLE *table-name*
{IGNOREFIELDS NO|YES}
[PART *integer* [PREFORMAT] *resume-spec*
{INDDN SYSREC|*ddname* [DISCARDDN *ddname*] |
INCURSOR *cursor-name*}
[NUMRECS *integer*] [WHEN SQL/DS='*table-name*' |
(*field-selection-criteria*)

resume-spec:
RESUME YES|NO [REPLACE [REUSE] [*copy-spec*]]
[KEEPDICTIONARY]

field-selection-criteria:
{field-name | (start:end) = x'byte-string' | 'character-string'
|
G'graphic-string' | N'graphic-string'}

field-specification:
field-name {POSITION(start[:end])
[CHAR [BIT(length) *strip-spec* | MIXED *strip-spec* |
BLOBF [PRESERVE WHITESPACE] [BINARYXML] |
CLOBF [MIXED] [PRESERVE WHITESPACE] [CCSID 1200]|
DBCLOBF [PRESERVE WHITESPACE][CCSID 1200]]
[VARCHAR *strip-spec* | BIT |CCSID1200 |MIXED |
BLOBF [PRESERVE WHITESPACE] [BINARYXML] |
CLOBF [MIXED] [PRESERVE WHITESPACE] [CCSID 1200]
|
DBCLOBF [PRESERVE WHITESPACE] [CCSID 1200]]
[GRAPHIC [EXTERNAL] [length] *strip-spec*] |[CCSID 1200]|
[VARGRAPHIC *strip-spec*] |[CCSID 1200]|
[SMALLINT] | INTEGER [EXTERNAL [length]]]
[BIGINT | BINARY [(length)] | [DATE EXTERNAL] |
[BINARY[(length) *strip-spec*] | [VARBINARY *strip-spec*] |
[BINARY VARYING *strip-spec*] | [*decimal-spec*] |
[FLOAT [EXTERNAL] [(length)] | [DATE EXTERNAL] |
[TIMESTAMP EXTERNAL [length]] |
[TIMESTAMP WITH TIME ZONE EXTERNAL [length]] |
[ROWID | BLOB | CLOB [MIXED] [CCSID 1200] | DBCLOB]
[CCSID 1200] | [DECFLOAT (34) | (16) | EXTERNAL [length]] |
[XML [PRESERVE WHITESPACE] [BINARYXML]]
[NULLIF *field-selection-criteria*]
[DEFAULTID *field-selection-criteria*]

strip-spec:
[STRIP BOTH |TAILING | LEADING |
'*strip-char*' | X'strip-char']

decimal-spec:
[DECIMAL PACKED | ZONED |
EXTERNAL [(length,0) | (length,scale)]]

override-spec
[OVERRIDE(SYSTEMPERIOD|IDENTITY | TRANSID |
NONDETERMINISTIC]

MERGECOPY [LIST listdef-name |
TABLESPACE [dbname.] tablespace-name [DSNUM
ALL|integer]
[CLONE] [WORKDDN SYSUT1|ddname]
[NEWCOPY NO] [COPYDDN SYSCOPY |
COPYDDN (ddname1 [,ddname2]) | COPYDDN
(,ddname2) |
[RECOVERYDDN (ddname3 [,ddname4])]] |
[NEWCOPY YES] [COPYDDN SYSCOPY |
COPYDDN (ddname1 [,ddname2]) | COPYDDN
(,ddname2) |
[RECOVERYDDN (ddname3 [,ddname4])]]

MODIFY RECOVERY [LIST listdef-name |
TABLESPACE [dbname.] tablespace-name
[DSNUM ALL | integer] [CLONE]
{DELETE [AGE integer | (*)] | DATE integer | (*)] } |
RETAIN LAST (integer) | LOGLIMIT |
GDGLIMIT [LAST (integer) | LOGLIMIT]}

MODIFY STATISTICS [LIST listdef-name |
TABLESPACE [dbname.] tablespace-name |
INDEXSPACE [dbname.] indexspace-name |
INDEX [creator-id.] index-name]
{DELETE ALL|ACCESSPATH|SPACE}
{[AGE (integer) | (*)] | [DATE (integer) | (*)]}
{DELETEDS } {NOCOPYPEND}

OPTIONS

[PREVIEW | LISTDEFDD | TEMPLATEDD |
FILSZ | *event-spec*] |
OFF | KEY key-value

event-spec:

EVENT (ITEMERROR,HALT | ITEMERROR,SKIP [,]
[WARNING RC4 | RC0 | RC8])

QUIESCE

[LIST listdef-name |
{TABLESPACE [dbname.]tablespace-name [PART integer] |
TABLESPACESET [TABLESPACE [dbname.]tablespace-name]
[CLONE] [WRITE YES|NO] }

REBUILD

{INDEX [(creator-id.index-name [PART integer] , |
(ALL) *tablespace-spec* | LIST listdef-name] |
INDEXSPACE [([database.]indexspace-name
[PART integer] |
(ALL) tablespace-spec | LIST listdef-name] }
[SHRLEVEL REFERENCE | CHANGE *change-spec*] *drain-spec*
[CLONE] [SCOPE ALL | PENDING] [REUSE]
[SORTDEVT device-type] [SORTNUM integer] *stats-spec*
[FLASHCOPY NO | YES | CONSISTENT]
[FCCOPYDDN(ddname)]
[PARALLEL 0 | (num-subtasks)]
[RBALRSN_CONVERSION NONE | BASIC | EXTENDED]

tablespace-spec:

TABLESPACE [database.] tablespace-name [PART integer]

change-spec:

[MAXRO integer|DEFER]
[LONGLOG CONTINUE|TERM|DRAIN]
[DELAY 1200|integer]

drain-spec:

[DRAIN_WAIT *IRLMWRWT value* | *integer*]
[RETRY UTIMOUT value | integer] [RETRY_DELAY integer]

stats-spec:

[STATISTICS REPORT NO | YES] *correlation-stats-spec*
[UPDATE ALL | ACCESSPATH | SPACE | NONE]
[HISTORY ALL | ACCESSPATH SPACE | NONE]
[FORCEROLLUP YES | NO]

correlation-stats-spec:

[KEYCARD [FREQVAL NUMCOLS 1 COUNT 10] |
[FREQVAL NUMCOLS integer COUNT integer] |
[HISTOGRAM [NUMCOLS 1 NUMQUANTILES 100] |
[NUMCOLS integer [NUMQUANTILES 100 |integer]]

RECOVER

{LIST listdef-name |
object [DSNUM ALL|integer,] *list-options-spec* |
object [DSNUM ALL | integer] *recover-options-spec*|
object PAGE page-number [CONTINUE] }
[CLONE] [LOCALSITE | RECOVERYSITE]
[LOGRANGES YES|NO]

object:

{TABLESPACE [database.] tablespace-name |
INDEXSPACE [dataabse.] indexspace-name |
INDEX [creator-id.] index-name }

list-options-spec:

[BACKOUT NO | YES]
[TORBA x'byte-string' | TOLOGPOINT x'byte-string' |
[VERIFYSET YES | NO] [ENFORCE YES | NO]
[SCOPE UPDATED | ALL] |
[LOGONLY | *non-LOGONLY-options-spec*]

non-LOGONLY-options-spec:

[REUSE] [CONCURRENTCOPYONLY]
[PARALLEL [num-objects]]
[RESTOREBEFORE x'byte-string] [FROMDUMP [DUMPCLASS (dcl)]]
[FLASCOPY_PPRCP NO | PMNO |PMPREF |PMREQ]
[ALTERNATE_CP (*copy-pool*)]

recover-options-spec:

{[TOCOPY dataset [*image-copy-spec* | *tocopy-options-spec]*]
[TOLASTCOPY [REUSE] [CURRENTONLYCOPY]
[ENFORCE YES | NO]]
[TOLASTFULLCOPY [REUSE] [CURRENTCOPYONLY] [ENFORCE YES | NO]}

tocopy-options-spec:

[REUSE] [CURRENTCOPYONLY] [ENFORCE YES | NO]] |
[NOSYSCOPY |INLCOPY | FCCOPY] |
[FLASHCOPY_PPRCP NO|PMNO| PMPREF | PMREQ]

image-copy-spec:

[TOVOLUME [CATALOG | volser [TOSEQNO integer]]]

REORG INDEX

{[LIST listdef-name | *index-name-spec*] [REUSE] [CLONE]}
[[SHRLEVEL REFERENCE | NONE]
[*deadline-spec*] [*drain-spec*] [*change-spec*]
[FASTSWITCH YES|NO] [FORCE NONE| READERS | ALL]
[LEAFDISTLIMIT [integer] [REPORTONLY]
[UNLOAD CONTINUE | PAUSE | ONLY] [*stats-spec*]
[SORTDEVT *device-type*] | [SORTNUM *integer*]
[WORKDDN(SYSUT1) | (ddname)] [PREFORMAT]
[FLASHCOPY NO|YES|CONSISTENT] [FCCOPYDDN(ddname)
[RBALRSN_CONVERSION NONE | BASIC | EXTENDED]

indexname-spec:

{[INDEX [creator-id.] index-name |
INDEXSPACE [database.] index-name]
[PART integer] }

deadline-spec:

[DEADLINE NONE | timestamp | labeled-duration-expression]

drain-spec:

[DRAIN_WAIT integer] [RETRY integer]
[RETRY_DELAY integer]
[TIMEOUT TERM | ABEND]

change-spec:

[MAXRO integer | DEFER] [DRAIN ALL | WRITERS]
[LONGLOG CONTINUE | TERM | DRAIN]
[DELAY 1200 | integer] [LOGRANGES YES | NO]
[SWITCHTIME NONE | timestamp |
labeled-duration-expression]
[NEWMAXRO NONE | integer]

labeled-duration-expression:

[CURRENT_DATE |
CURRENT_TIMESTAMP [WITH TIME ZONE]]
[+ | -] constant [YEAR | YEARS | MONTH | MONTHS |
DAY | DAYS | HOUR | HOURS | MINUTE | MINUTES |
SECOND | SECONDS | MICROSECOND | MICROSECONDS]

stats-spec:

STATISTICS [REPORT NO | YES] [correlation-stats-spec]
[UPDATE ALL | ACCESSPATH | SPACE | NONE]
[INVALIDATECACHE NO | YES]
[HISTORY ALL | ACCESSPATH SPACE | NONE]
[FORCEROLLUP YES | NO]

correlation-stats-spec:

[KEYCARD [FREQVAL NUMCOLS 1 COUNT 10] |
[FREQVAL NUMCOLS integer
COUNT integer MOST | BOTH | LEAST] |
[HISTOGRAM [NUMCOLS 1 NUMQUANTILES 100] |
[NUMCOLS integer [NUMQUANTILES 100 |integer]]

REORG TABLESPACE

{LIST listdef-name [LISTPARTS n] |
[database.] tablespace-name
[PART(integer | integer1:integer2,)}
[CLONE] [REUSE] [SCOPE ALL | PENDING]
[REBALANCE SORTCLUSTER NO |SORTCLUSTER YES]
[LOG YES|NO]
[DROP_PART NO | YES]
[SORTDATA |SORTDATE NO [RECLUSTER YES | NO]]
[NOSYSREC] *copy-spec* [AUTOESTSPACE YED | NO]
[SHRLEVEL NONE |
SHRLEVEL REFERENCE [*deadline-spec*] [*drain-spec*]
[*change-spec*]
[FASTSWITCH YES | NO] [AUX NO | YES]
[FORCE NONE | READERS | ALL] [SORTNPSI AUTO | YES |
NO]
[OFFPOSLIMIT 10 | integer]
[INDREFLIMIT 10 | integer] [REPORTONLY]
[UNLOAD [CONTINUE | PAUSE]
[KEEPDICTIONARY] [*statistics-spec*] |
[PUNCHDDN SYSPUNCH | ddname]
[DISCARDDN SYSDISC | ddname]
[reorg tablespace options]
[DISCARD NOPAD [YES] | NO] [FROM-TABLE-SPEC]
[PARALLEL 0 | (num-subtasks)]

copy-spec

{COPYDDN(SYSCOPY) |ddname1 [,ddname2) | (,ddname2)
[RECOVERYDDN(ddname3 [,ddname4])
[FLASHCOPY NO |YES | CONSISTENT]
[FCCOPYDDN(ddname)]

deadline-spec

[DEADLINE NONE | timestamp | *labeled-duration-expression*]

drain-spec:
[DRAIN_WAIT integer] [RETRY integer] [RETRY_DELAY integer]
[TIMEOUT TERM | ABEND] [LOGRANGES YES | NO]
[DRAIN_ALLPARTS NO | YES]
[SWITCHTIME NONE | timestamp |
labeled-duration-expression]
[NEWMAXRO NONE | integer]

change-spec:
[MAXRO integer | DEFER] [DRAIN ALL | WRITERS]
[LONGLOG CONTINUE | TERM | DRAIN]
[DELAY 1200 | integer]

map-spec:
[MAPPINGTABLE table-name] |
[MAPPINGDATABASE database-name]

labeled-duration-expression:
[CURRENT_DATE |
CURRENT_TIMESTAMP [WITH TIME ZONE]]
[+ | -] constant [YEAR | YEARS | MONTH | MONTHS |
DAY | DAYS | HOUR | HOURS | MINUTE | MINUTES |
SECOND | SECONDS | MICROSECOND |
MICROSECONDS]

statistics-spec
STATISTICS [*stat-table-spec*] [*stat-index-spec*]
[REPORT NO | YES]
[UPDATE ALL |ACCESSPATH | SPACE | NONE]
[INVALIDATECACHE NO | YES]
[HISTORY ALL | ACCESSPATH | SPACE | NONE]
[FORCEROLLUP YES | NO]

stat-table-spec
TABLE [(ALL) [SAMPLE integer] |
TABLE (table-name) [*table-stats-spec*] [USE PROFILE]

table-stats-spec
SAMPLE *integer* [COLUMN ALL | (column-name,)]]
[COLGROUP(column-name) [*colgroup-stats-spec*]]
[INDEX(ALL) [*correlation-stats-spec*] |
(index-name] correlation-stats-spec)

colgroup-stats-spec:
[FREQVAL COUNT integer [MOST| BOTH | LAST]
[HISTOGRAM NUMQUANTILES 100 | integer]

stat-index-spec
[INDEX(ALL) [*correlation-stats-spec*] |
INDEX (index-name] *correlation-stats-spec*)

correlation-stats-specs:
[KEYCARD [FREQVAL NUMCOLS 1 COUNT 10] |
[FREQVAL NUMCOLS integer COUNT integer
[MOST | BOTH | LEAST]] |
[HISTOGRAM [NUMCOLS 1 NUMQUANTILES 100] |
[NUMCOLS integer [NUMQUANTILES 100 |integer]]

FROM-TABLE spec:
{FROM TABLE table-name [WHEN (*selection-condition-spec*)}

selection-condition-spec:
[predicate] | *selection condition*] [AND | OR]
[*predicate*] | selection condition]

predicate:
[basic | BETWEEN | IN | LIKE | NULL predicate]

reorg tablespace options
[UNLDDN SYSREC | ddname] [SORTDEVT device-type]
[SORTNUM integer] [PREFORMAT] [ROWFORMAT BRF | RRF]
[RBALRSN_CONVERSION NONE | BASIC | EXTENDED]

REPAIR
{OBJECT] [LOG YES|NO] [dbd-statement |locate-block |
set statement | *level-id statement* | *versions statement* |
catalog statement] [CLONE]}

level-id statement:
{LEVELID} [*indexname-spec*] |
[TABLESPACE [database.] tablespace-name]
[PART integer]

catalog statement:
{CATALOG TABLESPACE [database.] tablespace-name} [TEST]

index-name-spec:
{INDEX [creator-id.] index-name |
INDEXSPACE [database.] indexspace-name}

set statement:
{SET} tablespace-spec [PART integer]
[NOCOPYPEND | NORCVRPEND | NOCHECKPEND |
NOAUXWARN | NOAUXCHKP | NOAREORPENDSTAR |
NOAREORPEND | PRO | NOPRO] |
INDEX [(index-name [PART integer])|
ALL) [tablespace-spec] |
INDEXSPACE ([database.] indexspace-name
[PART integer] |
(ALL) tablespace-spec)
[NOCOPYPEND | NORCVRPEND | NORBDPEND |
NOCHECKPEND | NOAREORPENDSTAR |
NOAREORPEND | RBDPEND | PSRBDPEND]

tablespace-spec:
TABLESPACE [database.] tablespace-name
table-options-spec

locate block:
{LOCATE *tablespace-spec* |
INDEX index-name *index-options-spec* |
INDEXSPACE indexspace-name *index-options-spec*
[*verify statement* | *replace statement* |
delete statement | *dump statement*]
[SHRLEVEL CHANGE] |
LOB-tablespace-spec | *xml-tablespace-spec*

LOB-tablespace-spec:
{TABLESPACE} [database.] tspace-name ROWID 'x'byte-string'
VERSION x'byte-string' {*delete statement* | *dump statement*}

xml-tablespace-spec:
{TABLESPACE} [database.] xml-tablespace-name
DOCID x'byte-string' *delete statement*

table-options-spec:
{PAGE} x'byte-string' | [PART integer] PAGE integer |
RID x'byte-string' | KEY literal INDEX index-name

index-options-spec:
{PAGE} x'byte-string' | [PART integer] PAGE integer |

verify statement:
{VERIFY OFFSET 0 | integer | x'byte-string'
DATA x'byte-string' | 'character-string'}

replace statement:
{REPLACE RESET | OFFSET 0 | integer | x'byte-string'
DATA x'byte-string' | 'character-string'}

delete statement:
{DELETE} [DATAONLY]

dump statement:
{DUMP} [OFFSET 0 | integer | x'byte-string'
[LENGTH x'byte-string' | integer]
[PAGES x'byte-string' | integer | *] |
MAP [pages] | DATA [pages]

dbd statement:
{DBD} REBUILD | DIAGNOSE | TEST DATABASE database
[OUTDDN ddname] |
DROP DATABASE database DBID x'dbid'

REPORT

RECOVERY [TABLESPACE LIST listdef-name |
tablespacename-spec
[INDEX NONE|ALL] | *index-list-spec* [*info options*]] |
TABLESPACESET [TABLESPACE] *tablespace-name-spec*
[SHOWDSNS]

index-list-spec:

{INDEXSPACE} [database.] ixspace-name | LIST listdef-name
{INDEX} [creator-id.] index-name | LIST listdef-name

info-options:

[DSNUM ALL | integer] [CURRENT] [SUMMARY] [LOCALSITE]
[RECOVERYSITE] [ARCHLOG 1 | 2 | ALL]
[database.] tablespace-name

RESTORE SYSTEM

[ALTERNATE_CP (*copy-pool*)] [RESTORBEFORE X'*byte-string*]
[LOGONLY [SWITCH VCAT [SYSVALUEDDN(ddname)]]] |
[FROMDUMP [DUMPCLASS(dcl) [RSA('key-label')]
[TAPEUNITS[(num-tape-units)]]
[FLASHCOPY_PPRCP NO | PMNO|PMPREF | PMREQ]

RUNSTATS TABLESPACE

{LIST listdef-name | [database.] tablespace-name}
[statistics-spec | reset-spec] [PART integer]

statistics-spec:

[PART integer] [FORCEROLLUP NO | YES]
[INVALIDATECACHE NO | YES]
[table-spec]
[[INDEX(ALL) *correlation-stats-spec* |
[INDEX(*) | *correlation-stats-spec* |
[INDEX(index-name *correlation-stats-spec*)]]
[SHRLEVEL CHANGE REGISTER YES | NO | REFERENCE]
[REPORT NO | YES]
[UPDATE ALL | ACCESSPATH | SPACE | NONE]
[*history-spec*] [SORTDEVT device-type]

table-spec:
{TABLE}
(ALL) DELETE PROFILE | *sample-spec* USE PROFILE [INCLUDE NPI]
(table-name) *sample-spec column-spec* [*colgroup-spec*] |
USE PROFILE [INCLUDE NPI] |
column-spec [*colgroup-spec*] *prof-spec* |
DELETE PROFILE

sample-spec:
[SAMPLE 25 | integer] |
[TABLESAMPLE SYSTEM AUTO | numeric-literal
[REPEATABLE-integer]]

column-spec:
[COLUMN(ALL) | COLUMN(column-name,)] [SORTNUM integer]

colgroup-spec:
COLGROUP(column-name,) *colgroup-stats-spec*

colgroup-stats-spec:
[FREQVAL COUNT integer [MOST| BOTH | LEAST]
[HISTOGRAM NUMQUANTILES 100 | integer]

correlation-stats-spec:
[KEYCARD [FREQVAL NUMCOLS 1 COUNT 10] |
[FREQVAL NUMCOLS integer COUNT integer] |
[HISTOGRAM [NUMCOLS 1 NUMQUANTILES 100] |
[NUMCOLS integer [NUMQUANTILES 100 |integer]]

prof-spec:
[SET PROFILE [FROM EXISTING STATS]] | UPDATE PROFILE

history-spec:
[HISTORY NONE |ALL | ACCESSPATH | SPACE]

reset-spec:
[RESET ACCESSPATH [HISTORY ACCESSPATH]]

RUNSTATS INDEX
{LIST listdef-name *correlation-stats-spec* |
(index-name [PART integer) *correlation-stats-spec* |
(ALL) TABLESPACE [database.] tablespace-name
correlation-stats-spec}
[SHRLEVEL CHANGE | REFERENCE] [REPORT NO | YES]
[UPDATE ALL | ACCESSPATH | SPACE | NONE]
[SORTDEVT device-type] [SORTNUM integer]
[HISTORY NONE |ALL | ACCESSPATH | SPACE]
[FORCEROLLUP NO | YES]

correlation-stats-spec:
[KEYCARD [FREQVAL NUMCOLS 1 COUNT 10 MOST] |
[FREQVAL NUMCOLS integer COUNT integer]
[MOST | BOTH | LEAST] |
[HISTOGRAM [NUMCOLS 1 NUMQUANTILES 100] |
[NUMCOLS integer [NUMQUANTILES 100 |integer]]

STOSPACE STOGROUP ([stogroup-name, | *])

TEMPLATE
{template-name}
{DSN *name-expression* [*common-options*]
[*disk-options* | *tape-options*] [*SUBSYS-spec*] |
path-expression[paranthetical-expression]}

name-expression:
{ *qualifier-expression* } [paranthetical-expression]

qualifier-expression:
{character-expression | &variable [(start [,length])]

common-options:
[UNIT SYSALLDA | name] [MODELDCB] [BUFNO 30 | integer]
[DATACLAS name] [MGMTCLAS name] [STORCLAS name]
[RETPD integer | EXPDL 'date'] [VOLUMES(volser,)]
[VOLCNT integer] [UNCNT integer] [GDGLIMIT 99 | integer]
[DISP (NEW | OLD | SHR | MOD , DELETE | KEEP | CATLG |
UNCATLG , DELETE | KEEP | CATLG | UNCATLG)]
[LIMIT(n CYL | GB | MB , new-tamplate)] [TIME LOCAL |
UTC]

disk-options:
[SPACE CYL | SPACE(primary,secondary) CYL | TRK | MB]
[PCTPRIME 100 |integer] [MAXPRIME]
[NBRSECND 10 | integer] [DIR integer]
[DSNTYPE LIBRARY | PDS | HFS | NULL | BASIC | LARGE |
EXTREQ | EXTPREF] [EATTR]

tape-options:
[STACK NO | YES] [TRTCH NONE | COMP | NOCOMP]

SUBSYS-spec:
{SUBSYS name LRECL integer RECFM F | FB | V | VB}

path-expression:
{PATH pathname} [FILEDATA RECORD RECFM VB
LRECL 32756 |
FILEDATA TEXT | BINARY RECFM VB | V | FB | F
LRECL integer]
[PATHOPTS(ORDONLY | OCREAT,OWRONLY) |
PATHOPTS(ORDONLY | OCREAT | OWRONLY |
ONONBLOCK,)]
[PATHMODE (SIRUSR) | (SIRUSR | SIWUSR | SIXUSR |
SIRWXU | SIRGRP | SIWGRP | SIXGRP | SIRWXG |
SIROTH | SIWOTH | SIXOTH | SIRWXO ,)
[PATHDISP (KEEP,KEEP) |(KEEP | DELETE , KEEP , DELETE)]

UNLOAD

{DATA *from-table-spec* | *source-spec* from-table-spec|
LIST listdef-name *unload-spec* [CLONE] }

source-spec:
{TABLESPACE [database.] tablespace-name}
[PART integer | int1 : int2]
[FROMCOPY dataset-name [FROMVOLUME CATALOG |
vol-ser [FROMSEQNO n]] | FROMCOPYDDN ddname]

unload-spec:
[PUNCHDDN SYSPUNCH | ddname | template-name]
[UNLDDN SYSREC | ddname | template-name]
[EBCDIC | ASCII | UNICODE] [CCSID integer,] [NOSUBS]
[NOPAD] [SPANNED NO | YES] [FORMAT INTERNAL]
[DELIMITED] [COLDEL ',' | coldel] [CHARDEL '" ' | chardel]
[DECPT '.' | decpt] [FLOAT S390 | IEEE]
[MAXERR 1 | integer] [SHRLEVEL CHANGE ISOLATION CS
[SKIP LOCKED DATA] |
SHRLEVEL REFERENCE | CHANGE ISOLATION UR |
REGISTER NO | YES]
[DECFLOAT_ROUNDMODE
{ROUND_CEILING |ROUND_DOWN |ROUND_FLOOR |
ROUND_HALF_DOWN |ROUND_HALF EVEN |
ROUND_HALF_UP | ROUND_UP]
[IMPLICIT_TZ '*timezone-string*']
[PARALLEL 0 | num-subtasks]

FROM-TABLE-spec:

{FROM TABLE table-name}
[HEADER OBID | NONE | CONST 'string'] [SAMPLE decimal]

field-specification:

{field-name POSITION(* | start)}
CHAR[(length)] [TRUNCATE | DBCLOBF template-name | CLOBF template-name | BLOBF template-name] [BINARYXML] |
VARCHAR[(length)] [*strip-spec*| DBCLOBF template-name | CLOBF template-name | BLOBF template-name] [BINARYXML] |
GRAPHIC {EXTERNAL] [(length)] [TRUNCATE] |
VARGRAPHIC [(length)] *strip-spec* |
SMALLINT | INTEGER [EXTERNAL [(length)]] |
BIGINT | BINARY [(length) [TRUNCATE]] |
VARBINARY | BINARY VARYING |strip-spec |
DECIMAL [PACKED |ZONED | EXTERNAL] [(length)] |
FLOAT [EXTERNAL] [(length)] |DOUBLE | REAL |
DATE EXTERNAL [(length)] |
TIME EXTERNAL [(length)] |
TIMESTAMP EXTERNAL [(length)] |
timestamp with time zone-spec |
CONSTANT 'string | x'hex-string' | ROWID |
BLOB [(length)] [TRUNCATE] |
CLOB [(length)] [TRUNCATE] |
DBCLOB [(length)] [TRUNCATE] |
decfloat-spec | XML [BINARYXML]

strip-spec:

[STRIP BOTH |TRAILING |LEADING] ['strip-char'| x]strip-char']
[TRUNCATE]

timestamp with time zone-spec:

TIMESTAMP WITH TIME ZONE EXTERNAL [(length)]

decfloat spec:

DECFLOAT (34) | (16) | EXTERNAL [(length)]

selection condition:

[predicate|selection-condition)] [AND|OR]
[predicate|selection-condition)]

predicate:

[basic | BETWEEN | IN | LIKE | NULL predicate]

Stand-Alone Utilities

DSNJCNVB

(Converts BSDS to support 10,000 archive log volumes and 93 active log data sets per log copy)
//EXEC PGM=DSNJCNVB

DSNJCNVT

(Converts BSDS to support 10-byte RBA and LRSN fields)
//EXEC PGM=DSNJCNVT

DSNJLOGF

(preformat active log)
//EXEC PGM=DSNJLOGF

DSNJU003

(change log inventory)
//EXEC PGM=DSNJU003

NEWLOG statement:

{NEWLOG DSNAME=dataset-name
{*new active log* | *new archive log*}}
STARTIME=starttime,
ENDTIME=endtime
[,STARTRBA=startrba,ENDRBA=endrba]

new active log:

[,COPY1 | ,COPY2]
[,STARTRBA=startrba,ENDRBA=endrba]

new archive log:

[,COPY1VOL=vol-id |,COPY2VOL=vol-id]
,STARTRBA=startrba,ENDRBA=endrba,UNIT=unit-id
[,CATALOG=NO | YES]
[STRTLRSN=startlrsn,ENDLRSN=endlrsn]

DELETE statement:

{DELETE CCSIDS | DSNAME=dataset-name}
[,COPY1VOL=vol-id | ,COPY2VOL=vol-id]

CRESTART statement:

{CRESTART CANCEL | CREATE create-spec}

create-spec:

[,STARTRBS=startrba ,ENDRBA=endrba |
,ENDLRSN=endlrsn | ,SYSPITR=log-truncation-point |
,ENDTIME=log-truncation-timestamp |
,SYSPITRT=log=truncation-timestamp]
[CHKPT=chkptrba] [,CSRONLY | [,FORWARD=YES | NO]
[,BACKOUT=YES | NO]]

NEWCAT statement

{NEWCAT VSAMCAT=catalog-name}

DDF statement:

{DDF ip-spec | lu-spec | no-spec}

ip-spec:

[LOCATION=locname |PORT=port |RESPORT=resport |
SECPORT=secport | ALIAS=alias-name [: alias-port |
: alias-secport | : alias-port :alias-secport] |
IPNAME=ipname] |
[[IPV4=IPV4-address [,GRPIPV4=group-ipv4-addr]
[IPV6=IPV4-address [,GRPIPV6=group-ipv6-addr],]

lu-spec:

[LOCATION=locname | LUNAME=luname |
PASSWORD=password|GENERIC=gluname |PORT=port |
RESPORT=resport | ALIAS= alias-name [: alias-port]]

no-spec:
[NOPASSWD | NGENERIC | NOALIAS |
NOIPV4 , NGRPIPV4 | NOIPV6 , NGRPIVP6 |
NGRPIPV4 | NGRPIPV6 | NOIPNAME | NOLUNAME]

CHECKPT statement:
{CHECKPT STARTRBA=startrba [,CANCEL |
,ENDRBA=endrba ,TIME=time [,ENDLRSN=endlrsn]}

HIGHRBA statement:
{HIGHRBA [OFFLRBA=offlrba | STARTRBA=startrba
[,OFFLRBA=offlrba] ,TIME=time}

DELMBR statement:
{DELMBR [DEACTIV | DESTROY] MEMBERID=member-id}

RSTMBR statement:
{RSTMBR MEMBERID=member-id}

DSNJU004

(print log map)
//EXEC PGM=DSNJU004
[MEMBER * | DDNAME | (member-name,)]

DSN1COMP

(estimates space savings achieved by using DB2 data compression)

For table spaces:
{DSN1COMP} [32K | PAGESIZE (4K | 8K | 16K | 32K)]
[DSSIZE(integer G) | LARGE] [NUMPARTS(integer)]
[MAXROWS(integer)] [PCTFREE(integer)] [FULLCOPY]
[REORG] [ROWLIMIT(integer)] [MAXROWS(integer)]
[EXTNDICT(dictionary-name)]

For indexes:
{DSN1COMP} [LEAFLIM(integer)]

DSN1COPY

(used to copy DB2 related datasets)
{DSN1COPY} [CHECK] [32K | PAGESIZE(4K | 8K | 16K | 32K)]
[FULLCOPY | INCRCOPY | SEGMENT | INLCOPY]
[LARGE | LOB] [DSSIZE (integer G)]
[PIECESIZE(integer K | M | G)] [NUMPARTS(integer)]
[PRINT [hexadecimal-constant,hexadecimal-constant)
[EBDDIC| ASCII | UNICODE]]
[VALUE(string | hexadecimal-constant)]
[OBIDXLAT] [RESET]

DSN1LOGP

(formats recovery log)
[RBASTART(hex-constant)
[RBAEND (FFFFFFFFFFFFFFFFFFFF)| (hex-constant)] |
[LRSNSTART(hex-constant)
[LRSNEND (FFFFFFFFFFFFFFFFFFFF)| (hex-constant)]
[DATAONLY (NO) | (YES)] [SYSCOPY (NO) | (YES)]
[DBID(hex-constant)] [OBID(hex-constant)]
[PAGE(hex-constant)] [RID(hex-constant)]
[URID(hex-constant)] [LUWID(luwid)]
[TYPE(hex-constant) | SUBTYPE(hex-constant)]
[value/offset statement]
[SUMMARY (NO) | YES | ONLY [FILTER]) [CHECK(DATA)]

value/offset statement:
[VALUE(hex-constant) OFFSET(hex-constant)]

DSN1PRNT

(print VSAM datasets)
[[32K | PAGESIZE (4K | 8K | 16K | 32K)]
[FULLCOPY | INCRCOPY | INLCOPY] [LARGE | LOB]
[DSSIZE (integer G)]
[PIECESIZE(integer K | M | G)] [NUMPARTS(integer)]
[PRINT EBCDIC |(hex-constant,hex-constant)]
[EBCDIC | ASCII | UNICODE]
[VALUE(string |hex-constant)] [FORMAT [EXPAND]
[NODATE | NODATPGS]]

DSN1SDMP

(IBM software support might advise to use IFS selective dump)
{START TRACE (trace-parameters)
[SELECT function,offset,data-specification,]
[ACTION (action [x'00E60100') | (abend-code)) |
(STTRACE [,action [(x'00E60100') | (abend-code))]
[AFTER (1) |(integer)] [FOR (1) | (integer)]

second-trace-spec:
[ACTION2(action [(x'00E60100' | (abend-code)]]
[FILTER(ACE | EB)] [COMMAND command]
[AFTER2 (1) | (integer)] [FOR2 (1) | (integer)]
[SELECT2 function,offset,data-specification]

Authorities Required

The following table lists the authorities required by utility:

Utility	Authorization Required
BACKUP SYSTEM	SYSCTRL or SYSADM
CATENFM	Installation SYSADM
CATMAINT	Installation SYSADM
CHECK DATA	SYSADM, SYSCTRL, DBADM, DBCTRL, DBMAINT, DATAACCESS, or STATS
CHECK INDEX	SYSADM, SYSCTRL, System DBADM, DBADM, DBCTRL, DBMAINT, or STATS
CHECK LOB	SYSADM, SYSCTRL, System DBADM, DBADM, DBCTRL, DBMAINT, or STATS
COPY or COPYTOCOPY	SYSADM, SYSCTRL, System DBADMN, DBADM, DBCTRL, DBMAINT, or IMAGCOPY
DIAGNOSE	SYSADM, SYSCTRL, System DBADM, DATAACCESS, SQLADM, DBADM, DBCTRL, or REPAIR
DSNJCNVB	Data set access
DSNJU003	Data set access
DSNJU004	Data set access
DSN1COMP	Data set access
DSN1COPY	Data set access
DSN1LOGP	Data set access
DSN1PRNT	Data set access
DSN1SDMP	SYSADM, SYSOPR, SQLADM, TRACE, MONITOR1, or MONITOR2
EXEC SQL	None for stmt, but EXECUTE IMMEDIATE authority needed to run PREPARE
LISTDEF	SQLADM, DATAACCESS, System DBADM, SYSCTRL, SYSADM, SELECT on SYSINDEXES, SYSTABLES, SYSTABLESPACE and to run utility
LOAD	SYSADM, SYSCTRL, LOAD for database, STATS if STATISTICS specified, Table Ownership, DATAACCESS, DBCTRL or DBADM
MERGECOPY	SYSADM, SYSCTRL, System DBADM, DBADM, DBCTRL, DBMAINT, DATAACCESS or IMAGCOPY
MODIFY RECOVERY	SYSADM, SYSCTRL, System DBADM, DBADM, DBCTRL, DBMAINT, or IMAGCOPY for database
MODIFY STATISTICS	SYSADM, SYSCTRL, System DBADM, DBADM, DBCTRL. DBMAINT, SQLADM, STATS for database
OPTIONS	No privileges needed
QUIESCE	SYSADM, SYSCTRL, System DBADM, DBADM, DBCTRL, DBMAINT, or IMAGCOPY
REBUILD	SYSADM, SYSCTRL, System DBADM, DBCTRL, DBADM, RECOVERDB, DATAACCESS, or STATS if STATISTICS specified
RECOVER	SYSADM, SYSCTRL, System DBADM, DBADM, DBCTRL, DATAACCESS, or RECOVERDB
REORG	SYSADM, SYSCTRL, DBADM, DBCTRL, DATAACCESS, or REORG for database
REPAIR	SYSADM, SYSCTRL, DBADM, DBCTRL, DATAACCESS, or REPAIR for database
REPORT	SYSADM, SYSCTRL, System DBADM, DBADM, DBCTRL, DATAACCESS, or RECOVERDB for database

Utility	Authorization Required
RESTORE SYSTEM	SYSADM
RUNSTATS	SYSADM, SYSCTRL, System DBADM, DBADM, DBCTRL, DBMAINT, SQLADM, or STATS for database
STOSPACE	SYSADM, SYSCTRL, or STOSPACE
TEMPLATE	No privileges needed
UNLOAD	Ownership of the table, SELECT on table, DBADM for DB, DATAACCESS, SYSADM. Accessing catalog tables: System DBADM, SQLADM, ACCESSCTRL, SECADM or SYSCTRL

3. COMMANDS

DB2 Commands

-ACC or -ACCESS DATABASE

([[[*database-name*]], | * | *dbname1*:*dbname2* |
*dbname** | **dbname* | **dbname** |
dbstring1*dbstring2**])
SPACENAM ([[[*space-name*]], | * |
spacename1:*spacename2*|
*spacename** | **spacename* | **spacename** |
spacestring1*spacestring2**])
{PART ([*integer* | *integer1*:*integer2*],)}
MODE ([OPEN | NGBPDEP | STATS])

-ACTIVATE FUNCTION LEVEL

(function-level)
{ TEST }

-ALT BPOOL or -ALTER BUFFERPOOL

(*bpname*)
{{ VPSIZE (*integer*) | VPSIZEMIN ([* | *integer*]) |
VPSIZEMAX ([* | *integer*]), | FRAMESIZE ([4K | 1M | 2G]) |
VPSEQT (*integer*) | VPPSEQT (*integer*) |
DWQT (*integer*) | VDWQT (*integer1*,*integer2*) |
PGSTEAL ([LRU | FIFO | NONE]) | PGFIX ([NO | YES]) |
AUTOSIZE ([NO | YES]) | SPSIZE(*integer*) |
SPSEQT(*integer*) }}

-ALT GBPOOL or -ALTER GROUPBUFFERPOOL

([*gbpname* | *structure-name*])
{{ GBPCACHE ([YES | NO]) | AUTOREC ([YES | NO]) |
RATIO (*ratio*) |
CLASST (*class-threshold1*,*class-threshold2*) |
GBPOOLT (*integer*) | GBPCHKPT (*integer*) }}

-ALT UTIL or -ALTER UTILITY

(*utility-id*) [REBUILD | REORG]
{{ DEADLINE ([NONE | *timestamp*]) |
MAXRO ([*integer* | DEFER]) |
LONGLOG ([CONTINUE | TERM | DRAIN]) |
DELAY (*integer*) }}

-ARC LOG or -ARCHIVE LOG

{ MODE (QUIESCE) {TIME (*nnn*)} {WAIT ([NO | YES])} |
SCOPE ([MEMBER |GROUP]) |
CANCEL OFFLOAD }

-CAN THD or -CANCEL THREAD

[(*token*) | DDF THREAD ([*luwid* | *token*])]
{{ DUMP | LOCAL | NOBACKOUT | FORCE }}

-DIS ACCEL or -DISPLAY ACCEL

([* | *accelerator-name,*])
{{DETAIL | LIST ([ACTIVE | *]) | SCOPE ([LOCAL | GROUP]) |
MEMBER (*member-name*) }}

-DIS ARC or -DISPLAY ARCHIVE

-DIS BPOOL or -DISPLAY BUFFERPOOL

([ACTIVE | * | [[*bpname*]],])
{{ DETAIL (INTERVAL | *) | LIST (ACTIVE | *) |
LSTATS (ACTIVE | *) |
DBNAME ([* | *database-name*, |
name1:*name2* | *name**]) |
SPACENAM ([* | *space-name*, | *name1*:*name2* | *name**]) |
GBPDEP ([YES | NO]) CASTOWNR ([YES | NO]) }}

-DIS DB or -DISPLAY DATABASE

([*database-name*, | * | *dbname1*:*dbname2* | *dbname** |
dbname* | **dbname | **dbstring1*dbstring2**])
{ {USE | CLAIMERS | LOCKS | LPL | WEPR} |
SPACENAM ([*space-name*, | * |
spacename1:*spacename2* | *spacename** |
spacename* |spacename** |
spacestring1*spacestring2])
{USE | CLAIMERS | LOCKS | LPL | WEPR} {ONLY} | ONLY[1] }
{{ PART ([*integer*, | *integer1*:*integer2*]) | OVERVIEW[2] |
LIMIT ([50 | *integer* |*]) | AFTER | ACTIVE | *restrict-block* |
ADVISORY ([[ICOPY | AUXW | ARBDP | AREO* |
AREOR]],) }}

Notes: [1] If ONLY is specified without SPACENAM, only the LIMIT, AFTER, and RESTRICT keywords apply. [2] Specify OVERVIEW only with SPACENAM, LIMIT, and AFTER.

restrict-block:

RESTRICT ([[ACHKP | CHKP | COPY | GRECP | LPL | PRO |
RBDP | RECP | REORP | RO | RREPL | STOP | UT |
UTRO | UTRW | UTUT | UT* | WEPR]],)

-DIS DDF or -DISPLAY DDF

{{ ALIAS (*alias-name*) | DETAIL }}

-DIS DYNQUERY or -DISPLAY DYNQUERYCAPTURE

{ CNO ([* | *integer,*]) }

-DIS FUNC SPEC or -DISPLAY FUNCTION SPECIFIC

{ (*.* | {{*schema.specific-function-name* |
*schema.partial-name**}},) } { SCOPE (LOCAL | GROUP) }

-DIS GROUP or -DISPLAY GROUP

{DETAIL}

-DIS GBPOOL or -DISPLAY GROUPBUFFERPOOL

{{ ([* | {{*gbpname* | *structure-name*}},]) |
TYPE ([* | GCONN | MCONN | NOCACHE]) |
MDETAIL { (INTERVAL | *) } | GDETAIL { (INTERVAL | *) } |
CONNLIST ([NO |YES]) }}

-DIS LOC or -DISPLAY LOCATION

{ (* | [[*location-name* | *partial-location** | <*luname*> |
ipaddr]],) } | {DETAIL}

-DIS LOG or -DISPLAY LOG

-DIS PROC or -DISPLAY PROCEDURE

{ (*.* | [[*schema.procedure-name* |
*schema.partial-name** |
procedure-name | *partial-name**]],) }
{ SCOPE (LOCAL | GROUP) }

-DIS PROFILE or -DISPLAY PROFILE

-DIS RLIM or -DISPLAY RLIMIT

-DIS STATS or -DISPLAY STATS

([INDEXMEMORYUSAGE | IDXMEMUSE | IMU])
{ LIMIT ([integer | *]) }

-DIS THD or -DISPLAY THREAD

{ ([*connection-name* | *partial-connection**], | *) }
SCOPE ([LOCAL | GROUP])
{{ TYPE ([ACTIVE | INDOUBT | * | INACTIVE |
POSTPONED | PROC | SYSTEM]) }
{ LOCATION ([{{*location-name* | *partial-location**}}, | *]) |
LUWID ([[*luwid* | *partial-luwid** | *token*]],) |
ACCEL ([* | [[*accelerator-name*]],) } |
LIMIT ([512 | integer | *] | DETAIL |
RRSURID ([*rrs-urid*, | *])

-DIS TRACE or -DISPLAY TRACE

({* | PERFM | ACCTG | STAT | AUDIT | MONITOR})
DEST ([GTF | SMF | SRV | OP*n*],)
{{ *constraint-block* | *filtering-block* | DETAIL (*output-type*) |
COMMENT (*string*) | SCOPE ([LOCAL | GROUP]) |
RMID }} ASID (x'*dddd*')

constraint-block:

{{ PLAN ([* | *plan-name*,]) |
PKGLOC ([* | *package-location*, |
partial-package-location]) |
PKGCOL ([* | *package-collection-id*, |
partial-package-collection-id]) |
PKGPROG ([* | *package-program-name* |
partial-package-program-name]) |
AUTHID ([* | *auth-id*, | *partial-authid*]) |
CLASS ([* | *integer*,]) |
TNO ([* | *integer*,]) |
LOCATION ([* | *location-name*, | <*luname*> |
*partial <luname>** | *ipaddr* | *partial-ipaddr*]) |
USERID ([* | *userid*, | *partial-userid*)] |
APPNAME ([* | *application-name*, |
partial-application-name]) |
WKSTN ([* | *workstation-name*, | *partial-workstation-name*]) |
CONNID ([* | *connection-role-id*,| *partial-connection-role-id*]) |
CORRID ([* | *correlation-id*, | *partial-correlation-id*]) |
ROLE ([* | *connection-role*, | *partial-connection-role-id*]) |
AUDTPLCY[1] (*policy-name*,) }}

[1] You cannot specify CLASS or IFCID with AUDTPLCY.

filtering-block:
{{ XPLAN ([*plan-name,* | *partial-plan-name*]) |
XPKGLOC ([*package-location*, |
partial-package-location]) |
XPKGCOL ([*package-collection-id*, |
partial-package-collection-id]) |
XPKGPROG ([*package-program-name*, |
partial-package-program-name]) |
XAUTHID ([*authorization-id*, | *partial-authorization-id*]) |
XLOC ([*location-name*, | *partial-location-name* |
<luname> | *partial- <luname>** | *ipaddr* |
partial-ipaddr]) |
XUSERID ([*userid*, | *partial-userid*]) |
XAPPNAME ([*application-name*, | *partial-application-name*])
|
XWRKSTN ([*workstation-name*, | *partial-workstation-name*])
|
XCONNID ([*connection-role-id*, | *partial-connection-role-id*])
|
XCORRID ([*correlation-id*, | *partial-correlation-id*]) |
XROLE ([*connection-role*, | *partial-connection-role-id*]) }}

-DIS UTIL or -DISPLAY UTILITY
([*utility-id* | *partial-utility-id** | *])
{ MEMBER (*member-name*,) }

-MODIFY DDF
[ALIAS (*alias-name*) [ADD | DELETE | START | STOP |
CANCEL |
PORT (*port-name*) | SECPORT (*secport-name*) |
NPORT|NSECPORT | IPV4 (*ipv4-address*) |
IPV6 (*ipv6-address*) | NIPV4 | NIPV6] |
PKGREL ([BNDOPT | BINDPOOL | COMMIT]) |
SESSIDLE (session-idle-limit)]

-MOD TRA or -MODIFY TRACE
([PERFM | ACCTG | STAT | AUDIT | MONITOR])
CLASS ([* | *integer*,]) TNO (*integer*)
IFCID ([* | *ifcid-nbr*,]) { COMMENT (*string*) }

-REC BSDS or -RECOVER BSDS

-REC IND or -RECOVER INDOUBT
{ (*connection-name*) }
ACTION ([COMMIT | ABORT])
[ID ([*correlation-id*, | *]) | NID (*network-id*,) |
LUWID ([*luwid* | *token*],)]

-REC POST or -RECOVER POSTPONED
{CANCEL}

-REF DB2 or -REFRESH DB2
,EARLY

-RESET GENERIC or -RESET GENERICLU
[([[*luname* | *netid.luname*]], | *)]

-RESET IND or -RESET INDOUBT
[LUNAME ([luname, | *]) {FORCE} |
LOCATION (*location-name*,) |
IPADDR ([*ipaddr..port*, | *]) {FORCE} |
LUWID ([*luwid* | *token*],) {LOCATION (*location-name*)}]

-SET ARC or -SET ARCHIVE

{{ COUNT (*integer*) } { TIME ([*minutes* {,*seconds*} |
,*seconds* | 1440 | NOLIMIT]) |
DEFAULT }}

-SET LOG

{ SINGLE[1] | BOTH }
[LOGLOAD (*integer*) {CHKTIME (*integer*)} |
CHKTIME (*integer*) {LOGLOAD (*integer*)} |
SUSPEND | RESUME |
NEWLOG (*dataset-name*) COPY (*log-copy*)]

[1] LOGLOAD or CHKTIME is required with SINGLE.

-SET SYSPARM

[LOAD (DZNPARM | *load-module-name*) | RELOAD |
STARTUP]

-STA ACCEL or -START ACCEL

([* | *accelerator-name,*])
{{ ACCESS (* | MAINT | EXPLAINONLY) |
SCOPE (LOCAL | GROUP) |
MEMBER (*member-name*,) }}

-STA DB or -START DATABASE

([*dbname*, | * | *dbname1*:*dbname2* | *dbname** |
dbname* | **dbname | **dbstring1*dbstring2**])
{{SPACENAM ([*space-name*, | * | *spacename1*:*spacename2*
| *spacename** | **spacename* |
spacename | **spacestring1*spacestring2*])
{PART ([*integer* | *integer1*:*integer2*],)} }}
{CLONE} [ACCESS ([RW | RO | UT | RREPL | FORCE])]

-STA DB2 or -START DB2

{{PARM (*module-name*) | DECP([DSNCEDCP | *decp-name*]) |
ACCESS ({* | MAINT}) |
LIGHT ({NO | YES | NOINDOUBTS | CASTOUT}) |
MSTR (*jcl-substitution*) | DBM1 (*jcl-substitution*) |
DIST (*jcl-substitution*)}}

-STA DDF or -START DDF

-STA DYNQUERY or -START DYNQUERYCAPTURE

STBLGRP (*stabilization-group*)
{ THRESHOLD ([2 | *integer-constant*]) cache-snap-spec |
STMTID (*integer-constant*) |
STMTTOKEN (string-*constant*) }

cache-snap-spec:

{{CURSQLID (* | SQLID) | MONITOR (NO | YES) |
SCOPE (LOCAL | GROUP)}}

-STA FUNC SPEC or -START FUNCTION SPECIFIC

{ (*.* | [*schema.specific-function-name* |
*schema.partial-name**],) }
SCOPE ([LOCAL | GROUP])

-STA PROC or - START PROCEDURE

{ (*.* | [[*schema.procedure-name* |
*schema.partial-name** | *procedure-name* |
*partial-name**]],) }
SCOPE ([LOCAL | GROUP])

-STA PROFILE or -START PROFILE

-STA RLIM or -START RLIMIT

{ ID=*id* }

-STA TRA or -START TRACE

([PERFM | ACCTG | STAT | AUDIT | MONITOR])
{{ DEST ([[GTF | SMF | SRV | *OPn* | OPX]],) |
constraint-block | *filtering block* | RMID |
COMMENT (*string*) | SCOPE ([LOCAL | GROUP]) }}

constraint-block:

{{ PLAN ([* | *plan-name*, | *partial-plan-name*]) |
PKGLOC ([* | *package-location*, | *partial-package-location*])
| PKGCOL ([* | *package-collection-id*, |
partial-package-collection-id]) |
PKGPROG ([* | *package-program-name,* |
partial-package-program-name]) |
AUTHID ([* | *authorization-id*, | *partial-authorization-id*]) |
CLASS ([* | *integer*,]) |
LOCATION ([* | *location-name*, | <*luname*>|
partial <*luname*>* | *ipaddr* | *partial-ipaddr*]) |
USERID (* | *userid*, | *partial-userid*)] |
APPNAME ([* | *application-name*, |
partial-application-name]) |
WKSTN ([* | *workstation-name*, | *partial-workstation-name*]) |
CONNID ([* | *connection-role-id*,| *partial-connection-role-id*]) |
CORRID ([* | *correlation-id*, | *partial-correlation-id*]) |
ROLE ([* | *connection-role*, | *partial-connection-role-id*]) |
IFCID ([* | *ifcid*,]) | BUFSIZE ([* | *k_bytes*])
AUDTPLCY[1] (*policy-name*,) |
TDATA ([[CORRELATION | TRACE | CPU |
DISTRIBUTED]],) | ASID(x'*dddd*') }}

[1] You cannot specify CLASS or IFCID with AUDTPLCY.

filtering-block:

[1]See DISPLAY TRACE**.**

-STO ACCEL or -STOP ACCEL

([*|*accelerator-name*,])
{{ MODE (QUIESCE | FORCE) |
SCOPE (LOCAL | GROUP) | MEMBER (*member-name*) }}

-STO DB or -STOP DATABASE

([*database-name*, | * | *dbname1:dbname2* | *dbname** |
dbname* | **dbname | **dbstring1*dbstring2**])
{{ SPACENAM ([*space-name*, | * |
spacename1:*spacename2* | *spacename** |
spacename* | **spacename |
spacestring1*spacestring2])
{ PART ([*integer* | *integer1*:*integer2*],) } {CLONE} }}
{ AT (COMMIT) }

-STO DB2 or -STOP DB2

{{ MODE ([QUIESCE | FORCE]) |
CASTOUT ([YES | NO]) }}

-STO DDF or -STOP DDF

{ MODE ([QUIESCE | FORCE |
SUSPEND { CANCEL(*n*) | WAIT(*n*) }]) }

-STO DYNQUERY or -STOP DYNQUERYCAPTURE

{ CNO ([* | integer,]) } SCOPE ([LOCAL | GROUP])

-STO FUNC SPEC or -STOP FUNCTION SPECIFIC

{{ (*.* | [[*schema.specific-function-name* |
*schema.partial-name**]],) |
ACTION ([QUEUE | REJECT]) |
SCOPE ([LOCAL | GROUP]) }}

-STO PROC or -STOP PROCEDURE

{{ (*.* | [[*schema.procedure-name* | *schema.partial-name** |
procedure-name | *partial-name**]],) |
ACTION ([QUEUE | REJECT]) |
SCOPE ([LOCAL | GROUP]) }}

-STO PROFILE or -STOP PROFILE

-STO RLIM or -STOP RLIMIT

-STO TRAC or -STOP TRACE

([PERFM | ACCTG | STAT | AUDIT | MONITOR | *])
DEST ([[GTF | SMF | SRV | OP*n*]],)
{{ *constraint-block* | *filtering block* | RMID |
COMMENT (*string*) | SCOPE ([LOCAL | GROUP]) }}
ASID (x'*dddd*')

constraint-block and filtering block:

See DISPLAY TRACE**.**

-TERM UTIL or -TERM UTILITY

([*utility-id* | *partial-utility-id** | *])

DSN Commands

BIND PACKAGE

({*location-name.*} *collection-id*)
{{ OWNER (*authorization-id*) |
QUALIFIER (*qualifier-name*) |
member-block | *enable-block* |
{{ DEFER(PREPARE) | DEFER (INHERITFROMPLAN)] |
NODEFER(PREPARE) }} |
ACTION [(REPLACE) {REPLVER (*version-id*)} | (ADD)] |
CONCENTRATESTMT ([NO | YES]) |
CURRENTDATA ([NO | YES]) |
DBPROTOCOL ([DRDA | DRDACBF]) |
DEGREE ([1 | ANY]) | DESCSTAT([NO | YES]) |
dynamicrules-block | *encoding-block* |
EXPLAIN ([NO | YES | ONLY]) | FLAG ([I | W | E | C]) |
GETACCELARCHIVE ([NO | YES]) |
IMMEDWRITE ([INHERITFROMPLAN | NO | YES]) |
ISOLATION[1] ([CS | RR | RS | UR | NC]) |
KEEPDYNAMIC ([NO | YES]) | *reopt-block* |
OPTHINT (‘*hint-id*’) | PATH ([*schema-name* | USER],)}}
rounding-block |
QUERYACCELERATION ([NONE | ENABLE |
ENABLEWITHFALLBACK |ELIGIBLE | ALL}) |
RELEASE ([COMMIT | DEALLOCATE | INHERITFROMPLAN]) |
SQLERROR ([NOPACKAGE | CONTINUE | CHECK]) |
VALIDATE ([RUN | BIND]) |
EXTENDEDINDICATOR ([NO | YES] |
CONCURRENTACCESSRESOLUTION
([USECURRENTLYCOMMITTED |
WAITFOROUTCOME]) |
APREUSE ([NONE | ERROR | WARN]) |
APCOMPARE ([NONE | WARN | ERROR]) | GENERIC(‘*string*’)
|
BUSTIMESENSITIVE ([YES | NO]) |
SYSTIMESENSITIVE ([YES | NO]) |
ARCHIVESENSITIVE ([YES | NO]) |
APPLCOMPAT ([*function-level* | V11R1 | V10R1]) }}

Note: [1] The default for a remote package is CS.

member-block:

[MEMBER (*dbrm-member-name*)
{LIBRARY (*dbrm-library-name*)} |
COPY (*collection-id.package-id*)
{{ COPYVER (*version-id*) |
OPTIONS([COMPOSITE | COMMAND]) }} |
DEPLOY (*collection-id.package-id*) COPYVER (*version-id*)]

enable-block:

{ [ENABLE(*)] | [ENABLE | DISABLE] ([[BATCH | DLIBATCH |
DB2CALL | CICS | IMS | IMSBMP | IMSMPP |
REMOTE | RRSAF]],)
{{ DLIBATCH (*connection-name*,) | CICS (*applid*,) |
IMSBMP (*imsid*,) | IMSMPP (*imsid*,) }}

dynamicrules-block:

{ DYNAMICRULES ([RUN | BIND | DEFINEBIND | DEFINERUN
| INVOKEBIND | INVOKERUN]) }

encoding-block:

{ ENCODING ([ASCII | EBCDIC | UNICODE | *ccsid*]) }

reopt-block:

{ REOPT ([NONE[1] | [ALWAYS | VARS] | ONCE | AUTO]) }

Notes: [1] NOREOPT(VARS) can be a synonym of REOPT(NONE).

rounding-block:

{ ROUNDING ([CEILING | DOWN | FLOOR | HALFDOWN | HALFEVEN | HALFUP | UP]) }

BIND PLAN

(*plan-name*)[1]
{{ OWNER (*authorization-id*) |
QUALIFIER (*qualifier-name*) |
enable-block | *pklist-block* |
{ NODEFER (PREPARE) | DEFER (PREPARE) }
ACQUIRE ([USE | ALLOCATE]) |
ACTION ([(REPLACE) {RETAIN} | (ADD]) |
CACHESIZE (*decimal-value*) |
CURRENTDATA ([NO | YES]) |
CURRENTSERVER (*location-name*) |
DBPROTOCOL (DRDA) | DEGREE ([1 | ANY]) |
DISCONNECT ([EXPLICIT | AUTOMATIC | CONDITIONAL]) |
DYNAMICRULES ([RUN | BIND]) |
ENCODING ([ASCII | EBCDIC | UNICODE | *ccsid*]) |
EXPLAIN ([NO | YES]) | FLAG ([I | W | E | C]) |
IMMEDWRITE ([NO | YES]) |
ISOLATION ([CS | RR | RS | UR]) |
KEEPDYNAMIC ([NO | YES]) |
REOPT ([NONE[2] | [ALWAYS | VARS] | ONCE | AUTO]) |
OPTHINT ('*hint-id*') | PATH([[*schema-name* | USER]],) |
RELEASE ([COMMIT | DEALLOCATE]) |
ROUNDING ([CEILING | DOWN | FLOOR |
HALFDOWN | HALFEVEN | HALFUP | UP]) |
SQLRULES ([DB2 | STD]) | VALIDATE ([RUN | BIND]) |
CONCURRENTACCESSRESOLUTION
([USECURRENTLYCOMMITTED |
WAITFOROUTCOME]) |
PROGAUTH (DISABLE | ENABLE) }}

Notes: [1]When PLAN is omitted no plan is produced and no rows are inserted into PLAN_TABLE. [2]NOREOPT(VARS) can be a synonym of REOPT(NONE).

*enable-block***:**

{ [ENABLE (*)] | [ENABLE | DISABLE]
([[BATCH | DLIBATCH | DB2CALL | CICS | IMS |
IMSBMP | IMSMPP | RRSAF]],)
{{DLIBATCH ([[*connection-name*]],) | CICS ([[*applid*]],) |
IMSBMP ([[*imsid*]],) | IMSMPP ([[*imsid*]],) }

pklist-block:

PKLIST ([[{*location-name.* | *.} [*collection-id* | *] .
[*package-id* | *]]],)

BIND QUERY

{{ LOOKUP ([NO | YES]) |
EXPLAININPUTSCHEMA (*schema-name*) }}

DCLGEN TABLE

([*table-name* | *view-name*])
{{ OWNER (*owner-name*) | AT (*location-name*) }}
LIBRARY (*library-name* {(*member-name*)} {/*password*})
{{ ACTION ([ADD | REPLACE]) |
LANGUAGE ([PLI | C | IBMCOB | CPP]) | NAMES (*prefix*) |
STRUCTURE (*structure-name*) | {APOST | QUOTE} |
LABEL ([NO | YES]) | DBCSSYMBOL ([G | N]) |
DBCSDELIM ([YES | NO]) | COLSUFFIX ([NO | YES]) |
INDVAR ([NO | YES]) | RMARGIN ([STD | WIDE]) |
DCLBIT ([NO | YES]) }}

END

FREE STABILIZED DYNAMIC QUERY

STBLGRP ([*stabilization-group* | *)
SDQSTMTID (*statement-id*)
{{ PLANMMTSCOPE ([ALL | INVALID}) |
INVALID ONLY (NO | YES) }}

FREE PACKAGE

([{*location-name*.} [*collection-id* | *] . [*package-id* | *]
{.(*version-id* | *)},) | (*)]
{{ FLAG ([I | W | E | C]) |
PLANMGMTSCOPE ([ALL | INACTIVE | PREVIOUS |
ORIGINAL]) {INVALIDONLY (NO | YES)} }}

FREE PLAN

([*plan-name*, | *]) { FLAG ([I | W | E | C])}

FREE QUERY

filter-block package-block

filter-block

[FILTER (‘*filter-name*’) | PACKAGE (*package-name*) |
QUERYID (*number*) | QUERYID (ALL)]

package-block:

{*location-name.*} [*collection-id* | *] . [*package-id* | *]
{. ([*version-id* | *]) }

REBIND PACKAGE

ID-block
{{OWNER (*authorization-id*) |
QUALIFIER (*qualifier-name*) | *enable-block* |
plan-management-block | *acceleration-block* |
CONCENTRATESTMT ([NO | YES]) |
CURRENTDATA ([YES | NO]) |
DBPROTOCOL ([DRDA | DRDACBF]) |
{DEFER (PREPARE) | DEFER (INHERITFROMPLAN)} |
NODEFER (PREPARE)} |
DEGREE ([1 | ANY]) | DESCSTAT (NO | YES) |
DYNAMICRULES ([RUN | BIND | DEFINE | INVOKE]) |
ENCODING ([ASCII | EBCDIC |UNICODE | *ccsid*]) |
EXPLAIN ([YES| NO| ONLY]) | FLAG ([I | W | E | C]) |
IMMEDWRITE ([INHERITFROMPLAN | NO | YES]) |
ISOLATION ([RR | RS | CS | UR | NC]) |
KEEPDYNAMIC ([NO | YES]) |
REOPT (NONE[1] | [ALWAYS | VARS] | ONCE | AUTO) |
OPTHINT ('*hint-id*') |
{PATH ([*schema-name* | USER],) | PATHDEFAULT} |
RELEASE ([COMMIT | DEALLOCATE |
INHERITFROMPLAN]) |
SWITCH ([PREVIOUS | ORIGINAL]) |
VALIDATE ([RUN | BIND]) |
EXTENDEDINDICATOR ([NO | YES]) |
CONCURRENTACCESSRESOLUTION
([USECURENTLYCOMMITTED |
WAITFOROUTCOME]) |
GENERIC ('*string*') |
BUSTIMESENSITIVE ([YES | NO]) |
SYSTIMESENSITIVE ([YES | NO]) |
ARCHIVESENSITIVE ([YES | NO]) |
APPLCOMPAT ([*function-level* | V11R1 | V10R1) }}

Notes: [1] NOREOPT(VARS) can be a synonym of REOPT(NONE).

ID-block:
[([{*location-name*.} [*collection-id* | *] . [*package-id* | *]
{.([*version-id* | *])}, | *])]

enable-block:
{ [ENABLE(*) | [ENABLE | DISABLE]
([[BATCH | DLIBATCH | DB2CALL | CICS | IMS |
IMSBMP | IMSMPP | REMOTE | RRSAF]],)
{{ DLIBATCH (*connection-name*,) | CICS (*applid*,) |
IMSBMP (*imsid*,) | IMSMPP (*imsid*,) }} }

plan-management-block:
{{ PLANMGMT ([BASIC | EXTENDED | OFF]) |
APREUSE (NONE | ERROR | WARN)
{APREUSESOURCE([CURRENT | PREVIOUS |ORIGINAL)} |
APCOMPARE (NONE | WARN | ERROR) |
APRETAINDUP (YES | NO) }}

REBIND PLAN

([*plan-name*, | *])
{{ COLLID (<u>*</u> | *collection-id*) |
OWNER (*authorization-id*) | QUALIFIER (*qualifier-name*) |
enable-block | *pklist-block* |
{NODEFER (PREPARE) | DEFER (PREPARE) } |
ACQUIRE ([USE | ALLOCATE]) |
CACHESIZE (*decimal-value*) |
CURRENTDATA ([NO | YES]) |
CURRENTSERVER (*location-name*) |
DBPROTOCOL (DRDA) | DEGREE ([1 | ANY]) |
DISCONNECT ([EXPLICIT | AUTOMATIC| CONDITIONAL]) |
DYNAMICRULES (RUN | BIND) |
ENCODING ([ASCII | EBCDIC | UNICODE | *ccsid*]) |
EXPLAIN ([NO | YES]) | FLAG([<u>I</u> | W | E | C]) |
IMMEDWRITE ([NO | YES]) |
ISOLATION ([RR | RS | CS | UR]) |
KEEPDYNAMIC ([<u>NO</u> | YES]) |
REOPT (<u>NONE</u>[1] | [ALWAYS| VARS] | ONCE | AUTO) |
OPTHINT ('*hint-id*') |
{PATH ([*schema-name* | USER],) | PATHDEFAULT} |
RELEASE ([COMMIT | DEALLOCATE]) |
ROUNDING (CEILING | DOWN | FLOOR | HALFDOWN |
HALFEVEN | HALFUP | UP) |
SQLRULES ([DB2 | STD]) |
VALIDATE ([RUN | BIND]) |
CONCURRENTACCESSRESOLUTION
([USECURRENTLYCOMMITTED |
WAITFOROUTCOME] |
PROGAUTH (<u>DISABLE</u> | ENABLE) }}

Notes: [1] NOREOPT(VARS) can be a synonym of REOPT(NONE).

enable-block:

{{ [<u>ENABLE(*)</u> | [ENABLE | DISABLE]
([[BATCH | DLIBATCH | DB2CALL | CICS | IMS | IMSBMP |
IMSMPP | RRSAF]],)
{{ DLIBATCH (*connection-name*,) | CICS (*applid*,) |
IMSBMP (*imsid*,) | IMSMPP (*imsid*,) }} }}

pklist-block:

{ PKLIST ({*location-name.* | *.}
[*collection-id* | *] . [*package-id* | *],) |
NOPKLIST }

REBIND TRIGGER PACKAGE

({*location-name*.} [*collection-id* | *] . [*package-id* | *]) |
options-block |
{{ CURRENTDATA ([NO | YES]) |
DESCSTAT ([NO | YES]) |
EXPLAIN ([YES | NO | ONLY]) |
FLAG ([I | W | E | C]) |
IMMEDWRITE ([NO | YES]) |
ISOLATION ([RR | RS | CS | UR | NC]) |
RELEASE ([COMMIT | DEALLOCATE]) |
SWITCH ([PREVIOUS | ORIGIONAL]) |
CONCURRENTACCESSRESOLUTION
([USECURRENTLYCOMMITTED |
WAITFOROUTCOME]) |
BUSTIMESENSITIVE ([YES | NO]) |
SYSTIMESENSITIVE ([YES | NO]) |
ARCHIVESENSITIVE ([YES | NO]) |
APPLCOMPAT ([*function-level* | V11R1 | V10R1]) }}

options-block:

{{ PLANMGMT ([BASIC | EXTENDED | OFF]) |
APREUSE ([NONE | ERROR | WARN])
{ APREUSESOURCE (CURRENT | PREVIOUS | ORIGINAL)
} |
APCOMPARE (NONE | WARN | ERROR)
APRETAINDUP (NONE | WARN | ERROR) }}

RUN

[PROGRAM (*program-name*) {PLAN (*plan-name*)} |
CP PLAN (*plan-name*)]
{{ LIBRARY (*library-name*) | PARMS (*parameter-string*) }}

SPUFI

CICS Attachment Facility

DSNC

{*destination*} *db2-command*

DSNC DISC or DSNC DISCONNECT

plan-name

DSNC DISPLAY

[PLAN *plan-name* |
[TRAN | TRANSACTION] *transaction-id* |
STATISTICS]
{*destination*}

DSNC MODIFY

[[DEST | DESTINATION] *old new* |
[TRAN | TRANSACTION] *transaction-id integer*]

DSNC STOP

[FORCE | QUIESCE]

DSNC STRT

{*ssid*}

TSO Commands

DSN

{{ SYSTEM
([DSN |
subsystem-name |
group-attachment-name |
subgroup-attachment-name]) |
RETRY ([0 |*integer*]) |
TEST (*integer*) |
GROUP ([YES | NO]) |
ASUSER (*userid*) }}

DSNH

INPUT(*data-set-name*) { *clist-parameter,* }

z/OS IRLM Commands

MODIFY *irlmproc*,ABEND
{, [DUMP | NODUMP] }

MODIFY *irlmproc*,DIAG
, [DELAY | PLOCK | ALL | NONE | HANG]

MODIFY *irlmproc*,PURGE
,db2name

MODIFY *irlmproc*,SET
, [DEADLOCK=*nnnn* |
LTE=*nnnn* |
MLT=*nnnnnU* |
PVT=*nnnn* |
TIMEOUT=*nnnn*,*subsystem-name* |
TRACE = [10|*nnn*]]

MODIFY *irlmproc*,STATUS
{, [*irlmx* | ALLD | ALLI | MAINT | STOR | TRACE] }

START
irlmproc,
{{ DEADLOK='*iiii,kkkk*' |
IRLMGRP='*irlm-group-name*' |
IRLMID=*n* |
IRLMNM=*irlmname* |
LOCKTABL=*irlmltnm* |
LTE=*nnnn* |
MAXCSA=*nnn* |
MAXUSRS=*nnn* |
PC= |
PGPROT=[YES | NO] |
SCOPE=[LOCAL | GLOBAL | NODISCON] |
TRACE [NO | YES] }},

STOP
irlmproc

TRACE CT
, [WTRSTART=*parmlibmem* {, [WRAP | NOWRAP] } |
WTRSTOP=*jobname* |
ON, COMP=*irlmssnm*
{ ,SUB=([DBM | EXP | INT | SLM | XCF | XIT]) } |
OFF]

IMS Commands

/CHANGE

SUBSYS [*subsystem-name*, |
subsystem-name OASN *schedule-number* |
ALL]
RESET

/DISPLAY

[SUBSYS [*subsystem-name*, | ALL] |
OASN SUBSYS [*subsystem-name*, | ALL]]

/SSR

subsystem-command

/START

SUBSYS [*subsystem-name*, | ALL]

/STOP

SUBSYS [*subsystem-name*, | ALL]

/TRACE

SET [ON|OFF]
{{ TABLE [ALL | SUBS] |
OPTION [NOLOG | LOG] }}

ADMTPROC Commands

MODIFY

admtproc,APPL=SHUTDOWN

MODIFY

admtproc,APPL=TRACE=[ON | OFF]

START

admtproc { ,TRACE=[ON | OFF] }

STOP

admtproc

4. SYSTEM TABLES

Catalog Tables

Legend:

SYSIBM.*tablename*		**(DSNDB06.*tablespacename*)**
SYSIBM.*indexname*	**Type**	**(index columns)**
table description		

SYSIBM.IPLIST		**(DSNDB06.SYSDDF)**
SYSIBM.DSNDUX01	**U**	**(LINKNAME,IPADDR)**
Allows multiple IP addresses to be specified for a LOCATION. You can insert, update, and delete rows in this table.		

This table contains the following columns:

Column Name/Data Type/Description
LINKNAME VARCHAR(24) NOT NULL This value is associated with the value in the LINKNAME column of the associated row in SYSIBM.LOCATIONS and SYSIBM.IPNAMES.
IPADDR VARCHAR(254) NOT NULL Contains the IP address or the domain name of the remote TCP/IP host. The column must contain member specific domain name when you use WLM Domain Name Server load balancing or the member specific VIPA name when you use Dynamic VIPA balancing. DB2 assumes the value is an IP address in dotted decimal format when the column contains a left justified character string with four numeric values delimited by decimal points. All other values are interpreted as TCP/IP gethostbyname socket call.
IBQREQD CHAR(1) NOT NULL WITH DEFAULT 'N' Whether the row came from the basic Machine Readable Material (MRM) tape: Y/N/other. If other, release dependency indicator.

SYSIBM.IPNAMES		**(DSNDB06.SYSDDF)**
SYSIBM.DSNFPX01	**P**	**(LINKNAME)**
Defines the remote DRDA server DB2 can access using TCP/IP. You can insert, update, and delete rows in this table.		

This table contains the following columns:

Column Name/Data Type/Description
LINKNAME VARCHAR(24) NOT NULL This value must match the value specified in the LINKNAME column of the associated row in SYSIBM.LOCATIONS.
SECURITY_OUT CHAR(1) NOT NULL WITH DEFAULT 'A' Defines the DRDA security option used when local DB2 SQL applications connect to any remote server associated with this TCP/IP host.

Column Name/Data Type/Description

- A (Already Verified) An outbound connection request. Contains an authorization ID (like the DB2 user's authorization ID or a translated ID and no password).
- D (Security Option of user ID and security-sensitive data encryption). An outbound connection request contains an authorization ID and no password. The authorization ID that is used for an outbound request is the DB2 user's or a translated ID depending on USERNAMES column.
- E (Security Option of user ID, password, and security-sensitive data encryption). Outbound connection requests contain authorization ID and password. Password obtained from SYSIBM.USERNAMES. USERNAMES column must specify "O".
- P (password). An outbound connection request contains an authorization ID and a password. Password obtained from SYSIBM.USERNAMES. Value of USERNAMES must be set to 'O'.
- R (RACF PassTicket). An outbound connection request contains a Userid and a RACF PassTicket. The value specified in the LINKNAME is used as the RACF PassTicket.

USERNAMES

CHAR(1) NOT NULL WITH DEFAULT

Controls the outbound authorization ID translation that occurs when an authorization ID is sent by DB2 to a remote server.

- O—Indicates an outbound ID is subject to translation. No translation or "come from" checking is performed on inbound IDs.
- blank—No translation occurs.

IBMREQD

CHAR(1) NOT NULL WITH DEFAULT 'N'

Whether the row came from the basic Machine Readable Material (MRM) tape: Y/N/other. If other, release dependency indicator.

IPADDR

VARCHAR(254)

Contains the IP address or domain name of a remote TCP/IP host. Specify as follows:

- If the column contains a left justified character string of 4 numeric values delimited by decimal points, DB2 assumes the value to be an IP address in dotted decimal format. For example, '756.45'321.09'.
- All other values are interpreted as a TCP/IP domain name resolved by the TCP/IP gethostbyname socket call.

SYSIBM.LOCATIONS		(DSNDB06.SYSDDF)
SYSIBM.DSNFCX01	P	(LOCATION)
A row associates a location with the TCP/IP or SNA network attributes for each remote server. You can insert, update, and delete rows in this table.		

This table contains the following columns:

Column Name/Data Type/Description

LOCATION
VARCHAR(128)
Unique location name for the accessible server.

LINKNAME
VARCHAR(24) NOT NULL
Identifies the VTAM or TCP/IP attributes associated with this location. For any LINKNAME specified, one or both of the following values must be true:
- A row exists in SYSIBM.LUNAMES whose LUNAME matches the value in LINKNAME (VTAM attributes).
- A row exists in SYSIBM.IPNAMES whose LINKNAME matches the value in LINKNAME (TCP/IP attributes

IBMREQD
CHAR(1) NOT NULL WITH DEFAULT 'N'
Whether the row came from the basic MRM tape: Y/N/other. If other, release dependency indicator

PORT
VARCHAR(96) NOT NULL WITH DEFAULT
If TCP/IP is used for outbound DRDA connections when the following statement is true, (a row exists in SYSIBM.IPNAMES where LINKNAME matches a row in SYSIBM.LOCATIONS), if the specified row is found, the value is:
- Blank—Default DRDA port (446) used.
- Non-blank—If the value is left justified 1-5 numeric, the value is assumed to be a TCP/IP port number. Any other value is assumed to be a TCP/IP service name.

TPN
VARCHAR(192) NOT NULL
Used only when the local DB2 begins an SNA conversation with another server. A zero length indicates default TPN value. For DRDA conversations, the DRDA default X'07F6C4C2' is used. Column is not used for DB2 private protocol. For SQL/DS, it contains the resource ID of the SQL/DS machine.

DBALIAS
VARCHAR(128) NOT NULL
Used to access remote db server. If blank, location name is used to access remote db server. It will not change the name of any db objects sent to the remote server that contains the location qualifier.

TRUSTED
CHAR(1) NOT NULL
Indicates whether the connection to the remote server can be trusted. This is restricted to TCP/IP only
- Y—Location is trusted. Access to the remote location requires trusted context defined at the remote location.
- N – Location is not trusted.

SECURE
CHAR(1) NOT NULL

Column Name/Data Type/Description

Indicates the use of the Secure Socket Layer (SSL) protocol for outbound DRDA connections when local DB2 applications connect to the remote database server using TCP/IP.

- Y – A secure connection using SSL is required for the outbound DRDA connection.
- N – A secure connection is not required for the outbound DRDA connection.

SYSIBM.LULIST		**(DSNDB06.SYSDDF)**
SYSIBM.DSNFLX01	**U**	**(LINKNAME,LUNAME)**
SYSIBM.DSNFLX02	**U**	**(LUNAME)**
Defines multiple LU names for each LOCATION. The same value for the LUNAME column cannot appear in both the SYSIBM.LUNAMES table and the SYSIBM.LULIST table. You can insert, update, and delete rows in this table.		

This table contains the following columns:

Column Name/Data Type/Description

LINKNAME

VARCHAR(24) NOT NULL

Value of the LINKNAME column in the SYSIBM.LOCATIONS table with which this row is associated. This value is also the value of the LUNAME column in SYSIBM.LUNAMES table. Values of the other columns in the SYSIBM.LUNAMES row apply to the LU identified by the LUNAME column in this row of SYSIBM.LULIST.

LUNAME

VARCHAR(24) NOT NULL

The VTAM logical unit name (LUNAME) of the remote database system. This LUNAME must not exist in the LUNAME column of SYSIBM.LUNAMES.

IBQREQD

CHAR(1) NOT NULL WITH DEFAULT 'N'

Whether the row came from the basic Machine Readable Material (MRM) tape: Y/N/other. If other, release dependency indicator.

SYSIBM.LUMODES		**(DSNDB06.SYSDDF)**
SYSIBM.DSNFMX01	**P**	**(LUNAME,MODENAME)**
Provides VTAM with conversation limits for each combination of LUNAME and MODENAME. This table is accessed only during the initial conversation limit negotiation between DB2 and a remote LU. This negotiation is called change-number-of-sessions (CNOS) processing. You can insert, update, and delete rows in this table.		

This table contains the following columns:

Column Name/Data Type/Description

LUNAME

VARCHAR(24) NOT NULL

LUNAME of server involved in CNOS processing.

MODENAME

VARCHAR(24) NOT NULL

Name of a logon mode description in the VTAM logon mode table.

CONVLIMIT

Column Name/Data Type/Description
SMALLINT NOT NULL Maximum number of active conversations between the local DB2 and server for this mode. Used to override the no. of the DSESLIM parameter in the VTAM APPL definition statement for this mode.
IBMREQD CHAR(1) NOT NULL WITH DEFAULT 'N' Whether the row came from the basic Machine Readable Material (MRM) tape: Y/N/other. If other, release dependency indicator.

SYSIBM.LUNAMES		**(DSNDB06.SYSDDF)**
SYSIBM.DSNFNX01	**P**	**(LUNAME)**
Defines each remote SNA client or server that communicates with DB2. You can insert, update, and delete rows in this table.		

This table contains the following columns:

Column Name/Data Type/Description
LUNAME VARCHAR(24) NOT NULL LU name of one or more accessible systems. Blank indicates undefined requesters. All other columns for a given row are for clients and servers associated with the rows LU name.
SYSMODENAME VARCHAR(24) NOT NULL WITH DEFAULT Mode used to establish intersystem conversations. Blank indicates default mode IBMDB2LM.
SECURITY_IN CHAR(1) NOT NULL WITH DEFAULT 'A' Defines the security options accepted by this DB2 when an SNA client connects. • V (Verify). An incoming connection must include a userid and password, a userid and RACF PassTicket, or a Kerberos security ticket. • A (Already Verified). A request does not need a password. If sent, the password is checked. An incoming request is accepted if it contains a user ID, a user ID and password, a user ID and RACF PassTicket, or a Kerberos security ticket. When USERNAMES contains I or B, RACF is not invoked for requests that contain a user ID only.
SECURITY_OUT CHAR(1) NOT NULL WITH DEFAULT 'A' Defines the security options that are used when local DB2 SQL applications connect to any remote server associated with this LUNAME. • A (Already Verified). An outbound connection request contains an authorization ID, e.g., the DB2 user's authorization ID and no password. • R (RACF PassTicket). An Outbound connection request contains a userid and a RACF PassTicket. The server's LU name is used as the RACF PassTicket application name. • P (Password). An outbound connection request contains an authorization ID and a password. The password is obtained from SYSIBM.USERNAMES table or RACF depending on the value of ENCRYPTPSWDS column. USERNAMES must equal 'B' or 'O'.

Column Name/Data Type/Description
ENCRYPTPSWDS CHAR(1) NOT NULL WITH DEFAULT 'N' Applies only to DB2 for z/OS partners. Provides connectivity with prior releases of DB2 that cannot support RACF PassTickets. Valid values: • N (No). Default. Passwords are not in internal RACF encrypted format. • Y (Yes). Outbound requests extract the encrypted password from RACF and send it to the server. Inbound requests treat the password as encrypted.
MODESELECT CHAR(1) NOT NULL WITH DEFAULT 'N' Whether to use SYSIBM.MODESELECT table. • N. Uses default mode IBMDB2LM or IBMRDB. • Y. Searches SYSIBM.MODESELECT for appropriate mode name.
USERNAMES CHAR(1) NOT NULL WITH DEFAULT Controls inbound and outbound authorization ID translation and "come-from" checking. • I. Inbound ID is subject to translation and "come from" checking. No translation is performed on outbound IDs. • O. Outbound ID is subject to translation. No translation or "come from" checking is performed on inbound IDs. • B. Both. Both inbound translation and "come from" checking and outbound translation are performed on request. • blank. No translation.
GENERIC CHAR(1) NOT NULL WITH DEFAULT 'N' Whether DB2 identifies itself to the partner LU by its real LU name or a generic LU name specified by this row. • N. Real VTAM name of this DB2. • Y. VTAM generic LU name for this DB2.
IBMREQD CHAR(1) NOT NULL WITH DEFAULT 'N' Whether the row came from the basic MRM tape: Y/N/other. If other, release dependency indicator.

SYSIBM.MODESELECT		**(DSNDB06.SYSDDF)**
SYSIBM.DSNFDX01	**P**	**(LUNAME,AUTHID,PLANNAME)**
Associates a mode name with any conversation created to support an outgoing SQL request. Each row represents one or more combinations of LUNAME, authorization ID, and application plan name. You can insert, update, and delete rows in this table.		

This table contains the following columns:

Name	Data Type/Description
AUTHID	VARCHAR(128) NOT NULL WITH DEFAULT Authorization ID of the request for data from another subsystem. Default is blank, which indicates that the specified MODENAME applies to all authorization IDs.
PLANNAME	VARCHAR(24) NOT NULL WITH DEFAULT Plan name associated with SQL request. Default is blank, which indicates that the specified MODENAME applies to all plan names.

Name	Data Type/Description
LUNAME	VARCHAR(24) NOT NULL WITH DEFAULT LU name associated with the SQL request.
MODENAME	VARCHAR(24) NOT NULL WITH DEFAULT Name of logon mode in the VTAM logon mode table. Used in support of the outgoing SQL request. If blank, default mode IBMDB2LM is used for DB2 private protocol connections and IBMRDB is used for DRDA connections.
IBMREQD	CHAR(1) NOT NULL WITH DEFAULT 'N' Whether the row came from the basic MRM tape: Y/N/other. If other, release dependency indicator.

SYSIBM.SYSAUDITPOLICIES		**(DSNDB06.SYSTSADT)**
SYSIBM.DSNAPX03	**U**	**(AUDITPOLICYNAME)**
Contains information about each audit policy. One row per audit policy.		

This table contains the following columns:

Name	Data Type/Description
AUDITPOLICYNAME	VARCHAR(128) NOT NULL Name of audit policy.
OBJECTSCHEMA	VARCHAR(128) NOT NULL WITH DEFAULT Schema of the audited object.
OBJECTNAME	VARCHAR(128) NOT NULL WITH DEFAULT Name of the object.
OBJECTTYPE	CHAR(1) NOT NULL WITH DEFAULT Type of the object: • A (Alias) • C (Clone table) • P (Implicit XML table) • T (Table) • blank (all of the above)
CREATEDTS	TIMESTAMP NOT NULL WITH DEFAULT Row insertion time.
ALTEREDTS	TIMESTAMP NO NULL WITH DEFAULT Row update time.
CHECKING	CHAR(1) NOT NULL WITH DEFAULT Indicates if authorization and authentication failures are audited. • A (audit all) • blank (audit none)
VALIDATE	CHAR(1) NOT NULL WITH DEFAULT Indicates if auditing is enabled for when a trusted connection is established or used by a different user. • A (audit all) • blank (audit none)
OBJMAINT	CHAR(1) NOT NULL WITH DEFAULT Table audited on first access by each unit of work. • A (audited during first operation) • blank (audit none)
EXECUTE	CHAR(1) NOT NULL WITH DEFAULT Indicates if auditing is enabled. • A (audit on the first of any operation) • C (audit when accessed by first insert, update, or delete)

Name	Data Type/Description
	• blank (no audit)
CONTEXT	CHAR(1) NOT NULL WITH DEFAULT Auditing enabled for utility. • A (audit all utilities) • blank (audit none)
SECMAINT	CHAR(1) NOT NULL WITH DEFAULT Auditing enabled for grants, revokes, and creation or alter of trusted context. • A (audit all) • blank (audit none)
SYSADMIN	VARCHAR(128) NOT NULL WITH DEFAULT Auditing enabled for administrative authority performing system administration tasks: • blank (audit none) • * (audit all) • I (installation SYSADM) • L (SYSCTRL) • O (SYSOPR) • R (Installation SYSOPR) • S (SYSADM) **Note:** Can be a concatenated string of all supported values.
DBADMIN	VARCHAR(128) NOT NULL WITH DEFAULT Auditing enabled for administrative authority performing data administration tasks: • blank (audit none) • * (audit all) • B (system DBADM) • C (DBCTRL) • D (DBADM) • E (SECADM) • G (ACCESSCTRL) • K (SQLADM) • M (DBMAINT) • P (PACKADM) • T (DATAACCESS) **Note:** Can be a concatenated string of all supported values.
DBNAME	VARCHAR(24) NOT NULL WITH DEFAULT Database name
COLLID	VARCHAR(128) NOT NULL WITH DEFAULT Package collection ID
DB2START	CHAR(1) NOT NULL WITH DEFAULT Automatically start audit policies at DB2 start up (Y or N).
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y. For other values, see release dependency indicator.
SYS_START	TIMESTAMP(12) NOT NULL System period temporal start time for transaction.
SYS_END	TIMESTAMP(12) NOT NULL System period temporal end time for transaction.
TRANS_START	TIMESTAMP(12) System period transaction timestamp.

SYSIBM.SYSAUTOALERTS		(DSNDB06.SYSTSATS
SYSIBM.DSNALX01	P	(ALERT_ID)
SYSIBM.DSNALX02	D	(HISTORY_ENTRY_ID)
SYSIBM.DSNALX03	D	(RETURN_CODE,ACTION)
SYSIBM.DSNALX04	D	(TARGET_QUALIFIER, TARGET_OBJECT, TARGET_PARTITION)
SYSIBM.DSNALX05	D	(CREATEDTS)
SYSIBM.DSNALX06	D	(STARTTS,RETURN_CODE)
Contains one row for each recommendation from autonomic procedures.		

This table contains the following columns:

Name	Data Type/Description
ALERT_ID	BIGINT NOT NULL GENERATED ALWAYS AS IDENTITY ID of this alert.
HISTORY_ENTRY_ID	BIGINT NOT NULL Procedure ID producing alert (see ADMIN_UTLPROCEDURES_HIST).
ACTION	VARCHAR(128) NOT NULL Type of action requested.
TARGET_QUALIFIER	VARCHAR(128) NOT NULL Database name to which the alert applies.
TARGET_OBJECT	VARCHAR(128) NOT NULL Table space name to which the alert applies.
TARGET_PARTITION	SMALLINT NOT NULL Partition number to which the alert applies. 0 (all) or object is not partitioned.
OPTIONS	VARCHAR(4000) Options to specify when the corresponding action is run: • USE PROFILE (Use options specified in profile). • TABLE (Options only apply for table). • COLUMNS (Options only apply for these columns). • SAMPLE (Sampling allowed). Tablespace too big.
CREATEDTS	TIMESTAMP NOT NULL WITH DEFAULT Time alert was issued.
DURATION	INTEGER Estimated time needed to run corresponding action.
STATUS	VARCHAR(32) Status of planned task: • OPEN (alert not yet resolved) • INPROGRESS (alert execution in progress) • COMPLETED (alert execution complete)
STARTTS	TIMESTAMP Start time of alert execution: NULL (not started)
ENDTS	TIMESTAMP End time of alert execution (NULL (not ended)
RETURN_CODE	INTEGER Automatic stored procedure return code:

Name	Data Type/Description
	• NULL (alert not resolved, stored procedure failed, no return code from stored procedure) • 0 (successful execution)
ERROR_MESSAGE	VARCHAR(1331) Why alert was not resolved successfully. • NULL (no return code from stored procedure) • none (successful execution)
OUTPUT	CLOB(2M) Automatic stored procedure output: • NULL (task not yet executed, execution failed, stored procedure does not write any output)
ROWID	ROWID NOT NULL GENERATED ALWAYS ROWID value for the CLOB.

SYSIBM.SYSAUTOALERTS_OUT		**(DSNDB06.SYSSTSATX)**
SYSIBM.DSNALX07	**U**	**(AUXID,AUXVER)**
Auxiliary table for the OUTPUT column of SYSIBM.SYSAUTOALERTS.		

This table contains the following columns:

Name	Data Type	Description
AUXID	VARCHAR(17)	ID of auxiliary data.
AUXVER	SMALLINT	Version of auxiliary data.
AUXVALUE	CLOB(2M)	Automatic stored procedure output.

SYSIBM.SYSAUTORUNS_HIST		**(DSNDB06.SYSTSPRH)**
SYSIBM.DSNPHX01	**P**	**(HISTORY_ENTRY_ID)**
SYSIBM.DSNPHX02	**D**	**(PROC_NAME,STARTTS)**
SYSIBM.DSNPHX03	**D**	**(STARTTS)**
Contains information about each autonomic procedure run (one row per autonomic procedure).		

This table contains the following columns:

Name	Data Type/Description
HISTORY_ENTRY_ID	BIGINT NOT NULL GENERATED ALWAYS AS IDENTITY History table entry ID.
PROC_NAME	VARCHAR(128) NOT NULL Autonomic stored procedure name that produced this entry.
STARTTS	TIMESTAMP Autonomic stored procedure start time.
ENDTS	TIMESTAMP Autonomic stored procedure end time.
OUTPUT	CLOB(2M) Autonomic stored procedure output.
ERROR_MESSAGE	VARCHAR(1331) Why autonomic stored procedure was unsuccessful. If empty, successful.
RETURN_CODE	INTEGER Autonomic stored procedure return code. • NULL (execution failed, procedure does not write return code).

Name	Data Type/Description
	• 0 (successful execution).
ROWID	ROWID NOT NULL GENERATED ALWAYS ROWID value for OUTPUT column.

SYSIBM.SYSAUTORUNS_HISTOU		(DSNDB06.SYSTSPHX)
SYSIBM.DSNPHX04	U	(AUXID,AUXVER)
Auxiliary table for the OUTPUT column of SYSIBM.SYSAUTOALERTS.		

This table contains the following columns:

Name	Data Type	Description
AUXID	VARCHAR(17)	ID of auxiliary data.
AUXVER	SMALLINT	Version of auxiliary data.
AUXVALUE	CLOB(2M)	Automatic stored procedure output.

SYSIBM.SYSAUTOTIMEWINDOWS		(DSNDB06.SYSTSATW)
SYSIBM.DSNTWX01	P	(WINDOW_ID)
Contains a row for every time period autonomic procedures can be run..		

This table contains the following columns:

Name	Data Type/Description
WINDOWID	BIGINT NOT NULL GENERATED ALWAYS AS IDENTITY Time ID of window.
DB2_SSID	CHAR(4) DB2 member name for running planned tasks. If NULL, run on any DB2 member.
MONTH_WEEK	CHAR(1) NOT NULL Interpretation of DAY column. M (day of month), W (day of week).
MONTH	INTEGER Month time window applies. NULL (all months). If MONTH_WEEK is W, MONTH must be NULL.
DAY	INTEGER Day of month or week. NULL (every day).
FROM_TIME	TIME Time window start time. NULL (any time). If TO_TIME is NULL, FROM_TIME is NULL.
TO_TIME	TIME Time window end time. NULL (any time). If FROM_TIME is NULL, TO_TIME is NULL.
ACTION	VARCHAR(256) Comma-separated list of allowed actions. NULL, all actions allowed.
MAX_TASKS	INTEGER Number of concurrent actions allowed. NULL, any number.

SYSIBM.SYSAUXRELS		(DSNDB06.SYSTSAUX)
SYSIBM.DSNOXX011	D	(TBOWNER,TBNAME)
SYSIBM.DSNOXX02	D	(AUXTBOWNER,AUXTBNAME)
Contains information for every auxiliary table created for a LOB column. A partitioned base table space must have one auxiliary table for each partition of each LOB column.		

This table contains the following columns:

Name	Data Type/Description
TBOWNER	VARCHAR(128) NOT NULL Authorization ID of base table owner.
TBNAME	VARCHAR(128) NOT NULL Base table name.
COLNAME	VARCHAR(128) NOT NULL LOB column name in base table.
PARTITION	SMALLINT NOT NULL If base table is partitioned, contains partition number. Otherwise, 0.
AUXTBOWNER	VARCHAR(128) NOT NULL Authorization ID of auxiliary table owner.
AUXTBNAME	VARCHAR(128) NOT NULL Auxiliary table name.
AUXRELOBID	INTEGER NOT NULL Internal identifier for relationship between auxiliary and base table.
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other. If other, release dependency indicator.
RELCREATED	CHAR(1) NOT NULL Release of DB2 used to create table. If created in DB2 9, this column is blank.

SYSIBM.SYSCHECKDEP		(DSNDB06.SYSTSCKD)
SYSIBM.DSNSDX01	P	(TBOWNER,TBNAME, CHECKNAME,COLNAME)
Contains information on each column reference in a check constraint. One row per column/check constraint combination.		

This table contains the following columns:

Name	Data Type/Description
TBOWNER	VARCHAR(128) NOT NULL Authorization ID of the owner for which the check constraint is defined.
TBNAME	VARCHAR(128) NOT NULL Name of the table for which the check constraint is defined.
CHECKNAME	VARCHAR(128) NOT NULL Check constraint name.
COLNAME	VARCHAR(128) NOT NULL Name of column referenced by the check constraint.

SYSIBM.SYSCHECKS		(DSNDB06.SYSTSCKS)
SYSIBM.DSNSCX01	P	(TBOWNER,TBNAME, CHECKNAME)
Contains information on each check constraint, one row for each check constraint.		

This table contains the following columns:

Name	Data Type/Description
TBOWNER	VARCHAR(128) NOT NULL Authorization ID of the owner of the table on which the table check constraint is defined.
CREATOR	VARCHAR(128) NOT NULL Authorization ID of the creator of the table check constraint.
DBID	SMALLINT NOT NULL Internal identifier of the database for the check constraint.
OBID	SMALLINT NOT NULL Internal identifier of the table check constraint.
TIMESTAMP	TIMESTAMP NOT NULL Time when the table check constraint was created.
RBA	CHAR(10) FOR BIT DATA NOT NULL The log RBA when the check constraint was created. Column contains BIT data.
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other. If other, release dependency indicator.
TBNAME	VARCHAR(128) NOT NULL Name of the table for which the check constraint is defined.
CHECKNAME	VARCHAR(128) NOT NULL The table check constraint is defined.
CHECKCONDITION	VARCHAR(7400) NOT NULL The text of the table check constraint.
RELCREATED	CHAR(1) NOT NULL Release of DB2 used to create table. If created before DB2 9, this value is blank.
ENVID	INTEGER IBM internal use.
PERIOD	CHAR(1) NOT NULL WITH DEFAULT Whether the row came from the basic MRM tape: Y/N/other. If other, release dependency indicator.

SYSIBM.SYSCHECKS2		(DSNDB06.SYSTSCHX)
SYSIBM.DSNCHX01	U	(TBOWNER,TBNAME, CHECKNAME)
Contains information about each check constraint for catalog tables created in or after DB2 V7. One row per check constraint.		

This table contains the following columns:

Name	Data Type/Description
TBOWNER	VARCHAR(128) NOT NULL Authorization ID of the owner of the table on which the table check constraint is defined.

Name	Data Type/Description
TBNAME	VARCHAR(128) NOT NULL Name of the table on which the check constraint is defined.
CHECKNAME	VARCHAR(128) NOT NULL Table check constraint name.
PATHSCHEMAS	VARCHAR(2048) NOT NULL SQL path at the time the check constraint was created. The path is used to resolve unqualified cast function names that are used in the constraint definition.
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other. If other, release dependency indicator.
RELCREATED	CHAR(1) NOT NULL Release of DB2 used to create table. If created before DB2 9, this value is blank.

SYSIBM.SYSCOLAUTH		**(DSNDB06.SYSTSFAU)**
SYSIBM.DSNACX01	**D**	**(CREATOR,TNAME,COLNAME)**
SYSIBM.DSNACX02	**D**	**(CREATOR,TNAME,TIMESTAMP)**
SYSIBM.DSNACX03	**D**	**(GRANTOR,GRANTORTYPE, CREATOR,TNAME,TIMESTAMP)**
SYSIBM.DSNACX04	**D**	**(GRANTEE,GRANTEETYPE, CREATOR,TNAME,TIMESTAMP)**
Defines UPDATE or REFERENCES privileges users have on individual columns of a table or view.		

This table contains the following columns:

Name	Data Type/Description
GRANTOR	VARCHAR(128) NOT NULL Authorization ID of the user who granted the privileges. Could also be PUBLIC or PUBLIC *.
GRANTEE	VARCHAR(128) NOT NULL Authorization ID of the user who holds the privilege, or the name of the application plan or package that uses the privilege.
GRANTEETYPE	CHAR(1) NOT NULL Type of grantee: • Blank (GRANTEE is an authorization ID). • P (GRANTEE is an application plan or package.
CREATOR	VARCHAR(128) NOT NULL Authorization ID of the owner of table or view on which the UPDATE privilege is held.
TNAME	VARCHAR(128) NOT NULL Name of the table or view.
TIMESTAMP	CHAR(12)/Internal use only.
DATEGRANTED	CHAR(6)/Unused.
TIMEGRANTED	CHAR(8)/Unused.
COLNAME	VARCHAR(128) NOT NULL Name of the column to which the UPDATE privilege applies.
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other. If other, release dependency indicator.
LOCATION	VARCHAR(128)/Unused.

Name	Data Type/Description
COLLID	VARCHAR(128) NOT NULL WITH DEFAULT If the GRANTEE is a package, its collection name. Blank otherwise.
CONTOKEN	CHAR(8) NOT NULL WITH DEFAULT FOR BIT DATA If the GRANTEE is a package, the consistency token of the DBRM from which the package was derived. Blank otherwise.
PRIVILEGE	CHAR(1) NOT NULL WITH DEFAULT Indicates privilege this row describes: • R (REFERENCES) • Blank (UPDATE)
GRANTEDTS	TIMESTAMP NOT NULL WITH DEFAULT Time when the GRANT was executed.
GRANTORTYPE	CHAR(1) NOT NULL WITH DEFAULT Indicates the type of grantor: • I (role) • Blank (authorization ID that is not a role.
SYS_START	TIMESTAMP(12) NOT NULL System period temporal start time for transaction.
SYS_END	TIMESTAMP(12) NOT NULL System period temporal end time for transaction.
TRANS_START	TIMESTAMP(12) System period transaction timestamp.

SYSIBM.SYSCOLDIST		**(DSNDB06.SYSSTATS)**
SYSIBM.DSNTNX01	**D**	**(TBOWNER,TBNAME,NAME)**
Contains distribution statistics in one or more rows for the first key column of an index key (populated by RUNSTATS). You can insert, update, and delete rows in this table.		

This table contains the following columns:

Name	Data Type/Description
FREQUENCY	SMALLINT/Unused.
STATSTIME	TIMESTAMP NOT NULL WITH DEFAULT The date and time when RUNSTATS last updated the statistics.
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other. If other, release dependency indicator.
TBOWNER	VARCHAR(128) NOT NULL The schema of the table that contains the column.
TBNAME	VARCHAR(128) NOT NULL Name of the table containing the column.
NAME	VARCHAR(128) NOT NULL Name of the column. If NUMCOLUMNS is > 1, contains first column name in the set associated with statistics.
COLVALUE	VARCHAR(2000) NOT NULL WITH DEFAULT FOR BIT DATA Contains the data of a frequently occurring value. Statistics are not collected on a ROWID column. Column contains bit data.
TYPE	CHAR(1) NOT NULL WITH DEFAULT 'F'

Name	Data Type/Description
	Type of statistics gathered: • C (cardinality) • F (frequent value) • H (histogram statistics) • N (non-padded frequent value)
CARDF	FLOAT NOT NULL WITH DEFAULT -1 • C (the number of distinct values for the column group) • H (the number of distinct values for the column group in a quantile indicated by QUANTILENO • Value is -1 if statistics have not been gathered.
COLGROUPCOLNO	VARCHAR(254) NOT NULL WITH DEFAULT FOR BIT DATA Identifies the set of columns associated with statistics. If associated with single column, length equal zero. Otherwise, column is an array of SMALLINT numbers with the dimension equal to NUMCOLUMNS.
NUMCOLUMNS	SMARTINT NOT NULL WITH DEFAULT 1 Number of columns associated with statistics.
FREQUENCYF	FLOAT NOT NULL WITH DEFAULT -1 Percentage of rows in a table containing the value in COLVALUE when the number is multiplied by 100. When TYPE='H', this is the percentage of rows in the table that falls in the quantile indicated by QUANTILENO whose range is limited by [LOWVALUE, HIGHVALUE]. Statistics are not collected on ROWID column.
QUANTILENO	SMALLINT NOT NULL Ordinary sequence number of a quantile in the whole consecutive value range, from low to high.
LOWVALUE	VARCHAR(2000) NOT NULL WITH DEFAULT FOR BIT DATA • H—This is the lower bound for the quantile indicated by QUANTILENO. • Not used if TYPE is not H.
HIGHVALUE	VARCHAR (2000) NOT NULL WITH DEFAULT FOR BIT DATA • H—This is the higher bound for the quantile indicated by QUANTILENO. • Not used if TYPE is not H.

SYSIBM.SYSCOLDISTSTATS		(DSNDB06.SYSSTATS)
SYSIBM.DSNTPX01	D	(TBOWNER,TBNAME,NAME, PARTITION)
Contains distribution statistics in zero or more rows per partition for the first key column of a partitioned index (populated by RUNSTATS). You can insert, update, and delete rows in this table.		

This table contains the following columns:

Name	Data Type/Description
FREQUENCY	SMALLINT (unused).
STATSTIME	TIMESTAMP NOT NULL WITH DEFAULT The date and time when RUNSTATS las updated the statistics.
IBMREQD	CHAR(1) Whether the row came from the basic MRM tape: Y/N/other. If other, release dependency indicator.
PARTITION	SMALLINT NOT NULL Partition number for the table space containing the table in which the column is defined.
TBOWNER	VARCHAR(128) NOT NULL The schema of the table that contains the column.
TBNAME	VARCHAR(128) NOT NULL Name of the table containing the column.
NAME	VARCHAR(128) NOT NULL Name of the column. If NUMCOLUMNS is > 1, contains the first column name in the set that is associated with statistics.
COLVALUE	VARCHAR(2000) NOT NULL FOR BIT DATA Contains the data of a frequently occurring value. Statistics are not collected on ROWID column. Column contains bit data.
TYPE	CHAR(1) NOT NULL WITH DEFAULT 'F' Type of statistics gathered: • C—(cardinality) • F—(frequent value) • H—Histogram statistics • N—Non-packed frequent value
CARDF	FLOAT NOT NULL WITH DEFAULT -1 C—The number of distinct values for the column group. H—The number of distinct values for the column group in a quantile indicated by QUANTILENO. Value is -1 if statistics have not been gathered.
COLGROUPCOLNO	VARCHAR(254) NOT NULL WITH DEFAULT FOR BIT DATA Identifies the set of columns associated with statistics. If a single column, length equal zero. Otherwise, column is an array of SMALLINT numbers with the dimension equal to NUMCOLUMNS.
NUMCOLUMNS	SMALLINT NOT NULL WITH DEFAULT 1 Number of columns associated with statistics.

Name	Data Type/Description
FREQUENCYF	FLOAT NOT NULL WITH DEFAULT -1 Gives the percentage of rows in the table with the value specified in COLVALUE when the number is multiplied by 100. For example, va value of 1 indicates 100%. A value of .153 indicates 15.3%. When TYPE='H', this is the percentage of rows in the table that falls in the quantile indicated by QUANTILENO whose range is limited by [LOWVALUE, HIGHVALUE]. Statistics are not collected for an index on a ROWID column. The value is -1 if statistics have not been gathered.
KEYCARDDATA	VARCHAR(1000) NOT NULL WITH DEFAULT FOR BIT DATA (Internal use only)
QUANTILENO	SMALLINT NOT NULL WITH DEFAULT -1 Ordinary sequence number of a quantile in the whole consecutive value range, from low to high.
LOWVALUE	VARCHAR(2000) NOT NULL WITH DEFAULT FOR BIT DATA H—This is the lower bound for the quantile indicated by QUANTILENO. Not used if TYPE is not H.
HIGHVALUE	VARCHAR(2000) NOT NULL WITH DEFAULT FOR BIT DATA H—This is the higher bound for the quantile indicated by QUANTILENO. Not used if TYPE is not H.

SYSIBM.SYSCOLDIST_HIST		**(DSNDB06.SYSHIST)**
SYSIBM.DSNHFX01	**D**	**(TBOWNER,TBNAME,NAME, STATISTIME)**
Contains distribution statistics history from SYSCOLDIST (populated by RUNSTATS). You can insert, update, and delete rows in this table.		

This table contains the following columns:

Name	Data Type/Description
STATSTIME	TIMESTAMP NOT NULL If RUNSTATS updated the statistics, the date and time they were last updated.
TBOWNER	VARCHAR(128) NOT NULL The schema of the table that contains the column.
TBNAME	VARCHAR(128) NOT NULL Name of the table that contains the column.
NAME	VARCHAR(128) NOT NULL Name of the column. If NUMCOLUMNS is > 1, this name identifies the first column name of the set of columns associated with the statistics.
COLVALUE	VARCHAR (2000) NOT NULL FOR BIT DATA Contains the data of a frequently occurring value. Statistics are not collected an index on a ROWID column. If the value has a non-

Name	Data Type/Description
	character data type, the data might not be printable.
TYPE	CHAR(1) NOT NULL WITH DEFAULT 'F' The type of statistics gathered: C – Cardinality F – Frequent value H – Histogram Statistics N – Non-padded frequent value
CARDF	FLOAT(8) NOT NULL WITH DEFAULT -1 • C – The number of distinct values for the column group. • H – The number of distinct values for the column group in a quantile indicated by QUANTILENO Value is -1 if statistics have not been gathered.
COLGROUPCOLNO	VARCHAR(254) NOT NULL FOR BIT DATA Identifies the set of columns associated with the statistics. If the statistics are only associated with a single column, the field contains a zero length. Otherwise, the field is an array of SMALLINT column numbers with a dimension equal to the value in NUMCOLUMNS.
NUMCOLUMNS	SMALLINT NOT NULL WITH DEFAULT -1 Identifies the number of columns associated with the statistics.
FREQUENCYF	FLOAT(8) NOT NULL WITH DEFAULT -1 Gives the percentage of rows in the table with the value specified in COLVALUE when the number is multiplied by 100. For example, a value of 1 indicates 100%. A value of .153 indicates 15.3%. When TYPE='H', this is the percentage of rows in table which falls in the quantile indicated by QUANTILENO whose range is limited by [LOWVALUE, HIGHVALUE]. Statistics are not collected for an index on a ROWID column. The value is -1 if statistics have not been gathered.
IBMREQD	CHAR(1) NOT NULL WITH DEFAULT 'N' Whether the row came from the basic MRM tape: Y/N/other.
QUANTILENO	SMALLINT NOT NULL WITH DEFAULT -1 Ordinary sequence number of a quantile in the whole consecutive value range, from low to high.
LOWVALUE	VARCHAR(2000) NOT NULL WITH DEFAULT FOR BIT DATA H—This is the lower bound for the quantile indicated by QUANTILENO. Not used if TYPE is not H.
HIGHVALUE	VARCHAR(2000) NOT NULL WITH DEFAULT FOR BIT DATA H—This is the higher bound for the quantile indicated by QUANTILENO. Not used if TYPE is not H.

SYSIBM.SYSCOLSTATS		(DSNDB06.SYSTSCOL)
SYSIBM.DSNTCX01	U	(TBOWNER,TBNAME,NAME, PARTITION)
Contains partition statistics for selected columns, one row for each partition (populated by RUNSTATS). You can insert, update, and delete rows in this table.		

This table contains the following columns:

Name	Data Type/Description
HIGHKEY	VARCHAR(2000) NOT NULL FOR BIT DATA Highest value of the column within the partition. Blank if stats not gathered or the column is an indicator column, a node ID column, or a column of an XML table. If column has non-char data, might be unprintable.
HIGH2KEY	VARCHAR(2000) NOT NULL FOR BIT DATA Second highest value of the column within the partition. Blank if stats not gathered or the column is an indicator column, a node ID column, or a column of an XML table. If column has non-char data, might be unprintable.
LOWKEY	VARCHAR(2000) NOT NULL FOR BIT DATA Lowest value of the column within the partition. Blank if stats not gathered or the column is an indicator column, a node ID column, or a column of an XML table. If column has non-char data, might be unprintable.
LOW2KEY	VARCHAR(2000) NOT NULL FOR BIT DATA Second lowest value of the column within the partition. Blank if stats not gathered or the column is an indicator column, a node ID column, or a column of an XML table. If column has non-char data, might be unprintable.
COLCARD	INTEGER NOT NULL Number of distinct column values in the partition.
STATSTIME	TIMESTAMP NOT NULL The date and time when RUNSTATS last updated the statistics. If value is '0001-01-02.00.00.00.000000' an ALTER TABLE changed length of VARCHAR column.
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other.
PARTITION	SMALLINT NOT NULL Partition number for the table space containing the table in which the column is defined.
TBOWNER	VARCHAR(128) NOT NULL Schema or qualifier of the table that contains the column.
TBNAME	VARCHAR(128) NOT NULL Name of the table containing the column.
NAME	VARCHAR(128) NOT NULL Name of the column.

Name	Data Type/Description
COLCARDDATA	VARCHAR(1000) NOT NULL FOR BIT DATA (internal use only)
STATS_FORMAT	CHAR(1) NOT NULL WITH DEFAULT Type of statistics generated: • Blank—If statistics have not been collected or VARCHAR column statistic values have been padded. • N—VARCHAR column statistics are not padded. • Updateable column.

SYSIBM.SYSCOLUMNS		**(DSNDB06.SYSTSCOL1)**
SYSIBM.DSNDCX01	**P**	**(TBCREATOR,TBNAME,NAME)**
SYSIBM.DSNDCX02	**D**	**(TYPESCHEMA,TYPENAME)**
SYSIBM.DSNDCX03	**D**	**(TBCREATOR,TBNAME)**
SYSIBM.DSNDCX04	**D**	**(TBCREATOR,TBNAME,COLNO)**
Defines the columns for each table and view, one row for every column.		

This table contains the following columns:

Name	Data Type/Description
NAME	VARCHAR(128) NOT NULL Name of the column.
TBNAME	VARCHAR(128) NOT NULL Name of the table or view that contains the column.
TBCREATOR	VARCHAR(128) NOT NULL Authorization ID of the owner of the table or view that contains the column.
COLNO	SMALLINT NOT NULL Numerical place of the column in the table or view. For example, 4 (out of 10).
COLTYPE	CHAR(8) NOT NULL Type of column: • INTEGER (full-word binary) • SMALLINT (half-word binary) • FLOAT (floating-point numeric) • CHAR (fixed-length character) • VARCHAR (variable-length character) • LONGVAR (variable-length character) • DECIMAL – packed decimal • GRAPHIC – fixed-length DBCS • VARG – variable-length DBCS • LONGVARG – variable-length DBCS • DATE – date • TIME – time • TIMESTMP – timestamp • TIMESTZ – timestamp with zone • BLOB – binary large object • CLOB – character large object • DBCLOB – double-byte character large object • ROWID – ROW ID data type • DISTINCT – distinct data type • XML – XML data type • BIGINT – Big integer • BINARY – Fixed-length binary string • VARBIN – Varying-length binary string

Name	Data Type/Description
	• DECFLOAT – Decimal floating point
LENGTH	SMALLINT NOT NULL The length attribute of the column or, in the case of a decimal column, its precision: • INTEGER – 4 • SMALLINT – 2 • FLOAT – 4 or 8 • CHAR – length of string • VARCHAR – max. length of string • LONGVAR – max. length of string • DECIMAL – precision of number • DECFLOAT – 8 or 16 • GRAPHIC – number of DBCS chars. • VARGRAPHIC – max. # of DBCS chars. • LONGVARG – max. # of DBCS chars. • DATE – 4 • TIME – 3 • TIMESTAMP WITHOUT TIME ZONE – Integral part of ((p+1)/2)+7 • TIMESTAMP WITH TIME ZONE – Integral part of ((p+1)/2)+9 • LOB – 4 length of field in base table, max. length of LOB in LENGTH2. • INLINE LOB – 4 length of field in base table, max. length of LOB in LENGTH2. • BLOB – 4 length of field in base table, max. length of LOB in LENGTH2. • CLOB – 4 length of field in base table, max. length of CLOB in LENGTH2. • DBCLOB – 4 length of field in base table, max. length of DBCLOB in LENGTH2. • ROWID – 17 • DISTINCT – length of source data type • XML – 6 • BIGINT – 8 • BINARY – Length of string • VARBINARY – Maximum length of string The number does not include the internal prefixes used to record the actual length and null state, where applicable.
SCALE	SMALLINT NOT NULL Scale of decimal data or fractional second digits of timestamp data, or 0 if not decimal or timestamp.
NULLS	CHAR(1) NOT NULL Whether the column can contain null values: Y/N. The value can be N for a view column derived from an expression or a function. However, such a column allows nulls when referenced in an outer select.
COLCARD	INTEGER NOT NULL (unused)
HIGH2KEY	VARCHAR(2000) NOT NULL FOR BIT DATA Second highest value of the column. Blank if statistics have not been gathered or if the column is an indicator column or a column of an auxiliary table. If key has non-character data type, data may not be printable.
LOW2KEY	VARCHAR(2000) NOT NULL FOR BIT DATA

Name	Data Type/Description
	Second lowest value of the column. Blank if statistics have not been gathered or if the column is an indicator column or a column of an auxiliary table. If key has non-character data type, data may not be printable.
UPDATES	CHAR(1) NOT NULL Whether column can be updated. Value is 'N' when: • Columns are for a read-only view • If columns are identified with AS IDENTITY and GENERATED ALWAYS • If column was derived from a function or expression • Column has a row ID data type (or a distinct type based on a row ID type).
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other. If other, release dependency indicator.
REMARKS	VARCHAR(762) NOT NULL A character string provided by the user with the COMMENT ON statement.
DEFAULT	CHAR(1) NOT NULL Meaningful for columns only if the SYSTABLES TYPE column is T (table) or G (created temporary table): • A (ROWID with GENERATE ALWAYS) • B (Depends on the data type. • D (ROWID with GENERATED BY DEFAULT) • E (Column is defined with the FOR EACH ROW ON UPDATE and GENERATED ALWAYS attributes). • F (Column is defined with the FOR EACH ROW ON UPDATE and GENERATED BY DEFAULT). • I (AS IDENTITY with GENERATED ALWAYS) • J (AS IDENTITY with GENERATED BY DEFAULT) • K (Column is defined for the implicit DOCID column for a base table that contains XML data). • L (Column is defined with the AS SECURITY LABEL attribute) • N (no default value) • Q (Column is defined with the AS ROW BEGIN attribute) • R (Column is defined with the AS ROW END attribute) • S (SQL ID of the process) • U (value of the USER special register at execution time) • Y (default of NULL if NULLS column=N, else default depends on data type. • X (AS TRANSACTION START ID) • 1 (string data) • 2 (floating point number) • 3 (decimal value) • 4 (integer value) • 5 (hex string)

Name	Data Type/Description
	• 6 (UX string) • 7 (graphic data type) • 8 (character data type) • 9 (DECFLOAT constant)
KEYSEQ	SMALLINT NOT NULL Columns relative position within the tables primary key, or 0 if it is not part of primary key.
FOREIGNKEY	CHAR(1) NOT NULL Applies to character columns only: • B (column contains bit data). • M (MIXED data) • S (indicates SBCS data if encoding scheme is UNICODE or MIXDATA field=YES on DSNTIPF. • blank (indicates either mixed data if encoding scheme is UNICODE or not UNICODE and value of MIXED DATA on installation panel DSNTIPS is YES. Or, SBCS data if the encoding scheme is not UNICODE and value of MIXED DATA on installation panel DSNTIPS is NO.
FLDPROC	CHAR(1) NOT NULL Whether column has a field proc: Y/N/blank. Only used for views defined before DB2 7.
LABEL	VARCHAR(90) NOT NULL Column label provided by the user with a LABEL ON statement; otherwise, an empty string.
STATSTIME	TIMESTAMP NOT NULL WITH DEFAULT The date and time when RUNSTATS last updated the statistics. The default value is '0001-01-01.00.00.00.000000'. A value of '0001-01-02.00.00.00.000000', indicates an ALTER TABLE statement was executed to change the length of a VARCHAR column. RUNSTATS should be run to update the statistics before they are used.
DEFAULTVALUE	VARCHAR(1536) NOT NULL WITH DEFAULT Meaningful for columns only if SYSTABLES TYPE column is T (table) or G (global temporary table). • When the DEFAULT column is 1, 2, 3, 4, 5, 6, 7, 8 or 9, it contains the default value. • If the default value is a string constant or hexadecimal constant, DEFAULT is 1, 5, 6, 7 or 8, • If the default value is a numeric constant, DEFAULT is 2, 3, 4. or 9 • When the DEFAULT column is S or U and the default value was specified when a new column was defined with the ALTER TABLE statement, this field contains the value of the CURRENT SQLID or USER special register at the time the ALTER TABLE statement was executed. • When the DEFAULT column is L and the column was added as a new column with the ALTER TABLE statement, this field

Name	Data Type/Description
	contains the security label of the user at the time the ALTER TABLE statement was executed.
COLCARDF	FLOAT NOT NULL WITH DEFAULT Estimated number of distinct values in the column. For an indicator, contains number of LOBs not null with length > 0. Value is -1 if statistics not gathered. Value is -2 if the column is a LOB column.
COLSTATUS	CHAR(1) NOT NULL WITH DEFAULT Indicates status of column definition: • blank (complete) • I (incomplete). LOB table space, auxiliary table, or index on auxiliary table not yet created.
LENGTH2	INTEGER NOT NULL WITH DEFAULT Maximum length of data retrieved from column: • 0 (not LOB or ROWID column) • 40 (ROWID column returned value is the length) • 1 to 2147483647 (for LOB column maximum value)
DATATYPEID	INTEGER NOT NULL WITH DEFAULT Internal identifier of data type.
SOURCETYPEID	INTEGER NOT NULL WITH DEFAULT Internal identifier of source type. 0 for built-in data types.
TYPESCHEMA	VARCHAR(128) NOT NULL WITH DEFAULT 'SYSIBM' If COLTYPE is distinct, schema of distinct type. Otherwise, SYSIBM.
TYPENAME	VARCHAR(128) NOT NULL WITH DEFAULT If COLTYPE is distinct, name of distinct type. Otherwise, same as COLTYPE column.
CREATEDTS	TIMESTAMP NOT NULL WITH DEFAULT Timestamp when column was created. A value of '0001-01-01.00.00.00.000000' is for columns created before DB2 6 migration.
STATS_FORMAT	CHAR(1) NOT NULL WITH DEFAULT Type of Statistics gathered: • Blank if no statistics gathered or if VARCHAR statistics have been padded. • N if VARCHAR statistics have not been padded. • This column is updateable.
PARTKEY_ COLSEQ	SMALLINT NOT NULL WITH DEFAULT The column's numeric position within the table's partitioning key. The value is 0 if it is not part of the partitioning key.
PARTKEY_ ORDERING	CHAR(1) NOT NULL WITH DEFAULT Order of the column in the partitioning key: • A (Ascending) • D (Descending)
ALTEREDTS	TIMESTAMP NOT NULL WITH DEFAULT Timestamp when alter occurred.
CCSID	INTEGER NOT NULL WITH DEFAULT

Name	Data Type/Description
	CCSID of the column. 0 if object was created before DB2 8 or is not a UNICODE defined VARBINARY.
HIDDEN	CHAR(1) NOT NULL WITH DEFAULT 'N' Indicated if column is hidden: • P—Partially hidden. From SELECT *. • N—Column is not hidden from all SQL statements.
RELCREATED	CHAR(1) NOT NULL Release of DB2 used to create table. Blank if created before DB2 9.
CONTROL_ID	INTEGER NOT NULL WITH DEFAULT Access control mask internal ID.
XML_TYPMOD_ID	INTEGER NOT NULL WITH DEFAULT XML type modifier ID.
PERIOD	CHAR(1) NOT NULL WITH DEFAULT Start or end of SYSTEM_TIME or BUSINESS_TIME: • B (start period BUSINESS_TIME) • C (end period BUSINESS_TIME EXCLUSIVE END TIME) • I (End period BUSINESS_TIME INCLUSIVE END TIME) • S (start period SYSTEM_TIME) • T (end period SYSTEM_TIME) • blank (not used)
GENERATED_ATTR	VARCHAR(128) NOT NULL Columns generated attribute: • A (GENERATED ALWAYS) • D (GENERATED BY DEFAULT) • blank (not applicable)
HASHKEY_COLSEQ	SMALLINT NOT NULL WITH DEFAULT Column's position within hash key.
ENCODING_SCHEME	CHAR(1) NOT NULL WITH DEFAULT 'E' Column encloding scheme: • A (ASCII) • E (EBCDIC) • U (UNICODE) • blank: Column has data type that does not have an encoding scheme or encoding scheme is the same as encoding scheme for table/view.

SYSIBM.SYSCOLUMNS_HIST		**(DSNDB06.SYSHIST)**
SYSIBM.DSNHEX01	**D**	**(TBCREATOR,TBNAME, NAME,STATISTIME)**
Contains statistical history information for every column of each table and view (populated by RUNSTATS). You can insert, update, and delete rows in this table.		

This table contains the following columns:

Name	Data Type/Description
NAME	VARCHAR(128) NOT NULL Name of the column.
TBNAME	VARCHAR(128) NOT NULL Name of the table or view that contains the column.
COLNO	SMALLINT NOT NULL

Name	Data Type/Description
	Numeric place or qualifier of the table or view that contains the column.
COLTYPE	CHAR(8) NOT NULL Type of column: • INTEGER (full-word binary) • SMALLINT (half-word binary) • FLOAT (floating point numeric) • CHAR (fixed length character) • VARCHAR (variable length character) • LONGVAR (variable length character) • DECIMAL (packed decimal) • GRAPHIC (fixed length DBCS) • VARG (variable length DBCS) • LONGVARG (variable length DBCS) • DATE (date) • TIME (time) • TIMESTMP (timestamp) • TIMESTZ (timestamp with zone) • BLOB (binary large object) • CLOB (character large object) • DBCLOB (double-byte character large object) • ROWID (ROW ID data type) • DISTINCT (distinct data type) • XML (XML data type) • BIGINT (big integer) • BINARY (fixed length binary string) • VARBIN (varying length binary string) • DECFLOAT (decimal floating string)
LENGTH	SMALLINT NOT NULL The length attribute of the column, or in the case of a decimal column, its precision: • INTEGER – 4 • SMALLINT – 2 • FLOAT – 4 or 8 • CHAR – length of string • VARCHAR – max. length of string • LONGVAR – max. length of string • DECIMAL – precision of number • GRAPHIC – number of DBCS chars. • VARGRAPHIC – max. # of DBCS chars. • LONGVARG – max. # of DBCS chars. • DATE – 4 • TIME – 3 • TIMESTMP – 10 • TIMESTAMP WITHOUT TIME ZONE – Integral part of (((p+1)/2 + 9 where p is precision of timestamp • TIMESTAMP WITH TIME ZONE – Integral part of (((p+1)/2 + 7 where p is precision of timestamp • BLOB – 4 length of field in base table, max. length of LOB in LENGTH2. • CLOB – 4 length of field in base table, max. length of CLOB in LENGTH2. • DBCLOB – 4 length of field in base table, max. length of DBCLOB in LENGTH2. • ROWID – 17 • DISTINCT - length of source data type • XML – 6 • BIGINT – 8

Name	Data Type/Description
	• BINARY – Length of string • VARBINARY – Maximum length of string • DECFLOAT – 8 or 16 The number does not include the internal prefixes used to record the actual length and null state, where applicable.
LENGTH2	INTEGER NOT NULL Maximum length of the data retrieved from the column. Possible values are: • 0—Not a LOB or ROWID column. • 40—For a ROWID column, the length of the returned value 1 to 2,147,483,647 bytes. For a LOB column, the maximum length of the LOB.
NULLS	CHAR(1) NOT NULL Whether the column can contain null values: N/Y.
HIGH2KEY	VARCHAR(2000) NOT NULL FOR BIT DATA Second highest value of the column. Blank if statistics have not been gathered, or the column is an indicator column or a column of an auxiliary table. If the column has a non-character data type, the data might not be printable.
LOW2KEY	VARCHAR(2000) NOT NULL FOR BIT DATA Second lowest value of the column. Blank if statistics have not been gathered, or the column is an indicator column or a column of an auxiliary table. If the column has a non-character data type, the data might not be printable.
STATSTIME	TIMESTAMP NOT NULL If RUNSTATS updated the statistics, the date and time when the last invocation of RUNSTATS updated the statistics. The default value is '0001-01-01.00.00.00.000000'. If the value is '0001-01-02.00.00.00.000000', this indicates that an ALTER TABLE statement was executed to change the length of a VARCHAR column.
COLCARDF	FLOAT(8) NOT NULL WITH DEFAULT -1 Estimated number of distinct values in the column. For an indicator column, this is the number of LOBs that are not null and have a length greater than zero. The value is -1 if statistics have not been gathered. The value is -2 if LOB column.
IBMREQD	CHAR(1) NOT NULL WITH DEFAULT 'N' Whether the row came from the basic MRM tape: Y/N/other. If other, release dependency indicator.
STATS_ FORMAT	CHAR(1) NOT NULL WITH DEFAULT Type of statistics gathered: • blank—No statistics gathered or if VARCHAR statistics have been padded. • N—If VARCHAR statistics have not been padded. • This column is updateable..

SYSIBM.SYSCONSTDEP		(DSNDB06.SYSTSCON)
SYSIBM.DSNCCX01	D	(BSCHEMA,BNAME,BTYPE)
SYSIBM.DSNCCX02	D	(DTBCREATOR,DTBNAME)
Contains dependencies on check constraints or user-defined defaults for column.		

This table contains the following columns:

Name	Data Type/Description
BNAME	VARCHAR(128) NOT NULL Object name on which dependency exists.
BSCHEMA	VARCHAR(128) NOT NULL Schema for object on which dependency exists.
BTYPE	CHAR(1) NOT NULL Object type on which dependency exists. • F (function instance)
DTBNAME	VARCHAR(128) NOT NULL Table name on which dependency applies.
DTBCREATOR	VARCHAR(128) NOT NULL The schema of the table to which the dependency applies.
DCONSTNAME	VARCHAR(128) NOT NULL For DTYPE: • C—Check constraint unqualified name • D—Column name
DTYPE	CHAR(1) NOT NULL Object type: • C—Check constraint • D—User-defined default constant
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other. If other, release dependency indicator.
DTBOWNER	VARCHAR(128) NOT NULL Authorization ID of the owner of the table or a zero length string for tables that were created before DB2 9.
OWNERTYPE	CHAR(1) NOT NULL Indicate the type of owner: • Blank—Authorization ID. • R—Role.

SYSIBM.SYSCONTEXT		(DSNDB06.SYSCONTX)
SYSIBM.DSNCTX01	U	(NAME)
SYSIBM.DSNCTX02	U	(SYSTEMAUTHID)
SYSIBM.DSNCTX03	P	(CONTEXTID)
SYSIBM.DSNCTX04	D	(DEFAULTROLE)
Defines trusted contexts, one row for each.		

This table contains the following columns:

Name	Data Type/Description
NAME	VARCHAR(128) NOT NULL Name of the trusted context.
CONTEXTID	INTEGER NOT NULL GENERATED ALWAYS AS IDENTITY Internal context ID.
DEFINER	VARCHAR(128) NOT NULL Authorization ID or role that defined the trusted context.

Name	Data Type/Description
DEFINERTYPE	CHAR(1) NOT NULL The type of definer: • L (role) • Blank (authorization ID)
SYSTEMAUTHID	VARCHAR(128) NOT NULL Primary authorization ID that is used to establish the connection. For remote requsts, this value is derived from the system user ID that is provided by an external entity, such as a middleware server. For local requests, this value depends on one of the following sources of the address space: • BATCH—USER parameter on JOB statement. • RRSAF—USER parameter on JOB statement or IBM RACF user TSO logon ID. • TSO—TSO logon ID
DEFAULTROLE	VARCHAR(128) NOT NULL Name of the trusted context default role.
OBJECTOWNERTYPE	CHAR(1) NOT NULL Specification of ROLE AS OBJECT OWNER on this trusted context: • L—ROLE AS OBJECT OWNER is specified. A role owns any object created in the trusted context. • blank—ROLE AS OBJECT OWNER is not specified. An authorization ID owns any object created in the trusted context.
CREATEDTS	TIMESTAMP The time when the trusted context is created.
ALTEREDTS	TIMESTAMP The time when the trusted context is last altered.
ENABLED	CHAR(1) NOT NULL The status of the trusted context: Y (enabled) or N (disabled).
ALLOWPUBLIC	CHAR(1) NOT NULL Whether the connection is allowed to be reused for PUBLIC: Y (allowed) N (not allowed).
AUTHENTICATE PUBLIC	CHAR(1) NOT NULL Whether automation is required for PUBLIC when ALLOWPUBLIC is Y: • Y—Authentication token is required for PUBLIC. For local requests, the token is the password. For remote requests, the token can be a password, RACF PassTicket, or a KERBEROS token. • N—Authentication is not required.
RELCREATED	CHAR(1) NOT NULL The release of DB2 that is used to create the object.

Name	Data Type/Description
IBMREQD	CHAR(1) NOT NULL A value of Y indicates that the row came from the basic machine-readable material (MRM) tape: Y/N/other.
REMARKS	VARCHAR(762) NOT NULL A character string provided using the COMMENT statement.
DEFAULTSECURITYLABEL	VARCHAR(24) NOT NULL Name of the context default RACF security label.
SYS_START	TIMESTAMP(12) NOT NULL System period temporal start time for transaction.
SYS_END	TIMESTAMP(12) NOT NULL System period temporal end time for transaction.
TRANS_START	TIMESTAMP(12) System period transaction timestamp.

SYSIBM.SYSCONTEXTAUTHIDS		**(DSNDB06.SYSCONTX)**
SYSIBM.DSNCDX01	**U**	**(CONTEXTID,AUTHID)**
SYSIBM.DSNCDX02	**D**	**(ROLE)**
Defines the authorization IDs that can be used by a trusted context. One row for each authorization ID/trusted context combination.		

This table contains the following columns:

Name	Data Type/Description
CONTEXTID	INTEGER NOT NULL The internal trusted context ID.
AUTHID	VARCHAR(128) NOT NULL The primary authorization ID that can reuse a connection or the RACF profile name that contains the primary authorization IDs in the identified trusted context.
AUTHENTICATE	CHAR(1) NOT NULL Whether authentication is required for the authorization ID in the AUTHID column: Y (required) or N (not required). When authorization is required, the token is the password for local requests. For remote requests, the token can be a password, a RACF PassTicket, or a Kerberos token.
ROLE	VARCHAR(128) NOT NULL The role for the authorization ID in the AUTHID column. The role supersedes the default role that is defined for the trusted context.
CREATEDTS	TIMESTAMP NOT NULL The time when the authorization ID is added to the trusted context.
IBMREQD	CHAR(1) NOT NULL A value of Y indicates that the row came from the basic machine-readable material (MRM) tape.
SYS_START	TIMESTAMP(12) NOT NULL System period temporal start time for transaction.
SYS_END	TIMESTAMP(12) NOT NULL

Name	Data Type/Description
	System period temporal end time for transaction.
TRANS_START	TIMESTAMP(12) System period transaction timestamp.

SYSIBM.SYSCONTROLS		(DSNDB06.SYSTSCTL)
SYSIBM.DSNCLX01	U	(SCHEMA,NAME)
SYSIBM.DSNCLX02	U	(CONTROL_ID)
SYSIBM.DSNCLX03	D	(TBSCHEMA,TBNAME)
SYSIBM.DSNCLX04	D	(TBSCHEMA,TBNAME,ENABLE)
SYSIBM.DSNCLX05	D	(TBSCHEMA,TBNAME,ENABLE, CONTROL_TYPE)
Contains information for each row permission and column mask. One row per row permission or column mask.		

This table contains the following columns:

Name	Data Type/Description
SCHEMA	VARCHAR(128) NOT NULL Row permission or column mask schema.
NAME	VARCHAR(128) NOT NULL Row permission or column mask name.
OWNER	VARCHAR(128) NOT NULL Row permission or column mask owner.
OWNERTYPE	CHAR(1) NOT NULL Owner type: blank (authorization ID) or L (role)
TBSCHEMA	VARCHAR(128) NOT NULL Schema of table row permission or column mask is defined.
TBNAME	VARCHAR(128) NOT NULL Name of table row permission or column mask is defined.
TBCORRELATION	VARCHAR(128) NOT NULL WITH DEFAULT Correlation name of table row permission or column mask is defined. Empty string – no correlation name.
COLNAME	VARCHAR(128) NOT NULL Column name column mask is defined. Blank if row permission.
COLNO	SMALLINT NOT NULL Column name column mask is defined. 0 if row permission.
CONTROL_ID	INTEGER NOT NULL GENERATED ALWAYS AS IDENTITY Internal access control ID.
CONTROL_TYPE	CHAR(1) NOT NULL Type of access control object: R (row permission) M (column mask).
ENFORCED	CHAR(1) NOT NULL Type of access enforced by row permission. A (all access).
IMPLICIT	CHAR(1) NOT NULL Row permission was implicitly created. N (row permission explicitly created or this is a column mask). Y (row permission implicitly created).
ENABLE	CHAR(1) NOT NULL

Name	Data Type/Description
	Row permission or column mask enabled for access control: N (not enabled) or Y (enabled).
STATUS	CHAR(1) NOT NULL Status of row permission or column mask: blank (row permission or column mask complete). R (error occurred generating row permission or column mask).
CREATEDTS	TIMESTAMP NOT NULL Create time of row permission or column mask.
RELCREATED	CHAR(1) NOT NULL Release of DB2 creating row permission or column mask.
ALTEREDTS	TIMESTAMP NOT NULL Alter time or row permission or column mask.
REMARKS	VARCHAR(762) NOT NULL Character string provided by COMMENT ON statement.
IBMREQD	CHAR(1) NOT NULL Y—Row came from basic machine-readable material (MRM). For other values, see release dependency indicators.
ENVID	INTEGER NOT NULL Environment internal identifier.
ROWID	ROWID Row identifier.
RULETEXT	CLOB(2MB) NOT NULL Source text of the search condition or expression from the CREATE PERMISSION or CREATE MASK statement. Set the appropriate precompiler option to fold ordinary tokens in a C or Java program to uppercase.
DESCRIPTOR	BLOB(2MB) NOT NULL Internal description of row permission or column mask.
SYS_START	TIMESTAMP(12) NOT NULL System period temporal start time for transaction.
SYS_END	TIMESTAMP(12) NOT NULL System period temporal end time for transaction.
TRANS_START	TIMESTAMP(12) System period transaction timestamp.
REGENERATETS	TIMESTAMP(12) NOT NULL Timestamp when object was regenerated.

SYSIBM.SYSCONTROLS_DESC		**(DSNDB06.SYSTSCTD)**
SYSIBM.DSNTRX02	**U**	**(AUXID,AUXVER)**
Auxiliary table for DESCRIPTOR LOB column of SYSCONTROLS table.		

This table contains the following column:

Column Name	Data Type	Description
AUXID	VARCHAR(17)	ID of auxiliary data.
AUXVER	SMALLINT	Version of auxiliary data.
AUXVALUE	BLOB(1G) NOT NULL WITH DEFAULT	IBM internal use

SYSIBM.SYSCONTROLS_RTXT		(DSNDB06.SYSTSCTR)
SYSIBM.DSNTRX01	U	(AUXID,AUXVER)
Auxiliary table for RULETEXT LOB column of SYSCONTROLS table.		

This table contains the following column:

Column Name	Data Type	Description
AUXID	VARCHAR(17)	ID of auxiliary data.
AUXVER	SMALLINT	Version of auxiliary data.
AUXVALUE	BLOB(1g) NOT NULL WITH DEFAULT	IBM internal use

SYSIBM.SYSCOPY		(DSNDB06.SYSTSCPY)
SYSIBM.DSNUCH01	D	(DBNAME,TSNAME,START_RBA, TIMESTAMP)
SYSIBM.DSNUCX01	D	(DSNAME)
Contains image copy information that is needed for recovery.		

This table contains the following columns:

Name	Data Type/Description
DBNAME	CHAR(8) NOT NULL Name of the database.
TSNAME	CHAR(8) NOT NULL Name of the target table space or index space.
DSNUM	INTEGER NOT NULL Data set number within table space. For partitioned table spaces, this value corresponds to the partition number for a single partition copy, or 0 for a copy of an entire partitioned table space or index space.
ICTYPE	CHAR(1) NOT NULL Operation type: • A—ALTER • B—REBUILD INDEX • D—CHECK DATA LOG(NO) • E—RECOVER (to current point) • F—COPY FULL YES • I—COPY FULL NO • L—SQL type of operation • M—MODIFY RECOVERY • P—RECOVER TCOPY/TORBA • Q—QUIESCE • R—LOAD REPLACE LOG(YES) • S—LOAD REPLACE LOG(NO) • T—TERM utility • V—REPAIR VERSION utility • W—REORG LOG(NO) • X—REORG LOG(YES) • Y—LOAD LOG(NO) • Z—LOAD LOG(YES)
ICDATE	CHAR(6) NOT NULL (unused)
START_RBA	CHAR(10) NOT NULL FOR BIT DATA 80-bit positive integer containing the RBA/LRSN (relative byte location in a non–

Name	Data Type/Description
	data-sharing environment) of a point in the DB2 recovery log. Column contains bit data. For ICTYPE: • I or F—Starting point for all updates since image copy was taken • J—RBA/LRSN of compression dictionary. • P—Point after log-application phase of point-in-time recovery. • Q—Point after all data sets have been successfully quiesced. • R or S—End of log before start of the LOAD utility and before any data changed. • T—The end of log when utility is terminated. • Other—End of log before start of RELOAD phase of LOAD or REORG utility
FILESEQNO	INTEGER NOT NULL Tape file sequence number of the copy.
DEVTYPE	CHAR(8) NOT NULL Device type the copy is on.
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other.
DSNAME	CHAR(44) NOT NULL For ICTYPE P, I, or F, DSNAME is the name of the data set. Otherwise, it contains the database name and table space, or index name. Blank for any row migrated from a release before DB2 4.
ICTIME	CHAR(6) Unused.
SHRLEVEL	CHAR(1) NOT NULL SHRLEVEL parameter on COPY (for ICTYPE F or I only): C (change), R (reference), blank (does not describe image copy).
DSVOLSER	VARCHAR(1784) NOT NULL Volume serial numbers of the data set; 6-byte numbers separated by commas. Blank if data set is catalogued.
TIMESTAMP	TIMESTAMP NOT NULL WITH DEFAULT Date and time when row was inserted (from ICDATE and ICTIME).
ICBACKUP	CHAR(2) NOT NULL WITH DEFAULT Specifies the type of image copy: • blank – local site primary copy • PC – FlashCopy copy • LB – local site backup copy • RP – recovery site primary copy • RB – recovery site backup copy
ICUNIT	CHAR(1) NOT NULL WITH DEFAULT Indicates device used for image copy: D (DASD), T (tape), or blank (medium is not tape or DASD).
STYPE	CHAR(1) NOT NULL WITH DEFAULT For ICTYPE=A: • A—A partition was added to the table. • B—MEMBER CLUSTER value changed. • C—A column was added to a table and an index in different scopes or a column

Name	Data Type/Description
	dropped from table, index compression activated/deactivated. • D—DSSIZE attribute altered. • E—The data set numbers of a base table and its associated clone table are exchanged. • F—Page size attribute altered. • G—An index was regenerated. • I—Inline length of LOB column altered. • L—The logging attribute of the table space was altered to LOGGED. • M—MAXPARTITIONS altered. • N—An index was altered to not padded. • O—The logging attribute of the table space was altered to NOT LOGGED. • P—An index was altered to padded. • R—A table was altered to rotate partitions. • S—SEGSIZE attribute altered. • V—Length of VARCHAR column increase. • X—A REORG dropped empty partitions. • Y—Index altered to be COPY YES. • Z—A column was altered that is in the key of an index that was versioned prior to DB2 8.
STYPE (continued)	ICTYPE=C: • L—The logging attribute of the table space was altered to LOGGED. • O—The logging attribute of the table space was altered to NOT LOGGED.
STYPE (continued)	ICTYPE=E: • B—RECOVER utility with BACKOUT keyword. • blank—RECOVER utility without BACKOUT keyword.
STYPE (continued)	ICTYPE=F: • blank—DB2 image copy. • C—DFSMS concurrent copy (“I” instance of the table space). • J—DFSMS concurrent copy. (“J” instance of the table space). RECOVER utility without BACKOUT keyword. • N—FlashCopy copy inconsistent. • Q—Sequential copy consistent. • S—LOAD REPLACE (NO). • T—FlashCopy consistent. • U—Sequential copy inconsistent. • V—ALTERED INDEX NOT PADDED. • W—REORG LOG (NO). • X—REORG LOG (YES).
STYPE (continued)	ICTYPE=L: • M—Mass DELETE, TRUNCATE TABLE, DROP TABLE, or ROTATE partition.
STYPE (continued)	ICTYPE=M and the MODIFY RECOVERY utility was executed to delete SYSCOPY and/or SYSLGRNX records, the value is R.
STYPE (continued)	ICTYPE=O:

Name	Data Type/Description
	• B—Tablespace or partition was in RRF but recovered to a PIT where it was BRF format. • R—Tablespace or partition converted to RRF using REORG or LOAD REPLACE.
STYPE (continued)	ICTYPE=P: • B—Recover to a point with the BACKOUT YES option. • C—Recover to a point-in-time without using logonly with consistency. • L—Recover to a point-in-time using logonly without consistency. • M—Recover to a point-in-time using logonly with consistency. • blank—Recover to a point-in-time without using logonly without consistency.
STYPE (continued)	ICTYPE=Q: • W—Option is WRITE (YES) for QUIESCE point.
STYPE (continued)	ICTYPE=R or S: • A—REORG pending status was reset. • T—First materialized default value for row change, timestamp column.
STYPE (continued)	ICTYPE=T: FCOPY FULL YES I COPY FULL NO
STYPE (continued)	ICTYPE=W or X: • A—REORG pending status was reset or REBALANCE. • H—Hash organization attributes altered. • T—First materializing default value for row change timestamp column.
STYPE (continued)	Other ICTYPE values, value is blank.
PIT_RBA	CHAR(10) NOT NULL WITH DEFAULT FOR BIT DATA • ICTYPE=P, field contains the LRSN for the point in the DB2 log or the stop location of a point-in-time-recovery. • ICTYPE=F or I and SHRLEVEL=C, this column contains the current RBA or LRSN that corresponds to the point in the DB2 log when the SHRLEVEL CHANGE copy completes. • ICTYPE=J, the RBA where compression dictionary written to log. • ICTYPE=M, the RBA for end-of-log when utility completed.
GROUP_MEMBER	CHAR(8) NOT NULL WITH DEFAULT DB2 data sharing member name of the DB2 subsystem that performed the operation. Blank if not part of DB2 data sharing at the time of operation.
OTYPE	CHAR(1) NOT NULL WITH DEFAULT 'T' Object type for recovery information: I (indexspace) or T (tablespace).
LOWDSNUM	INTEGER NOT NULL WITH DEFAULT Lowest partition number in range for SYSCOPY records created via REORG or

Name	Data Type/Description
	LOAD REPLACE to reset REORG pending. Version number of index for SYSCOPY records created via COPY of an index space.
HIGHDSNUM	INTEGER NOT NULL WITH DEFAULT Highest partition number in a range for SYSCOPY records created via REORG or LOAD REPLACE to reset REORG pending.
COPYPAGESF	FLOAT(8) NOT NULL WITH DEFAULT -1 Number of pages written to the copy data set.
NPAGESF	FLOAT(8) NOT NULL WITH DEFAULT -1 The number of pages in the table space or index at the time of INLINE COPY.
CPAGESF	FLOAT(8) NOT NULL WITH DEFAULT -1 Total number of changed pages.
JOBNAME	CHAR(8) NOT NULL WITH DEFAULT Job name of the utility.
AUTHID	CHAR(8) NOT NULL WITH DEFAULT Authorization ID of the utility.
OLDEST_VERSION	SMALLINT NOT NULL WITH DEFAULT When IC Type = B, F, I, S, W, or X, the version number for the oldest format of data for an object. The value is –1 for other values of IC Type.
LOGICAL_PART	INTEGER NOT NULL WITH DEFAULT The logical partition number.
LOGGED	CHAR(1) NOT NULL WITH DEFAULT Indicates the logging attribute of the table space at the time the SYSCOPY record is written. • Y—Table space has LOGGED attribute. • N—Table space has NOT LOGGED attribute. • Blank—The row was inserted prior to DB2 9. For a non-LOB table space or an index space, blank indicates that the logging attribute is LOGGED.
TTYPE	CHAR(8) NOT NULL WITH DEFAULT When ICTYPE=A and STYPE=B, then using previous value for MEMBER CLUSTER: Y/N. When ICTYPE=A and STYPE=C: • A—Column added to table. • D—Column dropped from table. • blank—Column added to table. • CMP=N: Index compression activated. • CMP=Y: Index compression deactivated.
TTYPE (continued)	When ICTYPE=A and STYPE=D then using previous DSSIZE value.
TTYPE (continued)	When ICTYPE=A and STYPE=F then using previous page size value.
TTYPE (continued)	When ICTYPE=A and STYPE=I then inline length of LOB column altered: • D—REORG decremented • N—REORG incremented
TTYPE (continued)	When ICTYPE=A and STYPE=M then table space changed: • I—Converted from single-table to partition-by-growth universal table space.

Name	Data Type/Description
	• n—Uses previous value of MAXPARTITIONS. • S—Converted from single-table to partition-by-growth universal table space.
TTYPE (continued)	When ICTYPE=A and STYPE=S then table space changed: • n—Uses previous value of SEGSIZE. • P—Converted from partitioned to range-partitioned universal table space.
TTYPE (continued)	When ICTYPE=E full recovery reset object: blank (YES) or N (NO).
TTYPE (continued)	When ICTYPE=A and STYPE=N, Q, T, or U, then FlashCopy was made by: • A—LOADRESUME LOG NO • B—REBUILD • C—COPY • D—CHECK DATA • E—LOAD SHRLEVEL CHANGE • L—LOAD • P—REPAIR • R—REORG TABLESPACE • S—REORG INDEX • T--COPYTOCOPY • W—REORG TABLESPACE LOG NO • X—REORG TABLESPACE LOG YES
TTYPE (continued)	When ICTYPE=I and TTYPE=S, then directory pages for the index image copy are in front of each partition, as indicated with a ‘V’ or ‘8’.
TTYPE (continued)	When ICTYPE=P,R,S,W,X then the row format is: • B–RBA changed to 6 byte format • RRF–Reordered Row Format • RRF I–Reordered Row Format and FORMAT INTERNAL specified • BRF–Basic Row Format • BRF I–Basic Row Format and FORMAT INTERNAL specified • E–RBA changed to 10 byte • F–REORG using FASTSWITCH • S–REORG w/o FASTSWITCH
TTYPE (continued)	When ICTYPE=M and STYPE = R then MODIFY RECOVERY utility deleted rows from SYSLGRNX: blank (YES)/N (NO).
TTYPE (continued)	When ICTYPE=T and TYPE = B then broken page detected during copy.
TTYPE (continued)	When ICTYPE=W or X and STYPE=H then prior value of HASHDATAPAGES.
TTYPE (continued)	When ICTYPE=Y or Z: ▪ I–FORMAT INTERNAL not used during LOAD ▪ blank–FORMAT INTERNAL used during LOAD.
TTYPE (continued)	When ICTYPE / STYPE = A/A, A/R, B, C, P, R, S, W, or X: Page format changed by ADD PARTITION, ROTATE PARTITION, CREATE, LOAD, REPLACE, REBUILD, REORG, or RECOVER:

Name	Data Type/Description
	• B—Page format changed to 6 byte RBA • E—Page format changed to 10 byte RBA
TTYPE (continued)	When ICTYPE=A and STYPE=P, tablespace converted from absolute to relative page numbering.
INSTANCE	SMALLINT NOT NULL WITH DEFAULT 1 When STYPE = 'E' and When STYPE = E and ICTYPE = A, INSTANCE indicates the data set instance number of a base object after an EXCHANGE statement completes. The value of the INSTANCE column for the last data exchange will match the value of the INSTANCE column for the SYSIBM.SYSTABLESPACE table. For an image copy, INSTANCE indicates the instance number of the current base objects (table and index).
RELCREATED	CHAR(1) NOT NULL WITH DEFAULT The release of DB2 used to create the object. Blank if created before DB2 9.
MODECREATED	CHAR(2) NOT NULL WITH DEFAULT Latest mode where DB2 subsystem has been migrated when the SYSCOPY row was written: • C (Conversion mode) • E (Enable New Function mode) • N (New Function mode)

SYSIBM.SYSCTXTTRUSTATTRS		**(DSNDB06.SYSCONTX**
SYSIBM.DSNCAX01	**U**	**(CONTEXTID,NAME, VALUE)**
Contains information on each list of attributes for a given trusted context. One row per list of attributes.		

This table contains the following columns:

Name	Data Type/Description
CONTEXTID	INTEGER NOT NULL The internal trusted context ID.
NAME	VARCHAR(128) NOT NULL Name of the trust attribute. Possible values including the following attributes: • An IPv4 address is represented as a dotted decimal IP address. An example of an IPv4 address is '9.112.46.111'. • An IPv6 address is represented as a colon hexadecimal address. An example of an IPv6 address is '2001:0DB8:0000:0000:0008:0800:200C:417A'. • A domain name which is converted to an IP address by the domain name server where a resulting IPv4 or IPv6 address is determined. • A job or started task name for local applications. If the job name ends with *, any job name that matches the characters prior to * in the specified job name are considered for establishing the trusted connection. A network access security zone name in the RACF® SERVAUTH class.

Name	Data Type/Description
VALUE	VARCHAR(254) NOT NULL The value of the trust attribute.
CREATEDTS	TIMESTAMP NOT NULL The time when the attribute is created.
IBMREQD	CHAR(1) NOT NULL A value of Y indicates that the row came from the basic machine-readable material (MRM) tape.
SYS_START	TIMESTAMP(12) NOT NULL System period temporal start time for transaction.
SYS_END	TIMESTAMP(12) NOT NULL System period temporal end time for transaction.
TRANS_START	TIMESTAMP(12) System period transaction timestamp.

SYSIBM.SYSDATABASE		**(DSNDB06.SYSTSDBA)**
SYSIBM.DSNDDH01	**U**	**(NAME)**
Defines each database. One row per database, except for database DSNDB01.		

This table contains the following columns:

Name	Data Type/Description
NAME	VARCHAR(24) Database name.
CREATOR	VARCHAR(128) NOT NULL Authorization ID of the owner of the database.
STGROUP	VARCHAR(128) NOT NULL Name of default storage group of database. Blank for system database.
BPOOL	CHAR(8) NOT NULL Name of default buffer pool of the table space. Blank for system table space.
DBID	SMALLINT NOT NULL Internal identifier for database. If more than 32511 databases have been created, the DBID is negative.
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other.
CREATEDBY	VARCHAR(128) NOT NULL WITH DEFAULT Authorization ID of user who created it.
ROSHARE	CHAR(1)/Unused.
TIMESTAMP	TIMESTAMP/Unused.
TYPE	CHAR(1) NOT NULL WITH DEFAULT Type of database: • blank—Not a work file database or a TEMP database. • W—A work file database. The database is DSNDB07, or it was created with the WORKFILE clause and used as a work file database by a member of a DB2 data sharing group Is a work file database.
GROUP_MEMBER	VARCHAR(24) NOT NULL WITH DEFAULT The DB2 data sharing member name of the DB2 subsystem that used this work file

Name	Data Type/Description
	database. Blank if the work file database was not created in a DB2 data sharing environment, or if the database is not a work file database (indicated by the TYPE column).
CREATEDTS	TIMESTAMP NOT NULL WITH DEFAULT Time the CREATE statement for this database was executed.
ALTEREDTS	TIMESTAMP NOT NULL WITH DEFAULT Time of the most recent ALTER DATABASE executed against this database.
ENCODING_ SCHEME	CHAR(1) NOT NULL WITH DEFAULT 'E' Default encoding scheme for the database: • E—EBCDIC • A—ASCII • U—UNICODE • blank—For DSNDB04, work files, and a TEMP database.
SBCS_CCSID	INTEGER NOT NULL WITH DEFAULT Default SBCS CCSID for the database. Databases created before DB2 5 or a TEMP database will contain zero.
DBCS_CCSID	INTEGER NOT NULL WITH DEFAULT Default DBCS CCSID for the database. Databases created before DB2 5 or a TEMP database will contain zero.
MIXED_CCSID	INTEGER NOT NULL WITH DEFAULT Default MIXED CCSID for the database. Databases created before DB2 5 or a TEMP database will contain zero.
INDEXBP	CHAR(8) NOT NULL WITH DEFAULT BP0 Name of default buffer pool for indexes.
IMPLICIT	CHAR(1) NOT NULL WITH DEFAULT 'N' Indicates whether the database was implicitly created: Y (implicitly created) or N (not implicitly created).
CREATORTYPE	CHAR(1) NOT NULL WITH DEFAULT Indicates the type of creator: • blank—Authorization ID. • L—Role
RELCREATED	CHAR(1) NOT NULL The release of DB2 used to create the database. Blank if created before DB2 9.

SYSIBM.SYSDATATYPES		**(DSNDB06.SYSTSDAT)**
SYSIBM.DSNODX01	**P**	**(SCHEMA,NAME)**
SYSIBM.DSNODX02	**U**	**(DATATYPEID)**
Defines distinct data types. One row for each user-defined type.		

This table contains the following columns:

Name	Data Type/Description
SCHEMA	VARCHAR(128) NOT NULL Schema of data type.
OWNER	VARCHAR(128) NOT NULL Owner of data type.
NAME	VARCHAR(128) NOT NULL Name of data type.
CREATEDBY	VARCHAR(128) NOT NULL

Name	Data Type/Description
	Authorization ID that created the data type.
SOURCESCHEMA	VARCHAR(128) NOT NULL Source data type schema.
SOURCETYPE	VARCHAR(128) NOT NULL Source type name.
METATYPE	CHAR(1) NOT NULL Class of data type: A (user-defined ordinary array type), L (user-defined associative array type), T (distinct type).
DATATYPEID	INTEGER NOT NULL Internal identifier of data type.
SOURCETYPEID	INTEGER NOT NULL Internal identifier of source type.
LENGTH	INTEGER NOT NULL Maximum length or precision of data type that is sourced from IBM-defined DECIMAL data type.
SCALE	SMALLINT NOT NULL One of the following: • Scale for DECIMAL data type • Fractional-second digits for TIMESTAMP with/without ZONE • For other data types: 0
SUBTYPE	CHAR(1) NOT NULL Subtype for data type, based on subtype of source type: • B—FOR BIT DATA • S—FOR SBCS DATA • M—FOR MIXED DATA • blank—Source type is not character type
CREATEDTS	TIMESTAMP NOT NULL Time data type was created.
ENCODING_ SCHEME	CHAR(1) NOT NULL Encoding scheme of data type: • E—EBCDIC • A—ASCII • U—UNICODE
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other.
REMARKS	VARCHAR(762) NOT NULL User-provided character string through COMMENT ON statement.
OWNERTYPE	CHAR(1) NOT NULL WITH DEFAULT Indicates the type of owner: • blank—Authorization ID. • L—Role
RELCREATED	CHAR(1) NOT NULL WITH DEFAULT The release of DB2 used to create the database. Blank if created before DB2 9.
INLINE_LENGTH	INTEGER NOT NULL WITH DEFAULT -1 Inline length of type if based on LOB source type. -1 Not applicable
ARRAYLENGTH	BIGINT NOT NULL WITH DEFAULT If ARRAY type, then max cardinality. If other data types, the value is 0.
ARRAYINDEXTYPEID	INTEGER NOT NULL WITH DEFAULT

Name	Data Type/Description
	Data type of index if data type is associative. If other data types, the value is 0.
ARRAYINDEXTYPELEN	BIGINT NOT NULL WITH DEFAULT Maximum length of array index for associate array type. If other data types, the value is 0.
ARRAYINDEXSUBTYPE	CHAR(1) NOT NULL WITH DEFAULT Subtype of array index: B (FOR BIT DATA), S (FOR SBCS DATA, M (FOR MIXED DATA), blank if array index is not a character type.

SYSIBM.SYSDBAUTH		**(DSNDB06.SYSTSDBU)**
SYSIBM.DSNADH01	**D**	**(GRANTEE,NAME,GRANTEETYPE)**
SYSIBM.DSNADH02	**D**	**(NAME)**
SYSIBM.DSNADX01	**D**	**(GRANTOR,NAME, GRANTORTYPE)**
Defines user privileges over databases. One or more rows for each user who is granted a database privilege for the database.		

This table contains the following columns:

Name	Data Type/Description
GRANTOR	VARCHAR(128) NOT NULL Authorization ID of user who granted the privileges, or PUBLIC or PUBLIC *.
GRANTEE	VARCHAR(128) NOT NULL Authorization ID of user who holds the privileges, or PUBLIC or PUBLIC *.
NAME	VARCHAR(24) NOT NULL Database name.
TIMESTAMP	CHAR(12)/Internal use only.
DATEGRANTED	CHAR(6)/Unused.
TIMEGRANTED	CHAR(8)/Unused.
GRANTEETYPE	CHAR(1) NOT NULL WITH DEFAULT Indicates the type of grantee: • blank—Authorization ID • L—Role
AUTHHOWGOT	CHAR(1) NOT NULL Authorization level of the user from whom the privileges were received: • blank—not applicable • C—DBCTL • D—DBADM • E—SECADM • G—ACCESSCTRL • L—SYSCTRL • M—DBMAINT • S—SYSADM
CREATETABAUTH	CHAR(1) NOT NULL Whether GRANTEE can create tables within the database: (see legend ** after table)
CREATETSAUTH	CHAR(1) NOT NULL Whether GRANTEE can create table spaces in the database: (see legend ** after table)
DBADMAUTH	CHAR(1) NOT NULL Whether GRANTEE has DBADM authority over database: (see legend ** after table)
DBCTRLAUTH	CHAR(1) NOT NULL

Name	Data Type/Description
	Whether GRANTEE has DBCTRL authority over database: (see legend ** after table)
DBMAINTAUTH	CHAR(1) NOT NULL Whether GRANTEE has DBMAINT authority over database: (see legend ** after table)
DISPLAYDBAUTH	CHAR(1) NOT NULL Whether GRANTEE can issue DISPLAY command for database: (see legend ** after table)
DROPAUTH	CHAR(1) NOT NULL Whether GRANTEE can issue the ALTER DATABASE and DROP DATABASE statements
IMAGCOPYAUTH	CHAR(1) NOT NULL Whether GRANTEE can use the COPY, MERGECOPY, MODIFY, and QUIESCE on the database: (see legend ** after table)
LOADAUTH	CHAR(1) NOT NULL Whether GRANTEE can LOAD tables in the database: (see legend ** after table)
REORGAUTH	CHAR(1) NOT NULL Whether GRANTEE can REORG table spaces and indexes in the database: (see legend ** after table)
RECOVERDBAUTH	CHAR(1) NOT NULL Whether the GRANTEE can use the RECOVER and REPORT utilities on table spaces in the database
REPAIRAUTH	CHAR(1) NOT NULL Whether GRANTEE can DIAGNOSE and REPAIR table spaces and indexes in the database: (see legend **)
STARTDBAUTH	CHAR(1) NOT NULL Whether the GRANTEE can use the START command against the database
STATSAUTH	CHAR(1) NOT NULL Whether GRANTEE can run CHECK and RUNSTATS on the database: (see legend ** after table)
STOPAUTH	CHAR(1) NOT NULL Whether GRANTEE can STOP the database: (see legend ** after table)
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other.
GRANTEDTS	TIMESTAMP NOT NULL WITH DEFAULT Time GRANT statement was executed.
GRANTORTYPE	CHAR(1) NOT NULL WITH DEFAULT Indicates the type of owner: • blank—Authorization ID • L—Role
SYS_START	TIMESTAMP(12)NOT NULL System period temporal start time for transaction
SYS_END	TIMESTAMP(12)NOT NULL System period temporal end time for transaction
TRANS_START	TIMESTAMP(12) System period transaction timestamp.

Legend **:

Blank=privilege not held; G=privilege held with GRANT option;
Y=privilege held without GRANT option.

SYSIBM.SYSDBRM		**(DSNDB06.SYSTSDBR)**
SYSIBM.DSNDBX01	**D**	**(PLNAME)**
SYSIBM.DSNDBX02	**P**	**(PLNAME,NAME)**
Contains information about each DBRM of each application plan. One row per DBRM per application plan.		

This table contains the following columns:

Name	Data Type/Description
NAME	VARCHAR(24) NOT NULL Name of the DBRM.
TIMESTAMP	CHAR(8) NOT NULL FOR BIT DATA Consistency Token.
PDSNAME	CHAR(132) NOT NULL Name of Partitioned Data Set (PDS) of which DBRM is a member.
PLNAME	VARCHAR(24) NOT NULL Application plan of which this DBRM is a part.
PLCREATOR	VARCHAR(128) NOT NULL Authorization ID of owner of application plan.
PRECOMPTIME	CHAR(8)/Unused.
PRECOMPDATE	CHAR(6)/Unused.
QUOTE	CHAR(1) NOT NULL SQL string delimiter for SQL statements in the DBRM: • N (apostrophe) • Y (quotation mark)
COMMA	CHAR(1) NOT NULL Decimal point representation for SQL statements in the DBRM: • N (period) • Y (comma)
HOSTLANG	CHAR(1) NOT NULL The host language used: • B—Assembler • C—OS/VS COBOL • D—C • F—FORTRAN • P—PL/1 • 2—VS COBOL II • 3—IBM COBOL • 4—C++
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other. If other, release dependency indicator.
CHARSET	CHAR(1) NOT NULL WITH DEFAULT Indicates if the system CCSID for SBCS was 290 (Katakana) at precompile: • A—No • K—Yes
MIXED	CHAR(1) NOT NULL WITH DEFAULT Indicates if MIXED option was in effect at precompile: Y/N.

Name	Data Type/Description
DEC31	CHAR(1) NOT NULL WITH DEFAULT Indicates if DEC31 option was in effect at precompile: blank (No), Y (Yes).
VERSION	VARCHAR(122) NOT NULL WITH DEFAULT Version identifier for the DBRM.
PRECOMPTS	TIMESTAMP NOT NULL WITH DEFAULT Time that DBRM was precompiled.
PLCREATORTYPE	CHAR(1) NOT NULL WITH DEFAULT Indicates the type of creator. • Blank—Authorization ID • L—Role
RELCREATED	CHAR(1) NOT NULL The release of DB2 used to create the object. Blank if created prior to V9.

SYSIBM.SYSDEPENDENCIES		**(DSNDB06.SYSTSDEP)**
SYSIBM.DSNONX01	**D**	**(BSCHEMA,BNAME,BTYPE, BCOLNAME)**
SYSIBM.DSNONX02	**D**	**(DSCHEMA,DNAME,DTYPE, DCOLNAME)**
Contains the dependencies between objects.		

This table contains the following columns:

Name	Data Type/Description
BNAME	VARCHAR(128) NOT NULL Name of the object on which another object is dependent. If BTYPE is 'F', the name is the specific name of the function. If BTYPE is 'W' or 'Z' then name of table.
BSCHEMA	VARCHAR(128) NOT NULL Schema or qualifier of the object on which another object is dependent.
BCOLNAME	VARCHAR(128) NOT NULL WITH DEFAULT Column name of the object on which another object is dependent.
BCOLNO	SMALLINT NOT NULL WITH DEFAULT Column number of the object on which another object is dependent.
BTYPE	CHAR(1) NOT NULL Type of object that is identified by BNAME, BSCHEMA, and BCOLNAME: • E—Instead of trigger • C—COLUMN • F—Function • G—Global temporary Table • I—Index • M—Materialized query table • Procedure • P—Partitioned tablespace • Q—Sequence • R—Tablespace • S—Synonym • T—Table • U—Distinct type • V—View • W—SYSTEM_TIME period • Z—BUSINESS_TIME period • 0—Alias

Name	Data Type/Description
BOWNER	VARCHAR(128) NOT NULL WITH DEFAULT Authorization ID of the owner of the object on which another object is dependent.
BOWNERTYPE	CHAR(1) NOT NULL Type of creator of the object on which another object is dependent: • L—Role • blank—Authorization ID that is not a role
DNAME	VARCHAR(128) NOT NULL Name of the object that has dependencies on another object.
DSCHEMA	VARCHAR(128) NOT NULL Schema or qualifier of the object that has dependencies on another object.
DCOLNAME	VARCHAR(128) NOT NULL Column name of the object that has dependencies on another object.
DCOLNO	SMALLINT NOT NULL WITH DEFAULT Column number of the object that has dependencies on another object.
DTYPE	CHAR(1) NOT NULL Type of the object that is identified by DNAME, DSCHEMA, DCOLNAME and DVERSION: • B—Basic trigger • C—Generated column • F—Function • I—Index • M—MQT • Procedure • V—View • X—Row permission • Y—Column mask • 1—Advanced trigger
DOWNER	VARCHAR(128) NOT NULL Authorization ID of the owner of the object that has dependencies on another object.
DOWNERTYPE	CHAR(1) NOT NULL Type of creator of the object that has dependencies on another object: • L—Role • blank—Authorization ID if not a role
IBMREQD	CHAR(1) NOT NULL A value of Y indicates that the row came from the basic machine-readable material (MRM) tape.
BAUTH	SMALLINT NOT NULL WITH DEFAULT Privilege ID
DVERSION	VARCHAR(122) NOT NULL WITH DEFAULT Version identifier of the object when the object identified in DSCHEMA and DNAME has a version; or 0.

SYSIBM.SYSDYNQRY		(DSNDB06.SYSTSDQY)
SYSIBM.DSNDQX01	P	(SDQ_STMT_ID,COPYID)
SYSIBM.DSNDQX02	D	(CURSCHEMA,QUERY_HASH, COPYID,RELBOUND)
SYSIBM.DSNDQX11	U	(STBLGRP,SDQ_STMT_ID, COPYID)
Contains information for the stabilization of access paths for dynamic SQL statements.		

This table contains the following columns:

Name	Data Type/Description
SDQ_STMT_ID	BIGINT NOT NULL GENERATED BY DEFAULT AS IDENTITY Identifier of the stabilized dynamic query.
STBLGRP	VARCHAR(128) NOT NULL The name of the stabilization group.
COPYID	SMALLINT NOT NULL The copy type of the stabilized runtime structures for the query. • The current copy. • 4—Invalid copy.
CURSQLID	VARCHAR(128) NOT NULL Current SQLID for the stabilized query.
CURSCHEMA	VARCHAR(128) NOT NULL Current schema for the stabilized query.
CURAPPLCOMPAT	VARCHAR(10) NOT NULL Current application compatibility for stabilized query.
QUERY_HASH	CHAR(16) NOT NULL FOR BIT DATA The hash key generated by the SQL statement text of the stabilized query.
QUERY_HASH_ VERSION	INTEGER NOT NULL Version of the query hash.
VALID	CHAR(1) NOT NULL Whether the stabilized query is valid. • A—ALTER statement changed table of base table of a view. For CREATE INDEX statement involved in data sharing this value does not invalidate query. • H—ALTER TABLE statement changed table or base table of view. If object created prior to DB2 5 the change invalidates the query. • N—Stabilized access path is not valid. • Y—Stabilized access path is valid.
LASTUSED	DATE NOT NULL Last execution date of stabilized query.
RELBOUND	CHAR(1) NOT NULL DB2 version where query was stabilized.
GROUP_MEMBER	VARCHAR(24) NOT NULL Data sharing member that updates row.
STBLTIME	TIMESTAMP NOT NULL Timestamp when statement was stabilized.
ROWID	CHAR(1) NOT NULL/Internal use.
STMTTEXT	CLOB(2M) NOT NULL SQL text and any attribute string.
DATA1	BLOB(2G) NOT NULL INLINE LENGTH(32031) Internal use.
DATA2	BLOB(2G) NOT NULL/Internal use.

Name	Data Type/Description
DATA3	BLOB(2G) NOT NULL/Internal use.
DATA4	BLOB(2G) NOT NULL/Internal use.
DATA5	VARCHAR(128) NOT NULL FOR BIT DATA Internal use.
DATA6	CHAR(8) NOT NULL FOR BIT DATA Internal use.
FUNCTION_LVL	VARCHAR(10) NOT NULL DB2 Function level when query was inserted.

SYSIBM.SYSDYNQRYDEP		**(DSNDB06.SYSTSDQD)**
SYSIBM.DSNDQX03	**D**	**(SDQ_STMT_ID,COPYID)**
SYSIBM.DSNDQX04	**D**	**(BQUALIFIER,BNAME,BTYPE, SDQ_STMT_ID,COPYID)**
SYSIBM.DSNDQX05	**D**	**(CLASS,AUTHID, AUTHID_TYPE)**
SYSIBM.DSNDQX12	**D**	**(STBLGRP,SDQ_STMT_ID, COPYID)**
Contains information about dependencies for stabilized dynamic query packages.		

This table contains the following columns:

Name	Data Type/Description
SDQ_STMT_ID	BIGINT NOT NULL Identifier of the stabilized dynamic query.
COPYID	SMALLINT NOT NULL The copy type of the stabilized runtime structures for the query. • 0—Current copy • 1—Previous copy • 2—Original copy
BQUALIFIER	VARCHAR(128) NOT NULL BTYPE = 'R', BNAME is the database BTYPE is 'F', 'O' or 'Q', the schema name for procedure or function or sequence. BTYPE = 'B' or 'C' BNAME is the table qualifier BTYPE = ' ', BNAME is a ROLE. Otherwise, the value is the schema of BNAME.
BNAME	VARCHAR(128) NOT NULL Name of an object the query is dependent on.
BTYPE	CHAR(1) NOT NULL Object identified by BNAME and BQUALIFIER: • B—BUSINESS_TIME • C—SYSTEM_TIME • E—INSTEAD OF trigger • F—User-defined function or cast function • G—Global temporary table • I—Index • M—Materialized query table • O—Stored procedure • P—Partitioned table space LARGE/DSSIZE • Q—Sequence object • R—Table space • S--Synonym • T--Table • U—Distinct type • V—View • W—SYSTEM_TIME period • Z—BUSINESS_TIME period

Name	Data Type/Description
	• 0—Alias
CLASS	CHAR(1) NOT NULL • A—Authorization dependency • D—DDL dependency
BAUTH	SMALLINT NOT NULL Privilege held when CLASS=A • 50—SELECTAUTH • 51—INSERTAUTH • 52—DELETEAUTH • 53—UPDATEAUTH • 64—EXECUTEAUTH • 263—USAGEAUTH • 291—READAUTH • 292—WRITEAUTH • 0—when class=D
AUTHID_TYPE	CHAR(1) NOT NULL Type of authorization indicated by AUTHID. • L—Name of ROLE • blank—Either CLASS=D or authorization-ID when CLASS=A
AUTHID	VARCHAR(128) NOT NULL Owner of privilege of object or zero Length string when CLASS=D. L Auth-id is a ROLE.
DBNAME	VARCHAR(2048) NOT NULL Database name if SDBADMAUTH=Y; otherwise blank.
BADMINAUTH	CHAR(1) NOT NULL Authority that allowed access to the object upon which the query is dependent when CLASS=A. • B: SDBADMAUTH • D: DBADMAUTH • G: ACCESSCTRLAUTH • K: SQLADMAUTH • L: SYSCTRLAUTH • S: SYSADMAUTH • T: DATAACCESSAUTH • blank: privilege not held
PUBLICAUTH	CHAR(1) NOT NULL • Y: privilege held by public bu user or role indicated in AUTHID. • blank: privilege not held by public or CLASS=D.
ALLOBJAUTH	CHAR(1) NOT NULL • Y: privilege held on all objects with schema by user or role in AUTHID. • blank: privilege not held on all objects within schema or CLASS=D.
QUERY_HASH	CHAR(16) FOR BIT DATA Hash key of statement if CLASS=D otherwise x'00'.
DMHE_PKNM_LNAME	VARCHAR(128) NOT NULL FOR BIT DATA Internal use.
DMHE_CONTK	CHAR(8) NOT NULL FOR BIT DATA Internal use.

SYSIBM.SYSDYNQRY_EXPL		(DSNDB06.SYSTSDQE)
SYSIBM.DSNDQX08	U	(AUXID,AUXVER)
Auxiliary table for DATA2 LOB column of SYSDYNQRY.		

This table contains the following column:

Column Name	Data Type	Description
AUXID	VARCHAR(17)	ID of auxiliary data.
AUXVER	SMALLINT	Version of auxiliary data.
AUXVALUE	BLOB(2G)	IBM internal use

SYSIBM.SYSDYNQRY_OPL		(DSNDB06.SYSTSDQO)
SYSIBM.DSNDQX10	U	(AUXID,AUXVER)
Auxiliary table for DATA4 LOB column of SYSDYNQRY.		

This table contains the following column:

Column Name	Data Type	Description
AUXID	VARCHAR(17)	ID of auxiliary data.
AUXVER	SMALLINT	Version of auxiliary data.
AUXVALUE	BLOB(2G)	IBM internal use

SYSIBM.SYSDYNQRY_SHTEL		(DSNDB06.SYSTSDQH)
SYSIBM.DSNDQX09	U	(AUXID,AUXVER)
Auxiliary table for DATA3 LOB column of SYSDYNQRY.		

This table contains the following column:

Column Name	Data Type	Description
AUXID	VARCHAR(17)	ID of auxiliary data.
AUXVER	SMALLINT	Version of auxiliary data.
AUXVALUE	BLOB(2G)	IBM internal use

SYSIBM.SYSDYNQRY_SPAL		(DSNDB06.SYSTSDQS)
SYSIBM.DSNDQX07	U	(AUXID,AUXVER)
Auxiliary table for DATA1 LOB column of SYSDYNQRY.		

This table contains the following column:

Column Name	Data Type	Description
AUXID	VARCHAR(17)	ID of auxiliary data.
AUXVER	SMALLINT	Version of auxiliary data.
AUXVALUE	BLOB(2G)	IBM internal use

SYSIBM.SYSDYNQRY_TXTL		(DSNDB06.SYSTSDQT)
SYSIBM.DSNDQX06	U	(AUXID,AUXVER)
Auxiliary table for STMTTEXT LOB column of SYSDYNQRY.		

This table contains the following column:

Column Name	Data Type	Description
AUXID	VARCHAR(17)	ID of auxiliary data.
AUXVER	SMALLINT	Version of auxiliary data.
AUXVALUE	BLOB(2G)	IBM internal use

SYSIBM.SYSDUMMY1	(DSNDB06.SYSEBCDC)
Contains one row (EBCDIC), used for SQL statements for which a table reference is required, but the contents of the table are unimportant.	

This table contains the following columns:

Name	Data Type	Description
IBMREQD	CHAR(1) NOT NULL	Whether the row came from the basic MRM tape: Y/N/other.

SYSIBM.SYSDUMMYA	(DSNDB06.SYSTSASC)
Contains one row (ASCII), used for SQL statements for which a table reference is required, but the contents of the table are unimportant.	

This table contains the following columns:

Name	Data Type	Description
IBMREQD	CHAR(1) NOT NULL	Whether the row came from the basic MRM tape: Y/N/other.

SYSIBM.SYSDUMMYE	(DSNDB06.SYSEBCDC)
Contains one row (EBCDIC), used for SQL statements for which a table reference is required, but the contents of the table are unimportant.	

This table contains the following columns:

Name	Data Type	Description
IBMREQD	CHAR(1) NOT NULL	Whether the row came from the basic MRM tape: Y/N/other.

SYSIBM.SYSDUMMYU	(DSNDB06.SYSTSUNI)
Contains one row (UNICODE), used for SQL statements for which a table reference is required, but the contents of the table are unimportant.	

This table contains the following columns:

Name	Data Type	Description
IBMREQD	CHAR(1) NOT NULL	Whether the row came from the basic MRM tape: Y/N/other.

SYSIBM.SYSENVIRONMENT		(DSNDB06.SYSTSENV)
SYSIBM.DSNOEX01	U	(ENVID)
Contains the environment variables when an object is created.		

This table contains the following columns:

Name	Data Type/Description
ENVID	INTEGER NOT NULL Internal identifier of the environment.
CURRENT_SCHEMA	VARCHAR(128) NOT NULL The current schema.
RELCREATED	CHAR(1) NOT NULL The release when the environment information is created.
PATHSCHEMAS	VARCHAR(2048) NOT NULL The schema path.

Name	Data Type/Description
APPLICATION_ ENCODING_CCSID	INTEGER NOT NULL The CCSID of the application environment.
ORIGINAL_ ENCODING_CCSID	INTEGER NOT NULL The original CCSID of the statement text string.
DECIMAL_POINT	CHAR(1) NOT NULL The decimal point indicator: • C (Comma) • P (Period)
MIN_DIVIDE_SCALE	CHAR(1) NOT NULL The minimum divide scale: • N (the usual rules apply for decimal division in SQL) • Y (Retain at least three digits to the right of the decimal point after any decimal division)
STRING_DELIMITER	CHAR(1) NOT NULL The string delimiter that is used in COBOL string constants: • A—Apostrophe (') • Q—Quote (")
SQL_STRING_ DELIMITER	CHAR(1) NOT NULL The SQL string delimiter that is used in string constants: • A—Apostrophe (') • Q—Quote (")
MIXED_DATA	CHAR(1) NOT NULL Whether mixed DBCS data is used: N (no) or Y (yes).
DECIMAL_ ARITHMETIC	CHAR(1) NOT NULL The rules that are to be used for CURRENT PRECISION and when both operands in a decimal operation have a precision of 15 or less: • DEC15—Specifies that the rules do not allow a precision greater than 15 digits • 2—DEC31 specifies that the rules allow a precision of up to 31 digits
DATE_FORMAT	CHAR(1) NOT NULL The date format: • I—ISO - yyyy-mm-dd • J—JIS - yyyy-mm-dd • U—USA - mm/dd/yyyy • E—EUR - dd.mm.yyyy • L—Locally defined by an installation exit routine
TIME_FORMAT	CHAR(1) NOT NULL The time format: • I—ISO - hh.mm.ss • J—JIS - hh.mm.ss • U—USA - hh:mm AM or hh:mm PM • E—EUR - hh.mm.ss • L—Locally defined by an installation exit routine
FLOAT_FORMAT	CHAR(1) NOT NULL The floating point format: • I—IEEE floating point format • S—System/390® floating point format
HOST_LANGUAGE	CHAR(8) NOT NULL

Name	Data Type/Description
	The host language: • ASM • C • CPP • IBMCOB • PLI • FORTRAN
CHARSET	CHAR(1) NOT NULL The character set: A Alphanumeric
FOLD	CHAR(1) NOT NULL FOLD is only applicable when HOST_LANGUAGE is C or CPP. Otherwise FOLD is blank. • N—Lower case letters in SBCS ordinary identifiers are not folded to uppercase • Y—Lower case letters in SBCS ordinary identifiers are folded to uppercase • blank—Not applicable
IBMREQD	CHAR(1) NOT NULL A value of Y indicates that the row came from the basic machine-readable material (MRM) tape. For other values, see Release dependency indicators.
ROUNDING	CHAR(1) NOT NULL WITH DEFAULT The rounding mode that is used when arithmetic and casting operations are performed on DECFLOAT data: • C (ROUND_CEILING) • D (ROUND_DOWN) • F (ROUND_FLOOR) • G (GOUND_HALF_DOWN) • E (ROUND_HALF_EVEN) • H (ROUND_HALF_UP) • U (ROUND_UP)
CREATEDTS	TIMESTAMP(12) NOT NULL The time when the row was inserted.
APPLCOMPAT	VARCHAR(10) NOT NULL WITH DEFAULT The application compatibility level that is associated with this environment.

SYSIBM.SYSFIELDS		**(DSNDB06.SYSTSFLD)**
SYSIBM.DSNDFX01	**D**	**(TBCREATOR,TBNAME,NAME)**
Contains information about each column that has a field procedure. One row per column that has a field procedure.		

This table contains the following columns:

Name	Data Type/Description
TBCREATOR	VARCHAR(128) NOT NULL Schema or qualifier of the table that contains the column.
TBNAME	VARCHAR(128) NOT NULL Name of table that contains the column.
COLNO	SMALLINT NOT NULL Numerical place of this column in table.
NAME	VARCHAR(128) NOT NULL Name of the column.
FLDTYPE	VARCHAR(24) NOT NULL

Name	Data Type/Description
	Data type of the encoded values in the field: • INTEGER (large integer) • SMALLINT (small integer) • FLOAT (floating point) • CHAR (fixed length character string) • VARCHAR (variable length character string) • DECIMAL (decimal) • GRAPHIC (fixed length graphic string) • VARG (variable length graphic string)
LENGTH	SMALLINT NOT NULL Length attribute of the field or for a decimal field, its precision: • INTEGER (4) • SMALLINT (2) • FLOAT (8) • CHAR (length of string) • VARCHAR (maximum length of string) • DECIMAL (precision of number) • GRAPHIC (number of DBCS characters) • VARG (maximum number of DBCS characters)
SCALE	SMALLINT NOT NULL Scale if FLDTYPE is DECIMAL, else 0.
FLDPROC	VARCHAR(24) NOT NULL For a row describing a field procedure, the name of the procedure.
WORKAREA	SMALLINT NOT NULL For a row describing a field procedure, size (in bytes), of work area required for encoding and decoding of the field procedure.
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other.
EXITPARML	SMALLINT NOT NULL For a row describing a field procedure, length of the field procedure parameter value block.
PARMLIST	VARCHAR(762) NOT NULL For a row describing a field procedure, the parameter list following FIELDPROC in the statement that created the column, with insignificant blanks removed.
EXITPARM	VARCHAR(1530) NOT NULL WITH DEFAULT FOR BIT DATA For a row describing a field procedure, the parameter value block of the field procedure.

SYSIBM.SYSFOREIGNKEYS		**(DSNDB06.SYSTSFOR)**
SYSIBM.DSNDRH01	**D**	**(CREATOR,TBNAME,RELNAME)**
Contains information for each column of a foreign key. One row for every column of every foreign key.		

This table contains the following columns:

Name	Data Type/Description
CREATOR	VARCHAR(128) NOT NULL Schema or qualifier of the table that contains the column.
TBNAME	VARCHAR(128) NOT NULL Name of the table containing the column.

Name	Data Type/Description
RELNAME	VARCHAR(128) NOT NULL Name of the constraint for which the column is part of the foreign key.
COLNAME	VARCHAR(128) NOT NULL Name of the column.
COLNO	SMALLINT NOT NULL Numerical place of column in its table.
COLSEQ	SMALLINT NOT NULL Numerical place of the column in the foreign key.
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other.

SYSIBM.SYSINDEXCLEANUP		**(DSNDB06.SYSTSIXC)**
SYSIBM.DSNICX01	**D**	**(DBNAME,INDEXSPACE)**
Contains information to control time windows when pseudo deleted entries should be cleaned up.		

This table contains the following columns:

Name	Data Type/Description
DBNAME	VARCHAR(24) Database name for the index.
INDEXSPACE	VARCHAR(24) Index space name
ENABLE_DISABLE	CHAR(1) NOT NULL Index clean up enabled/disabled: E – ENABLED D – DISABLED
MONTH_WEEK	CHAR(1) NOT NULL Meaning of DAY column: M – Day of month W – Day of week
MONTH	SMALLINT Month number
DAY	SMALLINT Day of week where Monday=1
START_TIME	TIME TIME to clean up – NULL indicates all times
END_TIME	TIME Local time of when cleanup can happen – NULL if START_TIME has NULL.

SYSIBM.SYSINDEXCONTROL	**(DSNDB06.SYSTSICO)**
Contains rows that specify time windows to control memory allocated for an index.	

This table contains the following columns:

Name	Data Type/Description
SSID	CHAR(4) NOT NULL DB2 subsystem. For data sharing, if NULL the row applies to all members.
PARTITION	SMALLINT Partition number. If NULL all partitions apply.
IXNAME	VARCHAR(128) NOT NULL Index name.
IXCREATOR	VARCHAR(128) NOT NULL Index creator.
TYPE	CHAR(1) NOT NULL WITH DEFAULT 'F'

Name	Data Type/Description
	Purpose for usage. F : Fast Traversal
ACTION	CHAR(1) NOT NULL WITH DEFAULT 'A' The action being performed. • F: Force fast traversal. • D: Disable fast traversal. • A: Automatic fast traversal.
MONTH_WEEK	CHAR(1) NOT NULL Meaning of DAY column. M : Day of month. W : Day of week.
MONTH	SMALLINT Month when window applies (1-12 or null).
DAY	SMALLINT Day of month/week depending On MONTH_WEEK value. If M : day of month or null. If W : values 1-7. If NULL : valid every day of month or every day of week based on MONTH_WEEK.
FROM_TIME	TIME Time of day when window starts. If null then no limitation.
TO_TIME	TIME Time of day when window ends. If null then no limitation.

SYSIBM.SYSINDEXES		**(DSNDB06.SYSTSIXS)**
SYSIBM.DSNDXX01	**P**	**(CREATOR,NAME)**
SYSIBM.DSNDXX02	**D**	**(DBNAME,INDEXSPACE)**
SYSIBM.DSNDXX03	**U**	**(TBCREATOR,TBNAME,CREATOR,NAME)**
SYSIBM.DSNDXX04	**D**	**(INDEXTYPE)**
SYSIBM.DSNDXX07	**D**	**(TBCREATOR,TBNAME)**
Defines index attributes. Contains one row for every index.		

This table contains the following columns:

Name	Data Type/Description
NAME	VARCHAR(128) NOT NULL Name of the index.
CREATOR	VARCHAR(128) NOT NULL Schema of the index.
TBNAME	VARCHAR(128) NOT NULL Name of table on which the index is defined.
TBCREATOR	VARCHAR(128) NOT NULL Schema of the table.
UNIQUERULE	CHAR(1) NOT NULL Whether index is unique: • C – Yes. Enforces uniqueness of a UNIUE constraint or hash key columns. • D – No (duplicates are allowed). • U – Yes. • P – Yes. Primary index (unique). Enforces referential integrity. • N – Yes. Defined with UNIQUE WHERE NOT NULL. • R – Yes. Enforces uniqueness of a non-

Name	Data Type/Description
	primary parent key. • G – Yes. Enforce uniqueness of values in a column defined as ROWID GENERATED BY DEFAULT. • X – Yes. Enforces the uniqueness of values in a column that contains XML values.
COLCOUNT	SMALLINT NOT NULL Number of columns in the key.
CLUSTERING	CHAR(1) NOT NULL Whether CLUSTER was specified when index was created: Y/N.
CLUSTERED	CHAR(1) NOT NULL Whether table is actually clustered by index: • N – No. Less than a significant number of rows are in clustering order, or statistics have not been gathered. • Y – Yes. More than a significant number of rows are in clustering order • blank – not applicable For SPARSE INDEX: Statistics based on actual index content. Updatable column that can be changed by the RUNSTATS utility.
DBID	SMALLINT NOT NULL Internal ID of the database.
OBID	SMALLINT NOT NULL Internal ID of the index fan set descriptor.
ISOBID	SMALLINT NOT NULL Internal ID of the index page set descriptor.
DBNAME	VARCHAR(24) NOT NULL Name of the database that contains the index.
INDEXSPACE	VARCHAR(24) NOT NULL Name of the index space.
FIRSTKEYCARD	INTEGER/Unused.
FULLKEYCARD	INTEGER/Unused.
NLEAF	INTEGER NOT NULL No. of active leaf pages in the index. The value is –1 if statistics have not been gathered.
NLEVELS	SMALLINT NOT NULL No. of levels in the index tree. The value is –1 if statistics have not been gathered.
BPOOL	CHAR(8) NOT NULL Name of the buffer pool used for the index.
PGSIZE	SMALLINT NOT NULL Contains the value 4, 8, 16, or 32 which indicates the size, in KB, of the leaf pages in the index.
ERASERULE	CHAR(1) NOT NULL Whether data sets are erased when dropped: Y/N. The value is meaningless if the index is partitioned
DSETPASS	VARCHAR(24)/Unused.
CLOSERULE	CHAR(1) NOT NULL Whether data sets are candidates for closure when the limit on the number of

Name	Data Type/Description
	open data sets is reached: Y/N.
SPACE	INTEGER NOT NULL No. of kilobytes of DASD storage allocated to index (from STOSPACE). For partitioned index space, the value is total kilobytes of DASD storage allocated to all partitions in the storage group.
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other.
CLUSTERRATIO	SMALLINT NOT NULL WITH DEFAULT %, multiplied by 100, of rows that are in clustering order. For partitioned index, weighted average of all index partitions in terms of the number of rows in the partition: • 0 – statistics not run. • –2 – index is for auxiliary table. For sparse index based on actual index content.
CREATEDBY	VARCHAR(128) NOT NULL WITH DEFAULT Primary authorization ID of the user who created the index.
IOFACTOR	SMALLINT NOT NULL/Internal use only.
PREFETCHFACTOR	SMALLINT/Unused.
STATSTIME	TIMESTAMP NOT NULL WITH DEFAULT The date and time when RUNSTATS last updated the statistics. The default value is 0001-01-01.00.00.00.000000.
INDEXTYPE	CHAR(1) NOT NULL WITH DEFAULT The index type: • 2 – Type-2 index or hash overflow index on non-partitioned table • blank – Type-1 index • D – Data-partitioned secondary index • P – Partitioning index
FIRSTKEYCARDF	FLOAT NOT NULL WITH DEFAULT –1 Number of distinct values of the first key column. If RUNSTATS not run value is -1.
FULLKEYCARDF	FLOAT NOT NULL WITH DEFAULT -1 Number of distinct values in the full key. If statistics not run value is -1
CREATEDTS	TIMESTAMP NOT NULL WITH DEFAULT Time when the CREATE statement was executed for the index.
ALTEREDTS	TIMESTAMP NOT NULL WITH DEFAULT Time the most recent ALTER INDEX was issued for this index.
PIECESIZE	INTEGER NOT NULL WITH DEFAULT Maximum size of a data set in KB for non-partitioning indexes. A value of (0) indicates that the index is a partitioning index or the index was created before DB2 5.
COPY	CHAR(1) NOT NULL WITH DEFAULT 'N' Whether COPY YES was specified for index: • Y – Yes. Index can be copied. SYSIBM.SYSLGRNX recording is enabled. • N – No. Index cannot be copied. SYSIBM.SYSLGRNX recording not

Name	Data Type/Description
	enabled.
COPYLRSN	CHAR(10) NOT NULL WITH DEFAULT X'00000000000000000000' FOR BIT DATA Value is RBA or LRSN (LRSN data sharing only). If index is currently COPY YES value is RBA or LRSN when created or altered to COPY YES. If index is currently COPY NO from create value is X'000000000000.' Otherwise, if COPY NO from an alter, COPYLRSN value is unchanged.
CLUSTERRATIOF	FLOAT NOT NULL WITH DEFAULT Percentage of rows that are in clustering sequence when multiplied by 100. For partitioning indexes it contains a weighted average for all partitions in terms of number of rows in the partition. • Statistics not gathered. • –2 index is for auxiliary table.
SPACEF	FLOAT(8) NOT NULL WITH DEFAULT -1 Kilobytes of DASD storage. The value is -1 if statistics have not been gathered. This is an updatable column.
REMARKS	VARCHAR(762) NOT NULL WITH DEFAULT A character field string provided by the user with the COMMENT ON statement.
PADDED	CHAR(1) NOT NULL WITH DEFAULT Indicates if keys within the index are padded for varying-length column data: • Y – The index contains varying-length character or graphic data and is PADDED. • N – The index contains varying-length character or graphic data and is NOT PADDED. • blank – The index does not contain varying-length character or graphic data. Value blank if created or altered prior to V8.
VERSION	SMALLINT NOT NULL WITH DEFAULT Version of data row format for the index. Zero indicates that a version-creating alter has never occurred against the index.
OLDEST_VERSION	SMALLINT NOT NULL WITH DEFAULT Version number describing the oldest format of data in the index space and any image copies of the index.
CURRENT_ VERSION	SMALLINT NOT NULL WITH DEFAULT Version number for the newest formal of data in the index space. Zero indicates index space has never had versioning. When version number hits max value, number will wrap back to one.
RELCREATED	CHAR(1) NOT NULL WITH DEFAULT Release of DB2 that was used to create the object, blank for indexes created prior to V8.
AVGKEYLEN	INTEGER NOT NULL WITH DEFAULT –1 Average length of keys within the index. Value is –1 if statistics not gathered.

Name	Data Type/Description
KEYTARGET_ COUNT	SMALLINT NOT NULL WITH DEFAULT The number of key-targets for an extended index. The value is 0 for a simple index.
UNIQUE_COUNT	SMALLINT NOT NULL WITH DEFAULT The number of columns or key-targets that make up the unique constraint of the index when other non-constraint enforcing columns or key-targets exist. Else 0.
IX_EXTENSION_ TYPE	CHAR(1) NOT NULL WITH DEFAULT Identifies the type of extended index: • N (Node ID index) • S (Index on a scalar expression) • T (Spatial index) • V (XML index) • blank (Simple index)
COMPRESS	CHAR(1) NOT NULL WITH DEFAULT 'N' Indicates whether index compression is active: N (not active or Y (active).
OWNER	VARCHAR(128) NOT NULL WITH DEFAULT Authorization ID of the owner of the index, empty string for indexes created in a DB2 release prior to V9.
OWNERTYPE	CHAR(1) NOT NULL WITH DEFAULT Indicates the type of owner: blank Authorization ID L Role
DATAREPEAT FACTORF	FLOAT NOT NULL WITH DEFAULT -1 The anticipated number of data pages that will be touched following an index key order. This statistic is only collected when the STATCLUS subsystem parameter is set to ENHANCED. This number is -1 if statistics have not been collected. The valid value is -1 or any value that is equal to or greater than 1. This is an updatable column.
ENVID	INTEGER NOT NULL WITH DEFAULT Internal environment identifier.
ROWID	ROWID NOT NULL GENERATED ALWAYS ROWID for LOB columns
HASH	CHAR(1) NOT NULL WITH DEFAULT N Hash overflow index: N/Y
SPARSE	CHAR(1) NOT NULL WITH DEFAULT N Sparse index: N/Y/X (excluded). Excluded means the index will not have an entry for key values with NULL.
PARSETREE	BLOB(1G) NOT NULL WITH DEFAULT IBM Internal use only
RTSECTION	BLOB(1G) NOT NULL WITH DEFAULT IBM Internal use only
DSSIZE	INTEGER NOT NULL Max size in KB of partitioned index otherwise 0.
PAGENUM	CHAR(1) NOT NULL WITH DEFAULT 'A' Format of partitioned index page numbers. A : Absolute addressing. R : Relative addressing. Blank : Object created prior to DB2 12
PARTKEY COLUMN	SMALLINT NOT NULL WITH DEFAULT Reserved

Name	Data Type/Description
STATUS	VARCHAR(30) NOT NULL WITH DEFAULT Reserved
INDEXSTATUS	SMALLINT/Reserved.
PARTITIONS	VARCHAR(765) NOT NULL/Reserved.
PQTY	INTEGER For user-managed data sets, the value is the primary space allocation in units of 4 KB storage blocks or -1. PQTY is based on a value of PRIQTY in the appropriate CREATE or ALTER TABLESPACE statement. Unlike PQTY, however, PRIQTY accepts space in 1 KB units. A value of -1 indicates that one of the following cases is true: • *)PRIQTY was not specified for a CREATE TABLESPACE statement or for any subsequent ALTER TABLESPACE statements. • *) -1 was the most recently specified value for PRIQTY, either on the CREATE TABLESPACE statement or a subsequent ALTER TABLESPACE statement. This column contains the null value when object created prior to DB2 12.
STORTYPE	CHAR(1) Type of storage allocation: • E: Explicit (storage group not used). • I: Implicit (storage group used). • Null if created prior to DB2 12.
STORNAME	VARCHAR(128) Name of storage group. Null if created prior to DB2 12.
VCATNAME	VARCHAR(24) Name of ICF catalog. Null if created prior to DB2 12.
FREEPAGE	SMALLINT Number of pages loaded before a page is left free.
PCTFREE	SMALLINT Percentage of each page left as free.
GBPCACHE	CHAR(1) Group buffer pool cache option specified for this index or partition. • Blank—Only changed pages are cached in the group buffer pool. • A—Changed and unchanged pages are cached in the group buffer pool. • N—No data is cached in the group buffer pool. This column contains the null value when the value
SECQTYI	INTEGER Secondary space allocation in units of 4 KB storage. For user-managed data sets, the value is the secondary space allocation in units of 4 KB blocks. This column contains the null value when the value is unknown for objects created prior to DB2 12.

Name	Data Type/Description
ENFORCED_ CONS	CHAR(1) NOT NULL WITH DEFAULT Whether the index is an enforcing non-unique constraint. • Blank—The index does not enforce a non-unique constraint. • F—The index enforces a foreign key for a temporal referential constraint.
IMPLICIT	CHAR(1) NOT NULL WITH DEFAULT Whether the index was implicitly created: • Blank—N/A • N—The index was explicitly created. • Y—The index was implicitly created.
REGENERATEDTS	TIMESTAMP(12) NOT NULL Timestamp when object was regenerated.

SYSIBM.SYSINDEXES_HIST		**(DSNDB06.SYSHIST)**
SYSIBM.DSNHHX01	**D**	**(TBCREATOR,TBNAME,NAME, STATSTTIME)**
SYSIBM.DSNHHX02	**D**	**(CREATOR,NAME)**
Contains history from SYSINDEXES (populated by RUNSTATS). You can insert, update, and delete rows in this table.		

This table contains the following columns:

Name	Data Type/Description
NAME	VARCHAR(128) NOT NULL Name of the index.
CREATOR	VARCHAR(128) NOT NULL The schema of the index.
TBNAME	VARCHAR(128) NOT NULL Name of the table on which the index is defined.
TBCREATOR	VARCHAR(128) NOT NULL The schema of the table.
CLUSTERING	CHAR(1) NOT NULL Whether CLUSTER was specified when the index was created: N/Y.
NLEAF	INTEGER NOT NULL WITH DEFAULT -1 Number of active leaf pages in the index. The value is -1 if statistics have not been gathered.
NLEVELS	SMALLINT NOT NULL WITH DEFAULT -1 Number of levels in the index tree. If the index is partitioned, it is the maximum of the number of levels in the index tree for all the partitions. -1 if statistics have not been gathered.
STATSTIME	TIMESTAMP NOT NULL If RUNSTATS updated the statistics, the date and time when the last invocation of RUNSTATS updated the statistics. The default value is '0001-01-01.00.00.00.000000'.
FIRSTKEYCARDF	FLOAT(8) NOT NULL WITH DEFAULT -1 Number of distinct values of the first key column. This number is an estimate if updated while collecting statistics on a single partition. -1 if statistics have not been gathered.

Name	Data Type/Description
FULLKEYCARDF	FLOAT(8) NOT NULL WITH DEFAULT -1 Number of distinct values of the key. The value is -1 if statistics have not been gathered.
CLUSTERRATIOF	FLOAT(8) NOT NULL Percentage of rows that are in clustering order. For a partitioning index, it is the weighted average of all index partitions in terms of the number of rows in the partition. • 0 – If statistics have not been gathered. • -2 – If the index is for an auxiliary table. For sparse index : based on actual index content
SPACEF	FLOAT(8) NOT NULL WITH DEFAULT -1 Number of kilobytes of DASD storage allocated to the index space partition. -1 if statistics have not been gathered.
IBMREQD	CHAR(1) NOT NULL WITH DEFAULT 'N' Whether the row came from the basic MRM tape: Y/N/other.
AVGKEYLEN	INTEGER NOT NULL WITH DEFAULT –1 Average length of key within the index. –1 if statistics never gathered. For sparse index: based on actual index content
DATAREPEATFACTORF	FLOAT NOT NULL WITH DEFAULT -1 The anticipated number of data pages that will be touched following an index key order. This statistic is only collected when the STATCLUS subsystem parameter is set to ENHANCED. -1 if statistics have not been collected. The valid value is -1 or any value that is equal to or greater than 1. For sparse index : based on actual index content

SYSIBM.SYSINDEXES_RTSECT		**(DSNDB06.SYSTSIXR)**
SYSIBM.DSNDXX06	**U**	**(AUXID,AUXVER)**
Auxiliary table for RTSECTION LOB column of SYSINDEXES.		

This table contains the following column:

Column Name	Data Type	Description
AUXID	VARCHAR(17)	ID of auxiliary data.
AUXVER	SMALLINT	Version of auxiliary data.
AUXVALUE	BLOB(1G) NOT NULL WITH DEFAULT	IBM internal use

SYSIBM.SYSINDEXES_TREE		**(DSNDB06.SYSTSIXT)**
SYSIBM.DSNDXX05	**U**	**(AUXID,AUXVER)**
Auxiliary table for PARSETREE LOB column of SYSINDEXES.		

This table contains the following column:

Column Name	Data Type	Description
AUXID	VARCHAR(17)	ID of auxiliary data.

Column Name	Data Type	Description
AUXVER	SMALLINT	Version of auxiliary data.
AUXVALUE	BLOB(1G) NOT NULL WITH DEFAULT	IBM internal use

SYSINDEXPART		**(DSNDB06.SYSTSIPT)**
SYSIBM.DSNDRX01	**R**	**(IXCREATOR,IXNAME,PARTITION)**
SYSIBM.DSNDRX02	**D**	**(STORNAME)**
SYSIBM.DSNDRX03	**D**	**(IXCREATOR,IXNAME)**
Defines index partitions. Contains one row for each un-partitioned index and one row for each partition of a partitioned index.		

This table contains the following columns:

Name	Data Type/Description
PARTITION	SMALLINT NOT NULL Partition number. 0 if index not partitioned for both PQTY and SQTY.
IXNAME	VARCHAR(128) NOT NULL Name of the index.
IXCREATOR	VARCHAR(128) NOT NULL The schema of the index.

PQTY	INTEGER NOT NULL
	• For user-managed data sets, the value is the primary space allocation in units of 4KB storage blocks or –1.
	• For user-specified values of PRIQTY other than –1, value is set to primary space allocation only if RUNSTATS INDEX w/UPDATE(ALL) or (SPACE) is ran; otherwise value is zero.
	• –1 indicates that either of the following is true:
	  ○ PRIQTY was not specified for CREATE INDEX statement or subsequent ALTER INDEX.
	  ○ -1 was the most recently specified value for PRIQTY either on CREATE INDEX or subsequent ALTER INDEX.

SQTY	SMALLINT NOT NULL
	• For user-managed datasets, the values is the secondary space allocation in units of 4KB storage blocks or –1.
	• For user-specified values of SECQTY other than –1, the value is set to the secondary space allocation only in RUNSTATS INDEX with UPDATE (ALL) or (SPACE) is ran, otherwise zero.
	• –1 indicates that either of the following is true:
	  ○ SECQTY was not specified for CREATE INDEX statement or subsequent ALTER INDEX.
	  ○ -1 was the most recently specified value for PRIQTY either on CREATE INDEX or subsequent ALTER INDEX

STORTYPE	CHAR(1) NOT NULL Type of storage allocation: • E (Explicit). STORNAME is an Integrated

Name	Data Type/Description
	Catalog Facility (ICF) catalog • I (Implicit) STORNAME names a storage group
STORNAME	VARCHAR(128) NOT NULL Name of storage group or ICF catalog used for space allocation. Blank for catalog indexes.
VCATNAME	VARCHAR(24) NOT NULL Name of ICF catalog used for space allocation.
CARD	INTEGER/Unused.
FAROFFPOS	INTEGER/Unused.
LEAFDIST	INTEGER NOT NULL 100 times the avg. no. of leaf pages between successive active leaf pages of the index. –1 if statistics have not been gathered -2 if the index is a node ID index or an XML index.
NEAROFFPOS	INTEGER/Unused.
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other.
LIMITKEY	VARCHAR(512) NOT NULL WITH DEFAULT FOR BIT DATA The high value of the limit key of the partition in an internal format. 0 if index is not partitioned or for a data-partitioned secondary index.
FREEPAGE	SMALLINT NOT NULL No. of pages loaded before a page is left as free space.
PCTFREE	SMALLINT NOT NULL Percentage of each leaf or nonleaf page left as free space.
SPACE	INTEGER NOT NULL WITH DEFAULT Number of kilobytes of DASD storage allocated to the index space partition (from STOSPACE). • STOSPACE or RUNSTATS has not been run or the data set for the index was created during first insert operation, or when LOAD utility was run. • –1 if the index was defined with the DEFINE NO clause • A non-negative value: Data sets for index space defined with the underlying data sets allocated.
STATSTIME	TIMESTAMP NOT NULL WITH DEFAULT If RUNSTATS updated the statistics, the date and time when the last invocation of RUNSTATS updated the statistics. The default value is 0001-01-01.00.00.00.000000.
INDEXTYPE	CHAR(1)/Unused.
GBPCACHE	CHAR(1) NOT NULL WITH DEFAULT Group buffer pool cache option specified for this index or index partition: • Blank—Only changed pages are cached in group buffer pool

Name	Data Type/Description
	• A—Changed and unchanged pages are cached in group buffer pool. • N—No date cached in group buffer pool.
FAROFFPOSF	FLOAT NOT NULL WITH DEFAULT -1 No. of referenced rows far from optimum position because of inserts into a full page. Not applicable for index of auxiliary tables. –1 if statistics are not gathered. -2 if the index is a node ID index or an XML index.
NEAROFFPOSF	FLOAT NOT NULL WITH DEFAULT -1 No. of referenced rows near from optimum position because of inserts into a full page. Not applicable for index of auxiliary tables. –1 if statistics are not gathered -2 if the index is a node ID index or an XML index.
CARDF	FLOAT NOT NULL WITH DEFAULT -1 No. of RIDs in the index that refer to data rows or LOBs. -1 Statistics never generated. For sparse index : based on actual index content
SECQTYI	INTEGER NOT NULL WITH DEFAULT Secondary space allocation in units of 4K. 0 if storage group not used.
IPREFIX	CHAR(1) NOT NULL WITH DEFAULT 'I' First character of the instance qualifier for this index's data set name. I or J are valid values.
ALTEREDTS	TIMESTAMP NOT NULL WITH DEFAULT Time of most recent ALTER INDEX execution. Otherwise '0001-01-01.00.00.00.000000.'
SPACEF	FLOAT(8) NOT NULL WITH DEFAULT -1 Kilobytes of DASD storage. The value is -1 if statistics have not been gathered.
DSNUM	INTEGER NOT NULL WITH DEFAULT -1 Number of data sets. -1 if statistics have not been gathered.
EXTENTS	INTEGER NOT NULL WITH DEFAULT -1 Number of data set extents. -1 if statistics have not been gathered.
PSEUDO_DEL_ ENTRIES	INTEGER NOT NULL WITH DEFAULT -1 Number of pseudo-deleted entries (entries that are logically deleted but still physically present in the index). For a non-unique index, value is the number of RIDs that are pseudo deleted. For a unique index, the value is the number of keys and RIDs that are pseudo deleted. The value is -1 if statistics have not been gathered.
LEAFNEAR	INTEGER NOT NULL WITH DEFAULT –1 Number of leaf pages physically near previous leaf page for successive active leaf pages. -1 if statistics have not been gathered.
LEAFFAR	INTEGER NOT NULL WITH DEFAULT -1 Number of leaf pages located physically far away from previous leaf pages for

Name	Data Type/Description
	successive active leaf pages accessed in an index scan. -1 if statistics have not been gathered.
OLDEST_VERSION	SMALLINT NOT NULL WITH DEFAULT The version number describing the oldest format of data in the index part and any image copies of the index part
CREATEDTS	TIMESTAMP NOT NULL WITH DEFAULT –1 Time the partition was created.
AVGKEYLEN	INTEGER NOT NULL WITH DEFAULT –1 Average length of keys within the index. –1 if statistics not gathered.
RBA_FORMAT	CHAR(1) NOT NULL WITH DEFAULT Format of RBA/LRSN: • B – basic 6-byte • E – extended 10-byte • U – undefined – DEFINE NO specified • blank – for migrated objects.
DSSIZE	INTEGER NOT NULL WITH DEFAULT Maximum size in KB of a partitioned index data set. 0 for a nonpartitioned index. NULL for indexes that were created beforeDB2 12.
PAGENUM	CHAR(1) NOT NULL WITH DEFAULT 'A' Format of pages for the index, indicating absolute or relative page numbering. • A—Absolute addressing so that PAGENUM contains the embedded partition number. • R—Relative addressing so that PAGENUM contains only the relative page number. NULL for objects created before DB2 12.
LIMITKEY_ EXTERNAL	VARCHAR(765)/Reserved.

SYSIBM.SYSINDEXPART_HIST		**(DSNDB06.SYSHIST)**
SYSIBM.DSNHGX01	**D**	**(IXCREATOR,IXNAME, PARTITION,STATSTIME)**
Contains index partitioning history of SYSINDEXPART (populated by RUNSTATS). You can insert, update, and delete rows in this table.		

This table contains the following columns:

Name	Data Type/Description
PARTITION	SMALLINT NOT NULL Partition number. Zero if index is not partitioned.
IXNAME	VARCHAR(128) NOT NULL Name of the index.
IXCREATOR	VARCHAR(128) NOT NULL The schema of the index.
PQTY	INTEGER NOT NULL • For user-managed data sets, the value is the primary space allocation in units of 4KB storage bocks or –1. • For user-specified values of PRIQTY other than –1, value is set to primary space

Name	Data Type/Description
	allocation only if RUNSTATS INDEX w/UPDATE(ALL) or (SPACE) is ran; otherwise value is zero. • A value of –1 indicates that either of the following is true: ○ PRIQTY was not specified for CREATE INDEX statement or subsequent ALTER INDEX. ○ -1 was the most recently specified value for PRIQTY either on CREATE INDEX or subsequent ALTER INDEX.
SECQTYI	INTEGER NOT NULL • For user-managed datasets, the values is the secondary space allocation in units of 4KB storage blocks or –1. • For user-specified values of SECQTY other than –1, the value is set to the secondary space allocation only in RUNSTATS INDEX with UPDATE (ALL) or (SPACE) is ran, otherwise zero. • A value of –1 indicates that either of the following is true: ○ SECQTY was not specified for CREATE INDEX statement or subsequent ALTER INDEX. ○ -1 was the most recently specified value for PRIQTY either on CREATE INDEX or subsequent ALTER INDEX
LEAFDIST	INTEGER NOT NULL WITH DEFAULT -1 100 times the average number of leaf pages between successive active leaf pages of the index. -1 if statistics have not been gathered.
SPACEF	FLOAT(8) NOT NULL WITH DEFAULT -1 Number of kilobytes of DASD storage allocated to the index space partition. -1 if statistics have not been gathered.
STATSTIME	TIMESTAMP NOT NULL If RUNSTATS updated the statistics, the date and time when the last invocation of RUNSTATS updated the statistics. The default value is '0001-01-01.00.00.00.000000'.
FAROFFPOSF	FLOAT(8) NOT NULL WITH DEFAULT -1 Number of referred to rows far from optimal position because of an insert into a full page. -1 if statistics have not been gathered. The column is not applicable for an index on an auxiliary table.
NEAROFFPOSF	FLOAT(8) NOT NULL WITH DEFAULT -1 Number of referred to rows near, but not at optimal position because of an insert into a full page. -1 if statistics have not been gathered. Not applicable for an index on an auxiliary table.
CARDF	FLOAT(8) NOT NULL WITH DEFAULT -1 Number of keys in the index that refer to data rows or LOBs. -1 if statistics have not been gathered.
EXTENTS	INTEGER NOT NULL WITH DEFAULT -1 Number of data set extents. -1 if statistics have not been gathered.

Name	Data Type/Description
PSEUDO_DEL_ ENTRIES	INTEGER NOT NULL WITH DEFAULT –1 Number of pseudo deleted entries. -1 if statistics have not been gathered.
DSNUM	INTEGER NOT NULL WITH DEFAULT -1 Data set number within the table space. For partitioned index spaces, this value corresponds to the partition number for a single partition copy, or 0 for a copy of an entire partitioned index space. The value is -1 if statistics have not been gathered.
IBMREQD	CHAR(1) NOT NULL WITH DEFAULT 'N' Whether the row came from the basic MRM tape: Y/N/other.
LEAFNEAR	INTEGER NOT NULL WITH DEFAULT -1 Number of leaf pages physically near previous leaf page for successive active leaf pages. -1 if statistics have not been gathered.
LEAFFAR	INTEGER NOT NULL WITH DEFAULT -1 Number of leaf pages located physically far away from previous leaf pages for successive (active leaf) pages accessed in an index scan. -1 if statistics have not been gathered.
AVGKEYLEN	INTEGER NOT NULL WITH DEFAULT –1 Average length of keys within the index. -1 if statistics have not been gathered.

SYSIBM.SYSINDEXSPACESTATS		**(DSNDB06.SYSTSISS)**
SYSIBM.DSNRTX02	**U**	**(DBID,ISOBID,PARTITION, INSTANCE)**
SYSIBM.DSNRTX03	**D**	**(CREATOR,NAME)**
Contains real-time statistics for index spaces. You can insert, update, and delete rows in this table.		

SYSIBM.SYSIXSPACESTATS_H	**(DSNDB06.SYSTSTSH)**
History table for SYSINDEXSPACESTATS for the business period temporal design.	

This table contains the following columns:

Name	Data Type/Description
UPDATESTATSTIME	TIMESTAMP NOT NULL WITH DEFAULT The timestamp that the row in the SYSINDEXSPACESTATS table is inserted or last updated.
NLEVELS	SMALLINT The number of levels in the index tree. A null value indicates that the number of levels is unknown.
NPAGES	INTEGER The number of pages that contain ony pseudo-deleted index entries. This is an updatable column.
NLEAF	INTEGER The number of leaf pages in the index. This is an updatable column.
NACTIVE	INTEGER The number of active pages in the index space or partition. This value is

Name	Data Type/Description
	equivalent to the number of pre-formatted pages. A null value indicates that the number of active pages is unknown.
SPACE	INTEGER The amount of space, in KB, that is allocated to the index space or partition. For multi-piece, linear page sets, this value is the amount of space in all data sets. A null value indicates the amount of space is unknown.
EXTENTS	SMALLINT The number of extents in the index space or partition. For multi-piece index spaces, this value is the number of extents for the last data sets. For a data set that is stripped across multiple volumes, the value is the number of logical extents. A null value indicates the number of extents is unknown.
LOADRLASTTIME	TIMESTAMP The timestamp that the LOAD REPLACE utility was last run on the index space or partition. A null value indicates that the LOAD REPLACE utility has never been run on the index space or partition or that the timestamp is unknown.
REBUILDLASTTIME	TIMESTAMP The timestamp that the REBUILD INDEX utility was last run on the index space or partition. A null value indicates that the timestamp that the REBUILD INDEX was last run is unknown.
REORGLASTTIME	TIMESTAMP The timestamp that the REORG INDEX utility was last run on the index space or partition, or create time if REORG INDEX never run.. A null value indicates that the timestamp is unknown.
REORGINSERTS	INTEGER The number of index entries that have been inserted into the index space or partition since the last time the REORG, REBUILD INDEX, or LOAD REPLACE utilities were run, or since object was created. A null value indicates that the number of inserted index entries is unknown.
REORGDELETES	INTEGER The number of index entries that have been deleted from the index space or partition since the last time the REORG, REBUILD INDEX, or LOAD REPLACE utilities were run, or since object was created. A null value indicates that the number of deleted index entries is unknown.

Name	Data Type/Description
REORGAPPENDINSERT	INTEGER The number of index entries that have a key value that is greater than the maximum key value in the index or partition that have been inserted into the index space or partition since the last time the REORG, REBUILD INDEX, or LOAD REPLACE utilities were run, or since object was created. A null value indicates that the number of inserted index entries is unknown.
REORGPSEUDODELETES	INTEGER The number of index entries that have been pseudo-deleted since the last REORG, REBUILD INDEX, or LOAD REPLACE on the index space or partition, or since object was created. A pseudo-delete is a RID entry that has been marked as deleted. A null value indicates that the number of pseudo-deleted index entries is unknown.
REORGMASSDELETE	INTEGER The number of mass deletes from a segmented or LOB table space, or the number of dropped tables from a segmented table space since the last time the REORG or LOAD REPLACE utilities were run, or since object was created. A null value indicates that the number of mass deletes is unknown.
REORGLEAFNEAR	INTEGER The net number of leaf pages located physically near previous pages for successive active leaf pages that occurred since the last REORG, REBUILD INDEX, or LOAD REPLACE, or since object was created. The distance between leaf pages is optimal if the difference is 1 and considered near if the distance is 2-16. A null value means that the value is unknown. A negative value is possible in some cases.
REORGLEAFFAR	INTEGER The net number of leaf pages located physically far away from previous leaf pages for successive active leaf pages that occurred since the last REORG, REBUILD INDEX, or LOAD REPLACE, or since object was created. The distance between leaf pages is optimal if the difference is 1 and considered far if the distance is greater than 16. A null value means that the value is unknown.

Name	Data Type/Description
REORGNUMLEVELS	INTEGER The number of levels in the index tree that were added or removed since the last REORG, REBUILD INDEX, or LOAD REPLACE, or since object was created. A null value means that the number of added or deleted levels is unknown.
STATSLASTTIME	TIMESTAMP The timestamp of the last time that the RUNSTATS utility is run on the table space or partition. A null value means that RUNSTATS has never been run on the index space or partition, or that the timestamp of the last RUNSTATS is unknown.
STATSINSERTS	INTEGER The number of records or LOBs that have been inserted into the table space or partition since the last time that the RUNSTATS utility was run, or since object was created. A null value indicates that the number of inserted records or LOBs is unknown.
STATSDELETES	INTEGER The number of index entries that have been deleted since the last RUNSTATS on the index space or partition, or since object was created. A null value means that the number of deleted index entries is unknown.
STATSMASSDELETE	INTEGER The number of times that the index or index space partition was mass deleted since the last RUNSTATS, or since object was created. A null value indicates that the number of mass deletes is unknown.
COPYLASTTIME	TIMESTAMP The timestamp of the last full image copy on the index space or partition. A null value means that COPY has never been run on the index space or partition, or that the timestamp of the last full image copy is unknown.
COPYUPDATEDPAGES	INTEGER The number of distinct types that have been updated since the last time that the COPY utility was run, or since object was created. A null value indicates that the number of updated pages is unknown.
COPYCHANGES	INTEGER The number of insert, update, and delete operations since the last time that the COPY utility was run, or since object was created. A null value indicates that the number of insert, update, and delete operations is unknown.

Name	Data Type/Description
COPYUPDATELRSN	CHAR(6) The LRSN or RBA of the first update that occurs after the last time the COPY utility was run. A null value indicates that the LRSN or RBA is unknown. NULL : LRSN/RBA unknown.
COPYUPDATETIME	TIMESTAMP The timestamp of the first update that occurs after the last time that the COPY utility was run. A null value indicates that the timestamp is unknown.
LASTUSED	DATE Date when index was used for SELECT, FETCH, searched UPDATE, searched DELETE, or RI.
IBMREQD	CHAR(1) NOT NULL A value of Y indicates that the row came from the basic machine-readable material (MRM) tape.
DBID	SMALLINT NOT NULL The internal identifier of the database
ISOBID	SMALLINT NOT NULL The internal identifier of the index space page set descriptor.
PSID	SMALLINT NOT NULL . The internal identifier of the table space page set descriptor for the table space that is associated with the index
PARTITION	SMALLINT NOT NULL The data set number within the index space. For partitioned index spaces, this value corresponds to the partition number for a single partition. For non-partitioned table spaces, this value is 0
INSTANCE	SMALLINT NOT NULL Indicates if the object is associated with data set 1 or 2. This is an updatable column.
TOTALENTRIES	BIGINT The number of entries, including duplicate entries, in the index space or partition. A null value indicates that the number of entries is unknown.
DBNAME	VARCHAR(24) NOT NULL The name of the database..
NAME	VARCHAR(128) NOT NULL The name of the index
CREATOR	VARCHAR(128) NOT NULL The schema of the index
INDEXSPACE	VARCHAR(24) NOT NULL The name of the index space
REORGINDEXACCESS	BIGINT Number of times index was used for SELECT, FETCH, searched UPDATE, searched DELETE, or RI, or since object creation. For hash overflow indexes, number of times DB2 used hash overflow index. NULL if unknown.

Name	Data Type/Description
DRIVETYPE	CHAR(3) NOT NULL WITH DEFAULT Drive type containing index or index partition: • HDD Hard Disk Drive • SDD Solid Disk Drive (if any volume of a multi-volume data set is SDD) For multi-piece linear page sets, drive type of first data set.
STATS101	BIGINT/Reserved for future IBM use.
GETPAGES	BIGINT Number of getpages since last reorg or since object created.
SYS_START	TIMESTAMP(12) NOT NULL System period temporal start time for transaction
SYS_END	TIMESTAMP(12) NOT NULL System period temporal end time for transaction
TRANS_START	TIMESTAMP(12) System period transaction timestamp.

SYSIBM.SYSINDEXSTATS		**(DSNDB06.SYSSTATS)**
SYSIBM.DSNTXX01	**C**	**(OWNER,NAME,PARTITION)**
Contains statistical information for each partition of a partitioned index. One row for each partition (populated by RUNSTATS). You can insert, update, and delete rows in this table.		

This table contains the following columns:

Name	Data Type/Description
FIRSTKEYCARD	INTEGER NOT NULL For the index partition, no. of distinct values of the first key column.
FULLKEYCARD	INTEGER NOT NULL For the index partition, no. of distinct values of the key.
NLEAF	INTEGER NOT NULL No. of active leaf pages in index partition.
NLEVELS	SMALLINT NOT NULL No. of levels in the partition index tree.
IOFACTOR	SMALLINT NOT NULL/Unused.
PREFETCHFACTOR	SMALLINT NOT NULL/Unused.
CLUSTERRATIO	SMALLINT NOT NULL %, multiplied by 100, of rows that are in clustering order for the index partition. Value is 0 if statistics have not been gathered.
STATSTIME	TIMESTAMP NOT NULL The date and time when RUNSTATS last updated the statistics.
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other.
PARTITION	SMALLINT NOT NULL Partition number of the index.
OWNER	VARCHAR(128) NOT NULL

Name	Data Type/Description
	Schema of the index.
NAME	VARCHAR(128) NOT NULL Name of the index.
KEYCOUNT	INTEGER NOT NULL Number of rows in the partition.
FIRSTKEYCARDF	FLOAT NOT NULL WITH DEFAULT -1 No. of distinct values of the first key column for the index partition.
FULLKEYCARDF	FLOAT NOT NULL WITH DEFAULT -1 No. of distinct values of the key for the index partition.
KEYCOUNTF	FLOAT NOT NULL WITH DEFAULT -1 Number of rows in the partition.
CLUSTERRATIOF	FLOAT NOT NULL WITH DEFAULT Percentage of rows that are in clustering sequence, when multiplied by 100, for this index partition. 0 if stats are not gathered.
FULLKEYCARDDATA	VARCHAR (1000) NOT NULL WITH DEFAULT FOR BIT DATA/Internal use only.
DATAREPEATFACTORF	FLOAT NOT NULL WITH DEFAULT -1 The anticipated number of data pages that will be touched following an index key order. This statistic is only collected when the STATCLUS subsystem parameter is set to ENHANCED. -1 if statistics have not been collected. The valid value is -1 or any value that is equal to or greater than 1.

SYSIBM.SYSINDEXSTATS_HIST		**(DSNDB06.SYSHIST)**
SYSIBM.DSNHIX01	**D**	**(OWNER,NAME,PARTITION, STATSTIME)**
Contains history of SYSINDEXSTATS, index statistics information. You can insert, update, and delete rows in this table.		

This table contains the following columns:

Name	Data Type/Description
NLEAF	INTEGER NOT NULL WITH DEFAULT -1 Number of active leaf pages in the index partition. The value is -1 if statistics have not been gathered.
NLEVELS	SMALLINT NOT NULL WITH DEFAULT -1 Number of levels in the partition index tree. The value is -1 if statistics have not been gathered.
STATSTIME	TIMESTAMP NOT NULL If RUNSTATS updated the statistics, the date and time when the last invocation of RUNSTATS updated the statistics. The default value is '0001-01-01.00.00.00.000000'.
PARTITION	SMALLINT NOT NULL Partition number of the index.
OWNER	VARCHAR(128) NOT NULL The schema of the index.
NAME	VARCHAR(128) NOT NULL Name of the index.

Name	Data Type/Description
FIRSTKEYCARDF	FLOAT(8) NOT NULL WITH DEFAULT -1 For the index partition, number of distinct values of the first key column. The value is -1 if statistics have not been gathered.
FULLKEYCARDF	FLOAT(8) NOT NULL WITH DEFAULT -1 For the index partition, number of distinct values of the key. The value is -1 if statistics have not been gathered.
KEYCOUNTF	FLOAT(8) NOT NULL WITH DEFAULT -1 Total number of RID'sin the partition. The value is -1 if statistics have not been gathered.
CLUSTERRATIOF	FLOAT(8) NOT NULL For the index partition, the value, when multiplied by 100, is the percentage of rows that are in clustering order. The value is 0 if statistics have not been gathered.
IBMREQD	CHAR(1) NOT NULL WITH DEFAULT 'N' Whether the row came from the basic MRM tape: Y/N/other.
DATAREPEATFACTORF	FLOAT NOT NULL WITH DEFAULT -1 The anticipated number of data pages that will be touched following an index key order. This statistic is only collected when the STATCLUS subsystem parameter is set to ENHANCED. -1 if statistics have not been collected. The valid value is -1 or any value that is equal to or greater than 1.

SYSIBM.SYSJARCLASS_SOURCE		**(DSNDB06.SYSJAUXB)**
SYSIBM.DSNJSX01	**U**	**(AUXID,AUXVER)**
Auxiliary table for the CLASS_SOURCE column of SYSIBM.SYSJARCONTENTS.		

This table contains the following column:

Column Name	Data Type	Description
AUXID	VARCHAR(17)	ID of auxiliary data.
AUXVER	SMALLINT	Version of auxiliary data.
AUXVALUE	BLOB(10M) NOT NULL	Contents of the class in the JAR file.

SYSIBM.SYSJARCONTENTS		**(DSNDB06.SYSJAVA)**
SYSIBM.DSNJCX01	**D**	**(JARSCHEMA,JAR_ID)**
Contains the Java class source for the installed JAR files. You can insert, update, and delete rows in this table.		

This table contains the following columns:

Name	Data Type/Description
JARSCHEMA	VARCHAR(128) NOT NULL The schema of the JAR file.
JAR_ID	VARCHAR(128) NOT NULL The name of the JAR file.
CLASS	VARCHAR(384) NOT NULL The class name contained in the JAR file.
CLASS_SOURCE_ROWID	ROWID NOT NULL GENERATED ALWAYS ID used to support CLOB data type.

Name	Data Type/Description
CLASS_SOURCE	CLOB(10M) NOT NULL The contents of the class in the JAR file.
IBMREQD	CHAR(1) NOT NULL WITH DEFAULT 'N' Whether the row came from the basic MRM tape: Y/N/other.

SYSIBM.SYSJARDATA		**(DSNDB06.SYSJAUXA)**
SYSIBM.DSNJDX01	**U**	**(AUXID,AUXVER)**
Auxiliary table for JAR_DATA LOB column of SYSIBM.SYSJAROBJECTS.		

This table contains the following column:

Column Name	Data Type	Description
AUXID	VARCHAR(17)	ID of auxiliary data.
AUXVER	SMALLINT	Version of auxiliary data.

SYSIBM.SYSJAROBJECTS		**(DSNDB06.SYSJAVA)**
SYSIBM.DSNJOX01	**P**	**(JARSCHEMA,JAR_ID)**
Contains binary large object representing the installed JAR files.		

This table contains the following columns:

Name	Data Type/Description
JARSCHEMA	VARCHAR(128) NOT NULL The schema of the JAR file.
JAR_ID	VARCHAR(128) NOT NULL The name of the JAR file.
OWNER	VARCHAR(128) NOT NULL Authorization ID of the owner of the JAR object.
JAR_DATA_ROWID	ROWID NOT NULL GENERATED ALWAYS ID used to support BLOB data type.
JAR_DATA	BLOB(100M) NOT NULL The contents of the JAR file.
PATH	VARCHAR(2048) NOT NULL The URL path of the source JAR file.
CREATEDTS	TIMESTAMP NOT NULL Time when the JAR object was created.
ALTEREDTS	TIMESTAMP NOT NULL Time when the JAR object was altered.
IBMREQD	CHAR(1) NOT NULL WITH DEFAULT 'N' Whether the row came from the basic MRM tape: Y/N/other.
OWNERTYPE	CHAR(1) NOT NULL WITH DEFAULT Indicates the type of owner: Blank Authorization ID L Role

SYSIBM.SYSJAVAOPTS		**(DSNDB06.SYSJAVA)**
SYSIBM.DSNJVX01	**U**	**(JARSCHEMA,JAR_ID)**
Contains build options used during INSTALL_JAR.		

This table contains the following columns:

Name	Data Type/Description
JARSCHEMA	VARCHAR(128) NOT NULL

Name	Data Type/Description
	The schema of the JAR file.
JAR_ID	VARCHAR(128) NOT NULL The name of the JAR file.
BUILDSCHEMA	VARCHAR(128) NOT NULL Schema name for BUILDNAME.
BUILDNAME	VARCHAR(128) NOT NULL Procedure used to create the routine.
BUILDOWNER	VARCHAR(128) NOT NULL Authorization ID used to create the routine.
DBRMLIB	VARCHAR(256) NOT NULL PDS name where DBRM is located.
HPJCOMPILE_OPTS	VARCHAR(512) NOT NULL HPJ compile options used to install the routine.
BIND_OPTS	VARCHAR(2048) NOT NULL Bind options used to install the routine.
POBJECT_LIB	VARCHAR(256) NOT NULL PDSE name where program object is located.
IBMREQD	CHAR(1) NOT NULL WITH DEFAULT 'N' Whether the row came from the basic MRM tape: Y/N/other.

SYSIBM.SYSJAVAPATHS		**(DSNDB06.SYSJAVA)**
SYSIBM.DSNJPX01	**U**	**(JARSCHEMA,JAR_ID, ORDINAL)**
SYSIBM.DSNJPX02	**D**	**(PE_JARSCHEMA,PE_JAR_ID)**
Contains the complete resolution path and dependencies for a JAR file.		

This table contains the following columns:

Name	Data Type/Description
JARSCHEMA	VARCHAR(128) NOT NULL The schema of the JAR file.
JAR_ID	VARCHAR(128) NOT NULL The name of the JAR file.
OWNER	VARCHAR(128) NOT NULL Authorization ID of the owner of the JAR object.
ORDINAL	SMALLINT NOT NULL The ordinal number of the path element within the JAR's Java path.
PE_CLASS_PATTERN	VARCHAR(2048) NOT NULL The pattern for the names of the classes that are to be searched for in this path element's JAR file.
PE_JARSCHEMA	VARCHAR(128) NOT NULL The schema of this path element's JAR file.
PE_JAR_ID	VARCHAR(128) NOT NULL The name of this path element's JAR file.
IBMREQD	CHAR(1) NOT NULL A value of Y indicates that the row came from the basic machine-readable material (MRM) tape.

SYSIBM.SYSKEYCOLUSE		(DSNDB06.SYSTSKYC)
SYSIBM.DSNCUX01	D	(TBCREATOR,TBNAME, CONSTNAME,COLSEQ)
Contains information for every column in a unique constraint (primary or unique key) from the SYSIBM.SYSTABCONST table. One row per column.		

This table contains the following columns:

Name	Data Type/Description
CONSTNAME	VARCHAR(128) NOT NULL Name of the constraint.
TBCREATOR	VARCHAR(128) NOT NULL Schema or qualifier of the owner of the table on which the constraint is defined.
TBNAME	VARCHAR(128) NOT NULL Name of the table on which the constraint is defined.
COLNAME	VARCHAR(128) NOT NULL Name of the column.
COLSEQ	SMALLINT NOT NULL Numeric position of the column in the key (the first position in the key is 1).
COLNO	SMALLINT NOT NULL Numeric position of the column in the table on which the constraint is defined.
IBMREQD	CHAR(1) NOT NULL WITH DEFAULT 'N' Whether the row came from the basic MRM tape: Y/N/other.
PERIOD	CHAR(1) NOT NULL WITH DEFAULT Start or end column for BUSINESS_TIME period: • B—Start • C—End with EXCLUSIVE • I—End with INCLUSIVE • blank—Mot applicable

SYSIBM.SYSKEYS		(DSNDB06.SYSTSKEY)
SYSIBM.DSNDKX01	U	(IXCREATOR,IXNAME,COLNAME)
SYSIBM.DSNDKX02	D	(IXCREATOR,IXNAME)
SYSIBM.DSNDKX03	U	(IXCREATOR,IXNAME,COLSEQ)
Defines which columns (keys) are in an index. One row per column.		

This table contains the following columns:

Name	Data Type/Description
IXNAME	VARCHAR(128) NOT NULL Name of the index.
IXCREATOR	VARCHAR(128) NOT NULL Schema or qualifier of the index.
COLNAME	VARCHAR(128) NOT NULL Name of the column of the key.
COLNO	SMALLINT NOT NULL Numeric position of column in the table.
COLSEQ	SMALLINT NOT NULL Numeric position of column in the key. Value is meaningless for an index that is based on expressions.
ORDERING	CHAR(1) NOT NULL Order of the column in the key:

Name	Data Type/Description
	• blank—Index is based on expressions or uses INCLUDE clause • A—Ascending • D—Descending • R—Random Order
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other.
PERIOD	CHAR(1) NOT NULL WITH DEFAULT Start or end column for BUSINESS_TIME period: • B—Start • C—End with EXCLUSIVE • I—End with INCLUSIVE • blank—Mot applicable
CREATEDTS	TIMESTAMP(12) NOT NULL WITH DEFAULT Time when the CREATE statement was executed for the index.

SYSIBM.SYSKEYTARGETS		**(DSNDB06.SYSTARG)**
SYSIBM.DSNRKX01	**P**	**(IXSCHEMA,IXNAME,KEYSEQ)**
SYSIBM.DSNRKX02	**D**	**(DATATYPEID, KEYSPEC_INTERNAL)**
Defines which key-target that are participating in an extended index definition. One row per participating key-target.		

This table contains the following columns:

Name	Data Type/Description
IXNAME	VARCHAR(128) NOT NULL Name of the index.
IXSCHEMA	VARCHAR(128) NOT NULL Qualifier of the index.
KEYSEQ	SMALLINT NOT NULL Numeric position of the key-target in the index.
COLNO	SMALLINT NOT NULL Numeric position of the column in the table if the expression is a single column. Otherwise the value is 0. For XML indexes, this field is also 0.
ORDERING	CHAR(1) NOT NULL Order of the key: A Ascending
TYPESCHEMA	VARCHAR(128) NOT NULL Schema of the data type.
TYPENAME	VARCHAR(128) NOT NULL Name of the data type.
DATATYPEID	INTEGER NOT NULL The internal ID of the data type.
SOURCETYPEID	INTEGER NOT NULL For a built-in data type, this column contains 0. For a distinct type, this column contains the internal ID of the built-in type on which the distinct type is based.
LENGTH	SMALLINT NOT NULL The length attribute of the key-target or its precision for a decimal key-target. Number does not include the internal prefixes that are used to record the actual length and null

Name	Data Type/Description
	states, when applicable. Listed by data type, then value of the LENGTH column. • INTEGER (4) • SMALLINT (2) • FLOAT (4 or 8) • CHAR (Length of the string) • VARCHAR (maximum length of the string) • DECIMAL (precision of the number) • GRAPHIC (number of DBCS characters) • VARGRAPHIC (maximum number of DBCS characters) • DATE (4) • TIME (3) • TIMESTAMP WITHOUT TIME ZONE—Integral part of ((p+1)/2+7) • TIMESTAMP WITH TIME ZONE—Integral part of ((p+1)/2+9) • BIGINT—8 • BINARY—Length of the string • VARBINARY—Maximum length of the string. • DECFLOAT—8 or 16
LENGTH2	INTEGER NOT NULL The maximum length of the data that is retrieved from the column. Possible values include the following values: • Not a ROWID column • 40—For a ROWID column, the length of the value that is returned
SCALE	SMALLINT NOT NULL The scale of decimal data, or number of fractional second digits for timestamp. SCALE contains 0 if the key is not a decimal key.
NULLS	CHAR(1) NOT NULL Whether the key can contain null values: N or Y. Y also indicates that the index is an XML index.
CCSID	INTEGER NOT NULL The CCSID of the key. CCSID contains 0 if the key is a non-character type key.
SUBTYPE	CHAR(1) NOT NULL SUBTYPE applies to character keys only and indicated the subtype of the data: • B (BIT data) • M (MIXED data) • S (SBCS data) • blank—non-character data
KEYSPEC_INTERNAL	VARCHAR(512) NOT NULL FOR BIT DATA Internal use.
CREATEDTS	TIMESTAMP NOT NULL The timestamp for when the key-target is created.
RELCREATED	CHAR(1) NOT NULL The release of DB2® in which the key-target is created.
IBMREQD	CHAR(1) NOT NULL

Name	Data Type/Description
	A value of Y indicates that the row came from the basic machine-readable material (MRM) tape.
DERIVED_FROM	VARCHAR(4000) NOT NULL For an index on a scalar expression, DERIVED_FROM contains the text of the scalar expression that is used to generate the key-target value. For an XML index, this is the XML pattern that is used to generate the key-target value. Otherwise DERIVED_FROM contains an empty string.
STATSTIME	TIMESTAMP NOT NULL WITH DEFAULT The timestamp of the most recent RUNSTATS. The default value is '0001-01-01.00.00.00.000000'. STATSTIME is an updatable column.
CARDF	FLOAT NOT NULL WITH DEFAULT -1 The number of distinct values for the key-target.
HIGH2KEY	VARCHAR(2000) NOT NULL WITH DEFAULT FOR BIT DATA The second highest key-value.
LOW2KEY	VARCHAR(2000) NOT NULL WITH DEFAULT FOR BIT DATA The second lowest key-value.
STATS_FORMAT	CHAR(1) NOT NULL WITH DEFAULT The type of statistics that are gathered: • N—VARCHAR column statistical values are not padded. • blank—Statistics have not been collected or VARCHAR column statistical values are padded.

SYSIBM.SYSKEYTARGETSTATS		**(DSNDB06.SYSSTATS)**
SYSIBM.DSNTKX01	**U**	**(IXSCHEMA,IXNAME, KEYSEQ,PARTITION)**
Contains partition statistics for selected key-targets. For each key-target, a row exists for each partition in the table (populated by RUNSTATS). You can insert, update, and delete rows in this table.		

This table contains the following columns:

Name	Data Type/Description
IXSCHEMA	VARCHAR(128) NOT NULL The qualifier of the index.
IXNAME	VARCHAR(128) NOT NULL The name of the index.
KEYSEQ	SMALLINT NOT NULL Numeric position of the key-target in the index.
HIGHKEY	VARCHAR(2000) NOT NULL WITH DEFAULT FOR BIT DATA The highest key value.
HIGH2KEY	VARCHAR(2000) NOT NULL WITH DEFAULT FOR BIT DATA The second highest key-value.
LOWKEY	VARCHAR(2000) NOT NULL WITH DEFAULT FOR BIT DATA The lowest key value.

Name	Data Type/Description
LOW2KEY	VARCHAR(2000) NOT NULL WITH DEFAULT FOR BIT DATA The second lowest key-value.
PARTITION	SMALLINT NOT NULL The partition number of the table space.
KEYCARDDATA	VARCHAR(1000) NOT NULL WITH DEFAULT FOR BIT DATA Internal use only.
STATSTIME	TIMESTAMP NOT NULL WITH DEFAULT The timestamp of the most recent RUNSTATS. The default value is '0001-01-01.00.00.00.000000'.
IBMREQD	CHAR(1) NOT NULL A value of Y indicates that the row came from the basic machine-readable material (MRM) tape.
STATS_FORMAT	CHAR(1) NOT NULL WITH DEFAULT The type of statistics that are gathered: • N—VARCHAR column statistical values are not padded. • blank—Statistics have not been collected or VARCHAR column statistical values are padded.
CARDF	FLOAT NOT NULL WITH DEFAULT -1 Number of distinct values for the key target.

SYSIBM.SYSKEYTARGETS_HIST		**(DSNDB06.SYSHIST)**
SYSIBM.DSNHKX01	**D**	**(IXSCHEMA,IXNAME, KEYSEQ,STATSTIME)**
Contains history from the SYSKEYTARGETS table. You can insert, update, and delete rows in this table.		

This table contains the following columns:

Name	Data Type/Description
IXNAME	VARCHAR(128) NOT NULL Name of the index.
IXSCHEMA	VARCHAR(128) NOT NULL Qualifier of the index.
KEYSEQ	SMALLINT NOT NULL Numeric position of the key-target in the index.
TYPESCHEMA	VARCHAR(128) NOT NULL Schema of the data type.
TYPENAME	VARCHAR(128) NOT NULL Name of the data type.
DATATYPEID	INTEGER NOT NULL The internal ID of the data type.
SOURCETYPEID	INTEGER NOT NULL For a built-in data type, this field contains 0. For a distinct type, this field contains the internal ID of the built-in type on which the distinct type is based.
LENGTH	SMALLINT NOT NULL The length attribute of the key-target or its precision for a decimal key-target. The number does not include the internal prefixes that are used to record the actual length and null states, when applicable. Listed by data type, then value of the LENGTH column.

Name	Data Type/Description
	• INTEGER—4 • SMALLINT—2 • FLOAT—4 or 8 • CHAR—The length of the string • VARCHAR—The maximum length of the string. • DECIMAL—The precision of the number. • GRAPHIC—The number of DBCS characters. • VARGRAPHIC—The maximum number of DBCS characters. • DATE—4 • TIME—3 • TIMESTAMP—10 • BIGINT—8 • BINARY—The length of the string • VARBINARY—The maximum length of the string. • DECFLOAT—8 or 16
LENGTH2	INTEGER NOT NULL The maximum length of the data that is retrieved from the column. Possible values include the following values: • 0—Not a ROWID column • 40—For a ROWID column, the length of the value that is returned
SCALE	SMALLINT NOT NULL The scale of decimal data. SCALE contains 0 if the key is not a decimal key.
NULLS	CHAR(1) NOT NULL Whether the key can contain null values: N/Y.
IBMREQD	CHAR(1) NOT NULL A value of Y indicates that the row came from the basic machine-readable material (MRM) tape.
STATSTIME	TIMESTAMP NOT NULL WITH DEFAULT The timestamp of the most recent RUNSTATS. The default value is '0001-01-01.00.00.00.000000'. STATSTIME is an updatable column.
CARDF	FLOAT NOT NULL WITH DEFAULT -1 The number of distinct values for the key-target.
HIGH2KEY	VARCHAR(2000) NOT NULL WITH DEFAULT FOR BIT DATA The second highest key-value.
LOW2KEY	VARCHAR(2000) NOT NULL WITH DEFAULT FOR BIT DATA The second lowest key-value.
STATS_FORMAT	CHAR(1) NOT NULL WITH DEFAULT The type of statistics that are gathered: • N—VARCHAR column statistical values are not padded • blank—Statistics have not been collects or VARCHAR column statistical values are padded

SYSIBM.SYSKEYTGTDIST		(DSNDB06.SYSSTATS)
SYSIBM.DSNTDX01	D	(IXSCHEMA,IXNAME,KEYSEQ)
Contains distribution statistical information for the first key-target of an extended index key (populated by RUNSTATS). You can insert, update, and delete rows in this table.		

This table contains the following columns:

Name	Data Type/Description
STATSTIME	TIMESTAMP NOT NULL WITH DEFAULT If the RUNSTATS utility updated the statistics, this column contains the date and time when the last invocation of RUNSTATS updated the statistics. The default value is '0001-01-01.00.00.00.000000'.
IBMREQD	CHAR(1) NOT NULL A value of Y indicates that the row came from the basic machine-readable material (MRM) tape.
IXSCHEMA	VARCHAR(128) NOT NULL The qualifier of the index.
IXNAME	VARCHAR(128) NOT NULL The name of the index.
KEYSEQ	SMALLINT NOT NULL The numeric position of the key-target in the index.
KEYVALUE	VARCHAR(2000) NOT NULL WITH DEFAULT FOR BIT DATA Contains the data of a frequently occurring value. If the value has a non-character data type, the data might not be printable.
TYPE	CHAR(1) NOT NULL WITH DEFAULT 'F' The type of statistics that are gathered: • C—Cardinality • F—Frequent value • N—Non-padded frequent value • H—Histogram statistics
CARDF	FLOAT NOT NULL WITH DEFAULT -1 • TYPE='C—CARDF contains the number of distinct values for the key group. • TYPE='H'—CARDF contains the number of distinct values for the key group in a quantile indicated by QUANTILENO.
KEYGROUPKEYNO	VARCHAR(254) NOT NULL WITH DEFAULT FOR BIT DATA KEYGROUPKEYNO contains a value that identifies the set of keys that are associated with the statistics. 0 if the statistics are only associated with a single key. If the statistics are associated with more than a single key, KEYGROUPKEYNO contains an array of SMALLINT key numbers with a dimension that is equal to the value in NUMKEYS.
NUMKEYS	SMALLINT NOT NULL WITH DEFAULT -1 The number of keys that are associated with the statistics – always 1 for XML index.

Name	Data Type/Description
FREQUENCYF	FLOAT NOT NULL WITH DEFAULT -1 • TYPE='F' or 'N'—FREQUENCYF contains the percentage of entries in the index that have the value that is specified in KEYVALUE when the number of entries is multiplied by 100. • When TYPE='H'—FREQUENCYF contains the percentage of entries in the index that have a value that is in the range of the quantile that is indicated in QUALTILENO.
QUANTILENO	SMALLINT NOT NULL WITH DEFAULT -1 QUANTILENO contains an ordinary sequence number of a quantile in the whole consecutive value range, from low to high.
LOWVALUE	VARCHAR(2000) NOT NULL WITH DEFAULT FOR BIT DATA TYPE='H', LOWVALUE contains the lower bound for the quantile that is in QUANTILENO. Not used if TYPE does not equal 'H'.
HIGHVALUE	VARCHAR(2000) NOT NULL WITH DEFAULT FOR BIT DATA When TYPE='H', HIGHVALUE contains the upper bound for the quantile that is in QUANTILENO. HIGHVALUE is not used if TYPE does not equal 'H'.

SYSIBM.SYSKEYTGTDISTSTATS		**(DSNDB06.SYSSTATS)**
SYSIBM.DSNTSX01	**D**	**(IXSCHEMA,IXNAME, KEYSEQ,PARTITION)**
Contains distribution statistical information for each partition for the first key-target of a data-partitioned secondary index. Zero or more rows per partition (populated by RUNSTATS). You can insert, update , and delete rows in this table.		

This table contains the following columns:

Column Name	Data Type/Description
STATSTIME	TIMESTAMP NOT NULL WITH DEFAULT If RUNSTATS updated the statistics, STATSTIME contains the timestamp of the most recent RUNSTATS. The default value is '0001-01-01.00.00.00.000000'.
IBMREQD	CHAR(1) NOT NULL A value of Y indicates that the row came from the basic machine-readable material (MRM) tape.
PARTITION	SMALLINT NOT NULL The partition number of the table space that contains the index in which the key is defined.
IXSCHEMA	VARCHAR(128) NOT NULL The qualifier of the index.
IXNAME	VARCHAR(128) NOT NULL The name of the index.
KEYSEQ	SMALLINT NOT NULL Numeric position of the key-target in the index.

Column Name	Data Type/Description
KEYVALUE	VARCHAR(2000) NOT NULL WITH DEFAULT FOR BIT DATA Contains the data of a frequently occurring value. If the value has a non-character data type, the data might not be printable.
TYPE	CHAR(1) NOT NULL WITH DEFAULT 'F' The type of statistics that are gathered: • C—Cardinality • F—Frequent value • N—Non-padded frequent value • H—Histogram statistics
CARDF	FLOAT NOT NULL WITH DEFAULT -1 • TYPE='C'—CARDF contains the number of distinct values for the key group. • TYPE='H'—CARDF contains the number of distinct values for the key group in the quantile that is in QUANTILENO.
KEYGROUPKEYNO	VARCHAR(254) NOT NULL WITH DEFAULT Identifies the set of keys that are associated with the statistics. If the statistics are only associated with a single key, KEYGROUPKEYNO contains a zero length value. Otherwise, KEYGROUPKEYNO contains an array of SMALLINT key numbers that have a dimension that is equal to the value in NUMKEYS.
NUMKEYS	SMALLINT NOT NULL WITH DEFAULT Identifies the number of keys that are associated with the statistics.
FREQUENCYF	FLOAT NOT NULL WITH DEFAULT -1 TYPE='F' or 'N', FREQUENCYF contains the percentage of entries in the index that have the value that is specified in KEYVALUE when the number of entries is multiplied by 100. TYPE='H', FREQUENCYF contains the percentage of entries in the index that have a value that is in the range of the quantile that is indicated in QUALTILENO.
QUANTILENO	SMALLINT NOT NULL WITH DEFAULT -1 QUANTILENO contains an ordinary sequence number of a quantile in the whole consecutive value range, from low to high.
LOWVALUE	VARCHAR(2000) NOT NULL WITH DEFAULT FOR BIT DATA TYPE='H' LOWVALUE is the lower bound for the quantile that is indicated in QUANTILENO. Not used if TYPE does not equal 'H'.
HIGHVALUE	VARCHAR(2000) NOT NULL WITH DEFAULT FOR BIT DATA TYPE='H' HIGHVALUE is the upper bound for the quantile that is indicated in QUANTILENO. Nnot used if TYPE does not equal 'H'.
KEYCARDDATA	VARCHAR(1000)/Internal use only

SYSIBM.SYSKEYTGTDIST_HIST		(DSNDB06.SYSHIST)
SYSIBM.DSNTDX02	D	(IXSCHEMA,IXNAME, KEYSEQ,STATSTIME)
Contains history from the SYSKEYTGTDIST table (populated by RUNSTATS). You can insert, update, and delete rows in this table.		

This table contains the following columns:

Column Name	Data Type/Description
STATSTIME	TIMESTAMP NOT NULL WITH DEFAULT If the RUNSTATS utility updated the statistics, this column contains the date and time when the last invocation of RUNSTATS updated the statistics. The default value is '0001-01-01.00.00.00.000000'.
IBMREQD	CHAR(1) NOT NULL A value of Y indicates that the row came from the basic machine-readable material (MRM) tape.
IXSCHEMA	VARCHAR(128) NOT NULL The qualifier of the index.
IXNAME	VARCHAR(128) NOT NULL The name of the index.
KEYSEQ	SMALLINT NOT NULL The numeric position of the key-target in the index.
KEYVALUE	VARCHAR(2000) NOT NULL WITH DEFAULT FOR BIT DATA KEYVALUE contains the data of a frequently occurring value. If the value has a non-character data type, the data might not be printable.
TYPE	CHAR(1) NOT NULL WITH DEFAULT 'F' The type of statistics that are gathered: • C—Cardinality • F—Frequent value • N—Non-padded frequent value • H—Histogram statistics
CARDF	FLOAT NOT NULL WITH DEFAULT -1 • TYPE='C'—CARDF contains the number of distinct values for the key group. • TYPE='H'—CARDF contains the number of distinct values for the key group in a quantile indicated by QUANTILENO.
KEYGROUPKEYNO	VARCHAR(254) NOT NULL WITH DEFAULT FOR BIT DATA Contains a value that identifies the set of keys that are associated with the statistics. Contains 0 if the statistics are only associated with a single key. If the statistics are associated with more than a single key, KEYGROUPKEYNO contains an array of SMALLINT key numbers with a dimension that is equal to the value in NUMKEYS.
NUMKEYS	SMALLINT NOT NULL WITH DEFAULT -1 The number of keys that are associated with the statistics.
FREQUENCYF	FLOAT NOT NULL WITH DEFAULT -1

Column Name	Data Type/Description
	• TYPE='F' or 'N', FREQUENCYF contains the percentage of entries in the index that have the value that is specified in KEYVALUE when the number of entries is multiplied by 100. • When TYPE='H', FREQUENCYF contains the percentage of entries in the index that have a value that is in the range of the quantile that is indicated in QUALTILENO.
QUANTILENO	SMALLINT NOT NULL WITH DEFAULT -1 QUANTILENO contains an ordinary sequence number of a quantile in the whole consecutive value range, from low to high.
LOWVALUE	VARCHAR(2000) NOT NULL WITH DEFAULT FOR BIT DATA When TYPE='H', LOWVALUE contains the lower bound for the quantile that is in QUANTILENO. Not used if TYPE does not equal 'H'.
HIGHVALUE	VARCHAR(2000) NOT NULL WITH DEFAULT FOR BIT DATA When TYPE='H', HIGHVALUE contains the upper bound for the quantile that is in QUANTILENO. Not used if TYPE does not equal 'H'.

SYSIBM.SYSLEVELUPDATES		**(DSNDB06.SYSTSLVH)**
SYSIBM.DSNLVX01	**D**	**(OPERATION_TYPE, EFFECTIVE_TIME)**
SYSIBM.DSNLVX02	**D**	**(EFFECTIVE_TIME)**
Contains information about function levels, catalog levels, and DB2 code levels of the DB2 subsystem or data sharing group.		

This table contains the following columns:

Column Name	Data Type/Description
FUNCTION_LVL	VARCHAR(10 NOT NULL Function level in effect when this row was inserted.
PREV_FUNCTION_LVL	VARCHAR(10) NOT NULL Previous function level.
HIGH_FUNCTION_LVL	VARCHAR(10) NOT NULL Highest activated function level.
CATALOG_LVL	VARCHAR(10) NOT NULL Type of operation. C : Catalog level change. F : Function level change. M : Code level change.
EFFECTIVE_TIME	TIMESTAMP(12) NOT NULL Timestamp when operation was performed.
EFFECTIVE_LRSN	CHAR(10) NOT NULL FOR BIT DATA RBA or LRSN.
OPERATION_TEXT	VARCHAR(256) NOT NULL Text of operation executed.
GROUP_MEMBER	VARCHAR(24) NOT NULL Data Sharing group member where operation was executed.

SYSIBM.SYSLOBSTATS		(DSNDB06.SYSSTATS)
SYSIBM.DSNLNX01	P	(DBNAME,NAME)
Contains information for each LOB table space (populated by RUNSTATS).		

This table contains the following columns:

Column Name	Data Type/Description
STATSTIME	TIMESTAMP NOT NULL Timestamp of statistical RUNSTATS update.
AVGSIZE	INTEGER NOT NULL Average LOB size, in bytes, in LOB table space.
FREESPACE	INTEGER NOT NULL Number of kilobytes of free space in LOB table space.
ORGRATIO	DECIMAL(5,2) NOT NULL Ratio of organization in the LOB table space. Value 1 indicating perfect organization. The greater the value exceeds 1, the greater the disorganization.
DBNAME	VARCHAR(24) NOT NULL Name of the database containing this LOB table space.
NAME	VARCHAR(24) NOT NULL Name of this LOB table space.
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other.

SYSIBM.SYSLOBSTATS_HIST		(DSNDB06.SYSHIST)
SYSIBM.DSNHJX01	D	(DBNAME,NAME, STATSTIME)
Contains history from SYSLOBSTATS (populated by RUNSTATS). You can insert, update, and delete rows in this table.		

This table contains the following columns:

Column Name	Data Type/Description
STATSTIME	TIMESTAMP NOT NULL Timestamp of RUNSTATS statistics update.
FREESPACE	INTEGER NOT NULL Number of pages of free space in the LOB table space.
ORGRATIO	DECIMAL(5,2) NOT NULL Ratio of organization in the LOB table space. A value of indicates perfect organization of the LOB table space. The greater the value exceeds 1, the more disorganized the LOB table space.
DBNAME	VARCHAR(24) NOT NULL Name of the database that contains the LOB table space named in NAME.
NAME	VARCHAR(24) NOT NULL Name of the LOB table space.
IBMREQD	CHAR(1) NOT NULL WITH DEFAULT 'N' Whether the row came from the basic MRM tape: Y/N/other.

SYSIBM.SYSOBDS		(DSNDB06.SYSALTER)
SYSIBM.DSNDOB01	D	(CREATOR,NAME,OBDTYPE)
SYSIBM.DSNDOB02	D	(DBID,PSID,RBA)
IBM internal use only table introduced in DB2 8 to hold information related to online ALTERs.		

This table contains the following columns:

Column Name	Data Type/Description
CREATOR	VARCHAR(128)/IBM internal use only.
NAME	VARCHAR(128)/IBM internal use only.
DBID	SMALLINT/IBM internal use only.
PSID	SMALLINT/IBM internal use only.
OBID	SMALLINT/IBM internal use only.
OBDTYPE	CHAR(1)/IBM internal use only.
VERSION	SMALLINT/IBM internal use only.
CREATEDTS	TIMESTAMP/IBM internal use only.
OBD	VARCHAR(30000)/IBM internal use only.
IBMREQD	CHAR(1) WITH DEFAULT 'N' FOR BIT DATA IBM internal use only.
RBA	CHAR(10) DEFAULT x'00000000000000000000' FOR BIT DATA
ROWID	ROWID/Row identifier.
OBD_IMAGE	BLOB(1G)
RELCREATED	CHAR(1)

SYSIBM.SYSOBD_AUX		(DSNDB06.SYSTSOBX)
SYSIBM.DSNDOB03	U	(AUXID,AUXVER)
Auxiliary table for OBD_IMAGE column of SYSIBM.SYSOBDS.		

This table contains the following column:

Column Name	Data Type	Description
AUXID	VARCHAR(17)	ID of auxiliary data.
AUXVER	SMALLINT	Version of auxiliary data.
AUXVALUE	BLOB(1G) NOT NULL	

SYSIBM.SYSOBJROLEDEP		(DSNDB06.SYSROLES)
SYSIBM.DSNRDX01	U	(DSCHEMA,DNAME,DTYPE)
SYSIBM.DSNRDX02	D	(ROLENAME)
Lists dependent objects for each role.		

This table contains the following columns:

Column Name	Data Type/Description
DEFINER	VARCHAR(128) NOT NULL The authorization ID or role that created the object.
DEFINERTYPE	CHAR(1) NOT NULL Type of definer: L - Role Blank - Authorization ID
ROLENAME	VARCHAR(128) NOT NULL Name of the role on which there is a dependency.
DSCHEMA	VARCHAR(128) NOT NULL Name of the schema of the dependent object.
DNAME	VARCHAR(128) NOT NULL Name of the dependent object.

Column Name	Data Type/Description
DTYPE	CHAR(1) NOT NULL The type of the dependent object in DNAME: • **B** Trigger • **D** Database • **E** Distinct type • **F** User-defined function • **I** Index • **J** JAR file • **L** Role • **M** Materialized query table • **N** Trusted context • Stored procedure • **Q** Sequence • **R** Table space • **S** Storage group • **T** Table • **V** View • **X** Row permission • **Y** Column mask • 0 Alias
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other.

SYSIBM.SYSPACKAGE		**(DSNDB06.SYSTSPKG)**
SYSIBM.DSNKKX01	**U**	**(LOCATION,COLLID,NAME, VERSION)**
SYSIBM.DSNKKX02	**P**	**(LOCATION,COLLID,NAME, CONTOKEN)**
Defines each package. One row per package.		

This table contains the following columns:

Column Name	Data Type/Description
LOCATION	VARCHAR(128) NOT NULL Always contains blanks.
COLLID	VARCHAR(128) NOT NULL Name of the package collection or schema name for trigger package.
NAME	VARCHAR(128) NOT NULL Name of the package.
CONTOKEN	CHAR(8) NOT NULL FOR BIT DATA Consistency token for the package from a DBRM: the level as specified by the LEVEL option at precompile or the timestamp indicating when package program was precompiled.
OWNER	VARCHAR(128) NOT NULL Authorization ID of package owner. For a trigger package, the value is the authorization ID of the trigger owner which is current authorization ID (plan or package owner for static CREATE TRIGGER or current SQLID for dynamic CREATE TRIGGER).
CREATOR	VARCHAR(128) NOT NULL Authorization ID of the owner of the creator of the package version. For trigger package: if dynamic SQL, then primary authorization ID of CREATE TRIGGER issuer;

Column Name	Data Type/Description
	if static SQL, then authorization ID of plan or package owner.
TIMESTAMP	TIMESTAMP NOT NULL Timestamp when package was created.
BINDTIME	TIMESTAMP NOT NULL Timestamp when package was last bound.
QUALIFIER	VARCHAR(128) NOT NULL Implicit name for unqualified tables, views, aliases, and indexes.
PKSIZE	INTEGER NOT NULL Size of the base section of the package, in bytes.
AVGSIZE	INTEGER NOT NULL Average size of those sections of the package containing SQL statements, in bytes.
SYSENTRIES	SMALLINT NOT NULL Number of enabled or disabled entries for this package in SYSIBM.SYSPKSYSTEM. 0, if all connections are enabled.
VALID	CHAR(1) NOT NULL Indicates whether the package is valid: • A—The ALTER statement changed the description of the table of a view referred to by the package, but package is still valid. • H—The ALTER TABLE statement changed the description of the table or base table of a view referred to by the package. The change invalidates the package for DB2 releases prior to V5. • N—NO • Y—YES
OPERATIVE	CHAR(1) NOT NULL Whether package can be allocated: Y/N.
VALIDATE	CHAR(1) NOT NULL Whether validity checking can be deferred until run time: • B—All checking must be performed at bind time • R—Checking deferred to runtime if tables, views, or privileges do not exist at bind time
ISOLATION	CHAR(1) NOT NULL Isolation level when package last bound or rebound: • R—RR (repeatable read) • S—CS (cursor stability) • T—RS (read stability) • U—UR (uncommitted read) • blank—Isolation level for the plan executing the package
RELEASE	CHAR(1) NOT NULL Value used for RELEASE when package last bound or rebound: • C—at COMMIT • D—at DEALLOCATION • I—inherited from plan • blank— not specified, and therefore the

Column Name	Data Type/Description
	value specified for the plan executing the package.
EXPLAIN	CHAR(1) NOT NULL Whether information on package statements was added to owner's PLAN_TABLE: Y/N.
QUOTE	CHAR(1) NOT NULL SQL string delimiter for SQL statements in the package: N (apostrophe) or Y (quotation mark).
COMMA	CHAR(1) NOT NULL Decimal point representation for SQL in the package: N (period) or Y (comma).
HOSTLANG	CHAR(1) NOT NULL Host language for the package's DBRM: • B—Assembler language • C—OS/VS COBOL • D—C • F—FORTRAN • P—PL/1 • 2—VS COBOL II or IBM COBOL Release I • 3—IBM COBOL • 4—C++ • Blank—for package: bound remotely, trigger (TYPE=T or 1, SQL procedure (TYPE=N) of non-inline SQL scalar function (TYPE=F).
CHARSET	CHAR(1) NOT NULL Indicates whether the system CCSID for SBCS was 290 (Katakana) at precompile: K (Yes) or A (No).
MIXED	CHAR(1) NOT NULL If mixed data was in effect at precompile: Y/N.
DEC31	CHAR(1) NOT NULL If DEC31 option was in effect when the package program was in precompile: Y/N.
DEFERPREP	CHAR(1) NOT NULL Indicates CURRENTDATA option when package was bound or rebound: • A—Data currency required and inhibit blocking for all cursors • B—Data currency not required for ambiguous cursors • C—Data currency required for ambiguous cursors • Blank—If package created before CURRENTDATA option was available
SQLERROR	CHAR(1) NOT NULL SQLERROR option on most recent subcommand that bound or rebound the package: C (CONTINUE) or N (NOPACKAGE)
REMOTE	CHAR(1) NOT NULL Package source: • C—Package created by BIND COPY • D—Package created by BINDCOPY with the OPTIONS (command). • K—Package copied from package originally bound on behalf of a remote

Column Name	Data Type/Description
	requester. • L—Package was copied with the OPTIONS (command) from a package originally bound by a remote requester. • N—Package bound locally from a DBRM • Y—Package bound on behalf of a remote requester.
PCTIMESTAMP	TIMESTAMP NOT NULL Date and time of program precompile. Or 0001-01-01-00.00.00.000000 if LEVEL precompile option is used or package came from a non-DB2 location.
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other..
VERSION	VARCHAR(122) NOT NULL Version identifier for the package. For trigger package(TYPE=T or 1), or created using BIND PACKAGE (TYPE=blank), value is blank
PDSNAME	VARCHAR(132) NOT NULL For package bound locally, the name of the PDS library in which the package's DBRM is a member. For package locally copied, the value in SYSPACKAGE PDSNAME for the source package. Otherwise, the product signature of the bind requester followed by requester's location name (if DB2) or requester's LUNAME enclosed in brackets.
DEGREE	CHAR(3) NOT NULL WITH DEFAULT The DEGREE option used when package last bound: • ANY—DEGREE(ANY) • 1—DEGREE(1) • blank—If package was migrated
GROUP_ MEMBER	VARCHAR(24) NOT NULL WITH DEFAULT The DB2 sharing member name of the DB2 subsystem that performed the most recent bind. Blank if the DB2 subsystem was not in a DB2 sharing environment when the bind was performed.
DYNAMICRULES	CHAR(1) NOT NULL WITH DEFAULT The DYNAMICRULES option used when the package was last bound: • B—BIND. Dynamic SQL statements are executed with DYNAMICRULES bind behavior. • D—DEFINEBIND. When package runs under active stored procedure or user-defined function, dynamic SQL in package is executed with DYNAMICRULES define behavior. When package is not run under an active stored procedure or user-defined function, dynamic SQL in package is executed with DYNAMICRULES bind behavior. • E—DEFINERUN. When package runs under active stored procedure or user-defined function, dynamic SQL in

Column Name	Data Type/Description
	package is executed with DYNAMICRULES defined behavior. When package is not run under an active stored procedure or user-defined function, dynamic SQL in package is executed with DYNAMICRULES run behavior. • H—INVOKEBIND. When package runs under active stored procedure or user-defined dynamic SQL in package is executed with DYNAMICRULES invoke behavior. When package is not run under an active stored procedure or user-defined function dynamic SQL in package is executed with DYNAMICRULES bind behavior. • I—INVOKERUN. When package runs under active stored procedure or user-defined dynamic SQL in package is executed with DYNAMICRULES invoke behavior. When package is not run under an active stored procedure or user-defined function dynamic SQL in package is executed with DYNAMICRULES run behavior. • R—RUN. Dynamic SQL is executed with DYNAMICRULES run behavior. • blank—DYNAMICRULES is not specified for the package. Package uses the DYNAMICRULES value of the plan calling the package at execution time.
REOPTVAR	CHAR(1) NOT NULL WITH DEFAULT 'N' Whether the access path was determined again at execution time using input variables: • A—Bind option REOPT(AUTO) indicates that the access path is determined multiple times at execution time depending on the parameter value. • N—Bind option NOREOPT(NONE) access path determination at BIND. • Y—Bind option REOPT(ALWAYS) access path determined at execution for SQL statements with variables. • 1—Bind option REOPT(ONCE) the access path is determined at execution time only once.
DEFERPREPARE	CHAR(1) NOT NULL WITH DEFAULT PREPARE processing deferred until OPEN is executed: • N—Bind option NODEFER (PREPARE); processing is not deferred. • Y—Bind option DEFER (PREPARE); processing is deferred. • I—Inherit value from plan • Blank—Trigger package otherwise bind option not specified and inherited from plan.
KEEPDYNAMIC	CHAR(1) NOT NULL WITH DEFAULT 'N' Whether prepared dynamic statements are

Column Name	Data Type/Description
	to be purged at each commit point: • N—Bind options KEEPDYNAMIC (NO); statements are destroyed at commit. • Y—Bind option KEEPDYMANIC (YES); Prepared dynamic statements are kept past commit/rollback.
PATHSCHEMAS	VARCHAR (2048) NOT NULL WITH DEFAULT SQL path specified on BIND or REBIND. Used to resolve unqualified data type, function and stored procedures. If PATH bind option is not specified, value has zero length and DB2 uses default of SYSIBM, SYSFUNC, SYSPROC, or package qualifier.
TYPE	CHAR(1) NOT NULL WITH DEFAULT Type of package. Indicates how package was created: • F—Non-inline scalar function created by CREATE FUNCTION, ALTER FUNCTION, or BIND PACKAGE DEPLOY. • N—CREATE/ALTER PROCEDURE statement, or BIND PACKAGE DEPLOY command created package, and package is native SQL routine. • T—Basic TRIGGER • Blank—Bind package command created package • 1—An advanced TRIGGER.
DBPROTOCOL	CHAR(1) NOT NULL WITH DEFAULT 'D' Whether remote access is implemented with DRDA: D (DRDA) or C (DRDA with package-based continuous block fetch).
FUNCTIONTS	TIMESTAMP NOT NULL WITH DEFAULT Timestamp when function was resolved. Set by BIND and REBIND, not by AUTOBIND.
OPTHINT	VARCHAR(128) NOT NULL WITH DEFAULT Value of OPTHINT is bind option, identifying rows from authid. PLAN_TABLE to be used as input to the optimizer. Blank if no rows input to optimizer.
ENCODING_ CCSID	INTEGER NOT NULL WITH DEFAULT The CCSID corresponding to the encoding scheme or CCSID as specified for the bind option ENCODING. The Encoding Scheme specified on the bind command: • ccsid—The specified or derived CCSID. • 0—The default CCSID as specified on panel DSNTIPF at installation time. Used when the package was bound prior to V7.
IMMEDWRITE	CHAR(1) NOT NULL WITH DEFAULT Indicates when writes of updated group buffer pool dependent pages are to be done. This option is only applicable for data sharing environments. • I—Inherit value from plan. • N—Bind option IMMEDWRITE(NO) indicates normal write activity is done. • Y—Bind option IMMEDWRITE (YES) indicates that immediate writes are done for updated group buffer pool

Column Name	Data Type/Description
	dependent pages. • 1—Bind option IMMEDWRITE(PH1) indicates that updated group buffer pool dependent pages are written at or before phase one commit. • blank—A migrated package.
RELBOUND	CHAR(1) NOT NULL WITH DEFAULT The release when the package was bound or rebound. • blank—Bound prior to V7 • K—Bound on V7 • L—Bound on V8
CATENCODE	CHAR(1)/Unused.
REMARKS	VARCHAR(550) NOT NULL WITH DEFAULT A character string provided by the user with the COMMENT statement.
OWNERTYPE	CHAR(1) NOT NULL WITH DEFAULT Indicates the type of owner: • blank—Authorization ID • L—Role
ROUNDING	CHAR(1)NOT NULL WITH DEFAULT The ROUNDING option used when the package was last bound: • C—ROUND_CEILING • D—ROUND_DOWN • F—ROUND_FLOOR • G—ROUND_HALF_DOWN • E—ROUND_HALF_EVEN • H—ROUND_HALF_UP • U—ROUND_UP • blank—The package created in a DB2 release prior to V9.
DISTRIBUTE	CHAR(1) NOT NULL WITH DEFAULT 'N' DB2 should gather location names from SQL statements, and create remote packages for user (local bind only). • A—Yes • L—No, use list of location-names.
LASTUSED	DATE NOT NULL WITH DEFAULT Last date objects used.
CONCUR_ACC_RES	CHAR(1) NOT NULL CONCURRENTACCESSRESOLUTION option when package was bound or rebound: • W—WAITFOROUTCOME • U—USECURRENTLYCOMMITTED
EXTENDED INDICATOR	CHAR(1) NOT NULL WITH DEFAULT EXTENDEDINDICATOR bind option: N (no) or Y (yes).
COPYID	INTEGER/Not Used
PLANMGMT	CHAR(1) NOT NULL WITH DEFAULT PLANMGMT bind option • B—BASIC • E—EXTENDED • blank—OFF
PLANMGMTSCOPE	CHAR(1) NOT NULL WITH DEFAULT PLANMGMTSCOPE bind option • S—STATIC

Column Name	Data Type/Description
APREUSE	CHAR(1) NOT NULL WITH DEFAULT Bind option for APREUSE: • N—No, access paths not reused • W—DB2 tries to reuse access paths. Processing continues if not possible. • E—DB2 tries to reused access paths. Processing ends if not possible.
APRETAINDUP	CHAR(1) NOT NULL WITH DEFAULT APRETAINDUP bind option • Y—YES, all copies retained. • 0—NO, previous or original package copy still retained due to access path differences. • 1—NO, previous copy not retained. • 2—NO, original copy not retained.
SYSTIMESENSITIVE	CHAR(1) NOT NULL WITH DEFAULT 'N' SYSTIMESENSITIVE bind option: • Y—System time temporal tables • affected by special register. • N—System time temporal tables not Affected by special register. Special register: SET CURRENT TEMPORAL SYSTEM_TIME.
RECORDTEMPORALHIST	CHAR(1) NOT NULL WITH DEFAULT 'Y' Unused
BUSTIMESENSITIVE	CHAR(1) NOT NULL WITH DEFAULT 'N' Bind option for BUSTIMESENSITIVE: • Y—business-time temporal tables affected by SET CURRENT TEMPORAL BUSINESS_TIME • N—business-tiem temporal tables not affected by SET CURRENT TEMPORAL BUSINESS_TIME
APPLCOMPAT	VARCHAR(10) NOT NULL WITH DEFAULT Bind option for APPLCOMPAT: • V10R1—SQL statements have DB2 10 behavior. • V11R1—SQL statements have DB2 11 behavior. • *Function-level*—specified function-level dictates compatibility behavior Empty string : Defined prior to V10R1.
ARCHIVESENSITIVE	CHAR(1) NOT NULL WITH DEFAULT 'N' Bind option for ARCHIVESENSITIVE: • Y—SYSIBMADM.GET_ARCHIVE affects archive enabled tables. • N—SYSIBMADM.GET_ARCHIVE does not affects archive enabled tables.
EXTSEQNO	INTEGER NOT NULL WITH DEFAULT 0 Internal use
DESCSTAT	CHAR(1) NOT NULL WITH DEFAULT Bind option for DESCSTAT: • Y DB2 generates DESCRIBE SQLDA • N DB2 does not generate DESCRIBE SQLDA
ORIGIN	CHAR(1) NOT NULL WITH DEFAULT The origin of the EXPLAIN records: • A—Automatic bind

Column Name	Data Type/Description
	• B—BIND command • G—Explicit ALTER REGENERATE of the SQL procedure for the package • I—Implicit automatic regeneration of the SQL procedure for the package because of fallback, coexistence, or deployment to a lower releasewhere the routine is incompatible. • R—REBIND command • blank—The row existed before DB2 12.
APREUSE_NO_FL	VARCHAR(10) NOT NULL WITH DEFAULT The function level when the package was bound with APREUSE(NO), or blank if the package was bound before DB2 12, or not determined.
APREUSE_NO_TS	TIMESTAMP NOT NULL WITH DEFAULT The bind time when the package was bound with APREUSE(NO): 0001-01-01-00.00.00.000000 The package was bound before DB2 12.
CONC_STMT	CHAR(1) NOT NULL WITH DEFAULT 'N' Whether statement concentration is enabled: N (No) or Y (Yes). The default is N.
FUNCTION_LVL	VARCHAR(10) NOT NULL WITH DEFAULT The function level when the row was inserted.

SYSIBM.SYSPACKAUTH		**(DSNDB06.SYSTSPKA)**
DSNKAX01	**D**	**(GRANTOR,LOCATION,COLLID, NAME,GRANTORTYPE)**
DSNKAX02	**D**	**(GRANTEE,LOCATION,COLLID, NAME,BINDAUTH,COPYAUTH, EXECUTEAUTH,GRANTEETYPE)**
DSNKAX03	**D**	**(LOCATION,COLLID,NAME)**
Defines the privileges held by users over packages.		

This table contains the following columns:

Column Name	Data Type/Description
GRANTOR	VARCHAR(128) NOT NULL Authorization ID of user who granted the privileges. Could also be PUBLIC or PUBLIC *.
GRANTEE	VARCHAR(128) NOT NULL Authorization ID of user who holds privileges or name of an application plan that uses privileges.
LOCATION	VARCHAR(128) NOT NULL Always contains blanks.
COLLID	VARCHAR(128) NOT NULL Name of package collection.
NAME	VARCHAR(128) NOT NULL Name of package on which privileges are held. An (*) if privileges held on all packages in a collection.
CONTOKEN	CHAR(8)/Unused.
TIMESTAMP	TIMESTAMP NOT NULL Timestamp when privilege was granted.
GRANTEETYPE	CHAR(1) NOT NULL Type of grantee.

Column Name	Data Type/Description
	• Blank—An authorization ID. • L—Role. • P—An application plan.
AUTHHOWGOT	CHAR(1) NOT NULL Authorization level of the grantor: • blank—not applicable • A—PACKADM (on collection *) • C—DBCTRL • D—DBADM • E—SECADM • G—ACCESSCTRL • L—SYSCTRL • M—DBMAINT • P—PACKADM (on specific collection) • S—SYSADM • T—DATAACCESS
BINDAUTH	CHAR(1) NOT NULL Whether grantee can use the BIND or REBIND: (see legend ** after table)
COPYAUTH	CHAR(1) NOT NULL Whether grantee can COPY package: (see legend ** after table)
EXECUTEAUTH	CHAR(1) NOT NULL If grantee can run programs that use the package: (see legend ** after table)
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other..
GRANTORTYPE	CHAR(1) NOT NULL WITH DEFAULT Indicates the type of grantor: • blank—Authorization ID • L—Role
SYS_START	TIMESTAMP(12) NOT NULL System period temporal start time for transaction
SYS_END	TIMESTAMP(12) NOT NULL System period temporal end time for transaction
TRANS_START	TIMESTAMP(12) System period transaction timestamp.

Legend **:

Blank=privilege not held; G=privilege held with GRANT option; Y=privilege held without GRANT option.

SYSIBM.SYSPACKCOPY		**(DSNDB06.SYSTSPKC)**
SYSIBM.SYSPCX01	**U**	**(LOCATION,COLLID,NAME, CONTOKEN,COPYID)**
Contains metadata for old package copies (i.e. PREVIOUS and ORIGINAL).		

This table contains the following columns:

Column Name	Data Type/Description
LOCATION	VARCHAR(128) NOT NULL/Always blank.
COLLID	VARCHAR(128) NOT NULL Package collection name. Schema

Column Name	Data Type/Description
	name of trigger if trigger package.
NAME	VARCHAR(128) NOT NULL Package name.
CONTOKEN	CHAR(8) NOT NULL WITH DEFAULT FOR BIT DATA Package consistency token. If package derived from a DB2 DBRM: Level option or precompile timestamp in internal format.
OWNER	VARCHAR(128) NOT NULL Authorization ID of the owner
CREATOR	VARCHAR(128) NOT NULL Authorization ID of the creator.
TIMESTAMP	TIMESTAMP NOT NULL Create time
BINDTIME	TIMESTAMP NOT NULL Last bind time.
QUALIFIER	VARCHAR(128) NOT NULL Implicit qualifier for unqualified table view, index and alias names in static SQL statements.
PKSIZE	INTEGER NOT NULL Package base section size in bytes.
AVGSIZE	INTEGER NOT NULL Average size, in bytes, of those sections of the plan that contain SQL statements processed at bind time.
SYSENTRIES	SMALLINT NOT NULL Number of enabled or disabled entries for this package in SYSIBM.SYSPKSYSTEM. 0 - All types of connections are enabled.
VALID	CHAR(1) NOT NULL Package validity: • A—Alter statement on table or base table of referenced view did not invalidate package. • H—Alter statement on table or base table of referenced view invalidated the package for releases of DB2 prior to V5. • N—NO. • Y—YES.
OPERATIVE	CHAR(1) NOT NULL Whether package can be allocated: • N—NO (requires explicit BIND or REBIND). • Y—YES.
VALIDATE	CHAR(1) NOT NULL Validity checking can be deferred until run time: B—No, must be done at bind time. R—Yes
ISOLATION	CHAR(1) NOT NULL Isolation level at last bind or rebind: • R—RR(repeatable read) • C—CS(cursor stability) • T—RS(read stability)

Column Name	Data Type/Description
	• U—UR(uncommitted read) • blank—Not specified (uses plan isolation level)
RELEASE	CHAR(1) NOT NULL Package RELEASE value: • C—COMMIT • D—DEALLOCATE • blank—Not specified (uses plan release value)
EXPLAIN	CHAR(1) NOT NULL EXPLAIN option specified: N (NO) or Y (YES).
QUOTE	CHAR(1) NOT NULL SQL string delimiter: N (apostrophe) or Y (quotation mark).
COMMA	CHAR(1) NOT NULL Decimal point representation: N (period) or Y (comma).
HOSTLANG	CHAR(1) NOT NULL Host language for the package's DBRM: • B—Assembler language • C—OS/VS COBOL • D—C • F—Fortran • P—PL/I • 2—VS COBOL II or IBM COBOL Release 1 (formerly called COBOL/370) • 3—IBM COBOL (Release 2 or subsequent releases) • 4—C++ • blank—For remotely bound packages, trigger packages (TYPE='T'), SQL procedure packages (TYPE='N'), or non-inline SQL scalar function packages (TYPE='F').
CHARSET	CHAR(1) NOT NULL CCSID for SBCS data was 290 (Katakana) when the program was precompiled: K (YES) or A (NO).
MIXED	CHAR(1) NOT NULL Mixed data in effect during precompile: N (NO) or Y (YES).
DEC31	CHAR(1) NOT NULL DEC31 in effect during precompile: N (NO) or Y (YES).
DEFERPREP	CHAR(1) NOT NULL Bind CURRENTDATA option (data currency required): • A—Yes for all cursors • B—No for ambiguous cursors • C—Yes for ambiguous cursors. • Blank—package created before C URRENTDATA available
SQLERROR	CHAR(1) NOT NULL Most recent SQLERROR option: • C—CONTINUE

Column Name	Data Type/Description
	• N—NOPACKAGE
REMOTE	CHAR(1) NOT NULL Package source: • C—BIND COPY. • D—BIND COPY with the OPTIONS(COMMAND) option. • K—Copied from a package that was originally bound on behalf of a remote requester. • L—Copied with the OPTIONS(COMMAND) option from a package that was originally bound on behalf of a remote requester. • N—Locally bound from a DBRM. • Y—Bound on behalf of a remote requester.
PCTIMESTAMP	TIMESTAMP Precompile time. '0001-01-01-00.00.00.000000' if the LEVEL precompiler option was used, or if the package came from a non-DB2 location.
IBMREQD	CHAR(1) NOT NULL Row source: • Y—Machine-readable material (MRM) tape. • Else—see Release dependency indicators for values.
VERSION	VARCHAR(122) NOT NULL Package version identifier. (blank for triggers).

PDSNAME	VARCHAR(132) NOT NULL
	• Locally bound package, the DBRM PDS name.
	• Locally copied package, SYSPACKAGE.PDSNAME.
	• Bind requester product signature, followed by:
	  ○ Requester's location name if DB2,
	  ○ Requester's LU name (in angle brackets.

DEGREE	CHAR(3) NOT NULL WITH DEFAULT Package DEGREE option: • ANY—DEGREE)ANY) • 1 or blank—DEGREE(1)
GROUP_MEMBER	VARCHAR(24) NOT NULL WITH DEFAULT Package source: • C—BIND COPY. • D—BIND COPY with the OPTIONS(COMMAND) option. • K—Copied from a package that was originally bound on behalf of a remote requester. • L—Copied with the OPTIONS(COMMAND) option from

Column Name	Data Type/Description
	a package that was originally bound on behalf of a remote requester. • N—Locally bound from a DBRM. • Y—bound on behalf of a remote requester
DYNAMICRULES	CHAR(1) NOT NULL WITH DEFAULT DYNAMICRULES option: • B—BIND behavior. • D—DEFINEBIND behavior. • E—DEFINERUN behavior. • H—INVOKEBIND behavior. • I—INVOKERUN behavior. • R—RUN behavior. • blank—DYNAMICRULES is not specified, use value from plan.
REOPTVAR	CHAR(1) NOT NULL WITH DEFAULT 'N' Re-determine path at execution time using input variable values: • A—AUTO • N—NONE • Y—ALWAYS • 1—ONCE
DEFERPREPARE	CHAR(1) NOT NULL WITH DEFAULT Defer PREPARE processing until OPEN: • N—NO • Y—YES • Blank—Not specified, us plan option.
KEEPDYNAMIC	CHAR(1) NOT NULL WITH DEFAULT 'N' Keep prepared dynamic statement at each commit: • N—No, purge at commit • Y—Yes
PATHSCHEMAS	VARCHAR(2048) NOT NULL WITH DEFAULT SQL path used to resolve unqualified data type, function, and stored procedure names. Empty – default SQL path of: SYSIBM, SYSFUN, SYSPROC, package qualifier.
TYPE	CHAR(1) NOT NULL WITH DEFAULT Package type: • F - CREATE FUNCTION or ALTER FUNCTION statement, or a BIND PACKAGE DEPLOY command created the package, and this package is a non-inline SQL scalar function package. • N—CREATE PROCEDURE or ALTER PROCEDURE statement, or BIND PACKAGE DEPLOY command created the package, and this package is a native SQL routine package. • R—Reserved for IBM use. • T—CREATE TRIGGER statement created the package, and the package is a trigger package.

Column Name	Data Type/Description
	• blank—BIND PACKAGE command created the package.
DBPROTOCOL	CHAR(1) NOT NULL WITH DEFAULT 'P' Remote access is: • D—DRDA • P—DRDA capable of package based continuous block fetch.
FUNCTIONTS	TIMESTAMP NOT NULL WITH DEFAULT Function resolved time.
OPTHINT	VARCHAR(128) NOT NULL WITH DEFAULT Optimizer hint bind option.
ENCODING_CCSID	INTEGER NOT NULL WITH DEFAULT Encoding scheme CCSID: • ccsid—Specified or derived CCSID • 0—Default CCSID
IMMEDWRITE	CHAR(1) NOT NULL WITH DEFAULT When to write updated group buffer pool dependent pages: • N—Normal • Y—Immediate • 1—at or before phase 1 commit • blank—Migrated package
RELBOUND	CHAR(1) NOT NULL WITH DEFAULT Package bind or rebind release: • blank—prior to V7 • else—see Release dependency indicators for values
CATENCODE	CHAR(1)/Not used.
REMARKS	VARCHAR(550) NOT NULL WITH DEFAULT Comment
OWNERTYPE	CHAR(1) NOT NULL WITH DEFAULT Owner type: • blank—Authorization ID • L—Role
ROUNDING	CHAR(1) NOT NULL WITH DEFAULT Rounding option: • C—ROUND_CEILING • D—ROUND_DOWN • F—ROUND_FLOOR • G—ROUND_HALF_DOWN • E—ROUND_HALF_EVEN • H—ROUND_HALF_UP • U—ROUND_UP • blank—Package created in a DB2 release prior to V9.
DISTRIBUTE	CHAR(1) NOT NULL WITH DEFAULT 'N' Collect location names from SQL statements, and create remote packages: A (YES) or L (NO, use list of location names).

Column Name	Data Type/Description
LASTUSED	DATE NOT NULL WITH DEFAULT Date objects were last used.
CONCUR_ACC_RES	CHAR(1) NOT NULL Current access resolution: • blank—Not specified • W—WAITFOROUTCOME • U—USECURRENTLYCOMMITTED
EXTENDEDINDICATOR	CHAR(1) NOT NULL WITH DEFAULT Value of EXTENDEDINDICATOR bind option: N (NO) or Y (YES).
COPYID	INTEGER NOT NULL Package copy version: 1 (previous) or 2 (original).
PLANMGMT	CHAR(1) NOT NULL WITH DEFAULT Value of PLANMGMT bind option: • B—BASIC • E—EXTENDED • F—OFF • O—ON
PLANMGMTSCOPE	CHAR(1) NOT NULL WITH DEFAULT Value of PLANMGMTSCOPE bind option: S - STATIC
APREUSE	CHAR(1) NOT NULL WITH DEFAULT Bind option for APREUSE: • N—Access paths not reused. • E—DB2 tries to reuse access paths – will fail if unsuccessful.
APRETAINDUP	CHAR(1) NOT NULL WITH DEFAULT Value of APRETAINDUP bind option: • Y—YES specified. All copies were retained. • 0—NO specified; however, the • previous or original package copy is still retained due to access path differences. • 1—NO specified, and the previous package copy is not retained as the access paths • are identical to the current copy. • 2—NO specified, and the previous and original package copies are not retained as the access paths are identical to the current copy.
SYSTIMESENSITIVE	CHAR(1) NOT NULL WITH DEFAULT 'N' Bind option for SYSTIMESENSITIVE: • Y—System time temporal tables affected by special register. • N—System time temporal tables not affected by special register. • Special register: SET CURRENT TEMPORAL SYSTEM_TIME.
RECORDTEMPORALHIST	CHAR(1)/Unused
BUSTIMESENSITIVE	CHAR(1) NOT NULL WITH DEFAULT 'N' Bind option for BUSTIMESENSITIVE: • Y— business-time temporal tables affected by SET CURRENT TEMPORAL BUSINESS_TIME

Column Name	Data Type/Description
	• N—business-tiem temporal tables not affected by SET CURRENT TEMPORAL BUSINESS_TIME
APPLCOMPAT	VARCHAR(10) NOT NULL WITH DEFAULT Bind option for APPLCOMPAT: • V10R1—SQL statements have DB2 10 behavior. • V11R1—SQL statements have DB2 11 behavior.
ARCHIVESENSITIVE	CHAR(1) NOT NULL WITH DEFAULT 'N' Bind option for ARCHIVESENSITIVE: • Y—SYSIBMADM.GET_ARCHIVE affects archive enabled tables. • N—SYSIBMADM.GET_ARCHIVE does not affects archive enabled tables.
EXTSEQNO	INTEGER NOT NULL WITH DEFAULT 0 Internal use
DESCSTAT	CHAR(1) NOT NULL WITH DEFAULT Bind option for DESCSTAT: • Y—DB2 generates DESCRIBE SQLDA • N—DB2 does not generate DESCRIBE SQLDA
ORIGIN	CHAR(1) NOT NULL WITH DEFAULT The origin of the EXPLAIN records: • A—Automatic bind • B—BIND command • G—Explicit ALTER REGENERATE of the SQL procedure for the package • I—Implicit automatic regeneration of the SQL procedure for the package because of fallback, coexistence, or deployment to a lower release where the routine is incompatible. • R—REBIND command blank The row existed before DB2 12.
APREUSE_NO_FL	VARCHAR(10) NOT NULL WITH DEFAULT The function level when the package was bound with APREUSE(NO), or blank if the package was bound before DB2 12, or not determined.
APREUSE_NO_TS	TIMESTAMP NOT NULL WITH DEFAULT The bind time when the package was bound with APREUSE(NO): 0001-01-01-00.00.00.000000 The package was bound before DB2 12.
CONC_STMT	CHAR(1) NOT NULL WITH DEFAULT 'N' Whether statement concentration is enabled: N (NO) or Y (YES). N is the default value.
FUNCTION_LVL	VARCHAR(10) NOT NULL WITH DEFAULT The function level when the row was inserted.

SYSIBM.SYSPACKDEP		(DSNDB06.SYSTSPKD)
SYSIBM.DSNKDX01	D	(DLOCATION,DCOLLID,DNAME, DCONTOKEN)
SYSIBM.DSNKDX02	D	(BQUALIFIER,BNAME,BTYPE)
SYSIBM.DSNKDX03	D	(BQUALIFIER,BNAME,BTYPE, DTYPE)
Records the dependencies of packages on tables, views, synonyms, table spaces, indexes, aliases, functions, and stored procedures..		

This table contains the following columns:

Column-name	Data Type/Description
BNAME	VARCHAR(128) NOT NULL Name of an object the package is dependent on. If BTYPE=B or C, then table name on which period is defined.
BQUALIFIER	VARCHAR(128) NOT NULL • If BTYPE is R, the name of the database • If IBTYPE is F, O, or Q, the schema name. • If IBNAME is B or C, then table qualifier. • If BTYPE is L, the value is blank. • Otherwise, the value is the schema of BNAME.
BTYPE	CHAR(1) NOT NULL Object identified by BNAME and BQUALIFIER: • A—Alias • E—INSTEAD OF trigger • F—User-defined function or cast function • G—Global temporary table • H—Global variable • I—Index • M—Materialized query table • O—Stored procedure • P—Partitioned table space • Q—Sequence object • R—Table space • S—Synonym • T—Table • U—Distinct type • V—View • W—SYSTEM_TIME period • Z—BUSINESS_TIME period • 0—Sequence alias
DLOCATION	VARCHAR(128) NOT NULL Always contains blanks.
DCOLLID	VARCHAR(128) NOT NULL Name of the collection.
DNAME	VARCHAR(128) NOT NULL Name of the package (DBRM).
DCONTOKEN	CHAR(8) NOT NULL FOR BIT DATA Consistency token for the package from a DBRM: the level as specified by the LEVEL option at precompile or the timestamp indicating when package program was precompiled.
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other.
DOWNER	VARCHAR(128) NOT NULL WITH DEFAULT

Column-name	Data Type/Description
	Package owner
DTYPE	CHAR(1) NOT NULL WITH DEFAULT Package type: • F—Non-inline SQL scalar function • N—Native SQL routine package • O—Original copy of package • P—Previous copy of package • R—Reserved for IBM use or native SQL routine • T—Basic trigger package • 1—Advanced trigger package • blank—Not trigger package or a native SQL routine package
DOWNERTYPE	CHAR(1) NOT NULL WITH DEFAULT Indicates the type of owner of the package: • blank—Authorization ID • L—Role

SYSIBM.SYSPACKLIST		**(DSNDB06.SYSTSPKL)**
SYSIBM.DSNKLX01	**D**	**(LOCATION,COLLID,NAME)**
SYSIBM.DSNKLX02	**U**	**(PLANNAME,SEQNO,LOCATION, COLLID,NAME)**
Contains all packages bound into a plan via a package list. One row per package.		

This table contains the following columns:

Column Name	Data Type/Description
PLANNAME	VARCHAR(24) NOT NULL Name of the plan.
SEQNO	SMALLINT NOT NULL Sequence no. of this entry in the package list.
LOCATION	VARCHAR(128) NOT NULL Location of the package. Blank if local. Asterisk (*) if location is to be determined at run time.
COLLID	VARCHAR(128) NOT NULL Collection name for the package. (*) indicates collection name determined at runtime.
NAME	VARCHAR(128) NOT NULL Name of the package. (*) means the entire collection.
TIMESTAMP	TIMESTAMP NOT NULL Timestamp when row was created.
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other.

SYSIBM.SYSPACKSTMT		(DSNDB06.SYSTSPKS)
SYSIBM.DSNKSX01	U	(LOCATION,COLLID,NAME, CONTOKEN,SEQNO)
Contains one or more rows for every SQL statement belonging to a package. Rows where SEQNO, STMTNO, and SECTNO are zero are for internal use.		

This table contains the following columns:

Column Name	Data Type/Description
LOCATION	VARCHAR(128) NOT NULL Always contains blanks.
COLLID	VARCHAR(128) NOT NULL Name of the collection.
NAME	VARCHAR(128) NOT NULL Name of the package (DBRM).
CONTOKEN	CHAR(8) NOT NULL WITH DEFAULT FOR BIT DATA Consistency token for the package from a DBRM: the level as specified by the LEVEL option at precompile or the timestamp indicating when package program was precompiled.
SEQNO	INTEGER NOT NULL/Not used.
STMTNO	SMALLINT NOT NULL Statement number of corresponding statement in the package source program. Statement nos. > 32767 displayed as zero or a negative number, see STMTNOI for statement number. To convert a negative statement number, add 65536 to it.
SECTNO	SMALLINT NOT NULL Section number of the statement.
BINDERROR	CHAR(1) NOT NULL Whether an SQL error was detected at bind: Y (Yes) or N (No).
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other.
VERSION	VARCHAR(122) NOT NULL Version identifier for the package.
STMT	VARCHAR(3500) NOT NULL WITH DEFAULT FOR BIT DATA IBM internal use only
ISOLATION	CHAR(1) NOT NULL WITH DEFAULT Isolation level for the SQL statement specified in the WITH clause, or blank if WITH was not specified at the statement level (in which case, the isolation level defaults to that of the package or plan): • R—RR (repeatable read) • T—RS (read stability) • S—CS (cursor stability) • U—UR (uncommitted read) • L—RS isolation, with a lock clause • X—RR isolation, with a lock clause • blank—WITH clause not specified on statement. The isolation level is recorded in SYSPACKAGE.ISOLATION and in SYSPLAN.ISOLATION.

Column Name	Data Type/Description
STATUS	CHAR(1) NOT NULL WITH DEFAULT The status of the binding statement: • A—Distributed; uses DB2 private protocol. Statement is parsed and executed at the server using defaults for input variables. • B—Distributed; uses DB2 private protocol. Statement will be parsed and executed using input values. • C—Compiled; statement bound using defaults for input values. • D—Distributed – statement references a remote object using DB2 private protocol access. • E—Explain; statement is an EXPLAIN done at bind time using defaults for input values. • F—Parsed; statement did not successfully bind. The statement will be rebound at execution time using values for input variables during access path selection. • G—Compiled; statements bound successfully. REOPT was specified to rebind statement at execution using input values. • H—Parsed; statement is DDL or another SQL statement that did not bind successfully. Rebind will occur at execution using default input values. • I—Indefinite; statement is dynamic bound at execution using defaults for input values. • J—Indefinite; statement is dynamic bound at execution using input values. • K—Control; CALL statement • L—Bad; statement is in error, bind continues, but statement cannot be executed. • M—Parsed – statement references a table that is qualified with SESSION and was not bound because the table reference could be for a declared temporary table. • blank—Statement is non-executed or bound prior to Version 5.
ACCESSPATH	CHAR(1) NOT NULL WITH DEFAULT For static statements, indicates if the access path for the statement is based on user-specified optimization hints. For static statement indicates if access path was determined using hints. • H—Hints used. • blank—Hints not used or no access path associated with statement For dynamic statements, the value is blank.
STMTNOI	INTEGER NOT NULL WITH DEFAULT Statement number of corresponding

Column Name	Data Type/Description
	statement in package source program.
SECTNOI	INTEGER NOT NULL WITH DEFAULT Section number of statement.
EXPLAINABLE	CHAR(1) NOT NULL WITH DEFAULT Contains one of the following values: • Y—Indicates that the SQL statement can be used with the EXPLAIN function and may have rows describing its access path in the userid.PLAN_TABLE. • N—Indicates that the SQL statement does not have any rows describing its access path in the userid.PLAN_TABLE. • blank—Indicates that the SQL statement was bound prior to V7.
QUERYNO	INTEGER NOT NULL WITH DEFAULT -1 The query number of the SQL statement in the source program. SQL statements bound prior to V7 have a default value of -1. Statements bound in V7 or later use the value specified on the QUERYNO clause on SELECT, UPDATE, INSERT, DELETE, EXPLAIN, DECLARE CURSOR, or REFRESH TABLE statements. If the QUERYNO clause is not specified, the query number is set to the statement number.
ROWID	ROWID NULL GENERATED ALWAYS LOB ROW identifier.
STMT_ID	BIGINT NOT NULL Statement ID
STATEMENT	CLOB(2M) NOT NULL WITH DEFAULT INLINE LENGTH (15360) SQL statement
STMTBLOB	BLOB(2M) NOT NULL WITH DEFAULT
EXPANSION_REASON	INLINE LENGTH(7168) IBM internal use only CHAR(2) NOT NULL Used for static SQL statements referencing temporal or archive tables: • A—Statement uses SYSIBMADM.GET_ARCHIVE • B—Statement uses CURRENT TEMPORAL BUSINESS_TIME • S—Statement uses CURRENT TEMPORAL SYSTEM_TIME • SB—Statement uses CURRENT TEMPORAL SYSTEM_TIME and BUSINESS_TIME • blank—Dynamic statements or VALIDATE(RUN) was unsuccessful or statement bound without implicit query transformation.
QUERYID	BIGINT The unique identifier for locating records in the SYSIBM.SYSQUERY catalog table. The default value is -1 indicates that no QUERYID value was found for the SQL statement when the package was bound.
QUERY_HASH	CHAR(16) FOR BIT DATA

Column Name	Data Type/Description
	The hash key for locating records in the SYSIBM.SYSQUERY catalog table. The 0x0 default value indicates that no hash key was generated for the SQL statement when the package was bound.
QUERY_HASH_VERSION	INTEGER The hash version for locating records in the SYSIBM.SYSQUERY catalog table. The -1 default value indicates that no has version was generated for the SQL statement when the package was bound.

SYSIBM.SYSPACKSTMT_STMB		**(DSNDB06.SYSTSPVR)**
SYSIBM.DSNKSX02	**U**	**(AUXID,AUXVER)**
Auxiliary table for STMTBLOB LOB column of SYSPACKSTMT.		

This table contains the following column:

Column Name	Data Type	Description
AUXID	VARCHAR(17)	ID of auxiliary data.
AUXVER	SMALLINT	Version of auxiliary data.
AUXVALUE	BLOB(2M) NOT NULL WITH DEFAULT	IBM internal use.

SYSIBM.SYSPACKSTMT_STMT		**(DSNDB06.SYSTSPKX)**
SYSIBM.DSNPKX01	**U**	**(AUXID,AUXVER)**
Auxiliary table for STATEMENT LOB of SYSPACKSTMT.		

This table contains the following column:

Column Name	Data Type	Description
AUXID	VARCHAR(17)	ID of auxiliary data.
AUXVER	SMALLINT	Version of auxiliary data.
AUXVALUE	BLOB(2M) NOT NULL WITH DEFAULT	Complete text for SQL statement.

SYSIBM.SYSPARMS		**(DSNDB06.SYSTSPRM)**
SYSIBM.DSNOPX01	**U**	**(SCHEMA,SPECIFICNAME, ROUTINETYPE,ROWTYPE, ORDINAL_VERSION)**
SYSIBM.DSNOPX02	**D**	**(TYPESCHEMA,TYPENAME, ROUTINETYPE, CAST_FUNCTION, OWNER_SCHEMA, SPECIFICNAME)**
SYSIBM.DSNOPX03	**D**	**(TYPESCHEMA,TYPENAME)**
SYSIBM.DSNOPX04	**D**	**(SCHEMA,SPECIFICNAME, ROUTINETYPE,VERSION)**
Contains information for each parameter of a routine or multiple rows for table parameters. One row for each column of the table.		

This table contains the following columns:

Column Name	Data Type/Description
SCHEMA	VARCHAR(128) NOT NULL Routine schema.
OWNER	VARCHAR(128) NOT NULL Routine owner.
NAME	VARCHAR(128) NOT NULL

Column Name	Data Type/Description
	Routine name.
SPECIFICNAME	VARCHAR(128) NOT NULL Routine specific name.
ROUTINETYPE	CHAR(1) NOT NULL Type of routine: • F—User-defined function or cast function • P—Stored procedure
CAST_FUNCTION	CHAR(1) NOT NULL Is routine a cast function: Y/N
PARMNAME	VARCHAR(128) NOT NULL Parameter name
ROUTINEID	INTEGER NOT NULL Internal identifier of routine.
ROWTYPE	CHAR(1) NOT NULL Parameter type: • P—Input parm • O—Output parm • B—Both input and output parm • R—Result before casting • C—Result after casting • S—For input parms to user-defined functions sourced on a built-in function • X—Indicates that the row is not used to describe a particular parameter of the routine
ORDINAL	SMALLINT NOT NULL ROWTYPE = B, O, P, or S value is the number of the parameter in the routine signature. ROWTYPE = C or R value is 0. ROWTYPE = S with parameter CCSID specified, value is 0. ROWTYPE = X, value is 0.
TYPESCHEMA	VARCHAR(128) NOT NULL Schema of data type of parameter.
TYPENAME	VARCHAR(128) NOT NULL Name of data type of parameter.
DATATYPEID	INTEGER NOT NULL Internal identifier of data type of parameter.
SOURCETYPEID	INTEGER NOT NULL Internal identifier of source type. Value is 0 for built-in data type.
LOCATOR	CHAR(1) NOT NULL Indicates locator value, not actual value, is passed as input when routine is called: Y (locator value) or N (actual value).
TABLE	CHAR(1) NOT NULL Data type of column for table parameter: Y (table parameter)/N (not table parameter).
TABLE_COLNO	SMALLINT NOT NULL Column number of table for table parameters. Otherwise 0.
LENGTH	INTEGER NOT NULL Length of parameter, for DECIMAL value is precision.
SCALE	SMALLINT NOT NULL Scale of data type of parameter, or

Column Name	Data Type/Description
	number of fractional seconds in a timestamp.
SUBTYPE	CHAR(1) NOT NULL Subtype for distinct type, based on subtype of source type: B FOR BIT DATA S FOR SBCS DATA M FOR MIXED DATA blank Source type is an array type or not character type
CCSID	INTEGER NOT NULL CCSID of data type for character, graphic, date, time, and timestamp data types. When ROWTYPE = X and ORDINAL = 0, the CCSID column is the CCSID for all character and graphic string parameters. Value is 0 for datetime array.
CAST_FUNCTION_ID	INTEGER NOT NULL Internal identifier of function used to cast argument if function is sourced on another function or result. Otherwise value is 0. Not applicable for stored procedures.
ENCODING_SCHEME	CHAR(1) NOT NULL Encoding scheme of parameter: • A—ASCII • E—EBCDIC • U—Unicode • blank—Source type either array or is not character, graphic, or datetime type
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other.
VERSION	VARCHAR(122) NOT NULL WITH DEFAULT Version identifier for the routine. If value of ORIGIN is not 'I' or rows created prior to V9, column is zero length string.
OWNERTYPE	CHAR(1) NOT NULL WITH DEFAULT Indicates the type of owner: blank Authorization ID L Role

SYSIBM.SYSPENDINGDDL		**(DSNDB06.SYSTSPEN)**
SYSIBM.DSNPDX01	**D**	**(DBNAME,TSNAME, CREATEDTS,OPTION_SEQNO)**
SYSIBM.DSNPDX02	**D**	**(OBJSCHEMA,OBJNAME,OBJTYPE , CREATEDTS,OPTION_SEQNO)**
Contains information about which objects have pending definition changes. Entries only exist between when the pending option is executed and when the utility applies these pending changes.		

This table contains the following columns:

Column Name	Data Type/Description
DBNAME	VARCHAR(24) NOT NULL Database name

Column Name	Data Type/Description
TSNAME	VARCHAR(24) NOT NULL Table space name
DBID	SMALLINT NOT NULL Database ID.
PSID	SMALLINT NOT NULL Table space page set descriptor internal ID.
OBJSCHEMA	VARCHAR(128) NOT NULL Object qualifier.
OBJNAME	VARCHAR(128) NOT NULL Object name.
OBJOBID	SMALLINT NOT NULL Object internal ID.
OBJTYPE	CHAR(1) NOT NULL Object type: I – Index S – Table space T – Table
STATEMENT_TYPE	CHAR(1) NOT NULL Statement type: A – Alter R – Recover
OPTION_ENVID	INTEGER NOT NULL Environment internal ID
OPTION_KEYWORD	VARCHAR(128) NOT NULL If ALTER operation then the name of the pending change. If RECOVER operation then the value of RECOVER option.
OPTION_VALUE	VARCHAR(4000) NOT NULL If ALTER operation then the name of the pending change. If RECOVER operation then the value of RECOVER option.
OPTION_SEQNO	SMALLINT NOT NULL Pending option sequence within statement
CREATEDTS	TIMESTAMP(12) NOT NULL Pending option creation time.
RELCREATED	CHAR(1) NOT NULL Object creation DB2 release: (see Release dependency indicators.)
IBMREQD	CHAR(1) NOT NULL Source of row: Y – Machine-readable material(MRM) tape (else – see Release dependency indicators.)
ROWID	ROWID/LOB column ID.
STATEMENT_TEXT	CLOB(2M) NOT NULL Original statement source text.
COLNAME	VARCHAR(128) NOT NULL WITH DEFAULT Column name
PARTITION	SMALLINT NOT NULL Partition number
PARTITION_KEYWORD	VARCHAR(18) NOT NULL WITH DEFAULT Blank or ALTER

Column Name	Data Type/Description
COLUMN_KEYWORD	VARCHAR(18) NOT NULL WITH DEFAULT Value 'DROP' if the column must be dropped.
REORG_SCOPE_ LOWPART	SMALLINT Logical partition number of the lowest partition in the range for REORG to materialize pending changes. Adjacent logical partitions must be reorganized together to materialize pending definition changes. The value is 0 if the range is the entire partitioned table space or index space, or if the record is generated by the RECOVER utility. NULL when the value is unknown for pending definition changes executed prior to DB2 12.
REORG_SCOPE_ HIGHPART	SMALLINT Logical partition number of the highest partition in the range for REORG to materialize pending changes. Adjacent logical partitions must be reorganized together to materialize pending definition changes. The value is 0 if the range is the entire partitioned table space or index space, or if the record is generated by the RECOVER utility. NULL when the value is unknown for pending definition changes executed prior to DB2 12.

SYSIBM.SYSPENDINGDDLTEXT		**(DSNDB06.SYSTSPDT)**
SYSIBM.DSNPDX03	**U**	**(AUXID,AUXVER)**
Auxiliary table for STATEMENT_TEXT LOB column of SYSPENDINGDDL table.		

This table contains the following column:

Column Name	Data Type	Description
AUXID	VARCHAR(17)	ID of auxiliary data.
AUXVER	SMALLINT	Version of auxiliary data.
AUXVALUE	BLOB(1G) NOT NULL WITH DEFAULT	IBM internal use.

SYSIBM.SYSPENDINGOBJECTS		(DSNDB06.SYSTSPDO)
SYSIBM.DSNPOX01	D	(DBNAME,TSNAME, PARTITION,COLNAME)
SYSIBM.DSNPOX02	D	(OBJSCHEMA,OBJNAME, OBJTYPE)
SYSIBM.DSNPOX03	D	(DNAME,INDEXSPACE)
Contains name and OBID object information about objects that are pending creation. Entries only exist after their names have been created, and before they have been materialized.		

This table contains the following columns:

Column Name	Data Type/Description
DBNAME	VARCHAR(24) NOT NULL Database name
TSNAME	VARCHAR(24) NOT NULL Table space name
DBID	SMALLINT NOT NULL Database ID.
PSID	SMALLINT NOT NULL Table space page set descriptor internal ID.
PARTITION	SMALLINT NOT NULL Object target partition number.
COLNAME	VARCHAR(128) NOT NULL Column name in target base table space.
OBJSCHEMA	VARCHAR(128) NOT NULL Object qualifier.
OBJNAME	VARCHAR(128) NOT NULL Object name.
OBJTYPE	CHAR(1) NOT NULL Object type: I – Index S – Table space T – Table
INDEXSPACE	CHAR(8) NOT NULL Index space name. Empty string if object no an index.
OBJOBD	SMALLINT NOT NULL Object internal ID.
OBJPSID	SMALLINT NOT NULL Object page set descriptor internal ID. (0 – no page set descriptor)

SYSIBM.SYSPKSYSTEM		(DSNDB06.SYSTSPKY)
SYSIBM.DSNKYX01	D	(LOCATION,COLLID,NAME, CONTOKEN,SYSTEM,ENABLE)
Contains connection information to possible execution environments for each package.		

This table contains the following columns:

Column Name	Data Type/Description
LOCATION	VARCHAR(128) NOT NULL Blank.
COLLID	VARCHAR(128) NOT NULL Name of collection.
NAME	VARCHAR(128) NOT NULL Name of package.
CONTOKEN	CHAR(8) NOT NULL FOR BIT DATA Consistency token for the package from a DBRM:

Column Name	Data Type/Description
	the level as specified by the LEVEL option at precompile or the timestamp indicating when package program was precompiled.
SYSTEM	VARCHAR(24) NOT NULL Package environment: BATCH TSO batch CICS Customer Information Control System DB2CALL Call Attachment Facility DLIBATCH DLI batch support facility IMSBMP IMS BMP region IMSMPP IMS MPP or IFP region REMOTE remote application server
ENABLE	CHAR(1) NOT NULL Indicates if the connections represented by the row are enabled or disabled: Y/N.
CNAME	VARCHAR(60) NOT NULL Identifies connection or connections: If SYSTEM = BATCH or DB2CALL then blank. Blank, also, if row applies to all servers or connections for the environment. If SYSTEM = REMOTE, LU name for a database server, requester's location (if DB2), or requester's LUNAME. Otherwise, name of a single connection.
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other.

SYSIBM.SYSPLAN		**(DSNDB06.SYSTSPLN)**
SYSIBM.DSNPPH01	**P**	**(NAME)**
Contains information about each application plan. One row per plan.		

This table contains the following columns:

Column Name	Data Type/Description
NAME	VARCHAR(24) NOT NULL Name of the application plan.
CREATOR	VARCHAR(128) NOT NULL Authorization ID of owner of application plan.
BINDDATE	CHAR(6)/Unused.
VALIDATE	CHAR(1) NOT NULL Whether validity checking can be deferred until run time: • B—All checking must be performed during BIND • R—Checking deferred to run time if tables, views, or privileges do not exist at bind time.
ISOLATION	CHAR(1) NOT NULL Isolation level: • R—Repeatable read (RR) • T—Read stability (RS) • S—Cursor stability (CS) • U—Uncommitted read (UR)
VALID	CHAR(1) NOT NULL Whether plan is valid (can be run without rebinding): • N—No

Column Name	Data Type/Description
	• Y—Yes • A—Table has been altered, but no rebinding is needed • H—Table has been altered invalidating the plan for releases of DB2 prior to V5.
OPERATIVE	CHAR(1) NOT NULL Whether plan can be allocated: • N—No, explicit BIND or REBIND required • Y—Yes
BINDTIME	CHAR(8)/Unused.
PLSIZE	INTEGER NOT NULL Size of the base section of the plan in bytes.
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other.
AVGSIZE	INTEGER NOT NULL Average size of plan sections containing SQL statements processed at bind time.
ACQUIRE	CHAR(1) NOT NULL When resources are acquired: • A—at allocation • U—at first use
RELEASE	CHAR(1) NOT NULL When resources are released: • C—at commit • D—at deallocation
EXREFFERENCE	CHAR(1)/Unused.
EXSTRUCTURE	CHAR(1)/Unused.
EXCOST	CHAR(1)/Unused.
EXPLAN	CHAR(1) NOT NULL Whether plan was bound with EXPLAIN YES: Y/N.
EXPREDICATE	CHAR(1) NOT NULL CURRENTDATA option when package was bound: • B—Data currency not required and blocking allowed for ambiguous cursors. • C—Data currency required and blocking not allowed for ambiguous cursors. • N—Blocking not allowed for ambiguous cursors, plan created before CURRENTDATA option was available.
BOUNDBY	VARCHAR(128) NOT NULL WITH DEFAULT Primary authorization ID of binder of the plan.
QUALIFIER	VARCHAR(128) NOT NULL WITH DEFAULT Implicit qualifier for all unqualified table, view, index, and alias names in static SQL statements.
CACHESIZE	SMALLINT NOT NULL WITH DEFAULT Cache size in bytes. 0 indicates none used.
PLENTRIES	SMALLINT NOT NULL WITH DEFAULT Number of package list entries for this plan. Negative value indicates plan was bound after fall-back to a prior release.
DEFERPREP	CHAR(1) NOT NULL WITH DEFAULT Plan bound with DEFER(PREPARE): Y/N.
CURRENTSERVER	VARCHAR(128) NOT NULL WITH DEFAULT Current server specification when plan was

Column Name	Data Type/Description
	last bound. Blank if none was specified.
SYSENTRIES	SMALLINT NOT NULL WITH DEFAULT Number of enabled or disabled entries or rows in SYSPLSYSTEM for this plan. A negative of that number means the plan was bound after fall-back to a prior release. A negative value or zero means all connections are enabled.
DEGREE	CHAR(3) NOT NULL WITH DEFAULT The DEGREE option used when package last bound: • ANY—DEGREE(ANY) • DEGREE(1) • blank—a migrated package
SQLRULES	CHAR(1) NOT NULL WITH DEFAULT The SQLRULES option used when plan last bound: • D or blank—SQLRULES(DB2) • S—SQLRULES(STD) • blank—a migrated plan
DISCONNECT	CHAR(1) NOT NULL WITH DEFAULT The DISCONNECT option used when plan last bound: • E or blank—DISCONNECT(EXPLICLT) • A—DISCONNECT (AUTOMATIC) • C—DISCONNECT (CONDITIONAL) • blank—a migrated plan
GROUP_MEMBER	VARCHAR(24) NOT NULL WITH DEFAULT The DB2 data sharing member name of the DB2 subsystem that performed the most recent bind. This is blank if the DB2 subsystem was not part of a DB2 data sharing environment.
DYNAMICRULES	CHAR(1) NOT NULL WITH DEFAULT The DYNAMICRULES option used when the plan was last bound: • B—Bind. Dynamic SQL statements are executed with DYNAMIC RULES bind behavior. • Blank—Run. Dynamic SQL statements in the plan are executed with DYNAMICRULES run behavior.
BOUNDTS	TIMESTAMP NOT NULL WITH DEFAULT Timestamp the plan was bound.
REOPTVAR	CHAR(1) NOT NULL WITH DEFAULT 'N' Access path is determined again at execution time. • A—Bind option REOPT(AUTO), access path determined multiple times at execution. • N—Bind option REOPT(NONE) access path is determined at bind. • Y—Bind option REOPT (ALWAYS) access path is determined again at execution for SQL statements with variable values. • 1—Bind option REOPT (ONCE) access path is determined only once at execution time.

Column Name	Data Type/Description
KEEPDYNAMIC	CHAR(1) NOT NULL WITH DEFAULT 'N' Purging of dynamically prepared statements at commit. • N—Bind option KEEPDYNAMIC (NO): statements are purged. • Y—Bind option KEEPDYNAMIC (YES): statements are kept past commit, or rollback.
PATHSCHEMAS	VARCHAR(2048) NOT NULL WITH DEFAULT SQL path specified on BIND or REBIND. Used to resolve unqualified data type, function and stored procedures. If PATH bind option not specified, value has zero length and DB2 uses default of SYSIBM, SYSFUNC, SYSPROC, or plan qualifier.
DBPROTOCOL	CHAR(1) NOT NULL WITH DEFAULT 'P' Whether remote access for SQL with three-part names is implemented with DRDA or DB2 private: D (DRDA) or P (DB2 private protocol).
FUNCTIONTS	TIMESTAMP NOT NULL WITH DEFAULT Timestamp when function was resolved. Set by BIND and REBIND, not by AUTOBIND.
OPTHINT	VARCHAR(128) NOT NULL WITH DEFAULT Value of OPTHINT bind option, identifying rows from authid. PLAN_TABLE to be used as input to the optimizer. Blank if no rows input to optimizer.
ENCODING_CCSID	INTEGER NOT NULL WITH DEFAULT The CCSID corresponding to the encoding scheme or CCSID as specified for the bind option ENCODING. The Encoding Scheme specified on the bind command: • ccsid—The specified or derived CCSID. • 0—The default CCSID as specified on panel DSNTIPF at installation time. Used when the plan was bound prior to V7.
IMMEDWRITE	CHAR(1) NOT NULL WITH DEFAULT For data sharing environments, this indicates when writes of updated group buffer pool dependent pages are to be done. • N— Bind option IMMEDWRITE(NO) indicates normal write activity is done. • Y—Bind option IMMEDWRITE(YES) indicates that immediate writes are done for updated group buffer pool pages. • 1—Bind option IMMEDWRITE(PH1) indicates that updated group buffer pool dependent pages are written at or before phase one commit. • blank—A migrated package.
RELBOUND	CHAR(1) NOT NULL WITH DEFAULT The release when the package was bound or rebound. • Blank—Bound prior to V7. • K—Bound on V7. • L—Bound on V8.

Column Name	Data Type/Description
CATENCODE	CHAR(1)/Unused.
REMARKS	VARCHAR(762) NOT NULL WITH DEFAULT A character string provided by the user with the COMMENT statement.
CREATORTYPE	CHAR(1) NOT NULL WITH DEFAULT Indicates the type of creator: • blank—Authorization ID • L—Role
ROUNDING	CHAR(1) NOT NULL WITH DEFAULT The ROUNDING option used when the plan was last bound: • C —ROUND_CEILING • D—ROUND_DOWN • F—ROUND_FLOOR • G—ROUND_HALF_DOWN • E—ROUND_HALF_EVEN • H—ROUND_HALF_UP • U—ROUND_UP • blank—The plan was created in a DB2 release prior to V9.
LASTUSED	DATE NOT NULL WITH DEFAULT Last date objects were used.
CONCUR_ACC_RES	CHAR(1) NOT NULL CONCURRENTACCESSRESOLUTION when package bound or rebound • blank – Not specified • W – WAITFOROUTCOME • U – USECURRENTLYCOMMITTED
PROGAUTH	CHAR(1) NOT NULL WITH DEFAULT 'D' Should DB2 check if program is authorized to execute plan: D (DISABLE) or E (ENABLE).

SYSIBM.SYSPLANAUTH		**(DSNDB06.SYSTSPLA)**
SYSIBM.DSNAPH01	**D**	**(GRANTEE,NAME, EXECUTEAUTH,GRANTEETYPE)**
SYSIBM.DSNAPX01	**D**	**(GRANTOR,GRANTORTYPE)**
SYSIBM.DSNAPX02	**D**	**(NAME)**
Records the privileges held by users over application plans.		

This table contains the following columns:

Column Name	Data Type/Description
GRANTOR	VARCHAR(128) NOT NULL Authorization ID of user who granted the privileges.
GRANTEE	VARCHAR(128) NOT NULL Authorization ID of user who holds the privileges. Could be PUBLIC, for a grant to PUBLIC.
NAME	VARCHAR(24) NOT NULL Name of the application plan on which the privileges are held.
TIMESTAMP	CHAR(12) NOT NULL/Internal use only.
DATEGRANTED	CHAR(6)/Unused.
TIMEGRANTED	CHAR(8)/Unused.
GRANTEETYPE	CHAR(1) Indicates the type of grantee: • blank—Authorization ID

Column Name	Data Type/Description
	• L—Role
AUTHHOWGOT	CHAR(1) NOT NULL Authorization level of user from whom privileges were received: • Blank—not applicable • C—DBCTL • D—DBADM • E—SECADM • G—ACCESSCTRL • L—SYSCTRL • M—DBMAINT • S—SYSADM
BINDAUTH	CHAR(1) NOT NULL Whether GRANTEE can use BIND, REBIND, or FREE against the plan: • blank—privilege not held • G—privilege held with GRANT option • Y—privilege held without GRANT option
EXECUTEAUTH	CHAR(1) NOT NULL Whether GRANTEE can run programs that use the plan: • blank—privilege not held • G—privilege held with GRANT option • Y—privilege held without GRANT option
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other.
GRANTEDTS	TIMESTAMP NOT NULL WITH DEFAULT Time GRANT statement was executed.
GRANTORTYPE	CHAR(1) NOT NULL WITH DEFAULT Indicates the type of grantor: • blank—Authorization ID • L—Role
SYS_START	TIMESTAMP(12) NOT NULL System period temporal start time for transaction
SYS_END	TIMESTAMP(12) NOT NULL System period temporal end time for transaction
TRANS_START	TIMESTAMP(12) System period transaction timestamp.

SYSIBM.SYSPLANDEP		**(DSNDB06.SYSTSPLD)**
SYSIBM.DSNGGX01	**D**	**(BCREATOR,BNAME,BTYPE)**
SYSIBM.DSNGGX05	**D**	**(DNAME)**
Contains the dependencies of plans on tables, views, synonyms, table spaces, indexes, aliases, functions and stored procedures.		

This table contains the following columns:

Column Name	Data Type/Description
BNAME	VARCHAR(128) NOT NULL Name of object the plan depends on.
BCREATOR	VARCHAR(128) NOT NULL If BNAME is a table space, its database. If BNAME is a role, the value is blank Otherwise, schema of the BNAME
BTYPE	CHAR(1) NOT NULL Type of object BNAME:

Column Name	Data Type/Description
	• A—alias • E—Instead of trigger • F—user-defined function or cast function • G—Global temporary table • I—index • M—Materialized query table • O—stored procedure • P—partitioned table space • Q—Sequence object • R—table space • S—synonym • T—table • V—view
DNAME	VARCHAR(24) NOT NULL Name of the plan.
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other. If other, release dependency indicator.

SYSIBM.SYSPLSYSTEM		**(DSNDB06.SYSTSPLY)**
SYSIBM.DSNKPX01	**D**	**(NAME,SYSTEM,ENABLE)**
Contains connection information to possible execution environments for each plan.		

This table contains the following columns:

Column Name	Data Type/Description
NAME	VARCHAR(24) NOT NULL Plan name.
SYSTEM	VARCHAR(24) NOT NULL System environment: BATCH TSO batch DB2CALL DB2 call attachment facility CICS Customer Information Control System DLIBATCH DLI batch support facility IMSBMP IMS BMP region IMSMPP IMS MPP or IFP region
ENABLE	CHAR(1) NOT NULL Whether connections are enabled: Y/N.
CNAME	VARCHAR(60) NOT NULL Identifies the connection to which the row applies. Values can be: • Blank—if SYSTEM=BATCH or =DB2CALL • The name of a single connection if SYSTEM has any other value
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other.

SYSIBM.SYSPROFILE_TEXT		(DSNDB06.SYSTSPTX)
SYSIBM.DSNPRX02	U	(AUXID,AUXVER)
Auxiliary table for PROFILE_TEXT LOB column of SYSTABLES_PROFILES.		

This table contains the following column:

Column Name	Data Type	Description
AUXID	VARCHAR(17)	ID of auxiliary data.
AUXVER	SMALLINT	Version of auxiliary data.
AUXVALUE	BLOB(5M)	IBM internal use.

SYSIBM.SYSQUERY		(DSNDB06.SYSTSQRY)
SYSIBM.DSNQYX01	D	(QUERY_HASH,SCHEMA, SOURCE,QUERY_SEC_HASH)
SYSIBM.DSNQYX02	P	(QUERYID)
SYSIBM.DSNQYX03	D	(LOCATION,COLLECTION, PACKAGE,VERSION,SECTNO)
SYSIBM.DSNQYX04	D	(QUERY_SEC_HASH,SCHEMA, SOURCE)
Contains information for each query in a set of queries. One row per query.		

This table contains the following columns:

Column Name	Data Type/Description
QUERYID	BIGINT NOT NULL GENERATED BY DEFAULT AS IDENTITY Query ID
QUERY_HASH	CHAR(16) NOT NULL FOR BIT DATA Hash key.
SCHEMA	VARCHAR(128) NOT NULL Default schema name
QUERY_SEC_HASH	CHAR(16) NOT NULL FOR BIT DATA Hash key generated by modified statement text.
QUERY_HASH_ VERSION	INTEGER NOT NULL Query hash version
SOURCE	SMALLINT NOT NULL Query source: 0—Statement-level optimization hints.
USERFILTER	CHAR(8) NOT NULL Set of queries filter name or blank.
OTHER_OPTION	CHAR(128) NOT NULL Internal use only
PLAN_VALID	CHAR(1) NOT NULL Whether plan hints are valid: • Blank—No plan hints (optimization parameters exist in SYSQUERYOPTS • Y—Valid plan hint exists in SYSQUERYPLAN • N—Plan hint in SYSQUERYPLAN is invalid.
INVALID_REASON	INTEGER NOT NULL Reason code when plan invalid (PLAN_VALID = N) (-1 when PLAN_VALID = Y)
LOCATION	VARCHAR(128) NOT NULL/Not used.
COLLECTION	VARCHAR(128) NOT NULL

Column Name	Data Type/Description
	Originating query collection name or blank.
PACKAGE	VARCHAR(128) NOT NULL Originating query package name or blank
VERSION	VARCHAR(128) NOT NULL Package version or blank
AUTHID	VARCHAR(128) NOT NULL Authorization ID in effect when query was captured or blank.
BINDTIME	TIMESTAMP NOT NULL Time package was bound, or BIND QUERY was run.
RELBOUND	CHAR(1) NOT NULL DB2 release when package was bound or blank (See Release dependency indicators for values).
IBMREQD	CHAR(1) NOT NULL Row source: Y – Machine-readable material(MRM) Else see Release dependency indicators
STMTNO	INTEGER NOT NULL Statement number in package when SOURCE = 1 -1 when SOURCE = 0 or 2
SECTNO	INTEGER NOT NULL Section number in package when SOURCE = 1 -1 when SOURCE = 0 or 2
STMTTEXT	CLOB(2M) INLINE LENGTH 2048 Revised SQL statement
QUERYNO	INTEGER NOT NULL WITH DEFAULT '-1' Query number
CLIENT_USERID	VARCHAR(255) User ID of the client.
CLIENT_ WRKSTNNAME	VARCHAR(255) Client workstation name.
CLIENT_APPLNAME	VARCHAR(255) Client application name.
DB2_GENERATED_ROWID_ FOR_LOBS	ROWID Generated row identifier
SELECTVTY_ OVERRIDE	CHAR(1) NOT NULL Are SELECTIVITY overrides in effect for query: Y/N.
ACCESSPATH_HINT	CHAR(1) NOT NULL Access paths specified for matching statements: • Y—access path specified • N—access path hint specified • blank—access path might be specified but look at SYSIBM.SYSQUERYPLAN to be sure.
OPTION_OVERRIDE	CHAR(1) NOT NULL Whether optimization parameters are in effect for matching statements: Y/N/blank. If blank,

Column Name	Data Type/Description
	perhaps but look at SYSIBM.SYSQUERYOPTS to be sure.
SELECTIVITY_VALID	CHAR(1) NOT NULL Are selectivity overrides valid: • Y—yes. • N—no but they exist.
FUNCTION_LVL	VARCHAR(10) NOT NULL WITH DEFAULT The function level when the row was inserted.

SYSIBM.SYSQUERYOPTS		**(DSNDB06.SYSTSQRO)**
SYSIBM.DSNQPX01	**D**	**(QUERYID,COPYID)**
Contains optimization parameters for each query in SYSQUERY.		

This table contains the following columns:

Column Name	Data Type/Description
QUERYID	BIGINT NOT NULL ON DELETE CASCADE Query unique ID.
COPYID	SMALLINT NOT NULL Plan hint version • 0 – Current version • 1 – Previous version • 2 – Original version
REOPT	CHAR(1) NOT NULL REOPT bind option in effect for plan. • 1 – ONCE • A – AUTO • N – NONE • Y – ALWAYS • Blank – REOPT not specified
STARJOIN	CHAR(1) NOT NULL Star join enabled: • Y—Yes • N—No • blank—Star join not specified
MAX_PAR_DEGREE	INTEGER NOT NULL Maximum parallel degree -1 – Maximum parallel degree not specified
DEF_CURR_DEGREE	CHAR(3) NOT NULL Parallelism enabled: ONE – Disabled ANY – Enabled Blank - Disabled
SJTABLES	INTEGER NOT NULL Number of tables qualifying for star join processing -1 – star join not specified
OTHER_PARMS	VARCHAR(128) NOT NULL IBM internal use only
GROUP_MEMBER	VARCHAR(24) NOT NULL Applicable group member name blank – not specified
IBMREQD	CHAR(1) NOT NULL Row source: Y – Machine-readable material(MRM) Else see Release dependency indicators

SYSIBM.SYSQUERYPLAN		(DSNDB06.SYSTSQRP)
SYSIBM.DSNQNX01	D	(QUERYID,COPYID)
Plan hint information for queries in SYSQUERY.		

This table contains the following columns:

Column Name	Data Type/Description
QUERYID	BIGINT NOT NULL ON DELETE CASCADE Query unique ID.
COPYID	SMALLINT NOT NULL Plan hint version • 0—Current version • 1—Previous version • 2—Original version
PLAN_VALID	CHAR(1) NOT NULL Plan valid: N (NO) or Y (YES).
IBMREQD	CHAR(1) NOT NULL Row source: Y – Machine-readable material(MRM) Else see Release dependency indicators
QBLOCKNO	SMALLINT NOT NULL Query block identifier within a query.
PLANNO	SMALLINT NOT NULL Execution order of the query block step.
METHOD	SMALLINT NOT NULL Join method for step: • 0—First table, continuation of previous table, or not used. • 1—Nested loop join • 2—Merge scan join • 3—Hybrid join
CREATOR	VARCHAR(128) NOT NULL Creator of new table. Blank if METHOD = 3
TNAME	VARCHAR(128) NOT NULL Name of: • Materialized query table • Temporary table • Materialized view • Materialized table expression
TABNO	SMALLINT NOT NULL IBM internal use only.
ACCESSTYPE	CHAR(2) NOT NULL New table access method: • A—Accelerated table access • DI—Intersection of multiple DOCIDs • DU—Union of multiple DOCIDs • DX—XML index scan returning multiple DOCIDs • E—Direct row access using row change timestamp column • H—Hash overflow index • I—Index access • IN—Index scan • I1—One fetch index scan • M—Multiple index scan (followed by MX, MI, MU, or MH) • MH—Hash overflow index named in ACCESSNAME • MI—Intersection of multiple indexes • MU—Union of multiple indexes

Column Name	Data Type/Description
	• MX—Index scan on index named in ACCESSNAME. • N—Index scan or hash access with IN keyword. • NR—Range list access • P—Dynamic pair-wise index scan • R—table space scan • RW—Work file scan of a materialized user-defined table function • V—Buffer for an INSERT statement within a SELECT • Blank—Not applicable
MATCHCOLS	SMALLINT NOT NULL For ACCESSTYPE I, I1, N, NR, MX, or DX, the number of index keys that are used in an index scan; otherwise, 0.
ACCESSCREATOR	VARCHAR(128) NOT NULL For ACCESSTYPE I, I1, N, NR, MX, or DX, the creator of the index; otherwise, blank.
ACCESSNAME	VARCHAR(128) NOT NULL For ACCESSTYPE I, I1, H, MH, N, NR, MX, or DX, the name of the index; for ACCESSTYPE P, DSNPJW(*mixopseqno*) is the starting pair-wise join leg in MIXOPSEQNO; otherwise, blank.
INDEXONLY	CHAR(1) NOT NULL Index only access: Y (yes) or N (no).
SORTN_UNIQ	CHAR(1) NOT NULL New table sort needed to remove duplicates: Y (yes) or N (no).
SORTN_JOIN	CHAR(1) NOT NULL New table sorted for join method 2 or 4: Y (yes) or N (no).
SORTN_ORDERBY	CHAR(1) NOT NULL New table sorted for ORDER BY: Y (yes) or N (no).
SORTN_GROUPBY	CHAR(1) NOT NULL New table sorted for GROUP BY: Y (yes) or N (no).
SORTC_UNIQ	CHAR(1) NOT NULL Composite table sort needed to remove duplicates: Y (yes) or N (no).
SORTC_JOIN	CHAR(1) NOT NULL Composite table sorted for join method 2 or 4: Y (yes) or N (no).
SORTC_ORDERBY	CHAR(1) NOT NULL Composite table sorted for ORDER BY: Y (yes) or N (no).
SORTC_GROUPBY	CHAR(1) NOT NULL Composite table sorted for GROUP BY: Y (yes) or N (no).

Column Name	Data Type/Description
TSLOCKMOD	CHAR(3) NOT NULL Lock mode acquired on new table, table space, or table space partition. Isolation determined at bind time: • IS—Intent share lock • IX—Intent exclusive lock • S—Share lock • U—Update lock • X—Exclusive lock • SIX—Share with intent exclusive lock • N—UR isolation; no lock Isolation determined at run time: • NS—UR isolation, no lock; for CS, RS, or RR, an S lock. • NIS—UR isolation, no lock; for CS, RS, or RR, an IS lock. • NSS—UR isolation, no lock; for CS or RS, an IS lock; for RR, an S lock. • SS—UR, CS, or RS isolation, an IS lock; for RR, an S lock.
PREFETCH	CHAR(1) NOT NULL Data pages read in advance by prefetch: • D—Optimizer expects dynamic prefetch • S—Pure sequential prefetch • L—Prefetch through a page list • blank—Unknown or no prefetch
COLUMN_FN_EVAL	CHAR(1) NOT NULL When SQL aggregate function is evaluated: • R—While the data is being read from the table or index • S—While performing a sort to satisfy a GROUP BY clause • blank—After data retrieval and after any sorts
MIXOPSEQ	SMALLINT NOT NULL Sequence number of step in a multiple index operation. • 1, 2, ... *n*—For the steps of the multiple index procedure (ACCESSTYPE is MX, MI, MU, DX, DI, or DU), or the sequence number of range list access (ACCESSTYPE is 'NR'). • 0—Any other rows.
ACCESS_DEGREE	SMALLINT Number of parallel tasks or operations: • 0—Host variable is used • NULL—plan table has fewer than 43 columns or method does not apply.
ACCESS_PGROUP_ID	SMALLINT Parallel group id for accessing new table.
JOIN_DEGREE	SMALLINT Number of parallel operations or tasks for joining composite table with new table. • 0—Host variable is used • NULL—plan table has fewer than 43 columns or method does not apply.
JOIN_PGROUP_ID	SMALLINT

Column Name	Data Type/Description
	Parallel group identifier for joining composite table with new table: NULL – plan table has fewer than 43 columns or method does not apply.
SORTC_PGROUP_ID	SMALLINT Parallel group identifier for parallel sort of composite table: NULL – plan table has fewer than 43 columns or method does not apply.
SORTN_PGROUP_ID	SMALLINT Parallel group identifier for parallel sort of new table: NULL – plan table has fewer than 43 columns or method does not apply.
PARALLELISM_ MODE	CHAR(1) Bind time parallelism: • I—Query I/O parallelism • C—Query CP parallelism • NULL—plan table has fewer than 43 columns or method does not apply.
MERGE_ JOIN_COLS	SMALLINT Number of merge scan join columns. NULL – plan table has fewer than 43 columns or method does not apply
CORRELATION_ NAME	VARCHAR(128) Correlation name of table or view: NULL – No correlation name exists, plan table has fewer than 43 columns or method does not apply.
PAGE_RANGE	CHAR(1) NOT NULL WITH DEFAULT Table qualifies for page range screening: Y (yes) or blank (no).
JOIN_TYPE	CHAR(1) NOT NULL WITH DEFAULT Type of join: • F—FULL OUTER JOIN • L—LEFT OUTER JOIN • P—Pair-wise join • S—Star join • blank—INNER JOIN or no join
QBLOCK_TYPE	CHAR(6) NOT NULL WITH DEFAULT Type of SQL operation performed: • SELECT—SELECT • INSERT—INSERT • UPDATE—UPDATE • MERGE—MERGE • DELETE—DELETE • SELUPD—SELECT with FOR UPDATE OF • DELCUR—DELETE WHERE CURRENT OF CURSOR • UPDCUR—UPDATE WHERE CURRENT OF CURSOR • CORSUB—Correlated subselect or fullselect • TRUNCA—TRUNCATE • NCOSUB—Noncorrelated subselect or fullselect • TABLEX—Table expression • TRIGGR—WHEN clause on CREATE TRIGGER

Column Name	Data Type/Description
	• UNION—UNION • UNIONA—UNION ALL • INTERS—INTERSECT • INTERA—INTERSECT ALL • EXCEPT—EXCEPT • EXCEPTA—EXCEPT ALL
PRIMARY_ ACCESSTYPE	CHAR(1) NOT NULL WITH DEFAULT Direct row access attempted first: • D—DB2 tries to use direct row access with a rowed column, else uses the access path that is described in the ACCESSTYPE column of PLAN_TABLE. • T—Base table or result file is materialized into a work file, and the work file is accessed via sparse index access. • blank—No direct row access.
PARENT_ QBLOCKNO	SMALLINT NOT NULL QBLOCKNO of parent query block.
TABLE_TYPE	CHAR(1) Type of table: • B—Buffers for SELECT from INSERT, SELECT from UPDATE, SELECT from MERGE, or SELECT from DELETE statement. • C—Common table expression • F—Table function • I—The new table is generated from an IN-LIST predicate. • M—Materialized query table • Q—Temporary intermediate result table (not materialized). For the name of a view or nested table expression, a value of Q indicates that the materialization was virtual and not actual. Materialization can be virtual when the view or nested table expression definition contains a UNION ALL that is not distributed. • R—Recursive common table expression • S—Subquery (correlated or non-correlated) • T—Table • W—Work file
TABLE_ENCODE	CHAR(1) Encoding scheme of table: • A—ASCII • E—EBCDIC • U—Unicode • M—The table contains multiple CCSID sets
TABLE_SCCSID	SMALLINT NOT NULL WITH DEFAULT SBCS CCSID value of the table. If column TABLE_ENCODE is M, the value is 0.
TABLE_MCCSID	SMALLINT NOT NULL WITH DEFAULT Mixed CCSID value of the table. If the value of TABLE_ENCODE = M, the value is 0. If MIXED=NO in the DSNHDECP module, the value is -2.

Column Name	Data Type/Description
TABLE_DCCSID	SMALLINT NOT NULL WITH DEFAULT DBCS CCSID value of the table. If the value of TABLE_ENCODE = M, the value is 0. If MIXED=NO in the DSNHDECP module, the value is -2.
ROUTINE_ID	INTEGER NOT NULL WITH DEFAULT IBM internal use only
CTEREF	SMALLINT NOT NULL WITH DEFAULT Top-level query block number for common table expression.
PARENT_PLANNO	SMALLINT NOT NULL Plan number in the parent query block where a correlated subquery is invoked. Or, for non-correlated subqueries, plan number in the parent query block that represents the work file for the subquery.
MERGC	CHAR(1)/IBM internal use only
MERGN	CHAR(1)/IBM internal use only
EXPANSION_ REASON	CHAR(2) NOT NULL Used for static SQL statements referencing temporal or achive tables: • Statement uses SYSIBMADM.GET_ARCHIVE • Statement uses CURRENT TEMPORAL BUSINESS_TIME • S—statement uses CURRENT TEMPORAL SYSTEM_TIME • SB—statement uses CURRENT TEMPORAL SYSTEM_TIME and BUSINESS_TIME • blank—dynamic statements or VALIDATE(RUN) was unsuccessful or statement bound without implicit query transformation.

SYSIBM.SYSQUERY_AUX		**(DSNDB06.SYSTSQRA)**
SYSIBM.DSNQSX01	**U**	**(AUXID,AUXVER)**
Auxiliary table for STMTTEXT column of SYSQUERY.		

This table contains the following column:

Column Name	Data Type	Description
AUXID	VARCHAR(17)	ID of auxiliary data.
AUXVER	SMALLINT	Version of auxiliary data.
AUXVALUE	CLOB(2M)	Full text of query.

SYSIBM.SYSQUERYPREDICATE		**(DSNDB06.SYSTSQRE)**
SYSIBM.DSNQEX01	**P**	**(QUERYID,PREDNO)**
SYSIBM.DSNQEX02	**D**	**(QUERYNO)**
Predicate information for SYSQUERY rows.		

This table contains the following columns:

Column Name	Data Type/Description
QUERYID	BIGINT Query identifier
QUERYNO	INTEGER NOT NULL Statement query number.

Column Name	Data Type/Description
	• If row created from EXPLAIN it's the QUERYNO specified. • If not from EXPLAIN, DB2 assigns number based SQL statement in source program. If value exeeds 32767 the value is 0.
QBLOCKNO	SMALLINT NOT NULL Query block identifier within a query.
APPLNAME	VARCHAR(24) NOT NULL Plan name
PROGNAME	VARCHAR(128) NOT NULL Package name
PREDNO	INTEGER NOT NULL Predicate number
TYPE	CHAR(8) NOT NULL Predicate operation: AND, OR, EQUAL, RANGE, BETWEEN, IN, LIKE, NOT LIKE, EXISTS, NOTEXISTS, SUBQUERY, HAVING, OTHERS
LEFT_HAND_SIDE	VARCHAR(128) NOT NULL If the left hand side of the predicate is a table-column (LHS_TABNO > 0) then the COLUMN NAME. Other possible values: • VALUE • COLEXP • NONCOLEXP • CORSUB • NONCORSUB • SUBQUERY • EXPRESSION • blanks
LEFT_HAND_PNO	INTEGER NOT NULL If the left hand side of the predicate is a table column (LHS_TABNO > 0), then the COLUMN NAME. Other possible values: • VALUE • COLEXP • NONCOLEXP • CORSUB • NONCORSUB • SUBQUERY • EXPRESSION • Blanks
LHS_TABNO	SMALLINT NOT NULL Unique number to identify table reference within a query when LHS is a table column.
LHS_QBNO	SMALLINT NOT NULL Unique number to identify table reference within a query when LHS is a table column.
RIGHT_HAND_SIDE	VARCHAR(128) NOT NULL If the left hand side of the predicate is a table-column (LHS_TABNO > 0) then the COLUMN NAME. Other possible values: • VALUE • COLEXP • NONCOLEXP • CORSUB • NONCORSUB • SUBQUERY • EXPRESSION

Column Name	Data Type/Description
	• Blanks
RIGHT_HAND_PNO	INTEGER NOT NULL If predicate is AND/OR this is the second child predicate. Use PARENT_PNO to reconstruct predicate tree.
RHS_TABNO	SMALLINT NOT NULL Unique number to identify table reference within a query when RHS is a table column.
RHS_QBNO	SMALLINT NOT NULL Unique number to identify table reference within a query when RHS is a table column.
FILTER_FACTOR	FLOAT NOT NULL Estimated filter facor
BOOLEAN_TERM	CHAR(1) NOT NULL Can this predicate be used to determine the true value of the entire where clause.
SEARCHARG	CHAR(1) NOT NULL Can predicate be processed by data manager – otherwise RDS needs to process.
JOIN	CHAR(1) NOT NULL Can predicate be used as simple join between two tables
AFTER_JOIN	CHAR(1) NOT NULL Predicate evaluation process: A—After join D—During join blank—n/a
ADDED_PRED	CHAR(1) NOT NULL Transitive closure used
REDUNDANT_PRED	CHAR(1) NOT NULL Is the predicate redundant
DIRECT_ACCESS	CHAR(1) NOT NULL Direct row access via ROWID
KEYFIELD	CHAR(1) NOT NULL Doe the predicate the index key column for all applicable indexes considered by DB2
EXPLAIN_TIME	TIMESTAMP NOT NULL EXPLAIN time. • For cached statements it is a full precisioned timestamp. • For non-cached static it is the BIND timestamp. • For non-cached dynamic it is the EXPLAIN timestamp equal to char(16) appended by four zeroes.
CATEGORY	SMALLINT NOT NULL/IBM Internal use.
CATEGORY_B	SMALLINT NOT NULL/IBM Internal use.
TEXT	VARCHAR(2000) NOT NULL Transformed predicate text up to 2000 byte – the rest is truncated.
PRED_ENCODE	CHAR(1) NOT NULL WITH DEFAULT IBM internal use
PRED_CCSID	SMALLINT NOT NULL WITH DEFAULT IBM internal use
PRED_MCCSID	SMALLINT NOT NULL WITH DEFAULT IBM internal use
MARKER	CHAR(1) NOT NULL WITH DEFAULT Does predicate hold parameter markers, host-variables or special registers
PARENT_PNO	INTEGER NOT NULL

Column Name	Data Type/Description
	Parent predicate number. Value is 0 for root predicate in query block.
NEGATION	CHAR(1) NOT NULL/Not used.
LITERALS	VARCHAR(128) NOT NULL Literal values separated by colon
CLAUSE	CHAR(8) NOT NULL The position of the predicate: HAVING, ON, WHERE or SELECT
GROUP_MEMBER	VARCHAR(24) NOT NULL Member name where EXPLAIN was executed otherwise blank
ORIGIN	CHAR(1) NOT NULL WITH DEFAULT • Blank – generated by DB2 • C - column mask • R - Row permission • U - user specified
UNCERTAINTY	FLOAT(4) NOT NULL WITH DEFAULT Uncertainty factor of estimated filter factor. Zero indicates no uncertainty or not considered.
SECTNOI	INTEGER NOT NULL WITH DEFAULT Section number of statement in SYSPACKSTMT.
COLLID	VARCHAR(128) NOT NULL WITH DEFAULT Collection-id: • DSNDYNAMICSQLCACHE: from dynamic statement cache. • DSNEXPLAINMODEYES: SET CURRENT EXPLAIN MODE YES used by application. • DSNEXPLAINMODEEXPLAIN: SET CURRENT EXPLAIN MODE EXPLAIN used by application.
VERSION	• VARCHAR(122) NOT NULL WITH DEFAULT • Package version identifier

SYSIBM.SYSQUERYSEL		**(DSNDB06.SYSTSQRS)**
SYSIBM.DSNQLX01	**P**	**(QUERYID,PREDNO,INSTANCE)**
SYSITEM.DSNQLX02	**D**	**(QUERYID,PREDNO)**
Information about predicate selectivity for queries in SYSQUERY.		

This table contains the following columns:

Column Name	Data Type/Description
QUERYID	BIGINT Query identifier
QUERYNO	INTEGER NOT NULL Statement query number. • If row created from EXPLAIN it's the QUERYNO specified. • If not from EXPLAIN, DB2 assigns number based SQL statement in source program. • If value exeeds 32767 the value is 0.
QBLOCKNO	SMALLINT NOT NULL Query block identifier within a query.
APPLNAME	VARCHAR(24) NOT NULL Plan name
PROGNAME	VARCHAR(128) NOT NULL Package name

Column Name	Data Type/Description
SECTNOI	INTEGER NOT NULL WITH DEFAULT Section number of statement in SYSPACKSTMT
COLLID	VARCHAR(128) NOT NULL WITH DEFAULT Collection-id: DSNDYNAMICSQLCACHE: from dynamic statement cache. DSNEXPLAINMODEYES: SET CURRENT EXPLAIN MODE YES used by application. DSNEXPLAINMODEEXPLAIN: SET CURRENT EXPLAIN MODE EXPLAIN used by application.
VERSION	VARCHAR(122) NOT NULL WITH DEFAULT Package version identifier
PREDNO	INTEGER NOT NULL Predicate number
INSTANCE	SMALLINT NOT NULL Selectivity instance to group related selectivities.
SELECTIVITY	FLOAT NOT NULL Selectivity of predicate.
WEIGHT	FLOAT(4) NOT NULL Weight of selectivity instance. A value of .025 means the predicate will have this selectivity 25% of the executions
ASSUMPTION	VARCHAR(128) NOT NULL How was selectivity estimated: NORMAL: during normal selectivity OVERRIDE : Optimizer input to override selectivity estimation.
ORIGIN	CHAR(1) NOT NULL WITH DEFAULT Blank – generated by DB2 C - column mask R - Row permission U - user specified
UNCERTAINTY	FLOAT(4) NOT NULL WITH DEFAULT Uncertainty factor of estimated filter factor. Zero indicates no uncertainty or not considered.
SECTNOI	INTEGER NOT NULL WITH DEFAULT Section number of statement in SYSPACKSTMT
COLLID	VARCHAR(128) NOT NULL WITH DEFAULT Collection-id: DSNDYNAMICSQLCACHE: from dynamic statement cache. DSNEXPLAINMODEYES: SET CURRENT EXPLAIN MODE YES used by application. DSNEXPLAINMODEEXPLAIN: SET CURRENT EXPLAIN MODE EXPLAIN used by application.
VERSION	VARCHAR(122) NOT NULL WITH DEFAULT Package version identifier
INSERT_TIME	TIMESTAMP NOT NULL GENERATED ALWAYS AS ROW CHANGE TIMESTAMP Time when the row was inserted.
EXPLAIN_TIME	TIMESTAMP EXPLAIN time. For cached statements it is a full precisioned timestamp. For non-cached static it is the BIND timestamp. For non-cached dynamic it is the EXPLAIN timestamp equal to char(16) appended by four zeroes.
REMARKS	VARCHAR(762) IBM internal use

SYSIBM.SYSRELS		(DSNDB06.SYSTSREL)
SYSIBM.DSNDLX01	D	(REFTBCREATOR,REFTBNAME)
SYSIBM.DSNDLX02	D	(CREATOR,TBNAME)
SYSIBM.DSNDLX03	D	(IXOWNER,IXNAME)
SYSIBM.DSNDLX04	P	(CREATOR,TBNAME,RELNAME)
Defines referential constraints. One row per constraint.		

This table contains the following columns:

Column Name	Data Type	Description
CREATOR	VARCHAR(128) NOT NULL	Schema of the dependent table of relationship.
TBNAME	VARCHAR(128) NOT NULL	Name of dependent table of relationship.
RELNAME	VARCHAR(128) NOT NULL	Name of the constraint.
REFTBNAME	VARCHAR(128) NOT NULL	Name of parent table in relationship.
REFTBCREATOR	VARCHAR(128) NOT NULL	Schema of the parent table.
COLCOUNT	SMALLINT NOT NULL	Number of columns in foreign key.
DELETERULE	CHAR(1) NOT NULL	Type of delete rule for referential constraint: C CASCADE R RESTRICT N SET NULL A NO ACTION
IBMREQD	CHAR(1) NOT NULL	Whether the row came from the basic MRM tape: Y/N/other..
RELOBID1	SMALLINT NOT NULL WITH DEFAULT	Internal ID of constraint with respect to the database that contains the parent table.
RELOBID2	SMALLINT NOT NULL WITH DEFAULT	Internal ID of constraint with respect to the database that contains the dependent table.
TIMESTAMP	TIMESTAMP NOT NULL WITH DEFAULT	Date and time constraint was defined.
IXOWNER	VARCHAR(128) NOT NULL	Schema of unique non-primary index used for the parent key. Contains all 9's if enforcing index is dropped. Blank if enforcing index is primary index.
IXNAME	VARCHAR(128) NOT NULL	Name of unique non-primary index used for the parent key. Contains all 9's if enforcing index is dropped. Blank if enforcing index is primary index.
ENFORCED	CHAR(1) NOT NULL WITH DEFAULT 'Y'	Enforced by the system or not: Y – Enforced by system N – Not enforced by system (trusted)
CHECKEXISTINGDATA	CHAR(1)	Option for checking data:

Column Name	Data Type	Description
	NOT NULL WITH DEFAULT	I – Immediately check existing data N – Never check existing data T – Immediately chaeck existing data for a temporal referential constraint.
RELCREATED	CHAR(1) NOT NULL	Release of DB2 used to create object.

SYSIBM.SYSRESAUTH		**(DSNDB06.SYSGPAUT)**
SYSIBM.DSNAGH01	**D**	**(GRANTEE,QUALIFIER,NAME, OBJTYPE,GRANTEETYPE)**
SYSIBM.DSNAGX01	**D**	**(GRANTOR,QUALIFIER,NAME, OBTYPE,GRANTORTYPE)**
Defines CREATE IN and PACKADM ON privileges for collections; USAGE privileges for distinct types; and USE privileges for buffer pools, storage groups, and table spaces.		

This table contains the following columns:

Column Name	Data Type/Description
GRANTOR	VARCHAR(128) NOT NULL Authorization ID of user who granted the privilege.
GRANTEE	VARCHAR(128) NOT NULL Authorization ID of user who holds the privilege. Could also be public for a grant to PUBLIC.
QUALIFIER	VARCHAR(128) NOT NULL The qualifier of the table space (database name), if privilege is for table space OBTYPE = R. The schema name of the user-defined data type if privilege is for distinct type OBTYPE = D. The schema name of the JAR file OBTYPE = J. PACKADM if OBTYPE = C and authority held is PACKADM. Otherwise blank.
NAME	VARCHAR(128) NOT NULL Name of storage group, table space, buffer pool, collection, or distinct type. Could be ALL when USE OF ALL BUFFERPOOLS is granted.
GRANTEETYPE	CHAR(1) Internal use only.
AUTHHOWGOT	CHAR(1) NOT NULL Authorization level of user from whom privileges were received: • blank—not applicable • A—PACKADM, on collection * • C—DBCTRL • D—DBADM • E—SECADM • G—ACCESSCTRL • L—SYSCTRL • M—DBMAINT • P—PACKADM, on specific collection • S—SYSADMT DATAACCESS • T—DATAACCESS
OBTYPE	CHAR(1) NOT NULL Object type: • B—buffer pool

Column Name	Data Type/Description
	• C—collection • D—distinct type • S—storage group • R—table space • J—JAR (JAVA ARchive file)
TIMESTAMP	CHAR(12) Internal use only.
DATEGRANTED	CHAR(6) Unused.
TIMEGRANTED	CHAR(8) Unused.
USEAUTH	CHAR(1) NOT NULL Whether privilege is held with GRANT option: • G—privilege held with GRANT option • Y—privilege held without GRANT option PACKADM, when OBTYPE is a collection (C) and QUALIFIER is PACKADM. CREATE IN when OBTYPE is a collection (C) and QUALIFIER is blank.
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other.
GRANTEDTS	TIMESTAMP NOT NULL WITH DEFAULT Time GRANT statement was executed.
GRANTORTYPE	CHAR(1) NOT NULL WITH DEFAULT Indicates the type of grantor: blank Authorization ID L Role
SYS_START	TIMESTAMP(12) NOT NULL System period temporal start time for transaction
SYS_END	TIMESTAMP(12) NOT NULL System period temporal end time for transaction
TRANS_START	TIMESTAMP(12) System period transaction timestamp.

SYSIBM.SYSROLES		**(DSNDB06.SYSROLES)**
SYSIBM.DSNRLX01	**U**	**(NAME)**
Defines roles. One row per role.		

This table contains the following columns:

Column Name	Data Type/Description
NAME	VARCHAR(128) NOT NULL The name of the role.
DEFINER	VARCHAR(128) NOT NULL The authorization ID or role that defined this role listed in the NAME column.
DEFINERTYPE	CHAR(1) NOT NULL The type of definer: L Role blank Authorization ID
CREATEDTS	TIMESTAMP NOT NULL The time when the role is created.
RELCREATED	CHAR(1) NOT NULL The release of DB2® that is used to create the role.
REMARKS	VARCHAR(762) NOT NULL A character string that is provided using the COMMENT statement.

Column Name	Data Type/Description
IBMREQD	CHAR(1) NOT NULL A value of Y indicates that the row came from the basic machine-readable material (MRM) tape.

SYSIBM.SYSROUTINEAUTH		(DSNDB06.SYSTSRAU)
SYSIBM.DSNOAX01	D	(GRANTOR,SCHEMA, SPECIFICNAME, ROUTINETYPE, GRANTEETYPE, EXECUTEAUTH, GRANTORTYPE)
SYSIBM.DSNOAX02	D	(GRANTEE,SCHEMA, SPECIFICNAME, ROUTINETYPE, GRANTEETYPE, EXECUTEAUTH, GRANTEDTS)
SYSIBM.DSNOAX03	D	(SCHEMA,SPECIFICNAME, ROUTINETYPE)
Defines privileges that are held by users on routines. For example, user-defined functions, cast functions, or stored procedures.		

This table contains the following columns:

Column Name	Data Type/Description
GRANTOR	VARCHAR(128) NOT NULL Authorization ID of user who granted the privilege.
GRANTEE	VARCHAR(128) NOT NULL Authorization ID of user who holds the privilege or the name of the package or plan that uses the privilege. Value could also be PUBLIC.
SCHEMA	VARCHAR(128) NOT NULL Routine schema.
SPECIFICNAME	VARCHAR(128) NOT NULL Specific name of routine. Value of “*” if privilege is held on all routines in the schema.
GRANTEDTS	TIMESTAMP NOT NULL Time when grant was executed.
ROUNTINETYPE	CHAR(1) NOT NULL Routine type: • F—user-defined function or cast function • P—stored procedure
GRANTEETYPE	CHAR(1) NOT NULL Grantee type: • blank—authorization ID • L—Role • P—application plan or package. If COLLID is not blank grantee is a package. • R—internal use only.
AUTHHOWGOT	CHAR(1) NOT NULL Authorization level of the user from whom privileges were received. • blank—not applicable • 1—grantor had privilege on schema.* at time of grant. • E—SECADM • G—ACCESSCTRL • L—SYSCTRL • S—SYSADM • T—DATAACCESS

Column Name	Data Type/Description
EXECUTEAUTH	CHAR(1) NOT NULL If grantee can execute routine: • Y—privilege held without grant option. • G—privilege held with grant option.
COLLID	VARCHAR(128) NOT NULL Collection name, if grantee is a package. Otherwise blank.
CONTOKEN	CHAR(8) NOT NULL WITH DEFAULT FOR BIT DATA Consistency token of the DBRM, if grantee is a package. Otherwise blank.
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other.
GRANTORTYPE	CHAR(1) NOT NULL Indicates the type of grantor: • blank—authorization ID • L—Role
SYS_START	TIMESTAMP(12) NOT NULL System period temporal start time for transaction.
SYS_END	TIMESTAMP(12) NOT NULL System period temporal end time for transaction.
TRANS_START	TIMESTAMP(12) System period transaction timestamp.

SYSIBM.SYSROUTINES		**(DSNDB06.SYSTSROU)**
SYSIBM.DSNOFX01	**U**	**(NAME,PARM_COUNT, PARM_SIGNATURE,ROUTINETYPE, SCHEMA PARM1 THRU PARM30, VERSION)**
SYSIBM.DSNOFX02	**P**	**(SCHEMA,SPECIFICNAME, ROUTINETYPE,VERSION)**
SYSIBM.DSNOFX03	**D**	**(NAME,SCHEMA, CAST_FUNCTION,PARM_COUNT, PARM_SIGNATURE, PARM1)**
SYSIBM.DSNOFX04	**U**	**(ROUTINEID)**
SYSIBM.DSNOFX05	**D**	**(SOURCESCHEMA, SOURCESPECIFIC,ROUTINETYPE)**
SYSIBM.DSNOFX06	**D**	**(SCHEMA,NAME,ROUTINETYPE, PARM_COUNT)**
SYSIBM.DSNOFX07	**U**	**(NAME,PARM_COUNT, ROUTINETYPE,SCHEMA, PARM_SIGNATURE,PARM1 THRU PARM30,VERSION)**
SYSIBM.DSNOFX08	**D**	**(JARSCHEMA,JAR_ID)**
Defines routines. One row per routine. For example, a user-defined function, cast function, or stored procedure.		

This table contains the following columns:

Column Name	Data Type/Description
SCHEMA	VARCHAR(128) NOT NULL Routine schema.
OWNER	VARCHAR(128) NOT NULL Routine owner.
NAME	VARCHAR(128) NOT NULL Routine name.

Column Name	Data Type/Description
ROUTINETYPE	CHAR(1) NOT NULL Routine type: F user-defined function or cast function P stored procedure
CREATEDBY	VARCHAR(128) NOT NULL Primary Authorization ID of the user that created the routine.
SPECIFICNAME	VARCHAR(128) NOT NULL Specific name of routine.
ROUTINEID	INTEGER NOT NULL Internal identifier of routine.
RETURN_TYPE	INTEGER NOT NULL Internal identifier of result data type of function. If table function, value is –2.
ORIGIN	CHAR(1) NOT NULL Routine origin: E external user-defined function or stored procedure N Native SQL procedure Q SQL function U sourced on user-defined function or built-in function S system-generated function
FUNCTION_TYPE	CHAR(1) NOT NULL Function type: C column function S scalar function T table function blank stored procedure, ROUTINETYPE = P
PARM_COUNT	SMALLINT NOT NULL Number of parameters for routine.
LANGUAGE	VARCHAR(24) NOT NULL Routine implementation language; values are ASSEMBLE, PLI, COBOL, C COMPJAVA, JAVA, REXX, SQL. Otherwise blank, ROUTINETYPE = 'F', if ORIGIN is not 'E' or not 'Q'.
COLLID	VARCHAR(128) NOT NULL Package collection name used when routine is executed. If package collection used to execute is the same as package collection invoking routine value is blank.
SOURCESCHEMA	VARCHAR(128) NOT NULL For source built-in function value is SYSIBM. ORIGIN = U and ROUTINETYPE = F value is source user-defined function schema. Otherwise, blank.
SOURCESPECIFIC	VARCHAR(128) NOT NULL ORIGIN = U and ROUTINETYPE = F value is specific name of source user-defined function or source built-in function. Otherwise, blank.
DETERMINISTIC	CHAR(1) NOT NULL Deterministic option for external function or a stored procedure: Y Deterministic (consistent results) N Indeterminate (results may differ

Column Name	Data Type/Description
	given input values) blank ORIGIN is not E, for a function (ROUTINETYPE = F)
EXTERNAL_ACTION	CHAR(1) NOT NULL External action option for external function: E Function has external side effects. Number of invocations is important. N Function has no side effects. blank ORIGIN is not E, for a function (ROUTINETYPE = F) or it is a stored procedure (ROUTINETYPE = P).
NULL_CALL	CHAR(1) NOT NULL CALLED ON NOT NULL INPUT option for external function: Y Function should be called if any parameter value is NULL. N Function should not be called if any parameter value is NULL. blank ORIGIN is not E, for a function and ROUTINETYPE = F.
CAST_FUNCTION	CHAR(1) NOT NULL Indicates if routine is a cast function: Y (yes) or N (no).
SCRATCHPAD	CHAR(1) NOT NULL SCRATCHPAD option for external function: Y Function has a SCRATCHPAD. N Function has no SCRATCHPAD. blank ORIGIN is not E, for a function (ROUTINETYPE = F) or it is a stored procedure (ROUTINETYPE = P)
SCRATCHPAD_ LENGTH	INTEGER NOT NULL If ORIGIN is E, for a function (ROUTINETYPE = F) and NO SCRATCHPAD was not specified, value is the length of the SCRATCHPAD. Otherwise, value is 0.
FINAL_CALL	CHAR(1) NOT NULL FINAL CALL option for external function: Y Final call will be made to function. N Final call will not be made to function. blank ORIGIN is not E, for a function (ROUTINETYPE = F) or it is a stored procedure (ROUTINETYPE = P).
PARALLEL	CHAR(1) NOT NULL PARALLEL option for external function: A Parallel tasks can invoke function. D Parallel tasks cannot invoke function. blank ORIGIN is not E, for a function (ROUTINETYPE = F) or it is a stored procedure (ROUTINETYPE = P).
PARAMETER_STYLE	CHAR(1) NOT NULL PARAMETER STYLE option of external function or stored procedure: D DB2SQL standard convention of parameter passing to external functions or stored procedures.

Column Name	Data Type/Description
	G GENERAL standard convention of parameter passing to stored procedures. N GENERAL CALL WITH NULLS convention of parameter passing to stored procedures. J JAVA. All parameters are passed according to conventions for JAVA and SQLJ. blank If ORIGIN is not E.
FENCED	CHAR(1) NOT NULL Y Indicates routine runs separately from DB2 address space in a WLM managed DB2 address space. Blank origin = 'Q' or 'N'
SQL_DATA_ACCESS	CHAR(1) NOT NULL SQL statements allowed in the external function or stored procedure. N NO SQL. C CONTAINS SQL; SQL type cannot read or modify data. R READS SQL DATA; SQL can only read data. M MODIFIES SQL DATA; All SQL allowed, and SQL can read or modify data. blank Not applicable
DBINFO	CHAR(1) NOT NULL DBINFO option for external function or stored procedure: Y DBINFO parameter will be passed to external function or stored procedure. N DBINFO parameter will not be passed to external function or stored procedure.
STAYRESIDENT	CHAR(1) NOT NULL STAYRESIDENT option for routine, determines if routine should be deleted from memory once ended: Y Load module remains in memory once ended. N Load module is deleted from memory once ended. blank ORIGIN is not E.
ASUTIME	INTEGER NOT NULL Number of CPU service units allowed for each invocation of the routine. If routine consumes more than allowed, DB2 cancels routine. If value is zero, service units is unlimited.
WLM_ENVIRONMENT	VARCHAR(96) NOT NULL WLM environment name used to run routine when origin = 'N', name of WLM ENVIRONMENT FOR DEBUG MODE to be used when debugging native SQL procedure. For (ROUTINETYPE = P), a value of blank indicates stored procedure cannot be run. For ROUTINETYPE = 'F' or ORIGIN is not 'E' column is blank. (SPAS).

Column Name	Data Type/Description
WLM_ENV_FOR_ NESTED	CHAR(1) NOT NULL For nested routines indicates whether the calling routines address space is used to run the nested routine: Y Nested routine uses calling routine address space, 'WLM ENVIRONMENT (name.*)' was specified. N Nested routine uses address space other than calling routine, 'WLM ENVIRONMENT name' was specified). blank WLM ENVIRONMENT is blank.
PROGRAM_TYPE	CHAR(1) NOT NULL Whether routine runs as a Language Environment main routine or subroutine: M main routine S subroutine blank ORIGIN is not E
EXTERNAL_ SECURITY	CHAR(1) NOT NULL Authorization ID to be used if routine accesses resources protected by external security: D DB2 – authorization ID associated with WLM-established SPAS. U SESSION_USER – authorization ID of SQL user invoking routine. C DEFINER – authorization ID of the routine owner. blank ORIGIN is not E.
COMMIT_ON_ RETURN	CHAR(1) NOT NULL For ROUTINETYPE = P, if transaction is always committed immediately on successful return from this stored procedure: N Unit of work is continued. Y Unit of work is committed immediately. A Unit of work of autonomous procedure is committed immediately unlike other work of the calling application blank ROUTINETYPE = F.
RESULT_SETS	SMALLINT NOT NULL For ROUTINETYPE = P, max number of ad hoc result sets this stored procedure can return. For ROUTINETYPE = F or if no ad hoc results exist, value is zero.
LOBCOLUMNS	SMALLINT NOT NULL If ORIGIN = E, the number of LOB columns in parameter list for user-defined function. If no LOB columns in parameter list or ORIGIN is not E, value is zero.
CREATEDTS	TIMESTAMP NOT NULL Time CREATE for this routine was executed.
ALTERDTS	TIMESTAMP NOT NULL Time last ALTER for this routine was executed.
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic

Column Name	Data Type/Description
	MRM tape: Y/N/other. If other, release dependency indicator.
PARM1	SMALLINT Internal use only.
…..	Thru
PARM30	SMALLINT Internal use only.
IOS_PER_INVOC	FLOAT NOT NULL WITH DEFAULT -1 Estimated number of I/O's required to execute routine. –1 if estimate is unknown, optimizer uses 0.
INSTS_PER_INVOC	FLOAT NOT NULL WITH DEFAULT -1 Estimated number of machine instructions required to execute routine. –1 if estimate is unknown, optimizer uses 4,000.
INITIAL_IOS	FLOAT NOT NULL WITH DEFAULT -1 Estimated number of I/O's performed the first or last time routine is invoked. –1 if estimate is unknown, optimizer uses 0.
INITIAL_INSTS	FLOAT NOT NULL WITH DEFAULT -1 Estimated number of machine instructions performed the first or last time routine is invoked. –1 if estimate is unknown, optimizer uses 40,000.
CARDINALITY	FLOAT NOT NULL WITH DEFAULT -1 Predicted cardinality of routine. –1 if predicted cardinality is unknown, optimizer uses 10,000.
RESULT_COLS	SMALLINT NOT NULL WITH DEFAULT -1 Number of columns in the result table for table function. Otherwise, value is 1
EXTERNAL_NAME	VARCHAR(762) NOT NULL Path/module/function DB2 should load to execute routine. If ORIGIN is not E, column is blank.
PARM_SIGNATURE	VARCHAR(150) NOT NULL FOR BIT DATA Internal use only.
RUNOPTS	VARCHAR(762) NOT NULL WITH DEFAULT FOR BIT DATA Language Environment (LE) run-time options. Empty string indicates installation default LE run-time options used.
REMARKS	VARCHAR(762) NOT NULL WITH DEFAULT User-defined character string provided via COMMENT ON statement.
JAVA_SIGNATURE	VARCHAR(3072) NOT NULL WITH DEFAULT The signature of the jar file. Blank when PARAMETER STYLE is not JAVA
CLASS	VARCHAR(384) NOT NULL WITH DEFAULT The class name contained in the jar file. Blank when PARAMETER STYLE is not JAVA.
JARSCHEMA	VARCHAR(128) NOT NULL WITH DEFAULT The schema of the jar file. Blank when PARAMETER STYLE is not JAVA.

Column Name	Data Type/Description
JAR_ID	VARCHAR(128) NOT NULL WITH DEFAULT The name of the jar file. Blank when PARAMETER STYLE is not JAVA.
SPECIAL_REGS	CHAR(1) NOT NULL WITH DEFAULT 'I' The SPECIAL REGISTER option for a routine. I INHERIT SPECIAL REGISTERS D DEFAULT SPECIAL REGISTERS Blank ROUTINETYPE = 'F' or ORIGIN is not 'E' or 'Q'
NUM_DEP_MQTS	SMALLINT NOT NULL WITH DEFAULT Number of dependent materialized query tables. The value is '0' if row doesn't describe a user-defined table function, or if no materialized query tables are defined on the table function.
MAX_FAILURE	SMALLINT NOT NULL WITH DEFAULT -1 Allowable failure for this routine. If zero routine will never be stopped.
PARAMETER_CCSID	INTEGER NOT NULL WITH DEFAULT A CCSID that specifies how character, graphic, date, time, and timestamp data types for system generated parameters to the routine should be passed. Value is dependent on encoding scheme: ASCII, EBCDIC, and UNICODE
VERSION	VARCHAR(122) NOT NULL WITH DEFAULT The version identifier for a native SQL procedure (ORIGIN='N'), or non-inline SQL scalar function(ORIGIN='Q' and INLINE='N'. A zero length string for the rows that are created prior to V9 and for other rows.
CONTOKEN	CHAR(8) NOT NULL WITH DEFAULT FOR BIT DATA The consistency token for the routine. The column is set to X'20' if the value of ORIGIN is not 'N'
ACTIVE	CHAR(1) NOT NULL WITH DEFAULT Identifies the active version of the routine: Y The routine is the active version. N The routine is not the active version. blank The value of ORIGIN is not 'N' or the row was created prior to V9.
DEBUG_MODE	CHAR(1) NOT NULL WITH DEFAULT Identifies whether or not this routine is enabled for debugging: 1 This routine is enabled for debugging and can be debugged in a client debug session using the DB2 Unified Debugger. 0 This routine is not enabled for debugging. N This routine can never be enabled for debugging. blank The LANGUAGE is not specified as JAVA, the value of ORIGIN is not 'N', or the row was created prior to V9.

Column Name	Data Type/Description
TEXT_ENVID	INTEGER NOT NULL WITH DEFAULT Internal identifier of the environment. The column is 0 if the value of ORIGIN is not 'N' or if the row was created prior to V9.
TEXT_ROWID	ROWID NOT NULL GENERATED ALWAYS ID to support LOB columns for source text.
TEXT	CLOB(2M) NOT NULL WITH DEFAULT The source text of the CREATE statement or the ALTER statement with the body for the routine. The column is a zero-length string if the value of ORIGIN is not 'N' or if the row was created prior to V9.
OWNERTYPE	CHAR(1) NOT NULL WITH DEFAULT Indicates the type of owner: blank Authorization ID L Role
PARAMETER_ VARCHARFORM	INTEGER NOT NULL WITH DEFAULT A non-zero value that indicates the actual representation, to a LANGUAGE C routine, of any varying length string parameter that appears in the parameter list or RETURNS clause for that routine.
RELCREATED	CHAR(1) NOT NULL The release of DB2 that is used to create the object. Blank if created prior to V9.
PACKAGEPATH	VARCHAR(4096) The value of the PACKAGE PATH option of the CREATE FUNCTION, CREATE PROCEDURE, ALTER FUNCTION, or ALTER PROCEDURE statement that created or last changed the routine. PACKAGE PATH identifies the package path to use when the routine is executed. A blank value indicates the package path is the same as the package path of the program that invoked the routine.
SECURE	CHAR(1) NOT NULL WITH DEFAULT 'N' Routine is secured: N NO Y YES
SYSTEM_DEFINED	CHAR(1) NOT NULL WITH DEFAULT Routine is system defined blank NO S YES
INLINE	CHAR(1) NOT NULL WITH DEFAULT SQL function inline: Y YES N NO blank Not an SQL function
PARSETREE	BLOB(1G) NOT NULL WITH DEFAULT IBM internal use only
WRAPPED	Y if the routine is obfuscated and blank if not obfuscated.
REGENERATETS	The time when the object was regenerated

SYSIBM.SYSROUTINESTEXT		(DSNDB06.SYSPLUXA)
SYSIBM.DSNPLX01	U	(AUXID,AUXVER)
Auxiliary table for the TEXT LOB column of SYSROUTINES.		

This table contains the following column:

Column Name	Data Type	Description
AUXID	VARCHAR(17)	ID of auxiliary data.
AUXVER	SMALLINT	Version of auxiliary data.
AUXVALUE	CLOB(2M) NOT NULL WITH DEFAULT)	The source text of the CREATE PROCEDURE statement for the routine. TEXT can also hold the source text of the ALTER PROCEDURE statement for the routine if the routine is a native SQL procedure and the SQL procedure body is included in the ALTER PROCEDURE statement.

SYSIBM.SYSROUTINES_OPTS		(DSNDB06.SYSGRTNS)
SYSIBM.DSNROX01	U	(SCHEMA,ROUTINENAME, BUILDDATE,BUILDTIME)
Defines the build options for generated routines. A row for each generated routine. You can insert, update, and delete rows in this table.		

This table contains the following columns:

Column Name	Data Type	Description
SCHEMA	VARCHAR(128) NOT NULL	Schema of the routine.
ROUTINENAME	VARCHAR(128) NOT NULL	Name of the routine.
BUILDDATE	DATE NOT NULL WITH DEFAULT	Date the routine was built
BUILDTIME	TIME NOT NULL WITH DEFAULT	Time the routine was built
BUILDSTATUS	CHAR(1) NOT NULL WITH DEFAULT 'C'	Whether this version of the routine's options is the current version
BUILDSCHEMA	VARCHAR(128) NOT NULL	Schema name for BUILDNAME.
BUILDNAME	VARCHAR(128) NOT NULL	Procedure used to create the routine.
BUILDOWNER	VARCHAR(128) NOT NULL	Authorization ID used to create the routine.
IBMREQD	CHAR(1) NOT NULL WITH DEFAULT 'N'	Whether the row came from the basic MRM tape: Y/N/other. If other, release dependency indicator.
PRECOMPILE_OPTS	VARCHAR(765) NOT NULL WITH DEFAUTL	Precompiler options used to build the routine.
COMPILE_OPTS	VARCHAR(765) NOT NULL WITH DEFAULT	Compiler options used to build the routine.

Column Name	Data Type	Description
PRELINK_OPTS	VARCHAR(765) NOT NULL WITH DEFAULT	Prelink-edit options used to build the routine.
LINK_OPTS	VARCHAR(765) NOT NULL WITH DEFAULT	Link-edit options used to build the routine.
BIND_OPTS	VARCHAR (3072) NOT NULL WITH DEFAULT	Bind options used to build the routine.
SOURCEDSN	VARCHAR(765) NOT NULL WITH DEFAULT	Name of the source data set.
DEBUG_MODE	CHAR(1) NOT NULL	L Debugging is on or off for the object: 0 Debugging is off 1 Debugging is on

SYSIBM.SYSROUTINES_SRC		(DSNDB06.SYSGRTNS)
SYSIBM.DSNRSX01	D	(ROUTINENAME)
SYSIBM.DSNRSX02	U	(SCHEMA,ROUTINENAME, BUILDDATE,SEQNO)
Contains source for generated routines. You can insert, update, and delete rows in this table.		

This table contains the following columns:

Column Name	Data Type/Description
SCHEMA	VARCHAR(128) NOT NULL Schema of the routine.
ROUTINENAME	VARCHAR(128) NOT NULL Name of the routine.
BUILDDATE	DATE NOT NULL WITH DEFAULT Date the routine was built
BUILDTIME	TIME NOT NULL WITH DEFAULT Time the routine was built
BUILDSTATUS	CHAR(1) NOT NULL WITH DEFAULT 'C' Whether this version of the routine's source is the current version
SEQNO	INTEGER NOT NULL Number of the source statement piece in CREATESTMT.
IBMREQD	CHAR(1) NOT NULL WITH DEFAULT 'N' Whether the row came from the basic MRM tape: Y/N/other. If other, release dependency indicator.
CREATESTMT	VARCHAR(7500) NOT NULL Routine source statement.

SYSIBM.SYSROUTINES_TREE		(DSNDB06.SYSPLUXB)
SYSIBM.DSNPLX02	U	(AUXID,AUXVER)
Auxiliary table for PTREE LOB column of SYSROUTINES.		

This table contains the following column:

Column Name	Data Type	Description
AUXID	VARCHAR(17)	ID of auxiliary data.
AUXVER	SMALLINT	Version of auxiliary data.
AUXVALUE	BLOB(1G) NOT NULL WITH DEFAULT	IBM internal use.

SYSIBM.SYSSCHEMAAUTH		(DSNDB06.SYSTSSCM)
SYSIBM.DSNSKX01	D	(GRANTEE,SCHEMANAME, GRANTEETYPE)
SYSIBM.DSNSKX02	D	(GRANTOR,GRANTORTYPE)
Defines user privileges on individual schemas. One or more rows for each user granted a privilege on a particular schema.		

This table contains the following columns:

Column Name	Data Type/Description
GRANTOR	VARCHAR(128) NOT NULL Authorization ID of user who granted the privilege or SYSADM.
GRANTEE	VARCHAR(128) NOT NULL Authorization ID of user or group that holds the privilege. Value could also be PUBLIC.
SCHEMANAME	VARCHAR(128) NOT NULL Schema name or '*' for all schemas.
AUTHHOWGOT	CHAR(1) NOT NULL Authorization level of the user from whom privileges were received. • Grantor had privilege on all schemas at time of grant. • E—SECADM • G—ACCESSCTRL • L—SYSCTRL • S—SYSADM
CREATEINAUTH	CHAR(1) NOT NULL Authorization level of grantor CREATIN privilege on schema: blank Not applicable. G privilege held with GRANT option. Y privilege held without GRANT option.
ALTERINAUTH	CHAR(1) NOT NULL Whether grantee holds ALTERIN privilege on schema: blank privilege not held. G privilege held with GRANT option. Y privilege held without GRANT option.
DROPINAUTH	CHAR(1) NOT NULL Whether grantee holds DROPIN privilege on schema: blank privilege not held. G privilege held with GRANT option. Y privilege held without GRANT option.
GRANTEDTS	TIMESTAMP NOT NULL Time when grant was executed.
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other. If other, release dependency indicator.
GRANTEETYPE	CHAR(1) NOT NULL WITH DEFAULT Indicates the type of grantee: blank Authorization ID L Role
GRANTORTYPE	CHAR(1) NOT NULL WITH DEFAULT Indicates the type of grantor: blank Authorization ID L Role
SYS_START	TIMESTAMP(12) NOT NULL

Column Name	Data Type/Description
	System period temporal start time for transaction
SYS_END	TIMESTAMP(12) NOT NULL System period temporal end time for transaction
TRANS_START	TIMESTAMP(12) System period transaction timestamp.

SYSIBM.SYSSEQUENCEAUTH		**(DSNDB06.SYSSEQ2)**
SYSIBM.DSNWCX01	**D**	**(SCHEMA,NAME)**
SYSIBM.DSNWCX02	**D**	**(GRANTOR,SCHEMA, NAME, GRANTORTYPE)**
SYSIBM.DSNWCX03	**D**	**(GRANTEE,SCHEMA_NAME, GRANTEETYPE)**
Defines the privileges users have over sequences.		

This table contains the following columns:

Column Name	Data Type/Description
GRANTOR	VARCHAR(128) NOT NULL Authorization ID of user who granted privileges.
GRANTEE	VARCHAR(128) NOT NULL Authorization ID of user/group the holds privileges or application plan or package that uses privileges. PUBLIC for a grant to PUBLIC.
SCHEMA	VARCHAR(128) NOT NULL Sequence schema.
NAME	VARCHAR(128) NOT NULL Sequence name.
GRANTEETYPE	CHAR(1) NOT NULL Type of grantee: blank An authorization ID. L Role. P An application plan or package. R Internal use only.
AUTHHOWGOT	CHAR(1) NOT NULL Authorization level of the user from whom privileges were received. • blank—not applicable. • E—SECADM • G—ACCESSCTRL • L—SYSCTRL • S—SYSADM • T—DATAACCESS
ALTERAUTH	CHAR(1) NOT NULL Whether grantee holds ALTERIN privilege on schema: blank privilege not held. G privilege held with GRANT option. Y privilege held without GRANT option.
USEAUTH	CHAR(1) NOT NULL Indicates whether grantee holds USAGE privilege on sequence: blank privilege not held. G privilege held with GRANT option. Y privilege held without GRANT option
COLLID	VARCHAR(128) NOT NULL If GRANTEE is a package, its collection name. Otherwise, a string of length zero.

Column Name	Data Type/Description
CONTOKEN	CHAR(8) NOT NULL FOR BIT DATA If GRANTEE is a package, the consistency token of the DBRM from which the package was derived. Blank otherwise.
GRANTEDTS	TIMESTAMP NOT NULL Time when the GRANT statement was executed.
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other. If other, release dependency indicator.
GRANTORTYPE	CHAR(1) NOT NULL WITH DEFAULT Indicates the type of grantor: blank Authorization ID L Role
SYS_START	TIMESTAMP(12) NOT NULL System period temporal start time for transaction
SYS_END	TIMESTAMP(12) NOT NULL System period temporal end time for transaction
TRANS_START	TIMESTAMP(12) System period transaction timestamp.

SYSIBM.SYSSEQUENCES		**(DSNDB06.SYSSEQ)**
SYSIBM.DSNSQX01	**P**	**(SCHEMA,NAME)**
SYSIBM.DSNSQX02	**U**	**(SEQUENCEID)**
SYSIBM.DSNSQX03	**D**	**(SEQSCHEMA,SEQNAME)**
Defines identity columns or user-defined sequences. One row for each identity column or sequence.		

This table contains the following columns:

Column Name	Data Type/Description
SCHEMA	VARCHAR(128) NOT NULL The value of TBCREATOR from the entry for the identity column or the schema of the sequence or alias.
OWNER	VARCHAR(128) NOT NULL The value of TBCREATOR from the SYSCOLUMNS entry for the identity column or the owner of the sequence or alias.
NAME	VARCHAR(128) NOT NULL Name that DB2 generated for the identity column or the name of the sequence or alias.
SEQTYPE	CHAR(1) NOT NULL Type of entry: A For an alias I For an identity column S For a user-defined sequence X Implicitly created DOCID column for base table that contains XML data.
SEQUENCEID	INTEGER NOT NULL Internal identifier of the identity column, sequence or alias.
CREATEDBY	VARCHAR(128) NOT NULL The authorization ID under which the identity column, alias or sequence was created.
INCREMENT	DECIMAL(31,0) NOT NULL Increment value (positive or negative, within INTEGER scope). Value is 0 if an alias.

Column Name	Data Type/Description
START	DECIMAL(31,0) NOT NULL Start value. Value is 0 if an alias.
MAXVALUE	DECIMAL(31,0) NOT NULL Maximum value allowed. Value is 0 if an alias.
MINVALUE	DECIMAL(31,0) NOT NULL Minimum value allowed. Value is 0 if an alias.
CYCLE	CHAR(1) NOT NULL Whether cycling will occur when a boundary is reached: N (no), Y (yes), blank (alias).
CACHE	INTEGER NOT NULL Number of values to preallocate in memory for faster access. A value of 0 indicates that values are not to be preallocated. Blank is an alias.
ORDER	CHAR(1) NOT NULL If values must be generated in order: N (no), Y (yes), blank (alias).
DATATYPEID	INTEGER NOT NULL For a built-in data type, the internal ID of the built-in type. For a distinct type, the internal ID of the distinct type. Value is 0 if an alias
SOURCETYPEID	INTEGER NOT NULL For a built-in data type, 0. For a distinct type, the internal ID of the built-in data type upon which the distinct type is sourced. Value is 0 if an alias
CREATEDTS	TIMESTAMP NOT NULL Timestamp when the identity column, alias or sequence was created.
ALTEREDTS	TIMESTAMP NOT NULL Timestamp when the last ALTER was executed for identity column, alias or sequence.
MAXASSIGNEDVAL	DECIMAL(31,0) Last possible assigned value. Initialized to null when the sequence object is created. Updated each time the next chunk of *n* values is cached, where *n* is the value for CACHE. The LOAD utility updates AXASSIGNEDVAL for identity columns in the following ways: • For sequence or identity column values that aregenerated by the DB2 database manager, LOAD updates MAXASSIGNEDVAL each time that the next chunk of *n* values is cached. *n* is the CACHE column value. • For identity column values that are generated by theuser, LOAD updates MAXASSIGNEDVAL with the MAXVALUE column value or MINVALUE column value. MAXVALUE is used if the INCREMENT column value is positive. MINVALUE is used if the INCREMENT column value is negative. • LOAD with REPLACE sets MAXASSIGNEDVAL to null before loading any data. • LOAD with FORMAT INTERNAL does not update MAXASSIGNEDVAL. Value is 0 if an alias

Column Name	Data Type/Description
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other.
REMARKS	VARCHAR(762) NOT NULL Character string provided with the COMMENT ON statement. Blank for an Identity column.
PRECISION	SMALLINT NOT NULL WITH DEFAULT The precision defined for a sequence with a decimal or numeric type. • SMALLINT • INTEGER Or the actual precision specified by user • 0 Row created prior to DB2 8. Value is 0 if an alias
RESTARTWITH	DECIMAL(31,0) NULLABLE WITH DEFAULT The RESTART WITH value specified for a sequence during ALTER. NULL if no ALTER RESTART or row is an alias.
OWNERTYPE	CHAR(1) NOT NULL WITH DEFAULT Indicates the type of owner: blank Authorization ID L Role
RELCREATED	CHAR(1) NOT NULL The release of DB2 that is used to create the object. Blank if created prior to V9.
SEQSCHEMA	VARCHAR(128) NOT NULL WITH DEFAULT Schema of target sequence.
SEQNAME	VARCHAR(128) NOT NULL WITH DEFAULT Name of target sequence.

SYSIBM.SYSSEQUENCESDEP		**(DSNDB06.SYSSEQ2)**
SYSIBM.DSNSRX01	**D**	**(DCREATOR,DNAME, DCOLNAME)**
SYSIBM.DSNSRX02	**D**	**(BSCHEMA,BNAME,DTYPE)**
Contains the dependent objects for identity columns and sequences.		

This table contains the following columns:

Column Name	Data Type/Description
BSEQUENCEID	INTEGER NOT NULL Internal identifier of the identity column or sequence.
DCREATOR	VARCHAR(128) NOT NULL Authorization ID of the owner of the sequence or the table that contains the identity column or sequence.
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other. If other, release dependency indicator.
DNAME	VARCHAR(128) NOT NULL Name of the sequence or table containing the identity.
DCOLNAME	VARCHAR(128) NOT NULL Name of the identity column or sequence.
DTYPE	CHAR(1) NOT NULL WITH DEFAULT 'I' Type of object that is dependent on sequence: F (SQL function), I (identity column), X (implicit

Column Name	Data Type/Description
	DOCID column that is created on a base table with XML, blank (identity column created prior to V8).
BSCHEMA	VARCHAR(128) NOT NULL WITH DEFAULT Schema name of the sequence. String of length zero if created prior to V8.
BNAME	VARCHAR(128) NOT NULL WITH DEFAULT Sequence name. String of length zero if created prior to V8.
DSCHEMA	VARCHAR(128) NOT NULL WITH DEFAULT Qualifier of the object that is dependent on the sequence. String of length zero if created prior to V8.
DOWNER	VARCHAR(128) NOT NULL WITH DEFAULT The owner of the object that is dependent on this sequence. This will be a string of length zero for an object that was created prior to V9.
DOWNERTYPE	CHAR(1) NOT NULL WITH DEFAULT The type of owner: Blank An authorization ID L A role

SYSIBM.SYSSESSION		**(DSNDB06.SYSTSSES)**
SYSIBM.DSNSNX02	**P**	**(TOKEN)**
Stores the session token generated by the server and associated session data.		

This table contains the following columns:

Column Name	Data Type/Description
TOKEN	CHAR(40) NOT NULL FOR BIT DATA Session token.
CORRTKN	VARCHAR(256) FOR BIT DATA Extended client correlation token in use.
GV_FLAGS	CHAR(2) NOT NULL FOR BIT DATA Flags for internal classification of global variables like LOBs or arrays.
TOTAL	CHAR(4) NOT NULL Number of entries in SYSSESSION_EX that corresponds to session token.
SPECIAL_REGISTERS	VARCHAR(16000) NOT NULL FOR BIT DATA Special register values.
GLOBAL_VARIABLES	BLOB(2G) INLINE LENGTH(16000) Global variable values. Arrays and LOBs stored as locator values that reference a row in SYSSESSION_EX.
ROWID	NOT NULL/Generated ROWID.

SYSIBM.SYSSESSION_GV		**(DSNDB06.SYSTSSNL)**
SYSIBM.DSNSNX01	**U**	**(AUXID,AUXVER)**
Auxiliary table for GLOBAL_VARIABLES LOB column of SYSSESSION.		

This table contains the following column:

Column Name	Data Type	Description
AUXID	VARCHAR(17)	ID of auxiliary data.
AUXVER	SMALLINT	Version of auxiliary data.

Column Name	Data Type	Description
AUXVALUE	CLOB(2M) NOT NULL WITH DEFAULT	Trigger text.

SYSIBM.SYSSESSION_EX		(DSNDB06.SYSTSSNX)
SYSIBM.DSNSNX04	D	(TOKEN)
SYSIBM.DSNSNX05	U	(TOKEN,GVID,LOCATOR)
Contains global variable data of LOB or array type that corresponds to the locator stored in the GLOBAL_VARIABLES column of SYSSESSION.		

This table contains the following columns:

Column Name	Data Type/Description
TOKEN	CHAR(40) NOT NULL FOR BIT DATA Session token.
LOCATOR	CHAR(8) NOT NULL FOR BIT DATA Locator value corresponding to one of the global variables with datatype array/LOB.
HEADER	CHAR(89) NOT NULL FOR BIT DATA Array static descriptor header when locator value corresponds to array type.
GVID	CHAR(8) NOT NULL FOR BIT DATA Global variable identifier.
DATATYPE	CHAR(2) NOT NULL FOR BIT DATA Global variable data type.
CCSID	CHAR(2) FOR BIT DATA Global variable CCSID.
GVSCHEMA	VARCHAR(130) FOR BIT DATA Global variable schema name.
GVNAME	VARCHAR(130) FOR BIT DATA Global variable name.
DATA	BLOB(2G) INLINE LENGTH(30000) NOT NULL FOR BIT DATA Data value stored in the global variable.

SYSIBM.SYSSESSION_DATA		(DSNDB06.SYSTSSXL)
SYSIBM.DSNSNX03	U	(AUXID,AUXVER)
Auxiliary table for DATA LOB column of SYSSESSION_EX.		

This table contains the following column:

Column Name	Data Type	Description
AUXID	VARCHAR(17)	ID of auxiliary data.
AUXVER	SMALLINT	Version of auxiliary data.
AUXVALUE	CLOB(2M) NOT NULL WITH DEFAULT	Trigger text.

SYSIBM.SYSSESSION_STATUS		(DSNDB06.SYSTSSTA)
SYSIBM.DSNSNX02	U	(TOKEN)
Stores the session token and timestamp for when the corresponding session data was last referred.		

This table contains the following columns:

Column Name	Data Type/Description
TOKEN	CHAR(40) NOT NULL FOR BIT DATA Session token.
TOKEN_TS	CHAR(16) NOT NULL FOR BIT DATA Timestamp when last referred.
TOKEN_MEMBER	CHAR(16) NOT NULL FOR BIT DATA Data sharing member where row was last accessed.
STATUS	CHAR(4) FOR BIT DATA Status of session corresponding to the token (timed-out, clean disconnect, . .).

SYSIBM.SYSSTATFEEDBACK		(DSNDB06.SYSTSFSB)
SYSIBM.DSNSFX01	D	(TBCREATOR,TBNAME, IXCREATOR,IXNAME,COLNAME, COLGROUPCOLNO, NUMCOLUMNS,TYPE)
SYSIBM.DSNSFX02	D	(TBCREATOR,TBNAME)
SYSIBM.DSNSFX03	D	(IXCREATOR,IXNAME)
Contains information about missing or conflicting catalog statistics used by the optimizer.		

This table contains the following columns:

Column Name	Data Type/Description
TBCREATOR	VARCHAR(128)/Table creator.
TBNAME	VARCHAR(128)/Table name.
IXCREATOR	VARCHAR(128)/Index creator.
IXNAME	VARCHAR(128)/Index name.
COLNAME	VARCHAR(128)/Column name.
NUMCOLUMNS	SMALLINT Number of columns in column group
COLGROUPCOLNO	VARCHAR(254) FOR BIT DATA Hex representation identifies set of columns associated with the statistics. Value is 0 for a single column, otherwise an array os smallint column numbers with dimension equal to NUMCOLUMNS.
TYPE	CHAR(1) Type of statistics to collect: • C—Cardinality • F—frequency • H—histogram • I—index • T—table
DBNAME	VARCHAR(24)/Database name.
TSNAME	VARCHAR(24)/Tablespace name.
REASON	CHAR(8) Reason for recommended statistics: ▪ BASIC—Basic statistics for table or index is missing. ▪ KEYCARD—Index key columns cardinality missing.

Column Name	Data Type/Description
	▪ LOWCARD—Cardinality for column is low value indicating data skew. ▪ NULLABLE—Distribution statistics not available for nullable column. ▪ DEFAULT—A predicate references a probable default value. ▪ RANGEPRD—Histogram statistics missing for range predicate. ▪ PARALLEL—Parallelism could not be improved by uniform partitioning of key ranges. ▪ CONFLICT—Other statistic conflicts. ▪ COMPFFIX—Multi-column statistics needed an index compound filter factor. ▪ STALE—Statistic appears likely to be out of sync with other statistics, based on comparison of the time that it was collected to statistics collection times for related objects.
BLOCK_RUNSTATS	CHAR(1) Is the row used when tools collect statistics based on recommendations. New rows will get BLANK upon insert by DB2. This is an updatable column.
REMARKS	VARCHAR(254)/Free form text.
LASTDATE	DATE Last date recommendations updated

SYSIBM.SYSSTMT		**(DSNDB06.SYSTSSTM)**
SYSIBM.DSNPSX01	**D**	**(PLNAME,NAME)**
SYSIBM.DSNPSX02	**D**	**(PLNAME,NAME,SEQNO)**
Contains statement text for each SQL statement of each DBRM. One or more rows for each SQL statement.		

This table contains the following columns:

Column Name	Data Type/Description
NAME	VARCHAR(24)/Name of the DBRM.
PLNAME	VARCHAR(24) NOT NULL Application plan name.
PLCREATOR	VARCHAR(128) NOT NULL Schema of the application plan.
SEQNO	INTEGER NOT NULL Sequence number of this row with respect to a statement of the DBRM. The number starts with zero. IBM internal use only if SEQNO, STMTNO and SECTNO are zero.
STMTNO	SMALLINT NOT NULL Statement no. of SQL statement in source program. Statement nos. > 32767 are displayed as zero, see STMTNOI for statement number.
SECTNO	SMALLINT NOT NULL Plan section no. containing the SQL statement.
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other.

Column Name	Data Type/Description
TEXT	VARCHAR(3800) NOT NULL WITH DEFAULT FOR BIT DATA Text or portion of the text of the SQL statement.
ISOLATION	CHAR(1) NOT NULL WITH DEAFULT Isolation level for the SQL statement. R RR (repeatable read) T RS (read stability) S CS (cursor stability) U UR (uncommitted read) L RS isolation with a lock-clause X RR isolation with a lock-clause blank The WITH clause was not specified on this statement
STATUS	CHAR(1) NOT NULL WITH DEFAULT The status of binding the statement. A Distributed; uses DB2 private protocol. Statement is parsed and executed at the server using defaults for input variables. B Distributed; uses DB2 private protocol. Statement will be parsed and executed using input variables. C Compiled; statements bound successfully using defaults for input values. D Distributed – statement references a remote object using DB2 private protocol access. E Explain; statements is an EXPLAIN done at bind time using defaults for input values. F Parsed statement did not successfully bind. Validate (RUN) was used to rebind statement at execution using input values. G Compiled; statement bound successfully. REOPT was specified to rebind statement at execution using input values. H Parsed; statement is DDL or another SQL statement that did not bind successfully. Rebind will occur at execution using default input values. I Indefinite; statement is dynamic bound at execution using default for input values. J Indefinite; statement is dynamic bound at execution using input values. K Control; CALL statement. L Bad; statement is in error, bind continues, but statement cannot be executed. M Parsed – statement references a table that is qualified with SESSION and was not bound because the table reference could be for a declared temp table that will not be defined until the package or plan is run. blank Statement is non-executable or bound prior to V5.
ACCESSPATH	CHAR(1) NOT NULL WITH DEFAULT For static statement indicates if access path was determined using hints. H hints used blank hints not used, no access path associated with statement, or dynamic statements.

Column Name	Data Type/Description
STMTNOI	INTEGER NOT NULL WITH DEFAULT Statement number of corresponding statement in source program.
SECTNOI	INTEGER NOT NULL WITH DEFAULT Section number of statement.
EXPLAINABLE	CHAR(1) NOT NULL WITH DEFAULT Contains one of the following values: Y Indicates that the SQL statement can be used with the EXPLAIN function and may have rows describing its access path in the userid.PLAN_TABLE. N Indicates that the SQL statement does not have any rows describing its access path in the userid.PLAN_TABLE. blank Indicates that the SQL statement was bound prior to V7.
QUERYNO	INTEGER NOT NULL WITH DEFAULT -1 The query number of the SQL statement in source program. SQL statements bound prior to V7 have a default value of -1. Statements bound in V7 or later use the value specified on the QUERYNO clause on SELECT, UPDATE, INSERT, DELETE, EXPLAIN, and DECLARE CURSOR statements. If the QUERYNO clause is not specified, the query number is set to the statement number.
PLCREATORTYPE	CHAR(1) NOT NULL WITH DEFAULT Indicates the type of creator: blank Authorization ID L Role

SYSIBM.SYSSTOGROUP		**(DSNDB06.SYSTSSTG)**
SYSIBM.DSNSSH01	**U**	**(NAME)**
Defines storage groups. One row for each.		

This table contains the following columns:

Column Name	Data Type/Description
NAME	VARCHAR(128) NOT NULL Name of the storage group.
CREATOR	VARCHAR(128) NOT NULL Schema of the storage group.
VCATNAME	VARCHAR(24) NOT NULL Integrated Catalog Facility (ICF) catalog name.
VPASSWORD	VARCHAR(24)/Not used.
SPACE	INTEGER NOT NULL No. of kilobytes of DASD allocated to storage group, as determined by last execution of STOSPACE utility.
SPCDATE	CHAR(5)/Unused.
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other..
CREATEDBY	VARCHAR(128) NOT NULL WITH DEFAULT Primary authorization ID of user who created the storage group.
STATSTIME	TIMESTAMP NOT NULL WITH DEFAULT Date and time of the last execution of the STOSPACE utility for this storage group.

Column Name	Data Type/Description
CREATEDTS	TIMESTAMP NOT NULL WITH DEFAULT Time that CREATE was executed for this storage group.
ALTEREDTS	TIMESTAMP NOT NULL WITH DEFAULT Time of the most recent ALTER STOGROUP executed for this storage group. If no ALTER STOGROUP has been executed ALTEREDTS and CREATEDTS will contain the same value.
SPACEF	FLOAT NOT NULL WITH DEFAULT Kilobytes of DASD storage for the storage group. Value is –1 if statistics have not been gathered. This is an updateable column.
DATACLAS	VARCHAR(24) NOT NULL Name of the SMS data class. Blank if data class is not used.
MGMTCLAS	VARCHAR(24) NOT NULL Name of the SMS management class. Blank if management class is not used.
STORCLAS	VARCHAR(24) NOT NULL Name of the SMS storage class. Blank if storage class is not used.
CREATORTYPE	CHAR(1) NOT NULL WITH DEFAULT Indicates the type of creator: blank Authorization ID L Role
RELCREATED	CHAR(1) NOT NULL The release of DB2 that is used to create the object. Blank if created prior to V9.

SYSIBM.SYSSTRINGS		**(DSNDB06.SYSSTR)**
SYSIBM.DSNSSX01	**P**	**(OUTCCSID,INCCSID,IBMREQD)**
Contains information needed to convert one coded character set to another. If IBMREQD=N, you can insert, update, and delete rows.		

This table contains the following columns:

Column Name	Data Type/Description
INCCSID	INTEGER NOT NULL Coded character set ID to translate from.
OUTCCSID	INTEGER NOT NULL Coded character set ID to translate to.
TRANSTYPE	CHAR(2) NOT NULL Indicates the type of conversion: GG GRAPHIC to GRAPHIC MM EBCDIC MIXED to EBCDIC MIXED MS EBCIDC MIXED to SBCS PM ASCII MIXED to EBCDIC MIXED PS ASCII MIXED to SBCS SM SBCS to EBCDIC MIXED SS SBCS to SBCS MP EBCDIC MIXED to ASCII MIXED PP ASCII MIXED to ASCII MIXED SP SBCS to ASCII MIXED
ERRORBYTE	CHAR(1) FOR BIT DATA (Nulls are allowed) Byte in translate table to indicate error. Null indicates the absence of an error byte.
SUBBYTE	CHAR(1) FOR BIT DATA (Nulls are allowed) Byte in conversion table used as substitution

Column Name	Data Type/Description
	character. Null indicates absence of substitution character.
TRANSPROC	VARCHAR(24) NOT NULL WITH DEFAULT The name of a module or blanks. If IBMREQD is 'N', a nonblank value is the name of a conversion procedure provided by the user. If IBMRED is 'Y', a nonblank value is the name of a DB2 module that contains DBCS conversion tables.
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other.
TRANSTAB	VARCHAR(256) NOT NULL WITH DEFAULT FOR BIT DATA Either a conversion table or an empty string.

SYSIBM.SYSSYNONYMS		**(DSNDB06.SYSTSSYN)**
SYSIBM.DSNDYX01	**U**	**(CREATOR,NAME)**
SYSIBM.DSNDYX02	**D**	**(TBCREATOR,TBNAME)**
Defines synonyms of tables or views. One row per synonym.		

This table contains the following columns:

Column Name	Data Type/Description
NAME	VARCHAR(128) NOT NULL Synonym for the table or view.
CREATOR	VARCHAR(128) NOT NULL Schema of the synonym.
TBNAME	VARCHAR(128) NOT NULL Name of the table or view.
TBCREATOR	VARCHAR(128) NOT NULL Schema of the table or view.
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other.
CREATEDBY	VARCHAR(128) NOT NULL WITH DEFAULT Primary authorization ID of user who created the synonym.
CREATEDTS	TIMESTAMP NOT NULL WITH DEFAULT Time the CREATE statement was executed for this synonym.
CREATORTYPE	CHAR(1) NOT NULL WITH DEFAULT Indicates the type of creator: blank Authorization ID L Role
RELCREATED	CHAR(1) NOT NULL The release of DB2 that is used to create the object. Blank if created prior to V9.

SYSIBM.SYSTABAUTH		(DSNDB06.SYSTSTAU)
SYSIBM.DSNATX01	D	(GRANTOR,GRANTORTYPE)
SYSIBM.DSNATX02	D	(GRANTEE,TCREATOR, TTNAME,GRANTEETYPE, UPDATECOLS,ALTERAUTH, DELETEAUTH,INDEXAUTH, INSERTAUTH,SELECTAUTH, UPDATEAUTH,CAPTUREAUTH, REFERENCESAUTH,REFCOLS, TRIGGERAUTH)
SYSIBM.DSNATX03	D	(GRANTEE,GRANTEETYPE, COLLID,CONTOKEN)
SYSIBM.DSNATX04	D	(TCREATOR,TTNAME)
SYSIBM.DSNATX05	D	(TCCREATOR,TTNAME, TIMESTAMP)
Defines the privileges users hold on tables and views.		

This table contains the following columns:

Column Name	Data Type/Description
GRANTOR	VARCHAR(128) NOT NULL Authorization ID of user who granted the privileges. Could also be PUBLIC or PUBLIC *.
GRANTEE	VARCHAR(128) NOT NULL Authorization ID of user who holds the privileges, or name of application plan or package that uses the privileges. PUBLIC for a GRANT to PUBLIC. PUBLIC * for a GRANT to PUBLIC AT ALL LOCATIONS.
GRANTEETYPE	CHAR(1) NOT NULL Type of grantee: L Role P An application plan or a package. The grantee is a package if COLLID is not blank blank An authorization ID
DBNAME	VARCHAR(24) NOT NULL If privileges were received from a user with DBADM, DBCTRL, or DBMAINT authority, this is the name of the database on which the GRANTOR has that authority. Otherwise blank.
SCREATOR	VARCHAR(128) NOT NULL If the row in SYSTABAUTH was created as a result of a CREATE VIEW statement, this is the schema of a table or view referenced in that statement. Otherwise, this is same as TCREATOR.
STNAME	VARCHAR(128) NOT NULL If the row in SYSTABAUTH was created as a result of a CREATE VIEW statement, this is the name of a table or view referenced in that statement. Otherwise, this is same as TTNAME.
TCREATOR	VARCHAR(128) NOT NULL Schema of the table or view.
TTNAME	VARCHAR(128) NOT NULL Name of the table or view.

Column Name	Data Type/Description
AUTHHOWGOT	CHAR(1) NOT NULL Authorization level of user from whom privileges were received: ▪ blank—not applicable ▪ B—System DBADM ▪ C—DBCTL ▪ D—DBADM ▪ E—SECADM ▪ G—ACCESSCTRL ▪ K—SQLADM ▪ L—SYSCTRL ▪ M—DBMAINT ▪ S—SYSADM ▪ T—DATAACCESS
TIMESTAMP	CHAR(12) NOT NULL/Internal use only.
DATEGRANTED	CHAR(6) NOT NULL/Unused.
TIMEGRANTED	CHAR(8) NOT NULL/Unused.
UPDATECOLS	CHAR(1) NOT NULL blank If UPDATEAUTH applies uniformly to all columns of table or view. * If UPDATEAUTH applies to some columns but not to others. In this case, rows will exist in SYSIBM.SYSCOLAUTH with equal timestamps and PRIVILEGE equal to blank; these indicate the columns on which update privileges were granted.
ALTERAUTH	CHAR(1) NOT NULL Whether GRANTEE can alter the table: (see legend **)
DELETEAUTH	CHAR(1) NOT NULL Whether GRANTEE can delete rows from the table or view: (see legend **)
INDEXAUTH	CHAR(1) NOT NULL Whether GRANTEE can create indexes on the table: (see legend **)
INSERTAUTH	CHAR(1) NOT NULL Whether GRANTEE can insert rows into the table or view: (see legend **)
SELECTAUTH	CHAR(1) NOT NULL Whether GRANTEE can select rows from the table or view: (see legend **)
UPDATEAUTH	CHAR(1) NOT NULL Whether GRANTEE can update rows in the table or view: (see legend **)
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other.
GRANTEELOCATION	VARCHAR(128) NOT NULL WITH DEFAULT Unused.
LOCATION	VARCHAR(128) NOT NULL WITH DEFAULT Unused.
COLLID	VARCHAR(128) NOT NULL WITH DEFAULT Collection name if GRANTEE is a package ID. Otherwise blank.
CONTOKEN	CHAR(8) NOT NULL WITH DEFAULT FOR BIT DATA Consistency token if GRANTEE is a package ID. Otherwise blank.
CAPTUREAUTH	CHAR(1) NOT NULL WOTH DEFAULT Unused.

Column Name	Data Type/Description
REFERENCES AUTH	CHAR(1) NOT NULL WITH DEFAULT Whether GRANTEE can create or drop referential constraints in which the table is a parent: (see legend **)
REFCOLS	CHAR(1) NOT NULL WITH DEFAULT blank If value of REFERENCEAUTH applies to all columns in the table. * If value of REFERENCESAUTH applies to some, but not all columns. In this case, rows exist in SYSIBM.SYSCOLAUTH with equal timestamps and PRIVILEGE = R indicating the columns on which reference privileges have been granted.
GRANTEDTS	TIMESTAMP NOT NULL WITH DEFAULT Time when the GRANT statement was executed.
TRIGGERAUTH	CHAR(1) NOT NULL WITH DEFAULT Whether GRANTEE can create triggers in which the table is named as the triggering table: (see legend **)
GRANTORTYPE	CHAR(1) NOT NULL WITH DEFAULT Indicates the type of grantor: blank Authorization ID L Role
SYS_START	TIMESTAMP(12) NOT NULL System period temporal start time for transaction
SYS_END	TIMESTAMP(12) NOT NULL System period temporal end time for transaction
TRANS_START	TIMESTAMP(12) System period transaction timestamp.

Legend **:

blank	privilege not held ; G privilege held with GRANT option
Y	privilege held without GRANT option

SYSIBM.SYSTABCONST		**(DSNDB06.SYSTSTBC)**
SYSIBM.DSNCNX01	**P**	**(TBCREATOR,TBNAME, CONSTNAME)**
SYSIBM.DSNCNX02	**D**	**(IXOWNER,IXNAME)**
Defines unique constraints (primary key or unique key) on tables created in DB2 7 or above. One row per unique constraint.		

This table contains the following columns:

Column Name	Data Type/Description
CONSTNAME	VARCHAR(128) NOT NULL Name of the constraint.
TBCREATOR	VARCHAR(128) NOT NULL Schema of the table on which constraint is defined.
TBNAME	VARCHAR(128) NOT NULL Name of the table on which the constraint is defined.
CREATOR	VARCHAR(128) NOT NULL Authorization ID under which the constraint was created.

Column Name	Data Type/Description
TYPE	CHAR(1) NOT NULL Type of constraint: F (foreign key), P (primary key), U (unique key).
IXOWNER	VARCHAR(128) NOT NULL Schema of the index enforcing the constraint or blank if index has not been created yet.
IXNAME	VARCHAR(128) NOT NULL Name of the index enforcing the constraint or blank if index has not been created yet.
CREATEDTS	TIMESTAMP NOT NULL Time when the statement to create the constraint was executed.
IBMREQD	CHAR(1) NOT NULL WITH DEFAULT 'N' Whether the row came from the basic MRM tape: Y/N/other. If other, release dependency indicator.
COLCOUNT	SMALLINT NOT NULL Number of columns in the constraint.
RELCREATED	CHAR(1) NOT NULL The release of DB2® that is used to create the object. Blank if created prior to V9.

SYSIBM.SYSTABLEPART		**(DSNDB06.SYSTSTPT)**
SYSIBM.DSNDPX01	**R**	**(DBNAME,TSNAME, PARTITION)**
SYSIBM.DSNDPX02	**D**	**(STORNAME)**
SYSIBM.DSNDPX03	**D**	**(DBNAME,TSNAME, LOGICAL_PART)**
SYSIBM.DSNDPX04	**D**	**(IXCREATOR,IXNAME)**
SYSIBM.DSNDPX05	**D**	**(DBNAME,TSNAME)**
Defines table space partitions. One row per un-partitioned table space. One row per partition of a partitioned table space.		

This table contains the following columns:

Column Name	Data Type/Description
PARTITION	SMALLINT NOT NULL Partition number. 0 if table space is not partitioned.
TSNAME	VARCHAR(24) NOT NULL Name of the table space.
DBNAME	VARCHAR(24) NOT NULL Name of the database containing the table space.
IXNAME	VARCHAR(128) NOT NULL Name of the partitioned index. Blank unless this is a table using index controlled partitioning.
IXCREATOR	VARCHAR(128) NOT NULL Schema of index. . Blank unless this is a table using index controlled partitioning.
PQTY	INTEGER NOT NULL For user-managed data-sets, value is the primary space allocation in units of 4KB storage blocks or –1. For user-specified values of PRIQTY the value is set to the primary space allocation only if RUNSTATS TABLESPACE w/UPDATE(ALL) or UPDATE(SPACE) is executes, otherwise 0.

Column Name	Data Type/Description
	A value of –1 indicates that: ▪ PRIQTY was not specified for CREATE TABLESPACE ▪ -1 was most recently specified value for PRIQTY
SQTY	SMALLINT NOT NULL For user-managed data-sets, value is the secondary space allocation in units of 4KB storage blocks or –1. For user-specified values of SECQTY the value is set to the primary space allocation only if RUNSTATS TABLESPACE w/UPDATE(ALL) or UPDATE(SPACE) is executes, otherwise 0. A value of –1 indicates that: ▪ SECQTY was not specified for CREATE TABLESPACE ▪ -1 was most recently specified value for SECQTY
STORTYPE	CHAR(1) NOT NULL Type of storage allocation: E Explicit, storage group is not used I Implicit, storage group is used
STORNAME	VARCHAR(128) NOT NULL Name of storage group used for space allocation. Blank if storage group not used, or for catalog table space.
VCATNAME	VARCHAR(24) NOT NULL Name of ICF catalog used for space allocation.
CARD	INTEGER NOT NULL No. of rows in the table space or partition. The value is 2147483647 if the no. of rows is ≥ 2147483647. –1 if stats not gathered. If a LOB table space, contains the number of LOBs in the table space.
FARINDREF	INTEGER NOT NULL No. of rows that have been relocated far from their original page. Value –1 if statistics not gathered; not applicable to a LOB table space.
NEARINDREF	INTEGER NOT NULL No. of rows that have been relocated near their original page. Value –1 if statistics not gathered; not applicable to a LOB table space.
PERCACTIVE	SMALLINT NOT NULL Percentage of space occupied by rows of data from active tables. –1 if stats not gathered. –2 if LOB table space. Not applicable for hash access.
PERCDROP	SMALLINT NOT NULL Percentage of space occupied by rows of dropped tables. Zero if a segmented table space; not applicable for an auxiliary table.
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other. If other, release dependency indicator.
LIMITKEY	VARCHAR(765) NOT NULL

Column Name	Data Type/Description
	High value of the partition in an external format. If table space was converted from index-controlled partitioning to table-controlled partitioning, the value is the highest possible value for an ascending key, or the lowest possible value for a descending key. 0 if table space is not partitioned. In DB2 11 date and time values are delimited by single quotes.
FREEPAGE	SMALLINT NOT NULL No. of pages loaded before a page is left as free space.
PCTFREE	SMALLINT NOT NULL Percentage of each page left as free space.
CHECKFLAG	CHAR(1) NOT NULL WITH DEFAULT C Table space partition in check-pending mode and some rows may violate referential constraints, table check constraints, or both. D Inline length of LOB column determined when altered. I Inline length of LOB column incremented when altered. blank Table space is not partitioned or does not contain rows that violate referential constraints, table check constraints, or both.
CHECKRID	CHAR(4)/Unused.
SPACE	INTEGER NOT NULL WITH DEFAULT FOR BIT DATA No. of kilobytes of DASD allocated to the table space partition, as determined by last execution of STOSPACE or RUNSTATS. The value is 0 if the table space is not related to storage groups, or if STOSPACE or RUNSTATS has not been run. Value is –1 if DEFINE NO clause was used.
COMPRESS	CHAR(1) NOT NULL WITH DEFAULT For a table space partition, whether the COMPRESS attribute for the partition is YES. For a nonpartitioned table space, whether the COMPRESS attribute for the table space is YES: blank no compression Y compression is defined
PAGESAVE	SMALLINT NOT NULL WITH DEFAULT %, multiplied by 100, of pages saved in the table space or partition as a result of data compression. Based on average row length. Zero if no savings from compression, or if stats not gathered.
STATSTIME	TIMESTAMP NOT NULL WITH DEFAULT The date and time when RUNSTATS last updated the statistics. The default value is 0001-01-01.00.00.00.000000.
GBPCACHE	CHAR(1) NOT NULL WITH DEFAULT Group buffer pool cache option specified for this table space or partition: blank Only changed pages are cached in the group buffer pool

Column Name	Data Type/Description
	A Changed and unchanged pages are cached in the group buffer pool N No data is cached in the group buffer pool. S Only changed system pages, such as space map pages that do not contain actual data values, are cached in the group buffer pool.
CHECKRID5B	CHAR(5) NOT NULL WITH DEFAULT FOR BIT DATA Blank if table or partition is not in check-pending status or if table space is not partitioned. Else it is the RID of the first row of the table space partition that may violate referential constraints, table check constraints, or both. Value is X'0000000000' indicating that any row may violate referential constraints.
TRACKMOD	CHAR(1) NOT NULL WITH DEFAULT Track page modifications in the space map pages: blank (yes) or N (no).
EPOCH	INTEGER NOT NULL WITH DEFAULT Number incremented each time an operation changes location of rows in a table.
SECQTYI	INTEGER NOT NULL WITH DEFAULT Secondary space allocation in 4KB units. Value is 0 if storage groups are not used.
CARDF	FLOAT NOT NULL WITH DEFAULT -1 Number of rows in table space or partition or if LOB table space number of LOBs. –1 if stats are not gathered.
IPREFIX	CHAR(1) NOT NULL WITH DEFAULT 'I' The first character of the instance qualifier for this index's data set name. Can contain I or J; default is I.
ALTEREDTS	TIMESTAMP NOT NULL WITH DEFAULT Time when most recent ALTER INDEX was executed for the index. Value is "00001-01-01.00.00.00.000000' if no ALTER INDEX has been executed.
SPACEF	FLOAT(8) NOT NULL WITH DEFAULT -1 Kilobytes of DASD storage. The value is -1 if statistics have not been gathered.
DSNUM	INTEGER NOT NULL WITH DEFAULT -1 Number of data sets. The value is -1 if statistics have not been gathered.
EXTENTS	INTEGER NOT NULL WITH DEFAULT -1 Number of data set extents. The value is -1 if statistics have not been gathered.
LOGICAL_PART	SMALLINT NOT NULL WITH DEFAULT Logical partition for table spaces created with either table-controlled partitioning or index-controlled partitioning. The physical partition number is kept in column PART and is zero for partitioned table spaces created prior to V8 and for nonpartitioned table spaces
LIMITKEY_ INTERNAL	VARCHAR(512) NOT NULL WITH DEFAULT FOR BIT DATA

Column Name	Data Type/Description
	Highest value of the limit key of the partition in an internal format. Value is 0 if index-controlled partitioning instead of table-controlled.
OLDEST_VERSION	SMALLINT NOT NULL WITH DEFAULT Version number of oldest format of data in the table part and any image copies at the part level.
CREATEDTS	TIMESTAMP NOT NULL WITH DEFAULT Time when the partition was created.
AVGROWLEN	INTEGER NOT NULL WITH DEFAULT -1 Average length of rows for the tables in the table space or part.
FORMAT	CHAR(1)NOT NULL WITH DEFAULT Indicates the format of the rows in the table space or partition: R Indicates reordered row format blank Indicates basic row format or a LOB table space
REORG_LR_TS	TIMESTAMP NOT NULL WITH DEFAULT The time when the REORG or LOAD REPLACE utility last occurred. The default value is '0001-01-01.00.00.00.000000'.
RELCREATED	CHAR(1) NOT NULL The release of DB2 that is used to create the object. Blank if created prior to V9.
HASHSPACE	BIGINT NOT NULL WITH DEFAULT Amount of space, in KB, specified at partition level to override value at table level for range partitioned universal table spaces. 0—Partition by growth table space, or no override provided
HASHDATAPAGES	BIGINT NOT NULL WITH DEFAULT Number of hash pages that correspond to the value of the HASHSPACE for each partition for range partitioned universal table spaces. 0 Partition by growth table space, or no override provided
RBA_FORMAT	CHAR(1) NOT NULL WITH DAFEULT RBA/LRSN format: B - basic 6-byte E - extended 10-byte U - unknown sinceDEFINE NO used.
PCTFREE_UPD	SMALLINT NOT NULL WITH DEFAULT Percentage of freespace reserved for updates for variable length records.
PCTFREE_UPD_CALC	SMALLINT NOT NULL WITH DEFAULT Percentage of freespace reserved for updates for variable length records – calculated by DB2 or utilities.
TYPE	CHAR(1) The type of partition. **blank** The table space was created without the LOB or MEMBER CLUSTER options. If the DSSIZE column is zero, the table space is not greater than 64 gigabytes. **G** The table space was defined with the MAXPARTITIONS option with the

Column Name	Data Type/Description
	underlying structure of a universal table space. **L** The table space can be greater than 64 gigabytes. **O** The table is a LOB tablespace. **P** Implicit table space created for XML columns. **R** Range-partitioned universal table space. NULL for table spaces that were created before DB2 12.
PAGENUM	CHAR(1) NOT NULL WITH DEFAULT 'A' Format of pages for the table space and indexes created on tables in the table space, indicating absolute or relative page numbering. **A : A**bsolute addressing so that PAGENUM contains the embedded partition number. **R : R**elative addressing so that PAGENUM contains only the relative page number. NULL for objects that were created before DB2 12.
BPOOL	CHAR(8) Buffer pool used for the partition. NULL for partitions that were created before DB2 12.
PGSIZE	SMALLINT Size of pages in the table space in kilobytes. NULL for table spaces that were created before DB2 12.
DSSIZE	CHAR(1) Maximum size on a partitioned index data set. 0 for a nonpartitioned index. NULL for table spaces that were created before DB2 DB2 12.
MEMBER_CLUSTER	CHAR(1) **Y :** MEMBER CLUSTER is specified for the tablespace. **blank** MEMBER CLUSTER is not specified for the tablespace. NULL for table spaces that were created before DB2 DB2 12.
COMPRESSRATIO	SMALLINT Average percentage of bytes saved by compression on each compressed data record in the partition when the table space is defined with the COMPRESS YES attribute. This calculation includes overhead bytes for each row. The value is based on an average row length and varies depending on the actual length of the data rows. The value is -1 or 0 in the following cases: **-1** This value has not been collected **0** No compression exists or the average compressed record length is the same as or longer than the uncompressed record length.

SYSIBM.SYSTABLEPART_HIST		(DSNDB06.SYSHIST)
SYSIBM.DSNHCX01	D	(DBNAME,TSNAME, PARTITION,STATSTIME)
Contains SYSTABLEPART history (populated by RUNSTATS). You can insert, update, and delete rows in this table.		

This table contains the following columns:

Column Name	Data Type/Description
PARTITION	SMALLINT NOT NULL Partition number. 0 if table space is not partitioned.
TSNAME	VARCHAR(24) NOT NULL Name of the table space.
DBNAME	VARCHAR(24) NOT NULL Name of the database that contains the table space.
PQTY	INTEGER NOT NULL For user-managed data-sets, value is the primary space allocation in units of 4KB storage blocks or –1. For user-specified values of PRIQTY the value is set to the primary space allocation only if RUNSTATS TABLESPACE w/UPDATE(ALL) or UPDATE(SPACE) is executes, otherwise 0. A value of –1 indicates that: • PRIQTY was not specified for CREATE TABLESPACE • -1 was most recently specified value for PRIQTY If a storage group is not used, the value is 0.
SECQTYI	INTEGER NOT NULL For user-managed data-sets, value is the secondary space allocation in units of 4KB storage blocks or –1. For user-specified values of SECQTY the value is set to the primary space allocation only if RUNSTATS TABLESPACE w/UPDATE(ALL) or UPDATE(SPACE) is executes, otherwise 0. A value of –1 indicates that: ▪ SECQTY was not specified for CREATE TABLESPACE ▪ -1 was most recently specified value for SECQTY If a storage group is not used, the value is 0.
FARINDREF	INTEGER NOT NULL WITH DEFAULT -1 Number of rows that have been relocated far from their original page. The value is –1 if statistics have not been gathered. Not applicable if the table space is a LOB table space.
NEARINDREF	INTEGER NOT NULL WITH DEFAULT –1 Number of rows that have been relocated near their original page. The value is -1 if statistics have not been gathered. Not applicable if the table space is a LOB table space.
PERCACTIVE	SMALLINT NOT NULL WITH DEFAULT –1 Percentage of space occupied by rows of data from active tables. The value is –1 if statistics have not been gathered. The value is -2 if the table space is a LOB table space.

Column Name	Data Type/Description
PERCDROP	SMALLINT NOT NULL WITH DEFAULT –1 Percentage of space occupied by rows of dropped tables. The value is -1 if statistics have not been gathered. The value is 0 for segmented table spaces. Not applicable if the table is an auxiliary table.
SPACEF	FLOAT(8) NOT NULL WITH DEFAULT –1 Number of kilobytes of DASD storage allocated to the table space partition. The value is –1 if statistics have not been gathered.
PAGESAVE	SMALLINT NOT NULL Percentage of pages saved in the table space or partition as a result of defining the table space with COMPRESS YES or other compression routines. The calculation includes overhead bytes for each row, the bytes required for dictionary, and the bytes required for the current FREEPAGE and PCTFREE specification. This calculation is based on an average row length. The value is 0 if there are no savings from using data compression, or if statistics have not been gathered. The value can be negative if data compression causes an increase in the number of pages in the data set.
STATSTIME	TIMESTAMP NOT NULL If RUNSTATS updated the statistics, the date and time when the last invocation of RUNSTATS updated the statistics. The default value is '0001-01-01.00.00.00.000000'.
CARDF	FLOAT(8) NOT NULL WITH DEFAULT –1 Number of rows in the table space or partition, or if the table space is a LOB table space, the number of LOBS in the table space. The value is -1 if statistics have not been gathered.
EXTENTS	INTEGER NOT NULL WITH DEFAULT -1 Number of data set extents. The value is -1 if statistics have not been gathered.
DSNUM	INTEGER NOT NULL WITH DEFAULT -1 Data set number within the table space. For partitioned table spaces, this value corresponds to the partition number for a single partition copy, or 0 for a copy of an entire partitioned table space or index space. The value is -1 if statistics have not been gathered.
IBMREQD	CHAR(1) NOT NULL WITH DEFAULT ‘N’ Whether the row came from the basic MRM tape: Y/N/other. If other, release dependency indicator.
AVGROWLEN	INTEGER NOT NULL WITH DEFAULT -1 Average length of rows for the tables in the table space or part.

SYSIBM.SYSTABLES		(DSNDB06.SYSTSTAB)
SYSIBM.DSNDTX01	P	(CREATOR,NAME)
SYSIBM.DSNDTX02	U	(DBID,OBID,CREATOR,NAME)
SYSIBM.DSNDTX03	D	(TBCREATOR,TBNAME)
SYSIBM.DSNDTX05	D	(DBNAME,TSNAME)
Defines tables, views, or aliases. One row for each table, view, or alias.		

This table contains the following columns:

Column Name	Data Type/Description
NAME	VARCHAR(128) NOT NULL Name of the table, view, or alias.
CREATOR	VARCHAR(128) NOT NULL Schema of the table, view, or alias.
TYPE	CHAR(1) NOT NULL Type of object: A alias C clone G created global temporary table H history table M materialized query table P Implicit table created for XML columns R Archive table T table V view X auxiliary table
DBNAME	VARCHAR(24) NOT NULL For a table or view, the name of the database that contains the table space named in TSNAME. For a temporary table, an alias or view of a view, this value is DSNDB06.
TSNAME	VARCHAR(24) NOT NULL For a table or view of 1 table, the name of the table space that contains the table. For a view of more than 1 table, the name of a table space that contains one of the tables. For a created temporary table, the value is SYSPKAGE. For a view of a view, the value is SYSVIEWS. For an alias it is SYSDBAUT.
DBID	SMALLINT NOT NULL Internal ID of database. 0 for view an alias, or a created temporary table. Non-zero if view has an INSTEAD OF trigger defined.
OBID	SMALLINT NOT NULL Internal ID of table. 0 for view, alias, or a created temporary table. Non-zero if view has an INSTEAD OF trigger defined.
COLCOUNT	SMALLINT NOT NULL No. of cols in table or view. 0 for alias.
EDPROC	VARCHAR(24) NOT NULL Name of edit-procedure. Blank for view or alias or a table without an edit procedure.
VALPROC	VARCHAR(24) NOT NULL Name of the validation procedure. Blank for view or alias or a table without a validation procedure.
CLUSTERTYPE	CHAR(1) NOT NULL Whether RESTRICT ON DROP applies: blank The table can be dropped.

Column Name	Data Type/Description
	Y The table (and its table space) cannot be dropped
CLUSTERID	INTEGER/Unused.
CARD	INTEGER/Unused.
NPAGES	INTEGER NOT NULL Total no. of pages on which rows of the table appear. –1 if stats not gathered or for view, alias or created temporary table or auxiliary. Updateable column
PCTPAGES	SMALLINT NOT NULL Percentage of total pages of table space that either contain rows of the table or are formatted for rows, even if the page contains none. If table space is segmented, the % of total pages in the set of segments assigned to the table. –1 if stats not gathered or for view, alias, created temporary table, or auxiliary table. Updateable column.
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other..
REMARKS	VARCHAR(762) NOT NULL Remarks provided in COMMENT statement, else empty string.
PARENTS	SMALLINT NOT NULL No. of relationships in which the table is a dependent. 0 for view, alias, or created temporary table, or materialized query table.
CHILDREN	SMALLINT NOT NULL No. of relationships in which the table is a parent. 0 for view, alias, or created temporary table, or materialized query table.
KEYCOLUMNS	SMALLINT NOT NULL No. of columns in the table's primary key. 0 for view, alias, or created temporary table.
RECLENGTH	SMALLINT NOT NULL Max. length of any row in table. Length is 8 + n + l, where: 8 accounts for header (6 bytes) and ID map entry (2 bytes) N is 10 if table has an edit procedure, else 0 L is sum of max. col. lengths Determine max. column length: Add 1 byte for null indicator for column that allows nulls. Add 2 bytes for length indicator for variable-length column. Use 4 bytes for length of a LOB column. Use 19 bytes for length of a ROWID column. RECLENGTH is 0 for auxiliary tables, views, and aliases.
STATUS	CHAR(1) NOT NULL I Table definition is incomplete because it lacks parent index. R An error occurred when an attempt was made to regenerate the internal representation of the view. X Table has a parent index. blank Table has no parent key or is a

Column Name	Data Type/Description
	catalog table, or row is for a view or alias; the definition of the table is complete.
KEYOBID	SMALLINT NOT NULL Internal DB2 ID of index that enforces uniqueness of table's primary key. 0 if not applicable.
LABEL	VARCHAR(90) NOT NULL Label provided in LABEL ON statement, else empty string.
CHECKFLAG	CHAR(1) NOT NULLWITH DEFAULT C Table space containing table is in check pending mode and some rows may violate referential constraints, table check constraints, or both; or the table is a materialized query table that might contain inconsistent data blank Table contains no rows that violate referential constraints, table check constraints, or both; or the table is a materialized query table that contains consistent data; or row is for view, alias, or temporary table space.
CHECKRID	CHAR(4) An EDIT PROCEDURE on this table is defined without row SENSITIVE attribute : value is 'FFFFFF00'. Other values indicates edit procedure is defined with SENSITIVITY.
AUDITING	CHAR(1) NOT NULL WITH DEFAULT Value of audit option: A AUDIT ALL C AUDIT CHANGE blank AUDIT NONE
CREATEDBY	VARCHAR(128) NOT NULL WITH DEFAULT Authorization ID of user who created the table, view, or alias.
LOCATION	VARCHAR(128) NOT NULL WITH DEFAULT Location name of object for alias. Blank for table, view, or an alias that was not defined with a three-part object name.
TBCREATOR	VARCHAR(128) NOT NULL WITH DEFAULT ▪ For an alias, the schema of the referred to table or view ▪ For a base table that is involved in a clone relationship, the name of the creator of the clone table ▪ For a clone table that is involved in a clone relationship, the name of the creator of the base table ▪ Otherwise, TBCREATOR is blank
TBNAME	VARCHAR(128) NOT NULL WITH DEFAULT ▪ For an alias, the name for the referred to table or view ▪ For a base table that is involved in a clone relationship, the name of the clone table ▪ For a clone table that is involved in a clone relationship, the name of the base table ▪ Otherwise, TBNAME is blank
CREATEDTS	TIMESTAMP NOT NULL WITH DEFAULT

Column Name	Data Type/Description
	The timestamp when the table, view, or alias was created.
ALTEREDTS	TIMESTAMP NOT NULL WITH DEFAULT For a table, the timestamp when the table was last altered. The value equals CREATEDTS if no ALTER occurred or if row refers to a view or alias.
DATACAPTURE	CHAR(1) NOT NULL WITH DEFAULT DATACAPTURE option: blank No Y Yes For created temporary tables DATACAPTURE is always blank.
RBA1	CHAR(10) NOT NULL WITH DEFAULT FOR BIT DATA Log RBA when table was created. Value is hex zeroes if log RBA unknown or object is a view, alias, or created temporary table. In data sharing, value is the LRSN. Value of 20 x'0' indicates the log rba is unknown
RBA2	CHAR(10) NOT NULL WITH DEFAULT FOR BIT DATA Log RBA when table was last altered. Value is hex zeroes if log RBA unknown or object is a view, alias, or created temporary table. Value equals RBA1 if no ALTER occurred. In data sharing, value is the LRSN. Value of 20 x'0' indicates the log rba is unknown
PCTROWCOMP	SMALLINT NOT NULL WITH DEFAULT % of rows compressed within the total number of active rows in the table. –1 if the row describes a view, alias, created temporary table or auxiliary table, or if statistics not gathered. Updateable Column
STATSTIME	TIMESTAMP NOT NULL WITH DEFAULT The date and time when RUNSTATS last updated the statistics. The default value is 0001-01-01.00.00.00.000000. For a created temporary table, the value is always the default.
CHECKS	SMALLINT NOT NULL WITH DEFAULT The number of check constraints defined on a table. 0 if the row describes a view or alias, created temporary table, or a materialized query table, or if no constraints are defined on the table.
CARDF	FLOAT NOT NULL WITH DEFAULT -1 Total no. of rows in table or LOBs in an auxiliary table. -1 if stats not gathered or for a view, alias, or created temporary table space. Updateable column.
CHECKRID5B	CHAR(5) NOT NULL WITH DEFAULT FOR BIT DATA Blank if the table or partition is not in check pending status, if the table space is not partitioned, or if the table is a created temporary table. Else, the RID of the first row of the table space partition that can

Column Name	Data Type/Description
	violate referential constraints, table check constraints, or both; or the value is x'0000000000', indicating that any row can violate referential constraints.
ENCODING_ SCHEME	CHAR(1) NOT NULL WITH DEFAULT 'E' Default encoding scheme for tables, views and local aliases. ▪ E—EBCDIC ▪ A—ASCII ▪ M—Multiple CCSID set or multiple encoding schemes ▪ U—Unicode ▪ blank— remote aliases Value of 'E' for tables in non-work file databases; and blank for databases created prior to V5 or the default database, DSNDB04.
TABLESTATUS	VARCHAR(30) NOT NULL WITH DEFAULT Indicates the reason for an incomplete table definition: F Table lacks a required BUSINESS_TIME WITHOUT OVERLAPS index on a foreign key. L Incomplete because an auxiliary table or auxiliary index has not been defined for a LOB column. P Incomplete because the table lacks a primary index. R Incomplete because the table lacks a required index on a row ID. U Incomplete because the table lacks a required index on a unique key. V An error occurred when an attempt was made to regenerate the internal representation of the view. blank Definition is complete.
NPAGESF	FLOAT(8) NOT NULL WITH DEFAULT –1 Number of pages used by the table. -1 if statistics have not been gathered or this is an auxiliary table.
SPACEF	FLOAT(8) NOT NULL WITH DEFAULT -1 Kilobytes of DASD storage. -1 if statistics have not been gathered.
AVGROWLEN	INTEGER Average length of rows for the tables in the table space. If the table space is compressed, the value is the compressed row length. If the table space is not compressed, the value is the uncompressed row length. -1 if statistics have not been gathered.
RELCREATED	CHAR(1) NOT NULL WITH DEFAULT Release of DB2 that was used to create the object:
NUM_DEP_MQTS	SMALLINT NOT NULL WITH DEFAULT No. of dependent materialized query tables. If row describes an alias or created temporary table, or if no materialized query tables are defined on the table, value is zero.

Column Name	Data Type/Description
VERSION	SMALLINT NOT NULL WITH DEFAULT The version of the data row format for this table. If a version-creating alter operation has never occurred against this table, value is zero. A value of 900 indicates an ALTER TABLE DROP COLUMN was executed.
PARTKEYCOLNUM	SMALLINT NOT NULL WITH DEFAULT The number of columns in the partitioning key. For tables that do not have partitioning or use index-controlled partitioning, value is zero. For tables that use table-controlled partitioning, the value is non-zero.
SPLIT_ROWS	CHAR(1) NOT NULL WITH DEFAULT Value is blank, except for VOLATILE tables, which have 'Y' in the field to tell DB2 to use index access on table whenever possible.
SECURITY_LABEL	CHAR(1) NOT NULL Only valid if the TYPE column is T or M. The value tells whether the table has multilevel security: blank (table does not) or R (table does).
OWNER	VARCHAR(128) NOT NULL WITH DEFAULT Schema of the table, view, or alias, blank for tables, views or aliases that were created in a DB2 release prior to V9.
APPEND	CHAR(1) NOT NULL WITH DEFAULT Indicates whether the APPEND option is specified for the table: Y (specified) or N (not specified).
OWNERTYPE	CHAR(1) NOT NULL WITH DEFAULT Indicates the type of owner: blank Authorization ID L Role
CONTROL	CHAR(1) NOT NULL WITH DEFAULT Access control enforcement: blank – None B – Both row and column C – Column R – Row
VERSIONING_ SCHEMA	VARCHAR(128) NOT NULL WITH DEFAULT Schema name of: ▪ History table if system-maintained temporal table with versioning. ▪ System maintained temporal table if history table ▪ Bland – No versioning
VERSIONING_ TABLE	VARCHAR(128) NOT NULL WITH DEFAULT Table name of: ▪ History table if system-maintained temporal table with versioning. ▪ System maintained temporal table if history table ▪ Bland – No versioning
HASHKEYCOLUMNS	SMALLINT NOT NULL WITH DEFAULT Number of columns in hash key. 0—View, alias or created temporary table
ARCHIVING_ SCHEMA	VARCHAR(128) NOT NULL WITH DEFAULT Schema name: ▪ Table is archive-enabled : schema name of archive table.

Column Name	Data Type/Description
	▪ Table is archive table : schema of archive-enabled table. ▪ Blank for others.
ARCHIVING_ TABLE	VARCHAR(128) NOT NULL WITH DEFAULT Table name: ▪ Table is archive-enabled : table name of archive table. ▪ Table is archive table : table name of archive-enabled table. ▪ Blank for others.
STATS_ FEEDBACK	CHAR(1) NOT NULL When a query qualifies for statistics collection based on DSNZPARM STATFDBK_SCOPE, this controls if statistics recommendations for this table are placed in SYSSTATFEEDBACK. Updatable column: Set this flag to 'Y' to enable collection.
REGENERATETS	TIMESTAMP(12) NOT NULL Timestamp when object was regenerated.

SYSIBM.SYSTABLESPACE		**(DSNDB06.SYSTSTSP)**
SYSIBM.DSNDSX01	**P**	**(DBNAME,NAME)**
Defines table spaces. One row per table space.		

This table contains the following columns:

Column-name	Data Type/Description
NAME	VARCHAR(24) NOT NULL Name of the table space.
CREATOR	VARCHAR(128) NOT NULL Schema of the table space.
DBNAME	VARCHAR(24) NOT NULL Name of database containing the table space.
DBID	SMALLINT NOT NULL Internal ID of the database.
OBID	SMALLINT NOT NULL Internal ID of table space file descriptor.
PSID	SMALLINT NOT NULL Internal ID of table space page set descriptor.
BPOOL	CHAR(8) NOT NULL Name of buffer pool used for the table space.
PARTITIONS	SMALLINT NOT NULL No. of partitions in the table space. 0 if table space is not a partitioned table space.
LOCKRULE	CHAR(1) NOT NULL Lock size of the table space: A any L LOB P page R row S table space T table X implicitly created XML table space
PGSIZE	SMALLINT NOT NULL Size of pages in table space in kilobytes.

Column-name	Data Type/Description
ERASERULE	CHAR(1) NOT NULL Whether data sets are erased when dropped: Y/N. No meaning if partitioned table space.
STATUS	CHAR(1) NOT NULL Availability status of table space: A available C definition incomplete because table space doesn't use table-controlled partitioning and a partitioning index has not been created. P table space in check-pending mode S table space in check-pending mode with scope less than entire table T definition incomplete because no table has been created
IMPLICIT	CHAR(1) NOT NULL Whether table space was created implicitly: Y/N.
NTABLES	SMALLINT NOT NULL No. of tables defined in the table space.
NACTIVE	INTEGER NOT NULL No. of active pages in the table space, or pages formatted for rows, even if the page contains none. 0 if stats not gathered.
DSETPASS	VARCHAR(24)/Unused.
CLOSERULE	CHAR(1) NOT NULL Whether data sets are candidates for closing when the limit on open data sets is reached: Y/N.
SPACE	INTEGER No. of kilobytes of DASD storage allocated to table space (from STOSPACE). 0 if table space not related to a storage group. If table space is partitioned, this is the no. of kilobytes of DASD allocated to all partitions that are storage-group defined.
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other.
ROOTNAME	VARCHAR(54)/Internal use only.
ROOTCREATOR	VARCHAR(24)/Internal use only.
SEGSIZE	SMALLINT NOT NULL WITH DEFAULT No. of pages in each segment of a segmented table space. 0 if table space is not segmented.
CREATEDBY	VARCHAR(128) NOT NULL WITH DEFAULT Authorization ID of user who created the table space.
STATSTIME	TIMESTAMP NOT NULL WITH DEFAULT The date and time when RUNSTATS last updated the statistics. The default value is 0001-01-01.00.00.00.000000. Updateable column.
LOCKMAX	INTEGER The maximum number of page or row locks per user to acquire for the table or the table space before escalating to the next locking level: 0 Lock escalation does not occur

Column-name	Data Type/Description
	n Specified maximum number -1 LOCKMAX SYSTEM is specified, lock escalation is determined by the value of the LOCKS PER TABLE(SPACE) installation parameter.
TYPE	CHAR(1) NOT NULL WITH DEFAULT Indicates the type of table space: blank Table space was created without any of the following options: DSSIZE, LARGE, LOB, and MEMBER CLUSTER. G Table space was defined with MAXPARTITIONS option with the underlying structure of a universal table space. L Table space can be greater than 64GB. O Table space was defined with the LOB option (table space is a LOB table space). P Implicit table space created for XML columns. R Range-partitioned universal table space.
CREATEDTS	TIMESTAMP NOT NULL WITH DEFAULT Time that the CREATE statement was executed for this table space.
ALTEREDTS	TIMESTAMP NOT NULL WITH DEFAULT Time the most recent ALTER TABLESPACE statement executed for this table space
ENCODING_ SCHEME	CHAR(1) NOT NULL WITH DEFAULT 'E' Default encoding scheme for the table space: E EBCDIC A ASCII U Unicode blank For table spaces in a work file or a TEMP database Value of 'E' for tables in a non-work file; blank for databases created prior to V5 or the default database, DSNDB04.
SBCS_CCSID	INTEGER NOT NULL WITH DEFAULT Default SBCS CCSID for the table space. Value of zero for databases created prior to V5 or a table space in a TEMP database.
DBCS_CCSID	INTEGER NOT NULL WITH DEFAULT Default DBCS CCSID for the table space. Value of 0 for a database created prior to V5 or a table space in a TEMP database.
MIXED_CCSID	INTEGER NOT NULL WITH DEFAULT Default MIXED CCSID for the table space. Value of 0 for a database created prior to V5 or a table space in a TEMP database.
MAXROWS	SMALLINT NOT NULL DEFAULT 255 Maximum number of rows DB2 will place on a data page; the default is 255. For a LOB table space, the value is 0 to indicate the column is not applicable.
LOCKPART	CHAR(1)/Unused.

Column-name	Data Type/Description
LOG	CHAR(1) NOT NULL WITH DEFAULT 'Y' Log changes to table space: Y Table space has LOGGED attribute. N Table space has NOT LOGGED attribute, logging is suppressed for all indexes, all auxiliary tables associated with tables in the table space. X LOB or XML table space has NOT LOGGED attribute.
NACTIVEF	FLOAT NOT NULL WITH DEFAULT -1 Number of active pages in table space (formatted for rows). -1 if stats not gathered. Updateable column.
DSSIZE	INTEGER NOT NULL WITH DEFAULT Max size of data set in kilobytes. 0 – Table space created prior to V10 and not converted to partitioned by growth table space.
OLDEST_ VERSION	SMALLINT NOT NULL WITH DEFAULT The version number of the oldest format of data in the table space and any image copies.
CURRENT_ VERSION	SMALLINT NOT NULL WITH DEFAULT The version number describing the newest format of data in the table space. Zero means table space has never had versioning. Number wraps to zero after version number hits maximum value.
AVGROWLEN	INTEGER NOT NULL WITH DEFAULT –1 Average length of rows for the tables in the table space or partition. If it is compressed, the value is the compressed row length. If not, the value is the uncompressed row length. –1 if statistics have not been gathered.
SPACEF	FLOAT NOT NULL WITH DEFAULT Kilobytes of DASD for the stogroup. –1 if statistics have not been gathered. Updateable column.
MAXPARTITIONS	SMALLINT NOT NULL WITH DEFAULT Identifies the maximum number of partitions to which the table space can grow. 0 if the table space is not partitioned or is range partitioned but not a universal table space.
CREATORTYPE	CHAR(1) NOT NULL Indicates the type of creator: Blank Authorization ID L Role
INSTANCE	SMALLINT NOT NULL WITH DEFAULT INSTANCE indicates the data set instance number of the current base object (table and index).
CLONE	CHAR(1) NOT NULL WITH DEFAULT Indicates whether the table space contains any objects that are involved in a clone relationship: Y Table space contains objects that are involved in a clone relationship

Column-name	Data Type/Description
	N Table space does not contain any objects that are involved in a clone relationship
RELCREATED	CHAR(1) NOT NULL The release of DB2® that is used to create the object. Blank if created prior to V9.
MEMBER_ CLUSTER	SMALLINT NOT NULL WITH DEFAULLT MEMBER CLUSTER specified: Y (YES) or blank (NO).
ORGANIZATION TYPE	CHAR(1) NOT NULL WITH DEFAULT Table space organization blank – Not known H – Hash
HASHSPACE	BIGINT NOT NULL WITH DEFAULT Space to be allocated in KB for: Table space if Partition by growth Partition if Range partitioned universal
HASHDATAPAGES	BIGINT NOT NULL WITH DEFAULT Number of hash data pages to preallocate for fixed hash space. Partition by growth – in table space Range partitioned universal – per partition 0 – non-hash table spaces or table spaces changed to hash and have not been reorganized
PAGENUM	CHAR(1) NOT NULL WITH DEFAULT 'A' Format of pages for the table space and indexes created on tables in the table space, indicating absolute or relative page numbering. **A :** Absolute addressing. **R :** Relative addressing. NULL for objects that were created before DB2 DB2 12.
INSERETALG	SMALLINT NOT NULL The insert algorithm level for tables in this table space. **0 :** The insert algorithm level for tables in this tablespace is determined by the DEFAULT_INSERT_ALGORITHM subsystem parameter. **1 :** The insert algorithm level for tables in this tablespace is the basic insert algorithm. **2 :** The insert algorithm level for tables in this tablespace is the fast insert algorithm when the MEMBER CLUSTER option is specified.
PQTY	INTEGER For user-managed data sets, the value is the primary space allocation in units of 4 KB storage blocks or -1. PQTY is based on a value of PRIQTY in the appropriate CREATE or ALTER TABLESPACE statement. Unlike PQTY, however, PRIQTY accepts space in 1 KB units. A value of -1 indicates that one of the following cases is true:

Column-name	Data Type/Description
	PRIQTY was not specified for a CREATE TABLESPACE statement or for any subsequent ALTER TABLESPACE statements. -1 was the most recently specified value for PRIQTY, either on the CREATE TABLESPACE statement or a subsequent ALTER TABLESPACE statement. NULL when unknown for objects created prior to DB2 12.
STORTYPE	CHAR(1) Type of storage allocation: E : Explicit (storage group not used). I : Implicit (storage group used). Null if created prior to DB2 12.
STORNAME	VARCHAR(128) Name of storage group. Null if created prior to DB2 12.
VCATNAME	VARCHAR(24) Name of ICF catalog. Null if created prior to DB2 12.
FREEPAGE	SMALLINT Number of pages loaded before a page is left free.
PCTFREE	SMALLINT Percentage of each page left as free.
COMPRESS	CHAR(1) Indicates the following: **Y :** Compression is defined for the tablespace **blank** No compression Null when the value is unknown for objects created prior to DB2 12.
GBPCACHE	CHAR(1) Group buffer pool cache option specified for this index or partition. **Blank** Only changed pages are cached in the group buffer pool. **A** Changed and unchanged pages are cached in the group buffer pool. **N** No data is cached in the group buffer pool. This column contains the null value when the value
TRACKMOD	CHAR(1) Whether to track the page modifications in the space map pages: **N / Y** **NULL** when the value is unknown for objects created prior to DB2 12.
SECQTYI	INTEGER Secondary space allocation in units of 4 KB storage. For user-managed data sets, the value is the secondary space allocation in units of 4 KB blocks. This column contains the null value when the value is unknown for objects created prior to DB2 12.

Column-name	Data Type/Description
PCTFREE_UPD	SMALLINT The percentage of free space that is reserved for updates to variable length records, as defined when the object as created or altered. This column contains the null value when the value is unknown for objects created prior to DB2 12.
PCTFREE_UPD_ CALC	SMALLINT The percentage of free space that is reserved for updates to variable length records, calculated by DB2 or utilities. NULL when the value is unknown for objects created prior to DB2 12.
COMPRESSRATIO	SMALLINT NOT NULL WITH DEFAULT -1 NOT NULL Average percentage of bytes saved by compression on each compressed data record in the table space when the table space is defined with the COMPRESS YES attribute. This calculation includes overhead bytes for each row. The value is based on an average row length and varies depending on the actual length of the data rows. **-1** This value has not been collected **0** No compression exists or the average compressed record length is the same as or longer than the uncompressed record length.

SYSIBM.SYSTABLESPACESTATS		**(DSNDB06.SYSTSTSS)**
SYSIBM.DSNRTX01	**U**	**(DBID,PSID,PARTITION, INSTANCE)**
Contains real-time statistics for table spaces. You can insert, update, and delete rows in this table.		

SYSIBM.SYSTABSPACESTATS_H	**(DSNDB06.SYSTSTSH)**
History table for SYSTABLESPACESTATS for the business period temporal design.	

This table contains the following columns:

Column Name	Data Type/Description
UPDATESTATSTIME	TIMESTAMP NOT NULL WITH DEFAULT The timestamp that the row in the TABLESPACESTATS table is inserted or updated.
NACTIVE	INTEGER The number of active pages in the table space or partition.
NPAGES	INTEGER The number of distinct pages with active rows in the partition or table space.
EXTENTS	SMALLINT The amount of space, in KB, that is allocated to the table space or partition.

Column Name	Data Type/Description
	For multi-piece, linear page sets, this value is the amount of space in all data sets. A null value indicates the amount of space is unknown.
LOADLASTTIME	TIMESTAMP The timestamp that the LOAD REPLACE utility was last run on the table space or partition. A null value indicates that the LOAD REPLACE utility has never been run on the table space or partition or that the timestamp is unknown.
REORGLASTTIME	TIMESTAMP The timestamp that the REORG utility was last run on the table space or partition or create timestamp if never REORGed. A null value indicates that the LOAD REPLACE utility has never been run on the table space or partition or that the timestamp is unknown.
REORGINSERTS	INTEGER The number of records or LOBs that have been inserted into the table space or partition or loaded into the table space or partition using the LOAD utility specified without the REPLACE option since the last time the REORG or LOAD REPLACE utilities were run. A null value indicates that the number of inserted records or LOBs is unknown.
REORGDELETES	INTEGER The number of records or LOBs that have been deleted from the table space or partition since the last time the REORG or LOAD REPLACE utilities were run or since object was created. A null value indicates that the number of deleted records or LOBs is unknown.
REORGUPDATES	INTEGER The number of rows that have been updated in the table space or partition since the last time the REORG or LOAD REPLACE utilities were run. A null value indicates that the number of updated rows is unknown.
REORGUNCLUSTINS	INTEGER The number of records that were inserted and not well clustered since the last REORG or LOAD REPLACE or since object was created. A record is well-clustered if the record is inserted into a page that is within 16 pages of the ideal candidate page. The clustering index determines the ideal candidate page. A null value indicates that the number of not well clustered pages is unknown.
REORGDISORGLOB	INTEGER The number of LOBs that were inserted since the last REORG or LOAD REPLACE, or

Column Name	Data Type/Description
	since object was created, that are not perfectly chunked. A LOB is perfectly chunked if the allocated pages are in the minimum number of chunks. A null value indicates that the number of not perfectly chunked LOBs is unknown.
REORGMASSDELETE	INTEGER The number of mass deletes from a segmented or LOB table space, or the number of dropped tables from a segmented table space since the last time the REORG or LOAD REPLACE utilities were run or since object was created. A null value indicates that the number of mass deletes is unknown.
REORGNEARINDREF	INTEGER The number of overflow records that are created and relocated near the pointer record since the last time the REORG and LOAD REPLACE utilities were run or since object was created. For non-segmented table spaces, a page is near the present page if the two page numbers differ by 16 or less. For segmented table spaces, a page is near the present page if the two page numbers differ by SEGSIZE*2 or less. A null value indicates that the number of overflow records that are near the pointer record is unknown.
REORGFARINDREF	INTEGER The number of overflow records that are created and relocated far from the pointer record since the last time the REORG and LOAD REPLACE utilities were run or since object was created. For non-segmented table spaces, a page is far from the present page if the two page numbers differ by more than 16. For segmented table spaces, a page is far from the present page if the two page numbers differ by at least (SEGSIZE*2) +1. A null value indicates that the number of overflow records that are near the pointer record is unknown.
STATSLASTTIME	TIMESTAMP The timestamp of the last time that the RUNSTATS utility is run on the table space or partition.
STATSINSERTS	INTEGER The number of records or LOBs that have been inserted into the table space or partition or loaded into the table space or partition using the LOAD utility specified without the REPLACE option since the last time that the RUNSTATS utility was run or since object was created. A null value indicates that the number of inserted records or LOBs is unknown.

Column Name	Data Type/Description
STATSDELETES	INTEGER The number of records or LOBs that have been deleted from the table space or partition since the last time that the RUNSTATS utility was run or since object was created. A null value indicates that the number of deleted records or LOBs is unknown.
STATSUPDATES	INTEGER The number of rows that have been updated in the table space or partition since the last time that the RUNSTATS utility was run or since object was created. A null value indicates that the number of updated rows is unknown.
STATSMASSDELETE	INTEGER The number of mass deletes from a segmented or LOB table space, or the number of tables that are dropped from a segmented table space, since the last time the RUNSTATS utility was run or since object was created. A null value indicates that the number of mass deletes is unknown.
COPYLASTTIME	TIMESTAMP The timestamp of the last full or incremental image copy of the table space or partition or since object was created. A null value indicates that the COPY utility has never been run on the table space or partition. A null value can also indicate that the timestamp of the last image copy is unknown.
COPYUPDATED PAGES	INTEGER The number of distinct types that have been updated since the last time that the COPY utility was run or since object was created. A null value indicates that the number of updated pages is unknown.
COPYCHANGES	INTEGER The number of insert, update, and delete operations, or the number of records loaded, since the last time that the COPY utility was run or since object was created. A null value indicates that the number of insert, update, and delete operations or the number of records loaded is unknown.
COPYUPDATELRSN	CHAR(10) FOR BIT DATA The LRSN or RBA of the first update that occurs after the last time the COPY utility was run or since object was created. A null value indicates that the LRSN or RBA is unknown.
COPYUPDATETIME	TIMESTAMP The timestamp of the first update that occurs after the last time that the COPY utility was run or since object was created. A null value indicates that the timestamp

Column Name	Data Type/Description
	is unknown. If value is x'FFFFFFFFFFFF' indicates RBA/LRSN exceeds the 6-byte limit.
IBMREQD	CHAR(1) NOT NULL A value of Y indicates that the row came from the basic machine-readable material (MRM) tape.
DBID	SMALLINT NOT NULL The internal identifier of the database. This column is used to map a DBID to its statistics.
PSID	SMALLINT NOT NULL The internal identifier of the table space page set descriptor. This column is used to map a PSID to its statistics.
PARTITION	SMALLINT NOT NULL The data set number within the table space. This column is used to map a data set number in a table space to its statistics. For partitioned table spaces, this value corresponds to the partition number for a single partition. For non-partitioned table spaces, this value is 0.
INSTANCE	SMALLINT NOT NULL WITH DEFAULT 1 Indicates if the object is associated with data set instance 1 or 2.
SPACE	BIGINT The amount of space, in KB, that is allocated to the table space or partition. For multi-piece, linear page sets, this value is the amount of space in all data sets. A null value indicates the amount of space is unknown.
TOTALROWS	BIGINT The number of rows or LOBs that are in the table space or partition.
DATASIZE	BIGINT The total number of bytes that row data occupy in the data rows or LOB rows.
UNCOMPRESSEDDATASIZE	BIGINT/Not used, always 0.
DBNAME	VARCHAR(24) NOT NULL The name of the database. This column is used to map a database to its statistics.
NAME	VARCHAR(24) NOT NULL The name of the table space. This column is used to map a table space to its statistics.
REORGCLUSTE RSENS	BIGINT Number of times data red by clustering sequence sensitive SQL statements since last REORG or LOAD REPLACE, or since object creation.
REORGSCANACCESS	BIGINT Number of times data accessed for SELECT, FETCH, searched UPDATE, or searched DELETE since last CREATE, LOAD

Column Name	Data Type/Description
	REPLACE or REORG. NULL – unknown
REORGHASHACCESS	BIGINT Number of times data accessed using hash access for SELECT, FETCH, searched UPDATE, searched DELETE, or RI since last CREATE, LOAD REPLACE or REORG. NULL – unknown
HASHLASTUSED	DATE Date when last hash access used for SELECT, FETCH, searched UPDATE, searched DELETE, or RI
DRIVETYPE	CHAR(3) NOT NULL WITH DEFAULT Drive type: HDD – Hard Disk Drive SDD – Solid Disk Drive (any volume SDD for Multi-volume data sets, first volume for multi-place linear page sets)
LPFACILITY	CHAR(1) If the disk control unit has high performance LP facility: N (no), Y (yes), NULL (unknown).
STATS01	BIGINT/Reserved for future IBM use.
UPDATESIZE	BIGINT Net number of bytes added/removed by UPDATEs since object was created or since last REORG/LOAD REPLACE
LASTDATACHANGE	TIMESTAMP/Last update
SYS_START	TIMESTAMP(12) NOT NULL System period temporal start time for transaction
SYS_END	TIMESTAMP(12) NOT NULL System period temporal end time for transaction
TRANS_START	TIMESTAMP(12) System period transaction timestamp.

SYSIBM.SYSTABLES_HIST		**(DSNDB06.SYSHIST)**
SYSIBM.DSNHDX01	**D**	**(CREATOR,NAME,STATSTIME)**
Contains SYSTABLES history (populated by RUNSTATS). You can insert, update, and delete rows in this table.		

This table contains the following columns:

Column Name	Data Type/Description
NAME	VARCHAR(128) NOT NULL Name of the table, view, or alias.
CREATOR	VARCHAR(128) NOT NULL Schema of the table, view, or alias.
DBNAME	VARCHAR(24) NOT NULL For a table, or a view of tables, the name of the database that contains the table space named in TSNAME. For a temporary table, an alias, or a view of a view, the value is DSNDB06.
TSNAME	VARCHAR(24) NOT NULL For a table, or a view of one table, the name of the table space that contains the table. For a view of more than one table, the name of a

Column Name	Data Type/Description
	table space that contains one of the tables. For a created temporary table, the value is SYSPKAGE. For a view of a view, the value is SYSVIEWS. For an alias, it is SYSDBAUT.
COLCOUNT	SMALLINT NOT NULL Number of columns in the table or view. The value is 0 if the row describes an alias.
PCTPAGES	SMALLINT NOT NULL WITH DEFAULT -1 Percentage of active table space pages that contain rows of the table. A page is termed active if it is formatted for rows, regardless of whether it contains any. If the table space is segmented, the percentage is based on the number of active pages in the set of segments assigned to the table. The value is -1 if statistics have not been gathered, or the row describes a view, alias, created temporary table, or auxiliary table.
PCTROWCOMP	SMALLINT NOT NULL WITH DEFAULT -1 Percentage of rows compressed within the total number of active rows in the table. This includes any row in a table space that is defined with COMPRESS YES. The value is –1 if statistics have not been gathered, or the row describes a view, alias, created temporary table, or auxiliary table.
STATSTIME	TIMESTAMP NOT NULL If RUNSTATS updated the statistics, the date and time when the last invocation of RUNSTATS updated the statistics. The default value is '0001-01-01.00.00.00.000000'. For a created temporary table, the value of STATSTIME is always the default value.
CARDF	FLOAT(8) NOT NULL WITH DEFAULT -1 Total number of rows in the table or total number of LOBs in an auxiliary table. The value is -1 if statistics have not been gathered or the row describes a view, alias, or created temporary table.
NPAGESF	FLOAT(8) NOT NULL WITH DEFAULT –1 Total number of pages on which rows of the partition appear. The value is -1 if statistics have not been gathered.
AVGROWLEN	INTEGER NOT NULL WITH DEFAULT -1 Average row length of the table specified in the table space. The value is -1 if statistics have not been gathered.
SPACEF	FLOAT(8) NOT NULL WITH DEFAULT -1 Kilobytes of DASD storage. The value is -1 if statistics have not been gathered. Updateable column.
IBMREQD	CHAR(1) NOT NULL WITH DEFAULT 'N' Whether the row came from the basic MRM tape: Y/N/other. If other, release dependency indicator.

SYSIBM.SYSTABLES_PROFILES		(DSNDB06.SYSTSTPF)
SYSIBM.DSNPRX01	P	(SCHEMA,TBNAME, PROFILE_TYPE)
Contains information for each profile associated with a table. One row per profile.		

This table contains the following columns:

Column Name	Data Type/Description
SCHEMA	VARCHAR(128) NOT NULL Schema (qualifier) for table
TBNAME	VARCHAR(128) NOT NULL Table name
PROFILE_TYPE	VARCHAR(32) NOT NULL Type of profile.
PROFILE_MODE	VARCHAR(32)/IBM internal use only.
PROFILE_TEXT	CLOB(1M)/Profile test.
ROWID	ROWID NOT NULL GENERATED ALWAYS ROWID for PROFILE_TEXT LOB
PROFILE_UPDATE	TIMESTAMP NOT NULL Time profile was inserted or updated.
PROFILE_USED	TIMESTAMP/IBM internal use only.

SYSIBM.SYSPROFILE_TEXT		(DSNDB06.SYSTSxxx)
SYSIBM.DSNPRX02	U	(AUXID,AUXVER)
Auxiliary table for PROFILE_TEXT LOB columns of SYSTABLES_PROFILES.		

This table contains the following column:

Column Name	Data Type	Description
AUXID	VARCHAR(17)	ID of auxiliary data.
AUXVER	SMALLINT	Version of auxiliary data.
AUXVALUE	CLOB(2M) NOT NULL WITH DEFAULT	Trigger text.

SYSIBM.SYSTABSTATS		(DSNDB06.SYSTABSTATS)
SYSIBM.DSNTTX01	C	(OWNER,NAME,PARTITION)
SYSIBM.DSNTTX02	D	(DBNAME,TSNAME, PARTITION)
Contains table space partition statistics. One row per partition (populated by RUNSTATS).		

This table contains the following columns:

Column Name	Data Type/Description
CARD	INTEGER NOT NULL Total number of rows in the partition.
NPAGES	INTEGER NOT NULL Total number of pages on which rows of the partition appear. Updateable column
PCTPAGES	SMALLINT NOT NULL Percentage of total active pages in the partition that contain rows of the table. Updateable column.
NACTIVE	INTEGER NOT NULL Number of active pages in the partition. Updateable column.
PCTROWCOMP	SMALLINT NOT NULL Percentage of rows compressed within the total

Column Name	Data Type/Description
	number of active rows in the partition. Updateable column
STATSTIME	TIMESTAMP NOT NULL The date and time when RUNSTATS last updated the statistics. Updateable column
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other. If other, release dependency indicator.
DBNAME	VARCHAR(24) NOT NULL Database containing table space named in TSNAME.
TSNAME	VARCHAR(24) NOT NULL Table space that contains the table.
PARTITION	SMALLINT NOT NULL Partition number of the table space that contains the table.
OWNER	VARCHAR(128) NOT NULL Schema of the table.
NAME	VARCHAR(128) NOT NULL Name of the table.
CARDF	FLOAT NOT NULL WITH DEFAULT -1 Total number of rows in the partition. Updateable column.

SYSIBM.SYSTABSTATS_HIST		**(DSNDB06.SYSHIST)**
SYSIBM.DSNHBX01	**D**	**(OWNER,NAME,PARTITION, STATSTIME)**
Contains SYSTABSTATS history (populated by RUNSTATS). You can insert, update, and delete rows in this table.		

This table contains the following columns:

Column Name	Data Type/Description
NPAGES	INTEGER NOT NULL Total number of pages on which rows of the partition appear.
STATSTIME	TIMESTAMP NOT NULL If RUNSTATS updated the statistics, the date and time RUNSTATS last updated the statistics.
DBNAME	VARCHAR(24) NOT NULL Database that contains the table space named in TSNAME.
TSNAME	VARCHAR(24) NOT NULL Table space that contains the table.
PARTITION	SMALLINT NOT NULL Partition number of the table space that contains the table.
OWNER	VARCHAR(128) NOT NULL Schema of the table.
NAME	VARCHAR(128) NOT NULL Name of the table.
CARDF	FLOAT(8) NOT NULL WITH DEFAULT –1 Total number of rows in the partition. The value is -1 if statistics have not been gathered.
IBMREQD	CHAR(1) NOT NULL WITH DEFAULT ‘N’ Whether the row came from the basic MRM tape: Y/N/other.

SYSIBM.SYSTRIGGERS		(DSNDB06.SYSTSTRG)
SYSIBM.DSNOTX01	U	(SCHEMA,NAME,SEQNO)
SYSIBM.DSNOTX02	D	(TBOWNER,TBNAME)
SYSIBM.DSNOTX03	D	(SCHEMA,TRIGNAME)
Defines triggers. One row per trigger.		

This table contains the following columns:

Column Name	Data Type/Description
NAME	VARCHAR(128) NOT NULL Name of the trigger and trigger package.
SCHEMA	VARCHAR(128) NOT NULL Schema of the trigger. This implicit or explicit qualifier for the trigger name is also used for the collection ID of the trigger package.
SEQNO	SMALLINT NOT NULL/Not used.
DBID	SMALLINT NOT NULL Internal identifier of the database for the trigger.
OBID	SMALLINT NOT NULL Internal identifier of the trigger.
OWNER	VARCHAR(128) NOT NULL Owner of the trigger.
CREATEDBY	VARCHAR(128) NOT NULL Primary Authorization ID of the user who created the trigger.
TBNAME	VARCHAR(128) NOT NULL Name of the table or view.
TBOWNER	VARCHAR(128) NOT NULL Qualifier of the name of the table to which this trigger applies.
TRIGTIME	CHAR(1) NOT NULL Time when triggered actions are applied to the base table, relative to the event that activated the trigger: B Trigger is applied before the event A Trigger is applied after the event I Trigger is applied instead of the event
TRIGEVENT	CHAR(1) NOT NULL Operation that activated the trigger: I Insert D Delete U Update
GRANULARITY	CHAR(1) NOT NULL Trigger is executed once per: S Statement R Row
CREATEDTS	TIMESTAMP NOT NULL Time when the CREATE statement was executed for this trigger. The time value is used in resolving functions, distinct types, and stored procedures. It is also used to order the execution of multiple triggers.
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other. If other, release dependency indicator.
TEXT	VARCHAR(6000) NOT NULL/Not used
REMARKS	VARCHAR(762) NOT NULL A character string provided by the user

Column Name	Data Type/Description
	with the COMMENT statement.
TRIGNAME	VARCHAR(128) NOT NULL/Not used
OWNERTYPE	CHAR(1) NOT NULL WITH DEFAULT Indicates the type of owner: blank Authorization ID L Role
ENVID	INTEGER NOT NULL WITH DEFAULT Internal environment identifier.
RELCREATED	CHAR(1) NOT NULL The release of DB2® that is used to create the object. Blank if created prior to V9.
SECURE	CHAR(1) NOT NULL WITH DEFAULT 'N' Triggered secured: N (no) or Y (yes).
ALTEREDTS	TIMESTAMP NOT NULL Time of trigger's last ALTER statement
ROWID	ROWID NULL GENERATED ALWAYS LOB ROWID
SQLPL	CHAR(1) NOT NULL WITH DEFAULT Does the trigger support SQL PL: **Y** Advanced trigger that supports SQL PL. **blank** Basic trigger that does not support SQL PL.
ALTEREDTS	TIMESTAMP NOT NULL When the trigger was last changed.
DEBUG_MODE	CHAR(1) NOT NULL WITH DEFAULT Whether the trigger is enabled for debugging: **1** This trigger is enabled for debugging, and can be debugged in a client debug session using the DB2 Unified Debugger. **0** This trigger is not enabled for debugging. **N** This trigger can never be enabled for debugging. **blank** This trigger is a basic trigger that cannot be debugged.
ASUTIME	INTEGER Number of CPU service units that are allowed for a single invocation of this trigger. If ASUTIME is zero, the number of CPU service units is unlimited. If the trigger consumes more CPU service units than the ASUTIME value allows, DB2 cancels the trigger.
WLM_ENVIRONMENT	VARCHAR(96) The WLM ENVIRONMENT FOR DEBUG MODE value in the trigger definition. This value is the name of the WLM environment that is used when a trigger is debugged. If this value is blank, the trigger cannot be debugged.
STATEMENT	CLOB(2M) NOT NULL WITH DEFAULT CREATE TRIGGER statement text.
VERSION	VARCHAR(122) NOT NULL The version identifier for a trigger. A zero length string for a basic trigger.
ORIGINAL_CONTOK	CHAR(8) NOT NULL FOR BIT DATA The consistency token for the trigger. The column is set to X'20' if the value of VERSION is a zero length string.
REGENERATETS	TIMESTAMP NOT NULL

Column Name	Data Type/Description
	Time when this version of the trigger was last regenerated.
ACTIVE	CHAR(1) NOT NULL Identifies the active version of the trigger: **Y** The version is the active version. **N** The version is not the active version. **blank** The value of VERSION is a zero length string.
WRAPPED	CHAR(1) NOT NULL **'Y'** The trigger text is obfuscated. **blank** The trigger text is not obfuscated.

SYSIBM.SYSTRIGGERS_STMT		**(DSNDB06.SYSTSTRT)**
SYSIBM.DSNOTX04	**U**	**(AUXID,AUXVER)**
Auxiliary table for STATEMENT LOB column of SYSTRIGGERS.		

This table contains the following columns:

Column Name	Data Type	Description
AUXID	VARCHAR(17)	ID of auxiliary data.
AUXREL	SMALLINT	Version of auxiliary data
AUXVALUE	CLOB(2M) NOT NULL WITH DEFAULT	Trigger text

SYSIBM.SYSUSERAUTH		**(DSNDB06.SYSUSER)**
SYSIBM.DSNAUH01	**D**	**(GRANTEE,GRANTEDTS, GRANTEETYPE)**
SYSIBM.DSNAUX02	**D**	**(GRANTOR,GRANTORTYPE)**
Defines user's system privileges.		

This table contains the following columns:

Column Name	Data Type	Description
GRANTOR	VARCHAR(128) NOT NULL	Authorization ID of the user who granted the privileges.
GRANTEE	VARCHAR(128) NOT NULL	Authorization ID of the user who holds the privileges. Could also be PUBLIC for a grant to PUBLIC.
TIMESTAMP	CHAR(12) NOT NULL	Internal use only.
DATEGRANTED	CHAR(6)	Unused
TIMEGRANTED	CHAR(8)	Unused
GRANTEETYPE	CHAR(1)	Unused
AUTHHOWGOT	CHAR(1) NOT NULL	Authorization level of user from whom privileges were received: • blank—not applicable • C—DBCTL • D—DBADM • E—SECADM • G—ACCESSCTRL • L—SYSCTRL • M—DBMAINT • O—SYSOPRS • S—SYSADM
ALTERBPAUTH	CHAR(1)	Unused

Column Name	Data Type	Description
BINDADDAUTH	CHAR(1) NOT NULL	Whether GRANTEE can use the BIND command with ADD option: (see legend**)
BSDSAUTH	CHAR(1) NOT NULL	Whether GRANTEE can issu the RECOVER BSDS command: (see legend**)
CREATEDBAAUTH	CHAR(1) NOT NULL	Whether GRANTEE can create databases and automatically receive DBADM authority over new databases: (see legend**)
CREATEDBCAUTH	CHAR(1) NOT NULL	Whether GRANTEE can create databases and automatically receive DBCTRL authority over the new databases: (see legend**)
CREATESGAUTH	CHAR(1) NOT NULL	Whether GRANTEE can create new storage groups: (see legend**)
DISPLAYAUTH	CHAR(1) NOT NULL	Whether GRANTEE can use DISPLAY: (see legend**)
RECOVERAUTH	CHAR(1) NOT NULL	Whether GRANTEE can use RECOVER INDOUBT command: (see legend**)
STOPALLAUTH	CHAR(1) NOT NULL	Whether GRANTEE can use the DB2 STOP: (see legend**)
STOSPACEAUTH	CHAR(1) NOT NULL	Whether GRANTEE can run STOSPACE utility: (see legend**)
SYSADMAUTH	CHAR(1) NOT NULL	Whether GRANTEE has system administration authority: (see legend**)
SYSOPRAUTH	CHAR(1) NOT NULL	Whether GRANTEE has system operator authority: (see legend**)
TRACEAUTH	CHAR(1) NOT NULL	Whether GRANTEE can use START and STOP TRACE commands: (see legend**)
IBMREQD	CHAR(1) NOT NULL	Whether the row came from the basic MRM tape: Y/N/other. If other, release dependency indicator.
MON1AUTH	CHAR(1) NOT NULL WITH DEFAULT	Whether GRANTEE can obtain IFC serviceability data: (see legend**)
MON2AUTH	CHAR(1) NOT NULL WITH DEFAULT	Whether GRANTEE can obtain IFC data: (see legend**)
CREATEALIAS AUTH	CHAR(1) NOT NULL WITH DEFAULT	Whether GRANTEE can use CREATE ALIAS statement: (see legend**)
SYSCTRLAUTH	CHAR(1) NOT NULL WITH DEFAULT	If GRANTEE has SYSCTRL authority: (see legend**) GRANTEE has GRANT option for either Y or G.

Column Name	Data Type	Description
BINDAGENTAUTH	CHAR(1) NOT NULL WITH DEFAULT	If GRANTEE has BINDAGENT privilege: (see legend**)
ARCHIVEAUTH	CHAR(1) NOT NULL WITH DEFAULT	Whether GRANTEE can use ARCHIVE LOG command: (see legend**)
CAPTURE1AUTH	CHAR(1)	Unused.
CAPTURE2AUTH	CHAR(1)	Unused.
GRANTEDTS	TIMESTAMP NOT NULL WITH DEFAULT	Time the GRANT statement was executed.
CREATETMTAB AUTH	CHAR(1) NOT NULL WITH DEFAULT	Indicates if the GRANTEE ha CREATMTABAUTH privilege (see legend**)
GRANTORTYPE	CHAR(1) NOT NULL WITH DEFAULT	Indicates the type of grantor: blank Authorization ID L Role
DEBUGSESSIONAUTH	CHAR(1) NOT NULL WITH DEFAULT	Whether the GRANTEE has DEBUGSESSION privilege: blank Privilege is not held G Privilege is held with the GRANT option Y Privilege is held without the GRANT option
EXPLAINAUTH	CHAR(1) NOT NULL WITH DEFAULT	GRANTEE can explain and prepare statements: blank – NO G - YES with GRANT option Y - YES without GRANT option
SQLADMAUTH	CHAR(1) NOT NULL WITH DEFAULT	GRANTEE has SQLADM authority: blank – NO G - YES with GRANT option Y - YES without GRANT option
SDBADMAUTH	CHAR(1) NOT NULL WITH DEFAULT	GRANTEE has DBADM authority: blank – NO Y - YES without GRANT option
DATAACCESSAUTH	CHAR(1) NOT NULL WITH DEFAULT	GRANTEE has DATAACCESS authority: blank – NO Y - YES without GRANT option
ACCESSCTRLAUTH	CHAR(1) NOT NULL WITH DEFAULT	GRANTEE has ACCESSCTRL authority: blank – NO Y - YES without GRANT option
CREATESECUREAUTH	CHAR(1) NOT NULL WITH DEFAULT	GRANTEE can create secure objects blank – NO

Column Name	Data Type	Description
		Y - YES without GRANT option
SYS_START	TIMESTAMP(12) NOT NULL	System period temporal start time for transaction
SYS_END	TIMESTAMP(12) NOT NULL	System period temporal en time for transaction
TRANS_START	TIMESTAMP(12)	System period transaction timestamp.

Legend **:

blank = privilege not held; G = privilege held with GRANT option; Y = privilege held without GRANT option.

SYSIBM.SYSVARIABLES		**(DSNDB06.SYSTSVAR)**
SYSIBM.DSNOVX01	**P**	**(SCHEMA,NAME)**
SYSIBM.DSNOVX04	**D**	**(TYPESCHEMA,TYPENAME)**
Contains one row for every created global variable.		

This table contains the following columns:

Column Name	Data Type	Description
VARID	BIGINT NOT NULL GENERATED ALWAYS AS IDENTITY	Identifier
SCHEMA	VARCHAR(128) NOT NULL	Schema name of global variable
NAME	VARCHAR(128) NOT NULL	Name of global variable
OWNER	VARCHAR(128) NOT NULL	Authorization-id of the owner of the global variable
OWNERTYPE	CHAR(1) NOT NULL	Type of owner: L - ROLE Blank – authorization-id
RELCREATED	CHAR(1) NOT NULL	The release of DB2® that is used to create the object.
CREATEDTS	TIMESTAMP NOT NULL	Timestamp when the global variable was created
TYPESCHEMA	VARCHAR(128) NOT NULL	Schema name for data type. Value is SYSIBm for built-in types.
TYPENAME	VARCHAR(128) NOT NULL	Name of data type.
DATATYPEID	INTEGER NOT NULL	Internal id for built-in type or distinct type.
SOURCETYPEID	INTEGER NOT NULL	For built-in data type : 0 For distinct type : internal id.
LENGTH	INTEGER NOT NULL	Maximum length of global variable.
SCALE	SMALLINT NOT NULL	Scale of global variable.
CCSID	INTEGER NOT NULL	CCSID of global variable.
DEFAULT	CHAR(3) NOT NULL	Default global value: N - no default value S - SQL authorization-id 1 - string constant 2 - floating-point constant 3 – decimal constant

Column Name	Data Type	Description
		4 – integer constant 5 – hexadecimal character 6 – UX string 7 – character string since global variable is graphic 8 – character string since global variable is character 10 – DECFLOAT constant If one of the following values, the default value is the value of the indicated special register when the default value is used (SET CURRENT): AES – APPLICATION EBCODING SCHEME ACT – CLIENT_ACCTNG APN – CLIENT_APPLNAME CID – CLIENT_USERID WSN – CLIENT_WRKSTNNAME DAT - DATE DBG – DEBUG MODE DEC – DECFLOAT ROUNDING MODE DEG – DEGREE EXP – EXPLAIN MODE LCT – LOCAL LC_CTYPE MTT – MAINTAINED TABLE TYPES FOR OPTIMIZATION MEM – MEMBER HNT – OPTIMIZATION HINT CPP – PACKAGE PATH CPS – PACKAGESET PTH – PATH PRC – PRECISION RFA – REFRESH AGE RVS – ROUTINE VERSION RUL – RULES SCH – SCHEMA SVR – SERVER TIM – TIME TST – TIMESTAMP STZ – SESSION TIME ZONE U - SESSION_USER
ROWID	ROWID NOT NULL GENERATED ALWAYS	ROWID of LOB column DAFAULTTEXT
DEFAULTTEXT	CLOB(2M) NOT NULL INLINE LENGTH(2000)	Text of default global variable.
DESCRIPTOR	BLOB(2M) NOT NULL INLINE LENGTH(2000)	IBM reserved.
ENVID	INTEGER NOT NULL	Internal identifier
REMARKS	VARCHAR(762) NOT	Comment entered for this

Column Name	Data Type	Description
	NULL	row.
IBMREQD	CHAR(1) NOT NULL	Whether the row came from the basic MRM tape: Y/N/other. If other, release dependency indicator.

SYSIBM.SYSVARIABLEAUTH		(DSNDB06.SYSTSVAU)
SYSIBM.DSNVAX01	D	(GRANTEE,GRANTEETYPE, SCHEMA,NAME)
SYSIBM.DSNVAX02	D	(GRANTOR, GRANTORTYPESCHEMA, NAME
SYSIBM.DSNVAX03	D	(SCHEMA,NAME)
Each privilege held by authorization IDs that has privileges on a global variable.		

This table contains the following columns:

Column Name	Data Type	Description
GRANTOR	VARCHAR(128) NOT NULL	Authorization ID of the user who granted the privileges.
GRANTORTYPE	CHAR(1) NOT NULL	Type of grantor: Blank : authorization-id L : Role
GRANTEE	VARCHAR(128) NOT NULL	Authorization ID of the user who holds the privileges.
GRANTEETYPE	CHAR(1) NOT NULL	Type of grantee. L : Role P : Package Blank: authorization-id
SCHEMA	VARCHAR(128) NOT NULL	Schema name of global variable.
NAME	VARCHAR(128) NOT NULL	Global variable name.
COLLID	VARCHAR(128) NOT NULL	Collection-id if grantee is package.
CONTOKEN	CHAR(8) NOT NULL FOR BIT DATA	Consistency token of DBRM if grantee is a package – otherwise blank
READAUTH	CHAR(1) NOT NULL	Privilege to read global variable: G : READ with grant option is held. Y : READ w/o grant option is held. Blank : no READ privilege is held.
WRITEAUTH	CHAR(1) NOT NULL	Privilege to write the global variable: G : WRITE with grant option is held. Y : WRITE w/o grant option is held. Blank : no WRITE privilege is held.
AUTHHOWGOT	CHAR(1) NOT NULL	Authorization level of user from whom privileges were received: ▪ E—SECADM

Column Name	Data Type	Description
		▪ G—ACCESSCTRL ▪ S—SYSADM ▪ T—DATAACCESS
GRANTEDTS	TIMESTAMP NOT NULL	Time when access was granted.
IBMREQD	CHAR(1) NOT NULL	Whether the row came from the Basic MRM tape: Y/N/other. If other, release dependency indicator.

SYSIBM.SYSVARIABLES_DESC		(DSNDB06.SYSTSVAD)
SYSIBM.DSNOVX03	U	(AUXID,AUXVER)
Auxiliary table for DESCRIPTOR LOB column of SYSVARIABLES.		

This table contains the following column:

Column Name	Data Type	Description
AUXID	VARCHAR(17)	ID of auxiliary data.
AUXVER	SMALLINT	Version of auxiliary data.
AUXVALUE	BLOB(2M) NOT NULL WITH DEFAULT	Trigger text.

SYSIBM.SYSVARIABLES_TEXT		(DSNDB06.SYSTSVAT)
SYSIBM.DSNOVX02	U	(AUXID,AUXVER)
Auxiliary table for DEFAULTEXT LOB column of SYSVARIABLES.		

This table contains the following columns:

Column Name	Data Type	Description
AUXID	VARCHAR(17)	ID of auxiliary data.
AUXREL	SMALLINT	Version of auxiliary data
AUXVALUE	CLOB(2M) NOT NULL WITH DEFAULT	Trigger text

SYSIBM.SYSVIEWDEP		(DSNDB06.SYSTSVWD)
SYSIBM.DSNGGX02	D	(BCREATOR,BNAME,BTYPE)
SYSIBM.DSNGGX03	D	(BSCHEMA,BNAME,BTYPE)
SYSIBM.DSNGGX04	D	(BCREATOR,BNAME,BTYPE, DTYPE)
SYSIBM.DSNGGX06	D	(DCREATOR,DNAME,TYPE)
Defines dependencies of views on tables, functions, and other views.		

This table contains the following columns:

Column Name	Data Type/Description
BNAME	VARCHAR(128) NOT NULL Name of the table or view on which the view is dependent. If object type is a function (BTYPE=F), name is specific function name.
BCREATOR	VARCHAR(128) NOT NULL Authorization ID of owner of the BNAME. For function value is the schema name of BNAME.
BTYPE	CHAR(1) NOT NULL Type of object BNAME: ▪ F—function

Column Name	Data Type/Description
	▪ G—Created global temp table ▪ M—Materialized query table ▪ T—Table ▪ V—view ▪ W—SYSTEM_TIME period ▪ Z—BUSINESS_TIME period
DNAME	VARCHAR(128) NOT NULL Name of the view.
DCREATOR	VARCHAR(128) NOT NULL Schema of the view.
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other.
BSCHEMA	VARCHAR(128) NOT NULL WITH DEFAULT Schema of BNAME.
DTYPE	CHAR(1) NOT NULL Type of table: ▪ F—function ▪ M—Materialized query table ▪ V—view
DOWNER	VARCHAR(128) NOT NULL WITH DEFAULT Authorization ID of the owner of the view, blank for views that were created in a DB2 release prior to V9.
OWNERTYPE	CHAR(1) NOT NULL WITH DEFAULT Indicates the type of owner: ▪ blank—Authorization ID ▪ L—Role
SYS_START	TIMESTAMP(12) NOT NULL System period temporal start time for transaction
SYS_END	TIMESTAMP(12) NOT NULL System period temporal end time for transaction
TRANS_START	TIMESTAMP(12) System period transaction timestamp.

SYSIBM.SYSVIEWS		**(DSNDB06.SYSTSVEW)**
SYSIBM.DSNVVX01	**U**	**(CREATOR,NAME,SEQNO, TYPE)**
Defines views, materialized query tables, and user-defined SQL functions. One or more rows for each.		

This table contains the following columns:

Column Name	Data Type/Description
NAME	VARCHAR(128) NOT NULL Name of the view.
CREATOR	VARCHAR(128) NOT NULL Schema of the object.
SEQNO	SMALLINT NOT NULL Not used.
CHECK	CHAR(1) NOT NULL Whether CHECK option was specified in the CREATE VIEW statement: N No C Yes with the *cascade* semantic Y yes with the *local* semantic The value is N when the view does not contain

Column Name	Data Type/Description
	a WHERE clause.
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other. If other, release dependency indicator.
TEXT	VARCHAR(1500) NOT NULL/Not used.
PATHSCHEMAS	VARCHAR(2048) NOT NULL WITH DEFAULT SQL path at time view was defined. Used to resolve unqualified data type and function names.
RELCREATED	CHAR(1) Release of DB2 used to create object.
TYPE	CHAR(1) NOT NULL Type of table: F(function), M (materialized query table), V (view).
REFRESH	CHAR(1) NOT NULL WITH DEFAULT Refresh mode; D (a materialized query table with a deferred refresh mode) or blank (not a materialized query table).
ENABLE	CHAR(1) NOT NULL WITH DEFAULT Indicates whether query optimization is enabled: Y (enabled), N (disabled), blank (not a MQT).
MAINTENANCE	CHAR(1) NOT NULL WITH DEFAULT Maintenance mode: S REFRESH = D, system maintained U REFRESH = D, user maintained Blank not a materialized query table
REFRESH_TIME	TIMESTAMP NOT NULL WITH DEFAULT For REFRESH = D and MAINTENANCE = S, the timestamp1 of the REFRESH TABLE statement that last refreshed the data. Otherwise default timestamp.
ISOLATION	CHAR(1) NOT NULL WITH DEFAULT Isolation level when the materialized query table is created or altered from a base table: R RR S CS T RS U UR Blank, not a materialized query table
SIGNATURE	VARCHAR (1024) NOT NULL WITH DEFAULT FOR BIT DATA Contains an internal description. Used for materialized query tables.
APP_ENCODING_CCSID	INTEGER NOT NULL WITH DEFAULT CCSID of the current application encoding scheme at the time the object was created. For objects created prior to V8, value = 0.
OWNER	VARCHAR(128) NOT NULL WITH DEFAULT Authorization ID of the owner of the view, blank for views that were created in a DB2 release prior to V9.
OWNERTYPE	CHAR(1) NOT NULL WITH DEFAULT Indicates the type of owner: blank Authorization ID

Column Name	Data Type/Description
	L Role
ENVID	INTEGER NOT NULL WITH DEFAULT Internal environment identifier
ROWID	ROWID NULL GENERATED ALWAYS LOB ROWID
STATEMENT	CLOB(2M) NOT NULL WITH DEFAULT CREATE VIEW statement text.
PARSETREE	BLOB(1G) NOT NULL WITH DEFAULT INLINE LENGTH(27670) IBM Internal use only.

SYSIBM.SYSVIEWS_STMT		**(DSNDB06.SYSTSVWT)**
SYSIBM.DSNVWX01	**U**	**(AUXID,AUXVER)**
Auxiliary table for STATEMENT LOB column of SYSVIEWS.		

This table contains the following column:

Column Name	Data Type	Description
AUXID	VARCHAR(17)	ID of auxiliary data.
AUXVER	SMALLINT	Version of auxiliary data.
AUXVALUE	CLOB(2M) NOT NULL WITH DEFAULT	View text.

SYSIBM.SYSVIEWS_TREE		**(DSNDB06.SYSTSVTR)**
SYSIBM.DSNVWX02	**U**	**(AUXID,AUXVER)**
Auxiliary table for PARSETREE LOB column of SYSTRIGGERS.		

This table contains the following column:

Column Name	Data Type	Description
AUXID	VARCHAR(17)	ID of auxiliary data.
AUXVER	SMALLINT	Version of auxiliary data.
AUXVALUE	BLOB(1G) NOT NULL WITH DEFAULT	IBM internal use.

SYSIBM.SYSVOLUMES		**(DSNDB06.SYSTSVOL)**
SYSIBM.DSNSSH02	**D**	**(SGNAME)**
Defines the volumes in each storage group. One row per volume.		

This table contains the following columns:

Column	Data Type/Description
SGNAME	VARCHAR(128) NOT NULL Name of the storage group.
SGCREATOR	VARCHAR(128) NOT NULL Authorization ID of the owner of the storage group. To determine the type of authorization ID for the storage group creator, see the CREATORTYPE column of the SYSIBM.SYSSTOGROUP catalog table.
VOLID	VARCHAR(18) NOT NULL Serial number of the volume or * if SMS managed.
IBMREQD	CHAR(1) NOT NULL Whether the row came from the basic MRM tape: Y/N/other.
RELCREATED	CHAR(1) NOT NULL

Column	Data Type/Description
	The release of DB2 that is used to create the object. Blank if created prior to V9.

SYSIBM.SYSXMLRELS		(DSNDB06.SYSXML)
SYSIBM.DSNXRX01	D	(TBOWNER,TBNAME, COLNAME)
SYSIBM.DSNXRS02	D	(XMLTBOWNER, XMLTBNAME)
Defines XML tables that are created for XML columns. One row for each created XML table.		

This table contains the following columns:

Column Name	Data Type/Description
TBOWNER	VARCHAR(128) NOT NULL Schema or qualifier of the base table.
TBNAME	VARCHAR(128) NOT NULL Name of the base table.
COLNAME	VARCHAR(128) NOT NULL Name of the XML column in the base table.
XMLTBOWNER	VARCHAR(128) NOT NULL Schema or qualifier of the XML table.
XMLTBNAME	VARCHAR(128) NOT NULL Name of the XML table.
XMLRELOBID	SMALLINT NOT NULL Internal identifier of the relationship between the base table and the XML table.
IBMREQD	CHAR(1) NOT NULL A value of Y indicates that the row came from the basic machine-readable material (MRM) tape.
CREATEDTS	TIMESTAMP NOT NULL Time when the XML table was created.
RELCREATED	CHAR(1) NOT NULL The release of DB2 that is used to create the object.

SYSIBM.SYSXMLSTRING		(DSNDB06.SYSXML)
SYSIBM.DSNXSX01	U	(STRINGID)
SYSIBM.DSNXSX02	U	(STRING)
Defines unique IDs used to condense XML data. One row per single string and its unique ID.		

This table contains the following columns:

Column Name	Data Type/Description
STRINGID	INTEGER NOT NULL GENERATED ALWAYS AS IDENTITY Unique ID for the string.
STRING	VARCHAR(1000) NOT NULL The string data.
IBMREQD	CHAR(1) NOT NULL A value of Y indicates that the row came from the basic machine-readable material (MRM) tape.

SYSIBM.SYSXMLTYPMOD		(DSNDB06.SYSTSXTM)
SYSIBM.DSNTMX01	P	(XML_TYPEMOD_ID)
Contains information about the XML type modifiers of XML columns. You can insert, update, and delete rows in this table.		

This table contains the following columns:

Column Name	Data Type/Description
XML_TYPEMOD_ID	INTEGER NOT NULL GENERATED ALWAYS AS IDENTITY Unique ID for the XML type modifier.
TYPE_ANNOTATION	CHAR(1) NOT NULL Type annotation flag Y – WITH type annotation N – with NO type annotation
CREATEDTS	TIMESTAMP NOT NULL Type modifier create time.
ALTEREDTS	TIMESTAMP NOT NULL Type modifier altered time.
RELCREATED	CHAR(1) NOT NULL Release of DB2 that created object (see Release dependency indicators for values).
IBMREQD	CHAR(1) NOT NULL Type modifier source: Y – Basic machine-readable material(MRM), (else see Release dependency indicators for values.
CREATEDBY	VARCHAR(128) NOT NULL Authorization ID of user who created database.

SYSIBM.SYSXMLTYPMSCHEMA		(DSNDB06.SYSTSXTS)
SYSIBM.DSNMSX01	U	(XML_TYPEMOD_ID, XSROBJECTID)
SYSIBM.DSNMSX02	D	(XSROBJECTID)
Contains XML schema information for an XML type modifier. One row per XML schema for an XML type modifier.		

This table contains the following columns:

Column Name	Data Type/Description
XML_TYPEMOD_ID	INTEGER NOT NULL XML type modifier ID
XSROBJECTID	INTEGER NOT NULL XML schema ID registered in XSR.
ELEMENT_NAMESPACE	INTEGER NOT NULL Namespace name of root element node.
ELEMENT_NAME	INTEGER NOT NULL Root element node local name string ID: 0 – not specified
CREATEDTS	TIMESTAMP NOT NULL Type modifier create time.
ALTEREDTS	TIMESTAMP NOT NULL Type modifier last altered time.
RELCREATED	CHAR(1) NOT NULL Release of DB2 that created object. (see Release dependency indicators for values.
IBMREQD	CHAR(1) NOT NULL Type modifier source

Column Name	Data Type/Description
	Y – Basic machine-readable material(MRM) (else see Release dependency indicators for values.

SYSIBM.USERNAMES		(DSNDB06.SYSDDF)
SYSIBM.DSNFEX01	**U**	**(TYPE,AUTHID,LINKNAME)**
Used to carry out an outbound ID translation or inbound ID translation and "come from" checking. You can insert, update, and delete rows in this table.		

This table contains the following columns:

Column Name	Data Type/Description
TYPE	CHAR(1) NOT NULL How the row is to be used: O outbound translation I inbound translation and "come from" checking S Outbound system AUTHID to establish a trusted connection.
AUTHID	VARCHAR(128) NOT NULL WITH DEFAULT Authorization ID to be translated. If blank, any auth ID.
LINKNAME	VARCHAR(24) NOT NULL Identifies the VTAM for TCP/IP network locations associated with this row. blank This name translation rule applies to any TCP/IP or SNA partner. nonblank A row exists in SYSIBM.LUNAMES whose LUNAME matches the value in LINKNAME. Or, a row exists in SYSIBM.IPNAMES whose LINKNAME matches the value in the USERNAMES LINKNAME column.
NEWAUTHID	VARCHAR(128) NOT NULL WITH DEFAULT Translated value of AUTHID. Blank specifies no translation. NEWAUTHID can be stored as encrypted data by calling DSNLEUSR stored procedure. To send the encrypted value of AUTHID across a network, one of the encryption security options in the SYSIBM.IPNAMES table should be specified.
PASSWORD	VARCHAR(128) NOT NULL WITH DEFAULT Password to accompany an outbound request, if passwords are not encrypted by RACF. If passwords are encrypted, or the row is for inbound requests, the column is not used. PASSWORD can be stored as encrypted data by calling DSNLEUSR stored procedure. To send the encrypted value of PASSWORD across the network, one of the encryption security options in the SYSIBM.IPNAMES table should be specified.
IBMREQD	CHAR(1) NOT NULL WITH DEFAULT 'N' Whether the row came from the basic MRM tape: Y/N/other.

SYSIBM.XSRANNOTATIONINFO		(DSNDB06.SYSXSR)
SYSIBM.XSRANNINFOIDX	U	(XSROBJECTID,ANNID)
IBM internal use table.		

This table contains the following columns:

Column Name	Data Type/Description
XSROBJECTID	INTEGER / IBM internal use only.
ANNID	INTEGER / IBM internal use only.
TABSCHEMA	VARCHAR(128) / IBM internal use only.
TABNAME	VARCHAR(128) / IBM internal use only.
ROWSET	INTEGER / IBM internal use only.
COLNAME	VARCHAR(30) / IBM internal use only.
COLTYPE	INTEGER / IBM internal use only.
INSTANCETYPE	INTEGER / IBM internal use only.
TRUNCATE	INTEGER / IBM internal use only.
EXPRESSION	VARCHAR(1024) / IBM internal use only.
CONDITION	VARCHAR(1024) / IBM internal use only.
CASTEXPRESSION	VARCHAR(20) / IBM internal use only.
RELCREATED	CHAR(1) / IBM internal use only.

SYSIBM.XSRCOMPONENT		(DSNDB06.SYSXSRA3)
SYSIBM.XSRXCC01	U	(AUXID,AUXVER)
Auxiliary table for the COMPONENT LOB column in SYSXSROBJECTCOMPONENTS.		

This table contains the following column:

Column Name	Data Type	Description
AUXID	VARCHAR(17)	ID of auxiliary data.
AUXVER	SMALLINT	Version of auxiliary data.
AUXVALUE	BLOB(30M)	Contents of the XML schema document.

SYSIBM.XSROBJECTCOMPONENTS		(DSNDB06.SYSXSR)
SYSIBM.XSRCOMP01	U	(XSRCOMPONENTID)
SYSIBM.XSRCOMP02	D	(TARGETNAMESPACE, SCHEMALOCATION)
Contains information for each component (document) in an XML schema. One row for each component. Rows in this able can only be changed using static SQL statements issued by the DB2-supplied XSR stored procedures.		

This table contains the following columns:

Column Name	Data Type/Description
XSRCOMPONENTID	INTEGER NOT NULL Internal identifier of the XML schema document. XSRCOMPONENTID is generated as an identity column.
TARGETNAMESPACE	INTEGER The value of the STRINGID column in SYSXMLSTRINGS when the target namespace URI of the primary XML schema document is stored in SYSXMLSTRINGS.
SCHEMALOCATION	INTEGER The value of the STRINGID column in SYSXMLSTRINGS when the schema

Column Name	Data Type/Description
	location URI of the primary XML schema document is stored in SYSXMLSTRINGS.
ROWID	ROWID NOT NULL GENERATED ALWAYS The ID that is used to support BLOB data type values.
COMPONENT	BLOB(30M) NOT NULL Contents of the XML schema document.
PROPERTIES	BLOB(5M) If available, additional property information of the XML schema document
CREATEDTS	TIMESTAMP NOT NULL The time that the XML schema document was registered.
STATUS	CHAR(1) NOT NULL WITH DEFAULT Registration status of the XML schema: C Complete I Incomplete
RELCREATED	CHAR(1) NOT NULL The release of DB2 that is used to create the object.

SYSIBM.XSROBJECTGRAMMAR		**(DSNDB06.SYSXSRA1)**
SYSIBM.XSRXOG01	**U**	**(AUXID,AUXVER)**
Auxiliary table for the GRAMMAR LOB column in SYSXSROBJECTS.		

This table contains the following column:

Column Name	Data Type	Description
AUXID	VARCHAR(17)	ID of auxiliary data.
AUXVER	SMALLINT	Version of auxiliary data.
AUXVALUE	BLOB(250M)	Internal binary representation of the XML schema.

SYSIBM.XSROBJECTHIERARCHIES		**(DSNDB06.SYSXSR)**
SYSIBM.XSRHIER01	**U**	**(XSROBJECTID, TARGETNAMESPACE, SCHEMALOCATION)**
SYSIBM.XSRHIER02	**D**	**(XSROBJECTID, TARGETNAMESPACE)**
Contains hierarchy relationship information for each component (document) in an XML schema. One row per component. Rows in this table can only be changed using static SQL statements issued by the DB2-supplied XSR stored procedure.		

This table contains the following columns:

Column Name	Data Type/Description
XSROBJECTID	INTEGER Internal identifier of the XML schema.
XSRCOMPONENTID	INTEGER Internal identifier of the XML schema document.
HTYPE	CHAR(1) Hierarchy type: D (document) or P (primary document).
TARGETNAMESPACE	INTEGER The value of the STRINGID column in SYSIBM.SYSXMLSTRINGS when the target namespace URI of the primary XML

Column Name	Data Type/Description
	schema document is stored in SYSIBM.SYSXMLSTRINGS.
SCHEMALOCATION	INTEGER The value of the STRINGID column in SYSIBM.SYSXMLSTRINGS when the schema location URI of the primary XML schema document is stored in SYSIBM.SYSXMLSTRINGS.
RELCREATED	CHAR(1) NOT NULL The release of DB2 that is used to create the object.

SYSIBM.XSROBJECTPROPERTY		**(DSNDB06.SYSXSRA2)**
SYSIBM.XSRXOP01	**U**	**(AUXID,AUXVER)**
Auxiliary table for the BLOB column PROPERTIES in SYSXSROBJECTS. It is in LOB table space SYSXSRA2.		

This table contains the following column:

Column Name	Data Type	Description
AUXID	VARCHAR(17)	ID of auxiliary data.
AUXVER	SMALLINT	Version of auxiliary data.
AUXVALUE	BLOB(5M)	Contents of the additional property information of the entire XML schema.

SYSIBM.XSROBJECTS		**(DSNDB06.SYSXSR)**
SYSIBM.XSROBJ01	**U**	**(XSROBJECTID)**
SYSIBM.XSROBJ02	**U**	**(IXSROBJECTSCHEMA, XSROBJECTNAME)**
SYSIBM.XSROBJ03	**D**	**(TARGETNAMESPACE, SCHEMALOCATION)**
SYSIBM.XSROBJ04	**D**	**(SCHEMALOCATION)**
Contains one row for each registered XML schema. Rows can only be changed using static SQL statements issued by the DB2-supplied XSR stored procedure.		

This table contains the following columns:

Column Name	Data Type/Description
XSROBJECTID	INTEGER NOT NULL Internal identifier of the XML schema. XSROBJECTID is generated as an identity column.
XSROBJECTSCHEMA	VARCHAR(128) NOT NULL Qualifier of the XML schema name. Always set to 'SYSXSR'.
XSROBJECTNAME	VARCHAR(128) NOT NULL Name of the XML schema.
TARGETNAMESPACE	INTEGER The value of the STRINGID column in SYSIBM.SYSXMLSTRINGS when the target namespace URI of the primary XML shcema document is stored in SYSIBM.SYSXMLSTRINGS.
SCHEMALOCATION	INTEGER The value of the STRINGID column in SYSIBM.SYSXMLSTRINGS when the schema location URI of the primary XML schema document is stored in

Column Name	Data Type/Description
	SYSIBM.SYSXMLSTRINGS.
ROWID	ROWID NOT NULL GENERATED ALWAYS The ID that is used to support BLOB data type values.
GRAMMAR	BLOB(250M) The internal binary representation of the XML schema.
PROPERTIES	BLOB(5M) Additional property information of the entire XML schema.
CREATEDBY	VARCHAR(128) NOT NULL Authorization ID under which the XML schema was created.
CREATEDTS	TIMESTAMP NOT NULL The time that the DB2-supplied stored procedure XSR_REGISTER was executed for the XML schema.
STATUS	CHAR(1) NOT NULL WITH DEFAULT Registration status of the XML schema: • C—Complete • I—Incomplete • T—Temporary
RELCREATED	CHAR(1) NOT NULL The release of DB2 that is used to create the object..
DECOMPOSITION	CHAR(1) Not Used
DECOMPOSITION_VERSION	VARCHAR(128) Not Used
REMARKS	VARCHAR(762) Character string that contains comments about this XML schema.

SYSIBM.XSRPROPERTY		**(DSNDB06.SYSXSRA4)**
SYSIBM.XSRXCP01	**U**	**(AUXID,AUXVER)**
Auxiliary table for the BLOB column COMPONENT in XSROBJECTCOMPONENTS.		

This table contains the following column:

Column Name	Data Type	Description
AUXID	VARCHAR(17)	ID of auxiliary data.
AUXVER	SMALLINT	Version of auxiliary data.
AUXVALUE	BLOB(5M)	Contents of the additional property information of the entire XML schema.

Directory Tables

The DB2 directory tables hold information that describes the structure of DB2 data, how it is stored, and how DB2 accesses it. The directory is maintained through SQL DDL statements, the BIND process, and utility executions.

Legend:

SYSIBM.Tablename	**(DSNDB01.Tablespacename)**
SYSIBM.Indexname	**Unique/Nonunique**
table description	

SYSIBM.DBDR		**(DSNDB01.DBD01)**
SYSIBM.DSNDB01X	**U**	**(DBID,SECTION)**
The DBDR table stores one row for every DBD section (database descriptor).		

This table contains the following columns:

Name	Data Type	Description
N/A	INTEGER	Not used.
DBID	SMALLINT	DBID of the database.
SECTION	SMALLINT	DBD section number.
DBD_ROWID	ROWID	ID used to support DBD_DATA.
DBD_DATA	BLOB(2 GB)	DBD data for the section.

SYSIBM.SCTR		**(DSNDB01.SCT02)**
SYSIBM.DSNSCT02	**U**	**(SCTNAME)**
Stores skeleton cursor tables (SKCT) information.		

This table contains the following columns:

Name	Data Type	Description
SCTLL	CHAR(14) FOR BIT DATA	Record length.
SCTNAME	CHAR(14) FOR BIT DATA	Plan name, section number, and sequence number.
SCTDATA	VARCHAR(4028)	Skeleton cursor table data.

SYSIBM.SPTR (SKPT Parent Record)		**(DSNDB01.SPT01)**
SYSIBM.DSNSPT01	**U**	**(SPTLOCID,SPTCOLID,SPTNAME, SPTCONID,SPTRESV,SPTSEC, SPTSEQ)**
SYSIBM.DSNSPT02	**U**	**(SPTVER,SPTLOCID,SPTCOLID, SPTNAME,SPTCONID,SPTRESV, SPTSEC,SPTSEQ)**
Stores skeleton package table (SKPT) information.		

This table contains the following columns:

Name	Data Type
SPTLL	INTEGER
SPTLOCID	VARCHAR(128)
SPTCOLID	VARCHAR(128)
SPTNAME	VARCHAR(128)
SPTCONID	CHAR(8) FOR BIT DATA
SPTRESV	CHAR(2) FOR BIT DATA
SPTSEC	CHAR(4) FOR BIT DATA
SPTSEQ	CHAR(2) FOR BIT DATA

Name	Data Type
SPTBODY	VARCHAR(1)
SPTVER	VARCHAR(64)
SPT_ROWID	ROWID
SPT_DATA	BLOB(2G) INLINE LENGTH(32142)
SPT_EXPLAIN	BLOB(2G)

SYSIBM.SYSDBD_DATA		**(DSNDB01.SYSDBDXA)**
SYSIBM.DSNDB1XA	**U**	**(AUXID,AUXVER)**
Auxiliary table that holds the LOB column for DBDR table and column DBD_DATA.		

This table contains the following column:

Column Name	Data Type	Description
AUXID	VARCHAR(17)	ID of auxiliary data.
AUXVER	SMALLINT	Version of auxiliary data.
AUXVALUE	BLOB(5M)	Contents of the additional property information of the entire XML schema.

Legend:

SYSIBM.*Tablename*		**(DSNDB06.*Tablespacename*)**
SYSIBM.*indexname*	**Type**	**(index columns)**
Table description		

SYSIBM.SYSLGRNX		**(DSNDB06.SYSLGRNX)**
SYSIBM.DSNLLX01	**U**	**(LGRDBID,LGRPSID,LGRPART, LGRMEMB,LGRLRSN)**
SYSIBM.DSNLLX02	**D**	**(LGRDBID,LGRPSID,LGRLRSN)**
Stores recovery log ranges that record the time an index space defined with COPY YES or a table space was open for updates.		

This table contains the following columns:

Name	Data Type	Description
LGRDBID	CHAR(2) FOR BIT DATA	DBID
LGRPSID	CHAR(2) FOR BIT DATA	OBID
LGRUCDT	CHAR(6) FOR BIT DATA	Date mmddyy
LGRUCTM	CHAR(8) FOR BIT DATA	Time hhmmssth
LGRSRBA	CHAR(10) FOR BIT DATA	Start RBA
LGRSPBA	CHAR(10) FOR BIT DATA	End RBA
LGRPART	SMALLINT FOR BIT DATA	Partition number
LGRLRSN	CHAR(10) FOR BIT DATA	Start LRSN for data sharing—otherwise system clock for open-for-update.
LGRELRSN	CHAR(10) FOR BIT DATA	End LRSN for data sharing—otherwise system clock for close-for-update.
LGRMEMB	CHAR(2) FOR BIT DATA	Data sharing member ID for data sharing—otherwise X'0000'

SYSIBM.SYSSPTSEC_DATA		(SYSDBDXA)
DSNSPDCA	U	(AUXID,AUXVER)
Auxiliary table holding column SPT_DATA for table SPTR.		

This table contains the following column:

Column Name	Data Type	Description
AUXID	VARCHAR(17)	ID of auxiliary data.
AUXVER	SMALLINT	Version of auxiliary data.
AUXVALUE	BLOB(2G)	Contents of the additional property information of the entire XML schema.

SYSIBM.SYSSPTSEC_EXPL		(SYSSPUXB)
DSNSPEXA	U	(AUXID,AUXVER)
Auxiliary table holding column SPT_EXPLAIN data for table SPTR.		

This table contains the following column:

Column Name	Data Type	Description
AUXID	VARCHAR(17)	ID of auxiliary data.
AUXVER	SMALLINT	Version of auxiliary data.
AUXVALUE	BLOB(2G)	Contents of the additional property information of the entire XML schema.

This table contains the following column:

Name	Data Type	Description
SPT_EXPLAIN	BLOB(2G)	SKPT section explain block contents.

SYSIBM.SYSUTIL		(SYSUTILX)
DSNLUX01	U	(USUID)
Stores information about DB2 utilities that have been started, but have not completed running.		

This table contains the following columns:

Name	Data Type	Description
USUUID	CHAR(16)	Utility-id.
USUJOBNM	CHAR(8)	Job name.
USUAUID	CHAR(8)	Authorization ID.
USURDATE	CHAR(4) FOR BIT DATA	Date.
USUREL	CHAR(3)	Release level when utility started.
USUIRQD	CHAR(1)	IBM required field.
USULSIZE	CHAR(4) FOR BIT DATA	List size.
USULCUR	CHAR(4) FOR BIT DATA	Object processed.
USUUTNAM	CHAR(8)	Name of executing utility.
USUPHASE	CHAR(8)	Phase of utility.
USUDSNU	CHAR(2) FOR BIT DATA	Data set number or piecesize number.
USUDSNU2	CHAR(2) FOR BIT DATA	Ending partition number if range processing.
USUSTATU	CHAR(1)	Status of utility.
USUTREQ	CHAR(1)	Has termination been requested Y/N.

Name	Data Type	Description
USUFORCE	CHAR(1)	Element of USO forced Y/N.
USURLOK	CHAR(1)	Reload successful Y/N.
USUCMPOK	CHAR(1)	Compatibility check passed Y/N.
USURSFLG	CHAR(1) FOR BIT DATA	Utility restriction flags.
USURTFLG	CHAR(1) FOR BIT DATA	Term settings.
USURSFLG2	CHAR(1) FOR BIT DATA	Utility flags.
USUPOS	CHAR(4) FOR BIT DATA	Relative USM position in SYSIN.
USUDONE	CHAR(8) FOR BIT DATA	Number of objects processed.
USUCKSUM	CHAR(4) FOR BIT DATA	USU checksum.
USUDBOB	CHAR(2) FOR BIT DATA	DBID.
USUPSID	CHAR(2) FOR BIT DATA	PSID for table space or index.
USUPSDD	CHAR(2) FOR BIT DATA	For RECOVER INDEX, the secondary PSID.
USUCATMGFRM	CHAR(1) FOR BIT DATA	Saved catalog level for the release level from which migration is done (taken from DBD01 header page).
USOOFLAG	CHAR(1) FOR BIT DATA	Flags for object properties.
USUDBNAM	CHAR(8)	Database name.
USUSPNAM	CHAR(8)	Table space name or index space name.
USUMEMBR	CHAR(8)	Data sharing member name.
USUOCATR	CHAR(1) FOR BIT DATA	Saved catalog releases level.
USUOCATV	CHAR(1) FOR BIT DATA	Saved catalog version level.
USUCATCV	CHAR(1) FOR BIT DATA	Saved migration mode.
USUOCATH	CHAR(1) FOR BIT DATA	Saved highest version of the catalog.
USUUDA	CHAR(150) FOR BIT DATA	Utility dependent data.
USURTIME	CHAR(4) FOR BIT DATA	Latest utility start time.
USURLSN	CHAR(6) FOR BIT DATA	Original utility start date.
USURDATO	CHAR(4) FOR BIT DATA	Original utility start date.
USURTIMO	CHAR(4) FOR BIT DATA	Original utility start time.
USURLSNO	CHAR(6) FOR BIT DATA	Original utility start LRSN.
USUR5	CHAR(10) FOR BIT DATA	IBM reserved.
USUCNTR	CHAR(4) FOR BIT DATA	Generic counter/value.

Name	Data Type	Description
USURLSNX	CHAR(10) FOR BIT DATA	Latest utility start LRSN.
USURLSOX	CHAR(10) FOR BIT DATA	Original utility start LRSN.
USUR6	CHAR(72) FOR BIT DATA	IBM reserved.
USUUSTRN	VARCHAR(27000) FOR BIT DATA	Utility dependent restart information.

SYSIBM.SYSUTILX		**(SYSUTILX)**
DSNLUX02	**U**	**(UTILID,SEQNO)**
SYSUTILX is dependent on SYSUTIL. This table holds information when SYSUTIL exceeds the record size.		

This table contains the following columns:

Name	Data Type	Description
UTILID	CHAR(16)	Utility-id.
SEQNO	SMALLINT	Sequence number.
RESV1	CHAR(12)	Reserved.
CHECKPOINT	VARCHAR(32000)	Overflow information from SYSUTIL holding checkpoint and/or restart information.

Other System Tables

Legend:

Creator.Tablename	**(Databasename.Tablespacename)**
Creator.Indexname	**Unique/Nonunique**
table description	

SYSACCEL.SYSACCELERATORS		**(DSNACCEL.SYSACCEL)**
DSNACC01	**U**	**(ACCELERATORNAME)**
Contains one row for each accelerator server.		

This table contains the following columns:

Name	Data Type/Description
ACCELERATORNAME	VARCHAR(128) NOT NULL Unique name for accelerator server.
LOCATION	VARCHAR(128) Location name for accelerator server.
ACCELERATORSRL	CHAR(64) FOR BIT DATA (internal use)

SYSACCEL.SYSACCELERATEDTABLES		**(DSNACCEL.SYSACCEL)**
DSNACT01	**U**	**(CREATOR,NAME,ACCELERATORNAME)**
SYSACCLERATEDTABLES describes each table selected for acceleration.		

This table contains the following columns:

Column Name	Data Type/Description
NAME	VARCHAR(128) NOT NULL Table name.
CREATOR	VARCHAR(128) NOT NULL Table schema.
ACCELERATORNAME	VARCHAR(128) NOT NULL Unique name for accelerator server.
REMOTENAME	VARCHAR(128) NOT NULL Base alias object.
REMOTECREATOR	VARCHAR(128) NOT NULL Base alias owner.
ENABLE	CHAR(1) NOT NULL Remote table enabled for query acceleration: Y (enabled) or N (disabled).
CREATEDBY	VARCHAR(128) NOT NULL Primary authorization ID of table creator.
CREATEDTS	TIMESTAMP NOT NULL WITH DEFAULT Time for CREATE statement.
ALTEREDTS	TIMESTAMP NOT NULL WITH DEFAULT Last alter for table.
REFRESH_TIME	TIMESTAMP NOT NULL WITH DEFAULT Last refresh for table.
SUPPORTLEVEL	SMALLINT NOT NULL (internal use)
ARCHIVE	CHAR(1) Archive status for accelerated table: • A (archived table). Accelerator has both active and archived data. • B (partially archived table). Accelerator can answer queries that contain active data.

Column Name	Data Type/Description
	• C (accelerator has active data). Archived data is in different accelerator. • X (table restored on different accelerator server). Accelerator cannot answer queries. • Blank—Table not archived.

SYSIBM.DSNPROGAUTH		(DSNMCCDB.DSNMDCTS)
DSNPROGAUTH_IDX	U	(PROGNAME,PLANNAME)

This table contains the following columns:

Name	Data Type/Description
PROGNAME	VARCHAR(24) NOT NULL Application program that can run the plan.
PLANNAME	VARCHAR(24) NOT NULL Application plan for program.
PROGMDCVAL	CHAR(16) FOR BIT DATA WITH DEFAULT X'00000000000000000000000000000000' Reserved.
PROGMDCPAD	CHAR(1) NOT NULL WITH DEFAULT (reserved)
CREATOR	VARCHAR(128) NOT NULL WITH DEFAULT CURRENT SQLID Base alias owner.
ENABLED	CHAR(1) NOT NULL WITH DEFAULT 'N' Indicates whether program authorization is enabled: Y (enabled) or N (disabled).
CREATEDTS	TIMESTAMP NOT NULL WITH DEFAULT Timestamp when row was inserted.
REMARKS	VARCHAR(762) User specified text.

5. USER TABLES

PLAN_TABLE

The Plan Table contains at least one row for each table that is referenced in each SQL statement being explained.

Column Name	Data Type	Description
QUERYNO	INTEGER	A number intended to identify the statement being explained. For a row produced by an EXPLAIN statement, specify the number in the QUERYNO clause, which is an optional part of the SELECT, INSERT, UPDATE, MERGE, and DELETE statements. Otherwise, DB2 assigns a number based on the source listing SQL line number. This numbering method is also used when the SQL is embedded in a compiled SQL function, native SQL procedure, or advanced trigger.
QBLOCKNO	SMALLINT	A number that identifies separate blocks (or sections) of the statement being explained. The values of the numbers are not in a specific order, nor are they necessarily consecutive.
APPLNAME	VARCHAR(24)	Name of the application plan being explained; blank for a dynamic EXPLAIN statement, compiled SQL function, native SQL procedure, or advanced trigger.
PROGNAME	VARCHAR(128)	Name of the DBRM or package being explained; blank for dynamic EXPLAIN statements. For an SQL statement that is embedded in a compiled SQL function, native SQL procedure, or advanced trigger, use the respective name.
PLANNO	SMALLINT	A number identifying the order in which the steps were executed within the QBLOCKNO.
METHOD	SMALLIINT	What join method or sort is used for the step identified by PLANNO: • First (outer) table accessed, continuation of previous table accessed, or not used. • 1—Nested loop join. For ech qualifying row in the outer (composite) table, all matching rows in the inner table are found and local predicates are applied. • 2—Merge scan join. Use the outer (composite) table join values to find all matches in the inner (new) table. Combine the matching rows and place in the result set. For the first outer table row, continue to read and process matching rows until the inner table join values exceed the outer table values. Read next outer table row for next row whose join values match or exceed the join values in the inner table. If the values are new, search the inner table for matches. If the values remain the same, reread the matching inner table rows. Repeat the process until there are no more rows in either table. • 3—Additional sorts needed by ORDER BY, GROUP BY, SELECT DISTINCT, UNION, a quantified predicate, or an IN predicate. This step does not access a new table.

Column Name	Data Type	Description
		• 4—Hybrid join. For each join column in the outer (composite) table, find qualifying index entries and RIDs in the inner table index. Append inner table RIDs to corresponding outer table data and place partial rows in an intermediate (new) table. Also place RIDs in a RID list. Sort intermediate (new) table and RID list in RID sequence. Use the sorted RID list to retrieve inner table rows and apply local predicates. Replace RID values in intermediate table with inner table row values.
CREATOR	VARCHAR(128)	Creator of new table (or materialized view) accessed in this step. Blank if METHOD is 3.
TNAME	VARCHAR(128)	Name of a table, materialized query table, created or declared temporary table, materialized view, or materialized table expression, accessed in this step. Blank if METHOD is 3. This column contains the temporary table name of the work file in the form DSNWFQB (qblockno) when an outer join is materialized or an intermediate result for a UNION ALL, INSERT ALL, or EXCEPT ALL. DSN_DIM_TBLX(qblockno) is used to represent the work file of a star join dimension table. DSN_SPIX_TBLX(qblockno) is used for a sparse index with a sideways table reference.
TABNO	SMALLINT	The sequence of the table in the FROM clause. Value is 0 if METHOD is 3. For IMS use.
ACCESSTYPE	CHAR(2)	Method used to access the new table: • DI—By an intersection of multiple DOCID lists to return the final DOCID list. • DU—By a union of multiple DOCID lists to return the final DOCID list. • DX—By an XML index scan on the index named in ACCESSNAME to return a DOCID list. • E—By direct row access using a row change timestamp column. • H—By hash access. • HN—By has access using an IN predicate or an IN predicate that DB2 generates. • I—By index (identified in ACCESSCREATOR and ACCESSNAME). • IN—By an index scan when the matching predicate contains an IN predicate and the IN-list is accessed through an in-memory table. • I1—By a one-fetch index scan. • M—By a multiple index scan (followed by MX, MI, or MU). • MH—By the hash overflow index named in ACCESSNAME. • MI—By an intersection of multiple indexes. • MU—By a union of multiple indexes. • MX—By an index scan on the index named in ACCESSNAME. • N—By an index scan when the matching predicate contains the IN keyword. • By a work file scan, as a result of a subquery. • NR—Range list access. • P—By a dynamic index anding scan. • R—By a table space scan.

Column Name	Data Type	Description
		• RW—By a workfile scan of the result of the materialized user-defined table function. • V—By buffers for an INSERT statement within a SELECT. • Blank—Not applicable to current row.
MATCHCOLS	SMALLINT	The number of index key columns used in an index scan (ACCESSTYPE=I, I1, N, MX, or DX). Otherwise 0.
ACCESSCREATOR	VARCHAR(128)	The creator of the index used in an index scan (ACCESSTYPE=I, I1, N, MX, or DX). Otherwise blank.
ACCESSNAME	VARCHAR(128)	The name of the index used in an index scan (ACCESSTYPE=I, I1, N, MX, or DX). Otherwise blank.
INDEXONLY	CHAR(1)	Indicates if only index entries are accessed for a step (PLANNO): Y/N.
SORTN_UNIQ	CHAR(1)	Whether a sort is performed on the new table to remove duplicate rows: Y/N.
SORTN_JOIN	CHAR(1)	Whether a sort is performed on the new (inner) table of join method 2 or 4: Y/N.
SORTN_ORDERBY	CHAR(1)	Whether an ORDER BY clause results in a sort on the new table: Y/N.
SORTN_GROUPBY	CHAR(1)	Whether a GROUP BY clause results in a sort on the new table: Y/N.
SORTC_UNIQ	CHAR(1)	Whether a sort is performed on the composite table to remove duplicate rows: Y/N.
SORTC_JOIN	CHAR(1)	Whether a sort is performed on the composite (outer) table of join method 1, 2, or 4: Y/N.
SORTC_ORDERBY	CHAR(1)	Whether an ORDER BY clause or quantified predicate results in a sort on the composite table: Y/N.
SORTC_GROUPBY	CHAR(1)	Whether a GROUP BY clause results in a sort on the composite table: Y/N.
TSLOCKMODE	CHAR(3)	The lock mode to be acquired on the new table, its table space, or table space partitions when isolation determined at BIND time: • IS—Intent Share • IX—Intext eXclusive • S—Share • U—Update • X—eXclusive • SIX—Share with Intext eXclusive • N—UR isolation When isolation determined at RUN time (not at BIND time) : • NS—If the isolation level at execution time is UR, then no lock is acquired; if the isolation level is CS, RS, or RR, a S table space lock is acquired. • NIS—If the isolation level at execution time is UR, then no lock is acquired; if the isolation level is CS, RS, or RR, then an IS table space lock is acquired. • NSS—If the isolation level at execution time is UR, then no lock is acquired; if the isolation level is CS or RS, an IS table space lock is acquired; if the isolation level is RR, an S table space lock is acquired. • SS—If the isolation level at execution time is UR, CS, or RS, an IS table space lock is acquired; if the isolation level is RR, an S table space lock is acquired. The data in this column is right justified. If the column contains a blank, no lock is acquired. When the ACCESSTYPE column is DS, DI, or DU, no lock is acquired for the XML index pages.
TIMESTAMP	CHAR(16)	Deprecated. Use EXPLAIN_TIME.
REMARKS	VARCHAR(762)	A column, initialized to zero length, which can be updated with up to 762 bytes of

Column Name	Data Type	Description
		comments. EXPLAIN(ONLY): reason code corresponding to SQL +395 can be: • 1-41—OPTHINT cannot be applied. • 42—Selectivity override invalid. • 43—Selectivity override error. • 44-49—Extended OPTHINT cannot be applied. COMPARE: unmatched PLAN_TABLE column name.
PREFETCH	CHAR(1)	A character that indicates whether data pages were read in advance by PREFETCH: • S—Pure sequential PREFETCH. • L—PREFETCH through a page list. • D—Optimizer expects dynamic PREFETCH. • U—List prefetch with an unsorted RID list. • Blank—Unknown, or no PREFETCH.
COLUMN_FN_EVAL	CHAR(1)	A character that indicates when an SQL column function is evaluated: • R—At retrieval time from table or index. • S—At sort time for GROUP BY. • X—When an OFFSET clause for an aggregate function is used at retrieval time from table or index. • Y—When an OFFSET clause for an aggregate function during a sort; blank after retrieval and after any sorts, or not applicable.
MIXOPSEQ	SMALLINT	The sequence of the steps in a multiple index operation (MX, MI, MU, DX, DI, or
VERSION	VARCHAR(122)	The version identifier for the package, compiled SQL function, or native SQL procedure; blank if not applicable or for a trigger package created before DB2 12, or without DB2 12 new function activated (TYPE='T'), or when the package is created using the BIND PACKAGE command (the initial version of the package). TYPE='blank' or is embedded in an advanced trigger body.
COLLID	VARCHAR(128)	The collection ID for the package: • DSNDYNAMICSQLCACHE—Indicates a cached statement. • DSNEXPLAINMODEYES—Indicates YES for CURRENT EXPLAIN MODE. • DSNEXPLAINMODEEXPLAIN—Indicates EXPLAIN for CURRENT EXPLAIN MODE, otherwise blank. OR, the schema name of the compiled SQL function, native SQL procedure, or advanced trigger.
ACCESS_DEGREE	SMALLINT	The number of parallel operations performed on the new table. Null if access is not executed in parallel. The number determined at bind time may not reflect the actual number at execution time.
ACCESS_PGROUP_ID	SMALLINT	The identifier of the parallel group for accessing the new table. A parallel group is the collective term for consecutive operations executed in parallel that has the same number of parallel tasks. Null if access is not executed in parallel.
JOIN_DEGREE	SMALLINT	The number of parallel operations used in joining the composite table with the new table. Determined at bind time, the actual number may be different. May be 0 if host variable is used.
JOIN_PGROUP_ID	SMALLINT	The identifier of the parallel group for joining the composite table with the new table. Could change in execution.
SORTC_PGROUP_ID	SMALLINT	Parallel group identifier for the parallel sort of the composite table.
SORTN_PGROUP_ID	SMALLINT	Parallel group identifier for the parallel

Column Name	Data Type	Description
		sort of the new table.
PARALLELISM_MODE	CHAR(1)	Kind of parallelism, if any, at bind time: • I—query I/O parallelism. • C—query CP parallelism. • X—query Sysplex parallelism. • Null-no parallelism.
MERGE_JOIN_COLS	SMALLINT	Number of columns joined during a merge scan join (METHOD = 2). Null if METHOD ¬= 2.
CORRELATION_NAME	VARCHAR(128)	Correlation name of the table or view specified in the statement, otherwise blank.
PAGE_RANGE	CHAR(1)	Whether the table qualifies for page range screening, so plans scan only the needed partitions: Y (Yes)/blank (No).
JOIN_TYPE	CHAR(1)	The type of join: • F—FULL OUTER JOIN • L—LEFT (or RIGHT) OUTER JOIN • P—Pair-wise join • S—STAR JOIN • Blank—INNER JOIN, or not applicable
GROUP_MEMBER	VARCHAR(24)	The member name of the DB2 that executed EXPLAIN. Blank for non-data sharing environment.
IBM_SERVICE_DATA	VARCHAR(254)	IBM internal information.
WHEN_OPTIMIZE	CHAR(1)	Indicates when access path was determined: • B—At bind time, using default filter factors for host variables, parameter markers, or special registers. REOPT(ALWAYS) or REOPT(ONCE) will allow run-time reoptimization with input variables. • R—At run time, bind option REOPT(ALWAYS) or REOPT(ONCE) was specified. • Blank—At bind time, using default filter factors.
QBLOCK_TYPE	CHAR(6)	Type of SQL operation performed for each query block: • SELECT (SELECT) • INSERT (INSERT) • UPDATE (UPDATE) • MERGE (MERGE) • DELETE (DELETE) • SELUPD (SELECT with FOR UPDATE clause) • DELCUR (DELETE WHERE CURRENT OF CURSOR) • UPDCUR (UPDATE WHERE CURRENT OF CURSOR) • CORSUB (Correlated subselect) • TRUNCA (TRUNCATE) • NCOSUB (Non correlated subselect) • TABLEX (Table expression) • TRIGGER (When clause on CREATE TRIGGER) • UNION (UNION) • UNIONA (UNION ALL) • INTERS (INTERSECT) • INTERA (INTERSECT ALL) • EXCEPT (EXCEPT) • EXCEPTA (EXCEPT ALL) • PRUNED (No access path generated—guarantee no rows qualify (like WHERE 0=1))
BIND_TIME	TIMESTAMP	Not used. Use EXPLAIN_TIME.
OPTHINT	VARCHAR(128)	A string identifying the row as an optimization hint. DB2 uses this row as input during access path selection.
HINT_USED	VARCHAR(128)	Contains the identifier (value of OPTHINT) when DB2 uses an optimization hint. • APREUSE—Access path reused.

Column Name	Data Type	Description
		• SYSQUERYPLAN query-id—Statement level access path hint used. • SYSQUERYSEL query-id—Predicate selectivity override used. • EXPLAIN PACKAGE: COPY copy-id—Explain package executed: 0 (current copy), 1 (previous copy), 2 (original copy) • "value"—PLAN_TABLE OPTHINT used.
PRIMARY_ ACCESSTYPE	CHAR(1)	Direct row access indicator. • D—DB2 tries direct row access at runtime. If unable, DB2 uses access described in ACCESSTYPE. • P—DPSI and part-level used. • S—Sparse index • T—Base table or result file is materialized. • blank—DB2 will not attempt direct row access.
PARENT_QBLOCKNO	SMALLINT	A number that indicates the QBLOCKNO of the parent query block.
TABLE_TYPE	CHAR(1)	The type of new table: • B—Buffers for an INSERT statement within a SELECT. • C—Common table function. • F—Table function. • I—New table generated from IN-list. • M—Materialized query table. • Q—Temporary intermediate result table (not materialized). • R—Recursive common table function. • S—Subquery. • T—Table. • W—Work file. The value of the column is null if the query uses GROUP BY, ORDER BY, or DISTINCT, which require an implicit sort.
TABLE_ENCODE	CHAR(1)	The encoding scheme of the table. If the table has a single ccsid set, possible values: • A (ASCII) • E (EBCDIC) • U (Unicode. Multiple CCSID set tables always have: • M—Multiple
TABLE_SCCSID	SMALLINT	The SBCS CCSID. If TABLE_ENCODE='M', the value is 0.
TABLE_MCCSID	SMALLINT	The Mixed CCSID. If TABLE_ENCODE='M', the value is 0.
TABLE_DCCSID	SMALLINT	The DBCS CCSID. If TABLE_ENCODE='M', the value is 0.
ROUTINE_ID	INTEGER	Values are for IBM only.
CTEREF	SMALLINT	The top level query block number if table is a common table expression.
STMTTOKEN	VARCHAR(240)	User-specified statement token.
PARENT_PLANNO	SMALLINT	Corresponds to the plan number in the parent query block where a correlated subquery is involved. Corresponds to the plan number in the parent query block that represents the work file for the subquery, for non-correlated subqueries.
BIND_EXPLAIIN_ONLY	CHAR(1)	Bind was done with EXPLAIN(ONLY).
SECTNOI	INTEGER	Section number of statement. Can be used to join to SYSPACKSTMT and SYSSTMT tables.
EXPLAIN_TIME	TIMESTAMP	Time of the explain. • All cached statements--When statement entered cache • Non-cached static statements—When statement was bound

Column Name	Data Type	Description
		• Non-cached dynamic statements—When EXPLAIN was executed using format of CHAR(16) of the time appended by 4 zeros.
MERGC	CHAR(1)	Indicates if the composite table is consolidated before the join. Y=Yes, N=No.
MERGN	CHAR(1)	Indicates if the new table is consolidated before the join. • Y—Yes. • N—No. • D—DPSI in merge operation. • U—DPSI without merge operation.
SCAN_DIRECTION	CHAR(1)	Index access direction: • F—Forward. • R—Reverse.
EXPANSION_REASON	CHAR(2)	Relates to archive/temporal tables; otherwise blank. Special registers control global variables/special registers: • A—SYSIBM.GET_ARCHIVE used. • B—CURRENT TEMPORAL BUSINESS_TIME used. • S—CURRENT TEMPORAL SYSTEM_TIME used. • SB—CURRENT TEMPORAL SYSTEM_TIME and CURRENT TEMPORAL BUSINESS_TIME used. • blank—no implicit query transformation.
PER_STMT_ID	BIGINT	Persistent statement identifier that is the same as: • STMT_ID in SYSIBM.SYSPACKSTMT for SQL statements in packages. • SDQ_STMT_ID in SYSIBM.SYSDYNQUERY for stabilized dynamic SQL statements.

DSN_COLDIST_TABLE

The Column Distribution Table (DSN_COLDIST_TABLE) contains non-uniform column group statistics that are obtained dynamically by the DB2 Optimizer.

Column Name	Data Type	Description
QUERYNO	INTEGER	See description in PLAN_TABLE p. 5-1.
APPLNAME	VARCHAR(128)	See description in PLAN_TABLE p. 5-1.
PROGNAME	VARCHAR(128)	See description in PLAN_TABLE p. 5-1.
COLLID	VARCHAR(128)	See description in PLAN_TABLE p. 5-1.
GROUP_ MEMBER	VARCHAR(128)	Member name of the DB2 that executed EXPLAIN. Blank for non-data sharing environment.
SECTNO	INTEGER	See description in PLAN_TABLE p. 5-1.
VERSION	VARCHAR(122)	See description in PLAN_TABLE p. 5-1.
EXPLAIN_TIME	TIMESTAMP	EXPLAIN timestamp.
SCHEMA	VARCHAR(128)	Schema of the table that contains the column.
TBNAME	VARCHAR(128)	Name of table that contains the column.
NAME	VARCHAR(128)	Name of column.
COLVALUE	VARCHAR(2000)	Contains data of a frequently occurring value in the column.
TYPE	CHAR(1)	Type of statistics: • C (Cardinality) • F (Frequent value) • H (Histogram) • T (Real-time table cardinality) • L (Real time unique index column cardinality) • P (real time partition cardinality
CARDF	FLOAT	• TYPE=C: Number of distinct values for the column group. • TYPE=H: Number of distinct values for the column group in a quantile (QUANTILENO).
COLGROUP COLNO	VARCHAR(254)	An array of SMALLIINT column numbers (0 for single column)
NUM COLUMNS	SMALLINT	Number of columns in group.
FREQUENCY	FLOAT	Percentage of rows in table with the value specified in COLVALUE
QUANTILENO	SMALLINT	Ordinary sequence number of quantile: L (column number) or P (partition number)
LOWVALUE	VARCHAR(2000)	If TYPE=H, then lower bound for the quantile, else not used.
HIGHVALUE	VARCHAR(2000)	If TYPE=H, then higher bound for the quantile, else not used.
EXPANSION_ REASON	CHAR(2)	Relates to archive/temporal tables; otherwise blank. Special registers control global variables/special registers: • A—SYSIBM.GET_ARCHIVE used. • B—CURRENT TEMPORAL BUSINESS_TIME used. • S—CURRENT TEMPORAL SYSTEM_TIME used. • SB—CURRENT TEMPORAL SYSTEM_TIME and CURRENT TEMPORAL BUSINESS_TIME used. • blank—No implicit query transformation.
PER_STMT_ID	BIGINT	See description in PLAN_TABLE p. 5-1.

DSN_DETCOST_TABLE

The Detailed Cost Table (DSN_DETCOST_TABLE) contains information about detailed cost estimation of the mini-plans in a query.

Column Name	Data Type	Description
QUERYNO	INTEGER	See description in PLAN_TABLE p. 5-1.
QBLOCKNO	SMALLINT	See description in PLAN_TABLE p. 5-1.
APPLNAME	VARCHAR(24)	See description in PLAN_TABLE p. 5-1.
PROGNAME	VARCHAR(128)	See description in PLAN_TABLE p. 5-1.
PLANNO	SMALLINT	See description in PLAN_TABLE p. 5-1.
OPENIO	FLOAT(4)	Do-at-open IO cost for non-correlated subquery.
OPENCPU	FLOAT(4)	Do-at-open CPU cost for non-correlated subquery.
OPENCOST	FLOAT(4)	Do-at-open total cost for non-correlated subquery.
DMIO	FLOAT(4)	IBM internal use only.
DMCPU	FLOAT(4)	IBM internal use only.
DMTOT	FLOAT(4)	IBM internal use only.
SUBQIO	FLOAT(4)	IBM internal use only.
SUBQCPU	FLOAT(4)	IBM internal use only
SUBCOST	FLOAT(4)	IBM internal use only.
BASEIO	FLOAT(4)	IBM internal use only.
BASECPU	FLOAT(4)	IBM internal use only.
BASETOT	FLOAT(4)	IBM internal use only.
ONECOMPROWS	FLOAT(4)	Number of rows qualified after applying local predicates.
IMLEAF	FLOAT(4)	Number of index leaf pages scanned by Data Manager.
IMIO	FLOAT(4)	IBM internal use only.
IMPREFH	CHAR(2)	IBM internal use only.
IMMPRED	INTEGER	IBM internal use only.
IMFF	FLOAT(4)	Filter factor of matching predicates only.
IMSRPRED	INTEGER	IBM internal use only.
IMFFADJ	FLOAT(4)	Filter factor of matching and screening predicates.
IMSCANCST	FLOAT(4)	IBM internal use only.
IMROWCST	FLOAT(4)	IBM internal use only.
IMPAGECST	FLOAT(4)	IBM internal use only.
IMRIDSORT	FLOAT(4)	IBM internal use only.
IMMERGCST	FLOAT(4)	IBM internal use only.
IMCPU	FLOAT(4)	IBM internal use only.
IMTOT	FLOAT(4)	IBM internal use only.
IMSEQNO	SMALLINT	IBM internal use only.
DMPREFH	CHAR(2)	IBM internal use only.
DMCLUDIO	FLOAT(4)	IBM internal use only.
DMNCLUDIO	FLOAT(4)	IBM internal use only
DMPREDS	INTEGER	IBM internal use only.
DMSROWS	FLOAT(4)	IBM internal use only.
DMSCANCST	FLOAT(4)	IBM internal use only.
DMCOLS	SMALLINT	Number of DM columns.
DMROWS	FLOAT(4)	Number of DM rows returned
RDSROWCST	FLOAT(4)	IBM internal use only.
DMPAGECST	FLOAT(4)	IBM internal use only.
DMDATAIO	FLOAT(4)	IBM internal use only.
DMDATACPU	FLOAT(4)	IBM internal use only.
DMDATATOT	FLOAT(4)	IBM internal use only.

Column Name	Data Type	Description
RDSROW	FLOAT(4)	Number of RDS rows returned.
SNCOLS	SMALLINT	Number of sort input columns.
SNROWS	FLOAT(4)	Number of sort input rows.
SNRECSZ	INTEGER	Record size for new table.
SNPAGES	FLOAT(4)	Page size for new table.
SNRUNS	FLOAT(4)	Number of runs generated for sort of new table.
SNMERGES	FLOAT(4)	Number of merges needed during sort.
SNIOCOST	FLOAT(4)	IBM internal use only.
SNCPUCOST	FLOAT(4)	IBM internal use only.
SNCOST	FLOAT(4)	IBM internal use only.
SNSCANIO	FLOAT(4)	IBM internal use only.
SNSCANCPU	FLOAT(4)	IBM internal use only.
SNSCANCOST	FLOAT(4)	IBM internal use only
SNCCOLS	SMALLINT	Number of columns as sort input for composite table.
SCROWS	FLOAT(4)	Number of rows as sort input for composite table.
SCRECSZ	FLOAT(4)	Record size for composite table.
SCPAGES	FLOAT(4)	Page size composite table.
SCRUNS	FLOAT(4)	Number of runs generated during sort of composite table.
SCMERGES	FLOAT(4)	Number of merges generated during sort of composite table.
SCIOCOST	FLOAT(4)	IBM internal use only.
SCCPUCOST	FLOAT(4)	IBM internal use only.
SCCOST	FLOAT(4)	IBM internal use only.
SCSCANIO	FLOAT(4)	IBM internal use only.
SCSCANCPU	FLOAT(4)	IBM internal use only.
SCSCAN COST	FLOAT(4)	IBM internal use only.
COMPCARD	FLOAT(4)	Total composite cardinality.
COMPIO COST	FLOAT(4)	IBM internal use only.
COMPCPU COST	FLOAT(4)	IBM internal use only.
COMPCOST	FLOAT(4)	Total cost.
JDINCOLS	SMALLINT	IBM internal use only.
EXPLAIN_ TIME	TIMESTAMP	EXPLAIN timestamp.
COSTBLK	INTEGER	IBM internal use only.
COSTSTOR	INTEGER	IBM internal use only.
MPBLK	INTEGER	IBM internal use only.
MPSTOR	INTEGER	IBM internal use only.
COMPOSITES	INTEGER	IBM internal use only.
CLIPPED	INTEGER	IBM internal use only.
PARTITION	INTEGER	IBM internal use only
TABREF	VARCHAR (64)	IBM internal use only.
MAX_COMPOSITES	INTEGER	IBM internal use only.
MAX_STOR	INTEGER	IBM internal use only.
MAX_CPU	INTEGER	IBM internal use only.
MAX_ELAP	INTEGER	IBM internal use only.
TBL_JOINED_THRESH	INTEGER	IBM internal use only.
STOR_USED	INTEGER	IBM internal use only.
CPU_USED	INTEGER	IBM internal use only.
ELAPSED	INTEGER	IBM internal use only.
MIN_CARD_ KEEP	FLOAT(4)	IBM internal use only.
MAX_CARD_KEEP	FLOAT(4)	IBM internal use only.
MIN_COST_ KEEP	FLOAT(4)	IBM internal use only.

Column Name	Data Type	Description
MAX_COST_KEEP	FLOAT(4)	IBM internal use only.
MIN_VALUE_KEEP	FLOAT(4)	IBM internal use only.
MIN_VALUE_CARD_KEEP	FLOAT(4)	IBM internal use only.
MIN_VALUE_COST_KEEP	FLOAT(4)	IBM internal use only.
MAX_VALUE_KEEP	FLOAT(4)	IBM internal use only.
MAX_VALUE_CARD_KEEP	FLOAT(4)	IBM internal use only.
MAX_VALUE_COST_KEEP	FLOAT(4)	IBM internal use only.
MIN_CARD_ CLIP	FLOAT(4)	IBM internal use only.
MAX_CARD_CLIP	FLOAT(4)	IBM internal use only.
MIN_COST_ CLIP	FLOAT(4)	IBM internal use only.
MAX_COST_CLIP	FLOAT(4)	IBM internal use only.
MIN_VALUE_CLIP	FLOAT(4)	IBM internal use only.
MIN_VALUE_CARD_CLIP	FLOAT(4)	IBM internal use only.
MIN_VALUE_COST_CLIP	FLOAT(4)	IBM internal use only.
MAX_VALUE_CLIP	FLOAT(4)	IBM internal use only.
MAX_VALUE_CARD_CLIP	FLOAT(4)	IBM internal use only.
MAX_VALUE_COST_CLIP	FLOAT(4)	IBM internal use only.
GROUP_ MEMBER	VARCHAR (24)	Data sharing group member name executing EXPLAIN.
PSEQIOCOST	FLOAT(4)	IBM internal use only.
PSEQCPU COST	FLOAT(4)	IBM internal use only.
PSEQCOST	FLOAT(4)	IBM internal use only.
PADJIOCOST	FLOAT(4)	IBM internal use only.
PADJCPU COST	FLOAT(4)	IBM internal use only.
PADJCOST	FLOAT(4)	IBM internal use only.
UNCERTAINTY	FLOAT(4)	Uncertainty factor of inner table index access.
UNCERTAINTY_1T	FLOAT(4)	Uncertainty factor of ONECOMPROWS column of table.
SECTNOI	INTEGER	See description in PLAN_TABLE p. 5-1.
COLLID	VARCHAR (128)	See description in PLAN_TABLE p. 5-1.
VERSION	VARCHAR (122)	See description in PLAN_TABLE p. 5-1.
IMNP	FLOAT(4)	IBM internal use only.
DMNP	FLOAT(4)	IBM internal use only.
IMJC	FLOAT(4)	IBM internal use only.
IMFC	FLOAT(4)	IBM internal use only.
IMJBC	FLOAT(4)	IBM internal use only.
IMJFC	FLOAT(4)	IBM internal use only.
CRED	INTEGER	IBM internal use only.
IXSCAN_SKIP_DUPS	CHAR(1)	Are duplicate index keys skipped during scan: Y/N.
IXSCAN_SKIP_SCREEN	CHAR(1)	Are key ranges disqualified by index screening skipped during index scan: Y: Disqualified key ranges skipped. N: Key ranges not skipped.
EARLY_CUT	CHAR(1)	Does fetching from table stop after first qualifying row: Y/N/blank. If blank, shows explain information from previous release.
EXPANSION_REASON	CHAR(2)	Relates to archive/temporal tables; otherwise blank. Special registers control global variables/special registers: ▪ A—SYSIBM.GET_ARCHIVE used. ▪ B—CURRENT TEMPORAL BUSINESS_TIME used. ▪ S—CURRENT TEMPORAL SYSTEM_TIME used. ▪ SB—CURRENT TEMPORAL SYSTEM_TIME and CURRENT TEMPORAL BUSINESS_TIME used.

Column Name	Data Type	Description
		▪ blank—no implicit query transformation.
BLOCK_FETCH	CHAR(1)	Indicates whether block fetch was used for the query. Y/N
PER_STMT_ID	BIGINT	See description in PLAN_TABLE p. 5-1.

DSN_FILTER_TABLE

The Filter Table (DSN_FILTER_TABLE) contains information about how predicates are used during query processing.

Column Name	Data Type	Description
QUERYNO	INTEGER	See description in PLAN_TABLE p. 5-1.
QBLOCKNO	SMALLINT	See description in PLAN_TABLE p. 5-1.
PLANNO	SMALLINT	See description in PLAN_TABLE p. 5-1.
APPLNAME	VARCHAR (24)	See description in PLAN_TABLE p. 5-1.
PROGNAME	VARCHAR (128)	See description in PLAN_TABLE p. 5-1.
COLLID	VARCHAR (128)	See description in PLAN_TABLE p. 5-1.
ORDERNO	INTEGER	Sequence number of evaluation.
PREDNO	INTEGER	Predicate number.
STAGE	CHAR(9)	Indicates stage at which predicate is evaluated: Matching, Screening, Pagerange, Stage 1, Stage 2.
ORDER CLASS	INTEGER	IBM Internal use only.
EXPLAIN_ TIME	TIMESTAMP	Explain Timestamp.
MIXOPSEQ NO	SMALLINT	IBM Internal use only.
REEVAL	CHAR(1)	IBM Internal use only.
GROUP_ MEMBER	VARCHAR (24)	Data sharing member executing EXPLAIN
SECTNOI	INTEGER	See description in PLAN_TABLE p. 5-1.
VERSION	VARCHAR (122)	See description in PLAN_TABLE p. 5-1.
PUSHDOWN	CHAR(1)	Indicates if the predicate is pushed down the Index Manager or Data Manager for evaluation: ▪ I – Index Manager. ▪ D – Data Manager. ▪ blank – no push down.
EXPANSION_ REASON	CHAR(2)	Relates to archive/temporal tables; otherwise blank. Special registers control global variables/special registers: ▪ A—SYSIBM.GET_ARCHIVE used. ▪ B—CURRENT TEMPORAL BUSINESS_TIME used. ▪ S—CURRENT TEMPORAL SYSTEM_TIME used. ▪ SB—CURRENT TEMPORAL SYSTEM_TIME and CURRENT TEMPORAL BUSINESS_TIME used. ▪ blank—no implicit query transformation.
PER_STMT_ID	BIGINT	See description in PLAN_TABLE p. 5-1.

DSN_FUNCTION_TABLE

The Function Table (DSN_FUNCTION_TABLE) contains a row describing how DB2 resolved the function reference for each user-defined function that is referenced to be an explainable statement.

Column Name	Data Type	Description
QUERYNO	INTEGER	See description in PLAN_TABLE p. 5-1.
QBLOCKNO	INTEGER	See description in PLAN_TABLE p. 5-1.
APPLNAME	VARCHAR (24)	See description in PLAN_TABLE p. 5-1.
PROGNAME	VARCHAR (128)	See description in PLAN_TABLE p. 5-1.
COLLID	VARCHAR (128)	See description in PLAN_TABLE p. 5-1.
GROUP_ MEMBER	VARCHAR (24)	The member name of the DB2 that executed EXPLAIN. Blank for non-data sharing environment.
EXPLAIN_TIME	TIMESTAMP	See description in PLAN_TABLE
SCHEMA_ NAME	VARCHAR (128)	Schema name of the function invoked in the explained statement.
FUNCTION_ NAME	VARCHAR (128)	Name of the function invoked in the explained statement.
SPEC_FUNC_ NAME	VARCHAR (128)	Specific name of the function invoked in the explained statement.
FUNCTION_ TYPE	CHAR (2)	Type of the function invoked in the explainable statement: ▪ CU - Column function. ▪ SU - Scalar function ▪ TU - Table function
VIEW_ CREATOR	VARCHAR (128)	Creator of view definition to which the specified function, e.g., FUNCTION_ NAME is referenced. Otherwise blank.
VIEW_NAME	VARCHAR (128)	Name of view definition to which specified function, i.e. FUNCTION_NAME is referenced. Otherwise blank.
PATH	VARCHAR (2048)	Value of the SQL path used to resolve the schema name of the function.
FUNCTION_ TEXT	VARCHAR (1500)	Function name and parameters of the referenced function. For function references over 1500 bytes, only the first 1500 bytes are contained.
FUNC_ VERSION	VARCHAR (122)	Version of non-inline SQL scalar function.
SECURE	CHAR(1)	Whether the user-defined function is secure.
SECTIONI	INTEGER	Section number of statement.
VERSION	VARCHAR (122)	See description in PLAN_TABLE p. 5-1.
EXPANSION_ REASON	CHAR(2)	Relates to archive/temporal tables; otherwise blank. Special registers control global variables/special registers: ▪ A—SYSIBM.GET_ARCHIVE used. ▪ B—CURRENT TEMPORAL BUSINESS_TIME used. ▪ S—CURRENT TEMPORAL SYSTEM_TIME used. ▪ SB—CURRENT TEMPORAL SYSTEM_TIME and CURRENT TEMPORAL BUSINESS_TIME used. ▪ blank—no implicit query transformation.
PER_STMT_ID	BIGINT	See description in PLAN_TABLE p. 5-1.

DSN_KEYTGTDIST_TABLE

The Key Target Distribution Table (DSN_KEYTGTDIST_TABLE) contains non-uniform index expression statistics that are obtained dynamically by the DB2 optimizer.

Column Name	Data Type	Description
QUERYNO	INTEGER	See description in PLAN_TABLE p. 5-1.
APPLNAME	VARCHAR (128)	See description in PLAN_TABLE p. 5-1.
PROGNAME	VARCHAR (128)	See description in PLAN_TABLE p. 5-1.
COLLID	VARCHAR (128)	See description in PLAN_TABLE p. 5-1.
GROUP_ MEMBER	VARCHAR (128)	Member name of the DB2 that executed the EXPLAIN.
SECTNOI	INTEGER	See description in PLAN_TABLE p. 5-1.
VERSION	VARCHAR (122)	See description in PLAN_TABLE p. 5-1.
EXPLAIN_ TIME	TIMESTAMP	EXPLAIN timestamp.
IXSCHEMA	VARCHAR (128)	Index qualifier.
IXNAME	VARCHAR (128)	Index name.
KEYSEQ	VARCHAR (128)	Numeric position of key-target.
KEYVALUE	VARCHAR(2000)	Data of a frequently occurring value.
TYPE	CHAR(1)	Type of statistics: C – Cardinality. F – Frequent value. H – Histogram. I – Real-time index statistics
CARDF	FLOAT	TYPE=C – Number of distinct values for column group. TYPE=H – Number of distinct values for column group in a quantile indicated by QUANTILENO.
KEYGROUP KEYNO	VARCHAR (254)	Array of SMALLINT values that identifies the set of keys associated with more than a single key.
NUMKEYS	SMALLINT	Number of keys associated with the statistics.
FREQUENCYF	FLOAT	Percentage of rows in table with the value that is specified in the COLVALUE column.
QUANTILENO	SMALLINT	Ordinary sequence number of a quantile in whole consecutive value range.
LOWVALUE	VARCHAR(2000)	TYPE=H – Lower bound for the quantile indicated by QUANTILENO column.
HIGHVALUE	VARCHAR(2000)	TYPE=H – Higher bound for the quantile indicated by QUANTILENO column.
EXPANSION_ REASON	CHAR(2)	Relates to archive/temporal tables; otherwise blank. Special registers control global variables/special registers: ▪ A—SYSIBM.GET_ARCHIVE used. ▪ B—CURRENT TEMPORAL BUSINESS_TIME used. ▪ S—CURRENT TEMPORAL SYSTEM_TIME used. ▪ SB—CURRENT TEMPORAL SYSTEM_TIME and CURRENT TEMPORAL BUSINESS_TIME used. ▪ blank—no implicit query transformation.
PER_STMT_ID	BIGINT	See description in PLAN_TABLE p. 5-1.

DSN_PGRANGE_TABLE

The Page Range Table (DSN_PGRANGE_TABLE) contains information about qualified partitions for all page range scans.

Column Name	Data Type	Description
QUERYNO	INTEGER	See description in PLAN_TABLE p. 5-1.
QBLOCKNO	SMALLINT	See description in PLAN_TABLE p. 5-1.
TABNO	SMALLINT	Table number which uniquely identifies the corresponding table reference within a query.
RANGE	SMALLINT	Sequence number of current page range.
FIRSTPART	SMALLINT	Starting partition in current page range.
LASTPART	SMALLINT	Ending partition in current page range.
NUMPARTS	SMALLINT	Number of partitions in current page range.
EXPLAIN_ TIME	SMALLINT	EXPLAIN timestamp.
GROUP_ MEMBER	VARCHAR (24)	Member name of DB2 subsystem that executed EXPLAIN.
SECTNOI	INTEGER	See description in PLAN_TABLE p. 5-1.
APPLNAME	VARCHAR (24)	See description in PLAN_TABLE p. 5-1.
PROGNAME	VARCHAR (128)	See description in PLAN_TABLE p. 5-1.
COLLID	VARCHAR (128)	See description in PLAN_TABLE p. 5-1.
VERSION	VARCHAR (122)	See description in PLAN_TABLE p. 5-1.
EXPANSION_ REASON	CHAR(2)	Relates to archive/temporal tables; otherwise blank. Special registers control global variables/special registers: ▪ A—SYSIBM.GET_ARCHIVE used. ▪ B—CURRENT TEMPORAL BUSINESS_TIME used. ▪ S—CURRENT TEMPORAL SYSTEM_TIME used. ▪ SB—CURRENT TEMPORAL SYSTEM_TIME and CURRENT TEMPORAL BUSINESS_TIME used. ▪ blank—no implicit query transformation.
PER_STMT_ID	BIGINT	See description in PLAN_TABLE p. 5-1.

DSN_PGROUP_TABLE

The Parallel Group Table (DSN_PGROUP_TABLE) contains information about the parallel groups in a query.

Column Name	Data Type	Description
QUERYNO	INTEGER	See description in PLAN_TABLE p. 5-1.
QBLOCKNO	SMALLINT	See description in PLAN_TABLE p. 5-1.
PLANNAME	VARCHAR (24)	See description in PLAN_TABLE p. 5-1.
COLLID	VARCHAR(128)	See description in PLAN_TABLE p. 5-1.
PROGNAME	VARCHAR(128)	See description in PLAN_TABLE p. 5-1.
EXPLAIN_ TIME	TIMESTAMP	EXPLAIN timestamp.
VERSION	VARCHAR(122)	See description in PLAN_TABLE p. 5-1.
GROUPID	SMALLINT	Parallel group identifier within the current query block.
FIRSTPLAN	SMALLINT	Plan number of first contributing mini-plan associated within this parallel group.
LASTPLAN	SMALLINT	Plan number of last mini-plan associated with this parallel group.
CPUCOST	FLOAT	Estimated total CPU cost in milliseconds.
IOCOST	FLOAT	Estimated total I/O cost in milliseconds.
BESTTIME	FLOAT	Estimated elapsed time.
DEGREE	SMALLINT	Degree of parallelism.
MODE	CHAR(1)	Parallel mode: I – I/O parallelism. C – CPU parallelism. X – multiple CPU SysspleX parallelism (highest level). N – No parallelism.
REASON	SMALLINT	Reason code for downgrading parallelism mode.
LOCALCPU	SMALLINT	Number of CPUs online when preparing the query.
TOTALCPU	SMALLINT	Total number of CPUs in Sysplex.
FIRSTBASE	SMALLINT	Table number of table that partitioning is performed on.
LARGETS	CHAR(1)	Y – Tablespace is large in this group.
PARTKIND	CHAR(1)	Partitioning type: L – Logical. P – Physical.
GROUPTYPE	CHAR(3)	Determines what operations this parallel group contains: A – Table Access. B – Join. C – Sort.
ORDER	CHAR(1)	Ordering requirement of parallel group: N – No order. T – Natural Order (already ordered via index). K – Key order (Sort required (parallel sort)).
STYLE	CHAR(4)	I/O format style of parallel group: blank – I/O parallelism. RIRO – Records IN, Records OUT. WIRO – Work file IN, Records OUT. WIWO – Work file IN, Work file OUT.
RANGEKIND	CHAR(1)	Range type: K – Key range. L – IN-list elements. P – Page range. R – Record range.

Column Name	Data Type	Description
NKEYCOLS	SMALLINT	Number of columns participating in key operation for parallel group.
LOWBOUND	VARCHAR (40)	Low bound of parallel group.
HIGHBOUND	VARCHAR (40)	High bound of parallel group.
LOWKEY	VARCHAR (40)	Low key of range if partitioned by key range.
HIGHKEY	VARCHAR (40)	High key of range if partitioned by key range.
FIRSTPAGE	CHAR(4)	First page in range.
LASTPAGE	CHAR(4)	Last page in range.
GROUP_ MEMBER	VARCHAR(24)	IBM internal use only.
HOST_ REASON	SMALLINT	IBM internal use only.
PARA_TYPE	CHAR(4)	IBM internal use only.
PART_INNER	CHAR(1)	IBM internal use only.
GRNU_ KEYRNG	CHAR(1)	IBM internal use only.
OPEN_ KEYRNG	CHAR(1)	IBM internal use only.
APPLNAME	VARCHAR (24)	See description in PLAN_TABLE p. 5-1.
SECTNOI	INTEGER	See description in PLAN_TABLE p. 5-1.
STRAW_ MODEL	CHAR(1)	IBM internal use only.
EXPANSION_REASON	CHAR(2)	Relates to archive/temporal tables; otherwise blank. Special registers control global variables/special registers: ▪ A—SYSIBM.GET_ARCHIVE used. ▪ B—CURRENT TEMPORAL BUSINESS_TIME used. ▪ S—CURRENT TEMPORAL SYSTEM_TIME used. ▪ SB—CURRENT TEMPORAL SYSTEM_TIME and CURRENT TEMPORAL BUSINESS_TIME used. ▪ blank— no implicit query transformation.
PER_STMT_ID	BIGINT	See description in PLAN_TABLE p. 5-1.

DSN_PREDICAT_TABLE

The Predicate Table (DSN_PREDICAT_TABLE) contains information about all the predicates in a query.

Column Name	Data Type	Description
QUERYNO	INTEGER	See description in PLAN_TABLE p. 5-1.
QBLOCKNO	SMALLINT	See description in PLAN_TABLE p. 5-1.
APPLNAME	VARCHAR (24)	See description in PLAN_TABLE p. 5-1.
PROGNAME	VARCHAR (128)	See description in PLAN_TABLE p. 5-1.
PREDNO	INTEGER	Predicate Number.
TYPE	CHAR(8)	Type of predicate operation.
LEFT_HAND_ SIDE	VARCHAR (128)	If the left hand side of the predicate is a table column (LHS_TABNO > 0) then column name or 'VALUE','COLEXP', 'NONCOLEXP', 'CORSUB', 'NONCORSUB', 'SUBQUERY', 'EXPRESSION', or blanks.
LEFT_HAND_ PNO	INTEGER	Second child predicate when compound predicate (and/or). Note: Use PARENT_PNO to reconstruct predicate tree when predicate tree consolidation occurs.
LHS_TABNO	SMALLINT	Uniquely identifying number of table reference within a query when the left hand side of the predicate is a table column.
LHS_QBNO	SMALLINT	Uniquely identifying number of the query block within a query when the left hand side of the predicate is a subquery.
RIGHT_HAND_ SIDE	VARCHAR (128)	If the right hand side of the predicate is a table column (LHS_TABNO > 0) then column name or 'VALUE', 'COLEXP', 'NONCOLEXP', 'CORSUB', 'NONCORSUB', 'SUBQUERY', 'EXPRESSION', or blanks.
RIGHT_HAND_ PNO	INTEGER	Second child predicate when compound predicate (and/or). Note: Use PARENT_PNO to reconstruct predicate tree when predicate tree consolidation occurs.
RHS_TABNO	CHAR(1)	Uniquely identifying number of table reference within a query when the right hand side of the predicate is a table column.
RHS_QBNO	CHAR(1)	Uniquely identifying number of the query block within a query when the right hand side of the predicate is a subquery.
FILTER_ FACTOR	FLOAT	Estimated filter factor.
BOOLEAN_ TERM	CHAR(1)	Predicate can be used to determine the truth value of the whole WHERE clause.
SEARCHARG	CHAR(1)	Predicate can be processed by DM vs. RDS.
JOIN	CHAR(1)	Whether the predicate can be used as simple join predicate.
AFTER_JOIN	CHAR(1)	Predicate evaluation phase: A – After Join. D – During Join. blank – Not applicable.
ADDED_ PRED	CHAR(1)	Predicate generated by transitive closure. B: Bubble up C: Correlation J: Join K: Like for expression based index L: Localization P: Push down R: Page range S: Simplification

Column Name	Data Type	Description
		T: Transitive closure blank DB2 did not add predicate
REDUNDANT_PRED	CHAR(1)	Redundant predicate.
DIRECT_ ACCESS	CHAR(1)	Predicate is direct access (i.e. uses ROWID).
KEYFIELD	CHAR(1)	Predicate includes the key column of the involved table for all applicable indexes.
EXPLAIN_ TIME	TIMESTAMP	Explain timestamp
CATEGORY	SMALLINT	IBM internal use only.
CATEGORY_B	SMALLINT	IBM internal use only.
TEXT	VARCHAR(2000)	Transformed predicate text.
PRED_ ENCODE	CHAR(1)	IBM internal use only.
PRED_CCSID	SMALLINT	IBM internal use only.
PRED_ MCCSID	SMALLINT	IBM internal use only.
MARKER	CHAR(1)	Predicate includes host variables, parameter markers, or special registers.
PARENT_ PNO	INTEGER	Parent predicate number, 0 if root.
NEGATION	CHAR(1)	Predicate is negated by NOT.
LITERALS	VARCHAR (128)	Literal value(s), separated by commas if more than one.
CLAUSE	CHAR(8)	Clause where the predicate exists: HAVING – Having clause. ON – On clause. WHERE – Where clause. SELECT – Select clause.
GROUP_ MEMBER	VARCHAR(24)	Member name of DB2 subsystem that executed EXPLAIN.
ORIGIN	CHAR(1)	Origin of predicate: blank – Generated by DB2. C – Column Mask. R – Row permission. U – Specified by user.
UNCERTAINTY	FLOAT	Uncertainty factor of the filter factor. Higher value, the more uncertain.
SECTNOI	INTEGER	See description in PLAN_TABLE p. 5-1.
COLLID	VARCHAR(128)	See description in PLAN_TABLE p. 5-1.
VERSION	VARCHAR(122)	See description in PLAN_TABLE p. 5-1.
EXPANSION_ REASON	CHAR(2)	Relates to archive/temporal tables; otherwise blank. Special registers control global variables/special registers: ▪ A—SYSIBM.GET_ARCHIVE used. ▪ B—CURRENT TEMPORAL BUSINESS_TIME used. ▪ S—CURRENT TEMPORAL SYSTEM_TIME used. ▪ SB—CURRENT TEMPORAL SYSTEM_TIME and CURRENT TEMPORAL BUSINESS_TIME used. ▪ blank—no implicit query transformation.
PER_STMT_ID	BIGINT	See description in PLAN_TABLE p. 5-1.

DSN_PREDICAT_SELECTIVITY_TABLE

The Predicate Selectivity Table (DSN_PREDICAT_SELECTIVITY_TABLE) contains information about the selectivity predicates that are used for access paths. It is used as input for BIND QUERY when selectivity override is specified.

Column Name	Data Type	Description
QUERYNO	INTEGER	See description in PLAN_TABLE p. 5-1.
QBLOCKNO	SMALLINT	See description in PLAN_TABLE p. 5-1.
APPLNAME	VARCHAR (24)	See description in PLAN_TABLE p. 5-1.
PROGNAME	VARCHAR (128)	See description in PLAN_TABLE p. 5-1.
SECTNOI	INTEGER	Section number of statement. Can be used to join to SYSPACKSTMT and SYSSTMT tables.
COLLID	VARCHAR(128)	See description in PLAN_TABLE p. 5-1.
VERSION	VARCHAR (122)	See description in PLAN_TABLE p. 5-1.
PREDNO	INTEGER	Predicate number within query.
INSTANCE	SMALLINT	Selectivity instance.
SELECTIVITY	FLOAT	Selectivity estimate.
WEIGHT	FLOAT(4)	Execution percentage having specified selectivity.
ASSUMPTION	VARCHAR (128)	How was selectivity estimated. NORMAL: normal assumption. OVERRIDE: based on an override.
INSERT_TIME	TIMESTAMP	Row change timestamp for when row was inserted/updated.
EXPLAIN_TIME	TIMESTAMP	See description in PLAN_TABLE p. 5-1.
REMARKS	VARCHAR (762)	IBM internal use.
EXPANSION_ REASON	CHAR(2)	Relates to archive/temporal tables; otherwise blank. Special registers control global variables/special registers: ▪ A—SYSIBM.GET_ARCHIVE used. ▪ B—CURRENT TEMPORAL BUSINESS_TIME used. ▪ S—CURRENT TEMPORAL SYSTEM_TIME used. ▪ SB—CURRENT TEMPORAL SYSTEM_TIME and CURRENT TEMPORAL BUSINESS_TIME used. ▪ blank—no implicit query transformation.
PER_STMT_ID	BIGINT	See description in PLAN_TABLE p. 5-1.

DSN_PTASK_TABLE

The Parallel Tasks Table (DSN_PTASK_TABLE) contains information about all the parallel tasks in a query.

Column Name	Data Type	Description
QUERYNO	INTEGER	See description in PLAN_TABLE p. 5-1.
QBLOCKNO	SMALLINT	See description in PLAN_TABLE p. 5-1.
PGDNO	SMALLINT	Paralle group identifier corresponding to DSN_PGROUP_TABLE
APPLNAME	VARCHAR (24)	See description in PLAN_TABLE p. 5-1.
PROGNAME	VARCHAR (128)	See description in PLAN_TABLE p. 5-1.
LPTNO	SMALLINT	Parallel task number
KEYCOLID	SMALLINT	Key column ID.
DPSI	CHAR(1)	Data Partition Secondary Index is used.
LPTLOKEY	VARCHAR (40)	Low key value for key column for parallel task (KEY range only).
LPTHIKEY	VARCHAR (40)	High key value for key column for parallel task (KEY range only).
LPTLOPAG	CHAR(4)	Low page information when partitioned by page range.
LPTHIPAG	CHAR(4)	Low page information when partitioned by page range.
LPTLOPG	CHAR(4)	Lower page bound for parallel task (Page range or DPSI only).
LPTHIPG	CHAR(4)	Upper page bound for parallel task (Page range or DPSI only).
LPTLOPT	SMALLINT	Lower bound partition for parallel task (Page range or DPSI only).
LPTHIPT	SMALLINT	Upper bound partition for parallel task (Page range or DPSI only).
KEYCOLDT	SMALLINT	Data type for key column (KEY range only).
KEYCOLPREC	SMALLINT	Precision/length for key column (KEY range only).
KEYCOLSCAL	SMALLINT	Scale for key column (KEY range with decimal data type only).
EXPLAIN_TIME	TIMESTAMP	EXPLAIN timestamp.
GROUP_MEMBER	VARCHAR (24)	Member name of DB2 subsystem that executed EXPLAIN.
SECTNOI	INTEGER	See description in PLAN_TABLE p. 5-1.
COLLID	VARCHAR (128)	See description in PLAN_TABLE p. 5-1.
VERSION	VARCHAR (122)	See description in PLAN_TABLE p. 5-1.
EXPANSION_ REASON	CHAR(2)	Relates to archive/temporal tables; otherwise blank. Special registers control global variables/special registers: ▪ A—SYSIBM.GET_ARCHIVE used. ▪ B—CURRENT TEMPORAL BUSINESS_TIME used. ▪ S—CURRENT TEMPORAL SYSTEM_TIME used. ▪ SB—CURRENT TEMPORAL SYSTEM_TIME and CURRENT TEMPORAL BUSINESS_TIME used. ▪ blank—no implicit query transformation.
PER_STMT_ID	BIGINT	See description in PLAN_TABLE p. 5-1.

DSN_QUERY_TABLE

The Query Table (DSN_QUERY_TABLE) contains information about an SQL statement, and displays the statement before and after query transformation.

Column Name	Data Type	Description
QUERYNO	INTEGER	See description in PLAN_TABLE p. 5-1.
TYPE	CHAR(8)	Type of data in NODE_DATA column.
QUERY_ STAGE	CHAR(8)	Stage during query transformation when this row is populated.
SEQNO	INTEGER	Sequence number of row when NODE_DATA exceeds size of its column.
NODE_ DATA	CLOB(2M)	XML data containing SQL statement and its query block, table, and column information.
EXPLAIN_ TIME	TIMESTAMP	EXPLAIN timestamp.
QUERY_ ROWID	ROWID	Statement ROWID.
GROUP_ MEMBER	VARCHAR(24)	Member name of DB2 subsystem that executed EXPLAIN.
HASH_KEY	INTEGER	Hash value of contents in NODE_DATA.
HAS_PRED	CHAR(1)	In NODE_DATA contains an SQL statement, indicates if statement contains a parameter marker literal, non-parameter marker literal, or no predicates.
SECTNOI	INTEGER	See description in PLAN_TABLE p. 5-1.
APPLNAME	VARCHAR(24)	See description in PLAN_TABLE p. 5-1.
PROG NAME	VARCHAR (128)	See description in PLAN_TABLE p. 5-1.
COLLID	VARCHAR (128)	See description in PLAN_TABLE p. 5-1.
VERSION	VARCHAR (122)	See description in PLAN_TABLE p. 5-1.
EXPANSION_REASON	CHAR(2)	Relates to archive/temporal tables; otherwise blank. Special registers control global variables/special registers: ▪ A—SYSIBM.GET_ARCHIVE used. ▪ B—CURRENT TEMPORAL BUSINESS_TIME used. ▪ S—CURRENT TEMPORAL SYSTEM_TIME used. ▪ SB—CURRENT TEMPORAL SYSTEM_TIME and CURRENT TEMPORAL BUSINESS_TIME used. ▪ Blank—No implicit query transformation.

DSN_QUERYINFO_TABLE

The Queryinfo Table (DSN_QUERYINFO_TABLE) contains information about: the eligibility of query blocks for automatic rewrite, MQT's being considered for eligible query blocks and reasons why these are not eligible, and acceleration of query blocks.

Column Name	Data Type	Description
QUERYNO	INTEGER	See description in PLAN_TABLE p. 5-1.
QBLOCKNO	SMALLINT	Query block number within the query.
QINAME1	VARCHAR (128)	When TYPE='A" REASON-CODE=0: the value of the accelerator server. REASON-CODE <>0: query not sent to an accelerator. Code indicates why.
QINAME2	VARCHAR (128)	TYPE=A and REASON-CODE=0 : accelerator server.
APPLNAME	VARCHAR (24)	See description in PLAN_TABLE p. 5-1
PROGNAME	VARCHAR (128)	See description in PLAN_TABLE p. 5-1
VERSION	VARCHAR (122)	See description in PLAN_TABLE p. 5-1
COLLID	VARCHAR (128)	See description in PLAN_TABLE p. 5-1.
GROUP_ MEMBER	VARCHAR (24)	Member name of DB2 subsystem that executed the EXPLAIN.
SECTNOI	INTEGER	See description in PLAN_TABLE p. 5-1.
SEQNO	INTEGER	Sequence number, if QI_DATE exceeds size.
EXPLAIN_TIME	TIMESTAMP	Explain timestamp.
TYPE	CHAR(8)	A: DB2 attempts to execute accelerator. Value in REASON_CODE indicated outcome.
REASON_CODE	SMALLINT	Associated with QI_DATA.
QI_DATA	CLOB(2M)	When TYPE=A: ▪ REASON-CODE<>0: description of REASON_CODE. ▪ REASON-CODE=0: value is query text upon conversion for the accelerator.
SERVICE_INFO	BLOB(2M)	IBM internal use.
QB_INFO_ROWID	ROWID	IBM internal use.
EXPANSION_REASON	CHAR(2)	Relates to archive/temporal tables; otherwise blank. Special registers control global variables/special registers: ▪ A—SYSIBM.GET_ARCHIVE used. ▪ B—CURRENT TEMPORAL BUSINESS_TIME used. ▪ S—CURRENT TEMPORAL SYSTEM_TIME used. ▪ SB—CURRENT TEMPORAL SYSTEM_TIME and CURRENT TEMPORAL BUSINESS_TIME used. ▪ Blank—no implicit query transformation.

DSN_SORT_TABLE

The Sort Table (DSN_SORT_TABLE) contains information about the sort operations required by a query.

Column Name	Data Type	Description
QUERYNO	INTEGER	See description in PLAN_TABLE p. 5-1.
QBLOCKNO	SMALLINT	See description in PLAN_TABLE p. 5-1.
PLANNO	SMALLINT	See description in PLAN_TABLE p. 5-1.
APPLNAME	VARCHAR(24)	See description in PLAN_TABLE p. 5-1.
PROGNAME	VARCHAR(128)	See description in PLAN_TABLE p. 5-1.
COLLID	VARCHAR (128)	See description in PLAN_TABLE p. 5-1.
SORTC	CHAR(5)	Reasons for sort of composite table. Reasons are by byte: ▪ Byte 1: G – Group by. ▪ Byte 2: J – Join. ▪ Byte 3: O – Order by. ▪ Byte 4: U – Uniqueness.
SORTN	CHAR(5)	Reasons for sort of new table. Reasons are by byte: ▪ Byte 1: G – Group by. ▪ Byte 2: J – Join. ▪ Byte 3: O – Order by. ▪ Byte 4: U – Uniqueness.
SORTNO	SMALLINT	Sequence number of sort.
KEYSIZE	SMALLINT	Sum of the length of sort keys.
ORDER CLASS	INTEGER	IBM internal use only.
EXPLAIN_ TIME	TIMETAMP	EXPLAIN timestamp.
GROUP_ MEMBER	VARCHAR (24)	Member name of DB2 subsystem that executed EXPLAIN.
SECTNOI	INTEGER	See description in PLAN_TABLE p. 5-1.
VERSION	VARCHAR (122)	See description in PLAN_TABLE p. 5-1.
EXPANSION_ REASON	CHAR(2)	Relates to archive/temporal tables; otherwise blank. Special registers control global variables/special registers: ▪ A—SYSIBM.GET_ARCHIVE used. ▪ B—CURRENT TEMPORAL BUSINESS_TIME used. ▪ S—CURRENT TEMPORAL SYSTEM_TIME used. ▪ SB—CURRENT TEMPORAL SYSTEM_TIME and CURRENT TEMPORAL BUSINESS_TIME used. ▪ Blank—no implicit query transformation.
PER_STMT_ID	BIGINT	See description in PLAN_TABLE p. 5-1.

DSN_SORTKEY_TABLE

The Sort Key Table (DSN_SORTKEY_TABLE) contains information about the sort keys for all the sorts required by a query.

Column Name	Data Type	Description
QUERYNO	INTEGER	See description in PLAN_TABLE p. 5-1.
QBLOCKNO	SMALLINT	See description in PLAN_TABLE p. 5-1.
PLANNO	SMALLINT	See description in PLAN_TABLE p. 5-1.
APPLNAME	VARCHAR (24)	See description in PLAN_TABLE p. 5-1.
PROGNAME	VARCHAR (128)	See description in PLAN_TABLE p. 5-1.
COLLID	VARCHAR (128)	See description in PLAN_TABLE p. 5-1.
SORTNO	SMALLINT	Sequence number of the sort.
ORDERNO	SMALLINT	Sequence number of the sort key.
EXPTYPE	CHAR(3)	Type of sort key.
TEXT	VARCHAR (128)	Sort key text.
TABNO	SMALLINT	Table number uniquely identifying the table reference within the query.
COLNO	SMALLINT	Column number uniquely identifying the column within the query.
DATATYPE	CHAR(18)	Data type of sort key.
LENGTH	INTEGER	Length of sort key.
CCSID	INTEGER	IBM internal use only.
ORDERCLASS	INTEGER	IBM internal use only.
EXPLAIN_ TIME	TIMETAMP	EXPLAIN timestamp.
GROUP_ MEMBER	VARCHAR (24)	Member name of DB2 subsystem that executed EXPLAIN.
SECTNOI	INTEGER	See description in PLAN_TABLE p. 5-1.
VERSION	VARCHAR (122)	See description in PLAN_TABLE p. 5-1.
EXPANSION_ REASON	CHAR(2)	Relates to archive/temporal tables; otherwise blank. Special registers control global variables/special registers: ▪ A—SYSIBM.GET_ARCHIVE used. ▪ B—CURRENT TEMPORAL BUSINESS_TIME used. ▪ S—CURRENT TEMPORAL SYSTEM_TIME used. ▪ SB—CURRENT TEMPORAL SYSTEM_TIME and CURRENT TEMPORAL BUSINESS_TIME used. ▪ Blank—no implicit query transformation.
PER_STMT_ID	BIGINT	See description in PLAN_TABLE p. 5-1.

DSN_STATEMENT_CACHE_TABLE

The Statement Cache Table (DSN_STATEMENT_CACHE_TABLE) contains a row describing how DB2 resolved the function reference for each user-defined function that is referred to be an explainable statement.

Column Name	Data Type	Description
STMT_ID	INTEGER	The statement ID, this value is the EDM unique token for the statement.
STMT_TOKEN	VARCHAR(240)	The statement token. You provide this value as an identification string.
COLLID	VARCHAR(128)	See description in PLAN_TABLE p. 5-4.
PROGRAM_ NAME	VARCHAR(128)	The name of the package or DBRM that performed the initial prepare for the statement.
INV_DROPALT	CHAR (1)	This column is not used.
INV_REVOKE	CHAR (1)	This column is not used.
INV_LRU	CHAR (1)	This column is not used.
INV_ RUNSTATS	CHAR (1)	This column is not used.
CACHED_TS	TIMESTAMP	The timestamp when the statement was stored in the dynamic statement cache.
USERS	INTEGER	The number of current users who have prepared or run the statement during their current unit of work.
COPIES	INTEGER	The number of copies of the statement owned by all threads in the system.
LINES	INTEGER	The precompiler line number from the initial prepare of the statement.
PRIMAUTH	VARCHAR(128)	The primary authorization id that did the initial prepare of the statement.
CURSQLID	VARCHAR(128)	The CURRENT SQLID that did the initial PREPARE of the statement.
BIND_QUALIFIER	VARCHAR(128)	The BIND qualifier. For unqualified table names, this is the object qualifier.
BIND_ISO	CHAR (2)	The BIND ISOLATION in effect for the statement: ▪ UR—Uncommitted Read. ▪ CS—Cursor Stability. ▪ RS—Read Stability. ▪ RR—Repeatable Read.
BIND_CDATA	CHAR (1)	The CURRENTDATA option in effect for the statement: ▪ Y—CURRENTDATA(YES). ▪ N—CURRENTDATA(NO).
BIND_DYNRL	CHAR (1)	The DYNAMICRULES option in effect for the statement: B DYNAMICRULES(BIND) . R DYNAMICRULES(RUN).
BIND_DEGRE	CHAR (1)	The CURRENT DEGREE special register value that is in effect for the statement: A CURRENT DEGREE(ANY). 1 CURRENT DEGREE(1).
BIND_SQLRL	CHAR (1)	The CURRENT RULES special register value that is in effect for the statement: D CURRENT RULES(DB2). S CURRENT RULES(SQL).
BIND_CHOLD	CHAR (1)	The WITH HOLD attribute of the prepare for this statement: Y Prepared WITH HOLD. N Prepared without WITH HOLD.
STAT_TS	TIMESTAMP	The timestamp of the statistics. This is the timestamp when IFCID 318 is started.

Column Name	Data Type	Description
STAT_EXEC	INTEGER	The number of times this statement has been run. For a statement with a cursor, this is the number of OPENs.
STAT_GPAG	INTEGER	The number of GETPAGE operations that are performed for the statement.
STAT_SYNR	INTEGER	The number of synchronous buffer reads that are performed for the statement.
STAT_WRIT	INTEGER	The number of buffer write operations that are performed for this statement.
STAT_EROW	INTEGER	The number of rows that are examined for this statement.
STAT_PROW	INTEGER	The number of rows that are processed for this statement.
STAT_SORT	INTEGER	The number of sorts for this statement.
STAT_INDX	INTEGER	The number of index scans for this statement.
STAT_RSCN	INTEGER	The number of table space scans for this statement.
STAT_PGRP	INTEGER	The number of parallel groups that are created for this statement.
STAT_ELAP	FLOAT	The accumulated elapsed time that is used for the statement.
STAT_CPU	FLOAT	The accumulated CPU time that is used for the statement.
STAT_SUS_SYNIO	FLOAT	The accumulated wait time for synchronous I/O operations for the statement.
STAT_SUS_ LOCK	FLOAT	The accumulated wait time for lock and latch requests for the statement.
STAT_SUS_ SWIT	FLOAT	The accumulated wait time for synchronous execution unit switch for the statement.
STAT_SUS_ GLCK	FLOAT	The accumulated wait time for global locks for this statement.
STAT_SUS_ OTHR	FLOAT	The accumulated wait time for read activity that is done by another thread.
STAT_SUS_OTHW	FLOAT	The accumulated wait time for write activity that is done by another thread.
STAT_RIDLIMT	INTEGER	The number of times a RID list was not used because the number of RIDs would have exceeded DB2 limits.
STAT_ RIDSTOR	INTEGER	The number of times a RID list was not used because there was not enough storage available to hold the list of RIDs.
EXPLAIN_TS	TIMESTAMP	The timestamp for when the statement cache table was populated.
SCHEMA	VARCHAR(128)	The value of the CURRENT SCHEMA special register.
STMT_TEXT	CLOB(2M)	The statement that is being explained.
STMT_ROWID	ROWID	The ROWID of the statement.
BIND_RO_TYPE	CHAR(1)	Current specification of REOPT option for statement: ▪ N—REOPT(NONE). ▪ 1—REOPT(ONCE). ▪ A—REOPT(AUTO). ▪ 0—Current plan is deemed optimal. No need for REOPT(AUTO).
BIND_RA_TOT	INTEGER	Total number of REBIND commands that have been issued for the dynamic statement because of the REOPT(AUTO) option.
GROUP_ MEMBER	VARCHAR(24)	Member name of DB2 subsystem that executed EXPLAIN.

Column Name	Data Type	Description
STAT_EXECB	BIGINT	Number of executions. For cursor operation the number of OPENs.
STAT_GPAGB	BIGINT	Number of get page operations.
STAT_SYNRB	BIGINT	Number of synchronous buffer reads.
STAT_WRITB	BIGINT	Number of buffer write operations.
STAT_EROWB	BIGINT	Number of rows examined.
STAT_PROWB	BIGINT	Number of rows processed.
STAT_SORTB	BIGINT	Number of sorts performed.
STAT_INDXB	BIGINT	Number of index scans.
STAT_RSCNB	BIGINT	Number of table space scans.
STAT_PGRPB	BIGINT	Number of parallel groups created.
STAT_ RIDLIMITB	BIGINT	Number of times RID limit was not used because of DB2 RID limit.
STAT_ RIDSTORB	BIGINT	Number of times RID list was not used because of storage.
LITERAL_REPL	CHAR(1)	Cached Statement where literal value replaced by “&”: ▪ R—Statement prepared with CONCETRATE STATEMENTS WITH LITERALS and literals replaced with “&”. ▪ D—Statement is duplicate statement instance with different literal reusability criteria. ▪ blank—Literal values not replaced.
STAT_SUS_ LATCH	FLOAT	Accumulated wait time for latch requests.
STAT_SUS_ PLATCH	FLOAT	Accumulated wait time for page latch requests.
STAT_SUS_ DRAIN	FLOAT	Accumulated wait time for drain latch requests.
STAT_SUS_ CLAIM	FLOAT	Accumulated wait time for claim count requests.
STAT_SUS_ LOG	FLOAT	Accumulated wait time for log writer requests.
EXPANSION_ REASON	CHAR(2)	Relates to archive/temporal tables; otherwise blank. Special registers control global variables/special registers: ▪ A—SYSIBM.GET_ARCHIVE used. ▪ B—CURRENT TEMPORAL BUSINESS_TIME used. ▪ S—CURRENT TEMPORAL SYSTEM_TIME used. ▪ SB—CURRENT TEMPORAL SYSTEM_TIME and CURRENT TEMPORAL BUSINESS_TIME used. ▪ blank—no implicit query transformation.
ACCELERATED	CHAR(10)	Identifies whether a cached dynamic statement was prepared for acceleration to an accelerator server. Possible values are: ‘NO’, ‘YES’, ‘NEVER’, ‘ACCEL-ONLY’.
STAT_ACC_ELAP	BIGINT	Accumulated elapsed time
STAT_ACC_CPU	BIGINT	Accumulated CPU time
STAT_ACC_ROW	BIGINT	Accumulated number of rows that are returned
STAT_ACC_BYTE	BIGINT	Accumulated number of bytes that are returned
STAT_ACC_1ROW	BIGINT	Time waited for the first row of the query result.
STAT_ACC_DB2	BIGINT	Total time the accelerator waited for DB2 to request query results.
STAT_ACC_EXEC	BIGINT	Accumulated execution time

Column Name	Data Type	Description
STAT_ACC_WAIT	BIGINT	Accumulated queue wait time
ACCEL_OFFLOAD_ ELIGIBLE	CHAR(1)	Statement is a candidate for acceleration (Y/N)
ACCELERATOR_NAME	VARCHAR(128)	Concatenated name of the accelerator server
STAT_SUS_CHILDLLOCKS	FLOAT	Accumulated wait time for child L-locks
STAT_SUS_OTHERLLOCKS	FLOAT	Accumulated wait time for other L-locks
STAT_SUS_PPPLOCKS	FLOAT	Accumulated wait time for P/P P-locks
STAT_SUS_PAGEPLOCKS	FLOAT	Accumulated wait time for page P-locks
STAT_SUS_OTHERPLOCKS	FLOAT	Accumulated wait time for other P-locks.
PER_STMT_ID	BIGINT	See description in PLAN_TABLE p. 5-1.
STBLGRP	VARCHAR(128)	Stabilization group name specified in a START DYNQRY command.
QUERY_HASH	FLOAT	Hash key generated by the statement text.
QUERY_HASH_VERSION	INTEGER	The version of QUERY_HASH
STABILIZED	CHAR(1)	Indicates whether the statement was stabilized.
APPLCOMPAT	CHAR(10)	Application compatibility level of a dynamic SQL statement.
CNO	BIGINT	Zero or command number for the dynamic query capture monitor
STAT_SUS_PIPE	FLOAT	Accumulated wait time for pipe requests

DSN_STATEMNT_TABLE

The Statement Table (DSN_STATEMNT_TABLE) contains a row with cost estimate, in service units, and milliseconds for processing of an explainable statement.

Column Name	Data Type	Description
QUERYNO	INTEGER	See description in PLAN_TABLE p. 5-1.
APPLNAME	VARCHAR(24)	See description in PLAN_TABLE p. 5-1.
PROGNAME	VARCHAR(128)	See description in PLAN_TABLE p. 5-1.
COLLID	VARCHAR(128)	See description in PLAN_TABLE p. 5-1.
GROUP_ MEMBER	VARCHAR(24)	The member name of the DB2 that executed EXPLAIN. Blank for non-data sharing environment.
EXPLAIN_TIME	TIMESTAMP	See description in PLAN_TABLE
STMT_TYPE	CHAR (6)	Type of statement being explained, i.e. SELECT, INSERT, UPDATE, DELETE, SELUPD, DELCUR, or UPDCUR.
COST_CATEGORY	CHAR (1)	Whether DB2 was forced to use default values to make cost estimates. ▪ A--Default values were not used. ▪ B--Some default values were used. See REASON for explanation of why defaults were used.
PROCMS	INTEGER	Estimated processor cost, in milliseconds. Value is rounded to next integer. Max value is 2147483647 milliseconds, i.e. 24.8 days. If max value is exceeded, max value is reported.
PROCSU	INTEGER	Estimated processor cost, in service units. Value is rounded to next integer. Max value is 2147483647 service units. If max value is exceeded, max value is reported.
REASON	VARCHAR(254)	Reason for putting an estimate in COST_CATEGORY_B: ▪ HAVING CLAUSE--A subselect in the SQL statement contains a HAVING clause. ▪ HOST VARIABLES--Statement uses host variables, parameter markers, or special registers. ▪ MATERIALIZATION--Statistics are missing because the statement uses materialized views or nested table expressions. ▪ PROFILED value--Profile mentoring is used (see DSN_PROFILE_TABLE). ▪ REFERENTIAL CONSTRAINTS--RI constraints of CASCADE or SET NULL exist on the target table of a DELETE. ▪ TABLE CARDINALITY--Cardinality statistics are missing for one or more of the tables used in the statement. ▪ TRIGGERS—triggers are defined on the target table of an INSERT, UPDATE, DELETE. ▪ UDF—user-defined functions used.
STMT_ENCODE	CHAR(1)	The encoding scheme of the table. If the table has a single CCSID set, possible values are: ▪ A—ASCII. ▪ E—EBCDIC. ▪ U—Unicode. Multiple CCSID set tables always have: ▪ M—Multiple.

Column Name	Data Type	Description
TOTAL_COST	FLOAT	Overall estimated cost of the statement. This cost should only be used for reference purposes.
SECTNOI	INTEGER	See description in PLAN_TABLE p. 5-1.
VERSION	VARCHAR(122)	See description in PLAN_TABLE p. 5-1.
EXPANSION_ REASON	CHAR(2)	Relates to archive/temporal tables; otherwise blank. Special registers control global variables/special registers: ▪ A—SYSIBM.GET_ARCHIVE used. ▪ B—CURRENT TEMPORAL BUSINESS_TIME used. ▪ S—CURRENT TEMPORAL SYSTEM_TIME used. ▪ SB—CURRENT TEMPORAL SYSTEM_TIME and CURRENT TEMPORAL BUSINESS_TIME used. ▪ blank—no implicit query transformation.
APCOMPARE_STATUS	CHAR(1)	The status of the access path comparison operation for the APCOMPARE option of a BIND or REBIND command S – Succeeded F – New access path does not match the previous access path or the access path comparison operation failed. N – No match found Blank – Default value. Either APCOMPARE is not used or was used prior to DB2 12 or this column was not added in time.
APREUSE_STATUS	CHAR(1)	Status of the access path reuse operation for the APREUSE option of a BIND or REBIND command S – Access path reuse succeeded F – Accpath reuse failed N – No match was found Blank – Default value. APREUSE was not used or APREUSE was used prior to DB2 12 or this column was not added in time.
APREUSE_VERSION	VARCHAR(122)	Version identifier for the package bind whose access path is reused. Blank is used when APREUSE_STATUS is blank.
APREUSE_COPYID	INTEGER	Copy number of identifier for the package whose access plan is being taken to be reused. The default value - 1 blank is used when APREUSE_STATUS is blank.
EXPLAIN_TYPE	CHAR(1)	The type of action that created the row: 'A' - Automatic bind 'B' - BIND command 'C' - EXPLAIN STATEMENT CACHE statement 'D' - Dynamic EXPLAIN statement 'R' - REBIND command 'S' - EXPLAIN STABILIZED DYNAMIC QUERY statement. blank – Default or row existed before this column was added to the table.
PER_STMT_ID	BIGINT	See description in PLAN_TABLE p. 5-1.
QUERY_HASH	CHAR(16)	Hash key generated by statement text.

DSN_STRUCT_TABLE

The Structure Table (DSN_STRUCT_TABLE) contains information about all the query blocks in a query.

Column Name	Data Type	Description
QUERYNO	INTEGER	See description in PLAN_TABLE p. 5-1.
QBLOCKNO	SMALLINT	See description in PLAN_TABLE p. 5-1.
APPLNAME	VARCHAR (24)	See description in PLAN_TABLE p. 5-1.
PROGNAME	VARCHAR (128)	See description in PLAN_TABLE p. 5-1.
PARENT	SMALLINT	Parent query block number.
TIMES	FLOAT	Estimated number of rows returned by DM.
ROWCOUNT	INTEGER	Estimated number of rows returned by RDS.
ATOPEN	CHAR(1)	Query block moved up for do-at-open processing: ▪ Y—Do at open. ▪ N—otherwise.
CONTEXT	CHAR(10)	Context of current query block.
ORDERNO	SMALLINT	Not used.
DOATOPEN_PARENT	SMALLINT	Do-at-open parent if query block is done-at-open.
QBLOCK_ TYPE	CHAR(6)	Type of query block.
EXPLAIN_ TIME	TIMETAMP	EXPLAIN timestamp.
QUERY_ STAGE	CHAR(8)	IBM internal use only.
GROUP_ MEMBER	VARCHAR (24)	Member name of DB2 subsystem that executed EXPLAIN.
ORIGIN	CHAR(1)	Origin of query block: ▪ blank—Generated by DB2. ▪ C—Column mask. ▪ R—Row permission. ▪ U—Specified by user.
SECTNOI	INTEGER	See description in PLAN_TABLE p. 5-1.
COLLID	VARCHAR (128)	See description in PLAN_TABLE p. 5-1.
VERSION	VARCHAR (122)	See description in PLAN_TABLE p. 5-1.
EXPANSION_ REASON	CHAR(2)	Relates to archive/temporal tables; otherwise blank. Special registers control global variables/special registers: ▪ A—SYSIBM.GET_ARCHIVE used. ▪ B—CURRENT TEMPORAL BUSINESS_TIME used. ▪ S—CURRENT TEMPORAL SYSTEM_TIME used. ▪ SB—CURRENT TEMPORAL SYSTEM_TIME and CURRENT TEMPORAL BUSINESS_TIME used. ▪ blank—no implicit query transformation.
PER_STMT_ID	BIGINT	See description in PLAN_TABLE.

DSN_STAT_FEEDBACK_TABLE

The Statistics Feedback Table (DSN_STAT_FEEDBACK_TABLE) contains recommendations for collecting missing or conflicting RUNSTATS information.

Column Name	Data Type	Description
QUERYNO	INTEGER	See description in PLAN_TABLE p. 5-1.
APPLNAME	VARCHAR (24)	See description in PLAN_TABLE p. 5-1.
PROGNAME	VARCHAR (128)	See description in PLAN_TABLE p. 5-1.
COLLID	VARCHAR (128)	See description in PLAN_TABLE p. 5-1.
GROUP_ MEMBER	VARCHAR (24)	Member name of DB2 subsystem that executed EXPLAIN.
EXPLAIN_TIME	TIMESTAMP	EXPLAIN timestamp.
SECTNOI	INTEGER	See description in PLAN_TABLE.
VERSION	VARCHAR (122)	See description in PLAN_TABLE .
TBCREATOR	VARCHAR (128)	Table creator.
TBNAME	VARCHAR (128)	Table name.
IXCREATOR	VARCHAR (128)	Index creator.
IXNAME	VARCHAR (128)	Index name.
COLNAME	VARCHAR (128)	Column name.
NUMCOLUMNS	SMALLINT	Number of columns in COLGROUP.
COLGROUPCOLNO	VARCHAR (254)	Hex representation of columns.
TYPE	CHAR(1)	Statistics to collect: C: cardinality F: frequency H: histogram I: index T: table
DBNAME	VARCHAR (24)	Database name.
TSNAME	VARCHAR (24)	Tablespace name.
REASON	CHAR (8)	Reason why statistics was recommended: ▪ BASIC—Basic statisics for column/index missing. ▪ CONFLICT—Another statistic contains a conflicting value. ▪ COMPFIX—Multi-column cardinality statistics needed to satisfy compound filter factor. ▪ DEFAULT—Predicate references value. ▪ KEYCARD—Index key cardinality missing. ▪ LOWCARD—Cardinality is low value (data can be skewed). ▪ NULLABLE—Distribution statistics not available. ▪ RANGEPRD—Histogram statistics not available. ▪ PARALLEL—Parallelism could be improved by uniform key range partitioning. ▪ STALE - Statistic appears likely to be out of sync.
REMARKS	VARCHAR (254)	Free form text.

DSN_VIEWREF_TABLE

The View Reference Table (DSN_VIEWREF_TABLE) contains information about all the views and MQTs that are used to process a query.

Column Name	Data Type	Description
QUERYNO	INTEGER	See description in PLAN_TABLE p. 5-1.
APPLNAME	VARCHAR (24)	See description in PLAN_TABLE p. 5-1.
PROGNAME	VARCHAR(128)	See description in PLAN_TABLE p. 5-1.
VERSION	VARCHAR(122)	See description in PLAN_TABLE p. 5-1.
COLLID	VARCHAR(128)	See description in PLAN_TABLE p. 5-1.
CREATOR	VARCHAR(128)	Authorization ID of the owner of object.
NAME	VARCHAR(128)	Name of object.
TYPE	CHAR(1)	Type of object: ▪ V—View. ▪ R—MQT replacing a base table via rewrite. ▪ M—MQT.
MQTUSE	SMALLINT	IBM internal use only.
EXPLAIN_ TIME	TIMETAMP	EXPLAIN timestamp.
GROUP_ MEMBER	VARCHAR (24)	Member name of DB2 subsystem that executed EXPLAIN.
SECTNOI	INTEGER	See description in PLAN_TABLE p. 5-1.
COLLID	VARCHAR(128)	See description in PLAN_TABLE p. 5-1.
EXPANSION_ REASON	CHAR(2)	Relates to archive/temporal tables; otherwise blank. Special registers control global variables/special registers: ▪ A—SYSIBM.GET_ARCHIVE used. ▪ B—CURRENT TEMPORAL BUSINESS_TIME used. ▪ S—CURRENT TEMPORAL SYSTEM_TIME used. ▪ SB—CURRENT TEMPORAL SYSTEM_TIME and CURRENT TEMPORAL BUSINESS_TIME used. ▪ blank—no implicit query transformation.

6. MISCELLANEOUS SYSTEM INFORMATION

SQLCA

SQLCA contains the following columns:

Name	Data Type	Description
SQLCAID	CHAR(8)	A dump “eye catcher” that contains SQLCA.
SQLCABC	INTEGER	Length of SQLCA: 136.
SQLCODE	INTEGER	SQL return code: • 0—Successful execution • Positive—Successful execution, but with an exception condition • Negative—Error condition
SQLERRML	SMALLINT	Length of SQLERRMC, ranging from 0 to 70. 0 means SQLERRMC value is not pertinent.
SQLERRMC	VARCHAR (70)	One or more tokens, separated by X’FF’, that are substituted for variables in error condition descriptions.
SQLERRP	CHAR(8)	Provides a product signature and if error, diagnostic information like name of module that found the error. First three (3) characters are always DSN for DB2 for z/OS.
SQLERRD(1)	INTEGER	For a sensitive static cursor, contains the number of rows in a result table when the cursor position is after the last row (that is, when SQLCODE is equal to + 100). SQLERRD(1) can also contain an internal error code.
SQLERRD(2)	INTEGER	For a sensitive static cursor, contains the number of rows in a result table when the cursor position is after the last row (that is, when SQLCODE is equal to + 100). SQLERRD(2) can also contain an internal error code.
SQLERRD(3)	INTEGER	Number of rows affected after INSERT, UPDATE, MERGE, or DELETE (but not rows affected by either triggers, referential integrity constraints, or inserted rows inserted by FOR PORTION OF clause for a BUSINESS_TIME_PERIOD). For the OPEN of a cursor for a SELECT with a data change statement or for a SELECT INTO, SQLERRD(3) contains the number of rows inserted by the embedded data change statement. • Set to 0 if SQL statement fails, indicating that all changes made in executing the statement were cancelled. • Set to –1 for mass delete from table in a segmented tablespace and the DELETE statement did not include selection criteria. • Set to -1 for a TRUNCATE statement. For a successful REFRESH TABLE statement, it contains the number of rows inserted into the MQT. For rowset-oriented FETCH statements,

Name	Data Type	Description
		contains the number of rows returned in the rowset. For SQLCODES –911 and –913, contains the reason code for the timeout or deadlock.
SQLERRD(4)	INTEGER	Generally, contains TOTAL_COST a short floating-point value that indicates a rough relative estimate of resources required. Does not reflect an estimate of time required. When preparing a dynamically defined SQL statement, you can use this field as an indicator of the relative cost of the prepared SQL statement. For a specific statement, it may vary with changes to stats in the catalog, and is also subject to change between releases of DB2 for z/OS.
SQLERRD(5)	INTEGER	Position or column of a syntax error for a PREPARE or EXECUTE IMMEDIATE statement.
SQLERRD(6)	INTEGER	Internal error code.
SQLWARN0	CHAR(1)	Blank if all indicators are blank. W if at least one other indicator contains W or Z.
SQLWARN1	CHAR(1)	W if value of string column was truncated when assigned to a host variable. Contains an N for non-scrollable cursors and an S for scrollable cursors after the OPEN CURSOR or ALLOCATE CURSOR statement. If subsystem parameter DISABSCL is set to YES, the field will not be set to N for non-scrollable cursors.
SQLWARN2	CHAR(1)	W if null values were eliminated from argument of an aggregate function; not always set to W for MIN function because its results are not dependent on elimination of null values.
SQLWARN3	CHAR(1)	W if the number of result columns is larger than the number of host variables. Contains a Z if fewer locators were provided in the ASSOCIATE LOCATORS statement than the stored procedure returned.
SQLWARN4	CHAR(1)	W if a prepared UPDATE or DELETE statement does not include a WHERE clause. For a scrollable cursor, contains a D for sensitive dynamic cursors, an I for insensitive cursors, and an S for sensitive static cursors after the OPEN CURSOR or ALLOCATE CURSOR statement; blank if cursor is not scrollable. If subsystem parameter DISABSCL = YES, the field will not be set to N for non-scrollable cursors.
SQLWARN5	CHAR(1)	W if SQL statement was not executed because it is not valid SQL in DB2 for z/OS. Contains a character value of 1 (read-only), 2 (read and delete), or 4 (read, delete, and update) to reflect capability of the cursor after the OPEN CURSOR or ALLOCATE CURSOR statement. If subsystem parameter DISABSCL = YES, the field will not be set to N for non-scrollable cursors

Name	Data Type	Description
SQLWARN6	CHAR(1)	W if the addition of a month or year duration to a DATE or TIMESTAMP value results in an invalid day (ex. June 31). Indicates the day was changed to the last day of the month to make the date valid.
SQLWARN7	CHAR(1)	W if 1 or more non-zero digits were eliminated from fractional part of a number used as the operand of a decimal multiply or divide operation.
SQLWARN8	CHAR(1)	W if a character that could not be converted was replaced with a substitute character.
SQLWARN9	CHAR(1)	W if arithmetic exceptions were ignored during COUNT or COUNT_BIG processing Is Z if stored procedure returns multiple result sets.
SQLWARNA	CHAR(1)	W if at least one character field of the SQLCA or SQLDA names or labels is invalid due to a character conversion error. Blank if all fields valid.
SQLSTATE	CHAR(5)	Return code for the outcome of the most recent execution of an SQL statement.

SQLDA

SQLDA contains the following columns:

Name	Data Type/Description
SQLDAID	CHAR(8) Usage in DESCRIBE and PREPARE: An “eye catcher” for dumps, contains ’SQLDA’. The 7th byte of the field is a flag that can be used to determine if more than one SQLVAR entry is needed for each column. For DESCRIBE CURSOR, the field is set to ‘SQLRS’. If a stored procedure cursor is declared WITH HOLD, high order bit of the 8th byte will contain a 1. For DESCRIBE PROCEDURE, it is set to ‘SQLPR’. Usage in FETCH, INSERT, OPEN, EXECUTE, and CALL: A plus sign (+) in the 6th byte indicates that SQLNAME contains an overriding CCSID. A ‘2’ in the 7th byte indicates that two SQLVAR entries were allocated for each column or parameter. A ‘3’ in the 7th byte indicates that three SQLVAR entries were allocated for each column or parameter. Although three entries are never needed on input to DB2, an SQLDA with three entries might be used when the SQLDA was initialized by a DESCRIBE or PREPARE INTO with a USING BOTH clause. Otherwise, SQLDAID is not used.
SQLDABC	INTEGER Usage in DESCRIBE and PREPARE: Length of SQLDA, equal to SQLN*44 + 16 Usage in FETCH, EXECUTE, and CALL: Length of the SQLDA, greater than or equal to SQLN*44+16.
SQLN	SMALLINT Usage in DESCRIBE and PREPARE INTO: Unchanged by DB2. SQLN must be set to a value greater than or equal to zero prior to statement execution. Indicates the total number of occurrences of SQLVAR. • For DESCRIBE INPUT, the number of parameter markers to be described. • For other DESCRIBEs or PREPARE INTO: the number of columns of the result, or a multiple of the columns of the result when there are multiple sets of SQLVAR entries because column labels are returned in addition to column names. Usage in FETCH, INSERT, OPEN, EXECUTE, and CALL: Total number of occurrences of SQLVAR. SQLN must be set to a value greater than or equal to zero.
SQLD	CHAR(5) Usage in DESCRIBE and PREPARE INTO: number of columns described by occurrences of SQLVAR. Double that number if USING BOTH is in the DESCRIBE or PREPARE INTO statement. Contains a 0 if the statement string is not a query. For DESCRIBE PROCEDURE contains number of result sets returned by the stored procedure. Contains a 0 if no result sets are returned. Usage in FETCH, INSERT, OPEN, EXECUTE, and CALL: number of host variables described by occurrences of SQLVAR.

Contents of SQLVAR arrays:

					Set of SQLVAR entries		
LOBs	Distinct types	7th Byte of SQLDAID	SQLD	Minimum for SQLN	First set (base)	Second Set (Extended)	Third Set (Extended)
USING BOTH clause not specified:							
No	No	Space	*N*	*n*	Column names or labels	Not Used	Not Used
Yes	Yes	2	*N*	*2n*	Column names or labels	LOBs, distinct types, or both	Not Used
USING BOTH clause was specified:							
No	No	Space	*2n*	*2n*	Column names	Labels	Not Used
Yes	No	2	*N*	*2n*	Column names	LOBs and labels	Not Used
No	Yes	3	*N*	*3n*	Column names	Distinct types	Labels
Yes	Yes	3	*N*	*3n*	Column names	LOBs and distinct types	Labels

Fields in an occurrence of a base SQLVAR array:

Name	Data Type/Description
SQLTYPE	SMALLINT Usage in DESCRIBE and PREPARE INTO: Data type of column or parameter and whether or not it allows null values. For a distinct type, the data type on which the distinct type was based is placed in this field. The base SQLVAR provides no indication that this part of the description of a distinct type. For VARCHAR with CCSID=1208 in EBCDIC table, the SQLTYPE reflects VARCHAR despite the column is VARBINARY. For VARGRAPHIC with CCSID=1200 in EBCDIC table, SQLTYPE reflects VARGRAPHIC despite the column is VARBINARY. Usage in FETCH, OPEN, EXECUTE, and CALL: Data type of host variable and whether an indicator variable is provided. Host variables for datetime values must be character string variables. For FETCH a datetime type code means fixed-length character string.
SQLLEN	SMALLINT Usage in DESCRIBE and PREPARE INTO: Defines length of column or parameter. For date/time columns, the length of the string representation of the value. Usage in FETCH, OPEN, and EXECUTE, CALL: Defines external length of the host variable. For LOBs, the value is 0 regardless of the length attributes of the LOB. Field SQLLONGLEN in the extended SQLVAR contains the length attribute. For XML AS BLOB, CLOB, or DBCLOB, sqllen is 0 for LOB types. For FETCH WITH CONTINUE, the eight byte of SQLLEN is used as a column-level indicator to show whether CONTINUE behavior should be enabled for this result column, if the column is of type BLOB, CLOB, DBCLOB, or XML. If the eight byte is X'01', DB2 will permit FETCH CURRENT CONTINUE operations to be performed against this result column.
SQLDATA	Pointer Usage in DESCRIBE and PREPARE INTO: Contains X'0000ZZZZ' for string columns or parameters, where ZZZZ is the associated CCSID. For bit data, ZZZZ is X'FFFF'. For date/time columns, contains the CCSID of the string representation of the datetime value. For DESCRIBE PROCEDURE contains result set locator value. For VARCHAR with CCSID=1208 in EBCDIC table, the SQLTYPE reflects VARCHAR despite the column is VARBINARY. For VARGRAPHIC with CCSID=1200 in EBCDIC table, SQLTYPE reflects VARGRAPHIC despite the column is VARBINARY. Usage in FETCH, OPEN, CALL, and EXECUTE: Address

Name	Data Type/Description
	of host variable.
SQLIND	Pointer Usage in DESCRIBE and PREPARE INTO: Reserved. Set to -1 for DESCRIBE PROCEDURE. Usage in FETCH, OPEN, CALL, and EXECUTE: Address of an associated indicator variable if SQLTYPE is odd.
SQLNAME	VARCHAR(30) Usage in DESCRIBE and PREPARE INTO: Name or label of column, or a string of length 0 if the name or label does not exist. For DESCRIBE INPUT, SQLNAME is not used. For DESCRIBE PROCEDURE contains the names of cursor(s) used by the stored procedure to return the result set. The values for SQLNAME appear in the order the cursors were opened by the stored procedure. Usage in FETCH, OPEN, CALL, and EXECUTE: Contains CCSID in the third and fourth byte. This applies if the 6th byte of SQLDAID = "+"; SQLTYPE indicates host variable is a string variable; the length of SQLNAME is 8; and the first two bytes are X'0000'. Usage in FETCH, OPEN, INSERT, and EXECUTE: The fifth through eighth bytes contain a binary integer that represents the dimension of the host-variable-array and the corresponding indicator-array if one is specified, if all the following are true: • The length of SQLNAME is 8 • The first two bytes of the data portion of SQLNAME are X'0000'. • 5th and 6th byte is a flag indicating the host variable type: x'0000' host variable x'0100' XML host variable x'0001' host variable array x'0101' XML host variable array x'0002' special host variable represents value for 'n' in multi-row insert. • 7th and 8th byte: If 6th byte is x'01', a binary small integer representing the dimension of host variable array and the corresponding indicator array if specified.
SQLLONGL SQLLONGLEN	INTEGER Usage in DESCRIBE and PREPARE INTO: length attributes of BLOB, CLOB, or DBCLOB. Usage in FETCH, OPEN, EXECUTE, and CALL: length attribute of BLOB, CLOB, or DBCLOB host variable. The number of bytes for BLOB or CLOB, the number of characters for DBCLOB.
	INTEGER Reserved
SQLDATAL SQLDATALEN	Pointer Usage in DESCRIBE and PREPARE: Not Used. Usage in FETCH, OPEN, EXECUTE, and CALL: Used only for LOB (BLOB, CLOB, and DBCLOB) host variables. If the value of the field is null, the actual length of the LOB is stored in the 4 bytes immediately before the start of the data, and SQLDATA points to the first byte of the field length. The actual length indicates the number of bytes for a BLOB or a CLOB, and the number of characters for a DBCLOB. If the value of the field is not null, the field contains a pointer to a 4-byte long buffer that contains the actual length *in bytes* (even for DBCLOBS) of the data in the buffer pointed to from the SQLDATA field in the matching base SQLVAR. Regardless of whether this field is used, field SQLLONGLEN must be set.

Name	Data Type/Description
SQLTNAME	VARCHAR(30)
SQLDATATYPE_ NAME	Usage in DESCRIBE and PREPARE INTO: For appropriate extended, SQLVAR is set accordingly. For distinct data type, value is fully qualified distinct data type name (truncated to 30 bytes). For a label, it contains the contents of the label. USING BOTH stores the distinct name in the first extended SQLVAR and the label in the second extended SQLVAR. Usage in FETCH, OPEN, EXECUTE, and CALL, not used.

SQL Limits

The following table shows the identifier items and SQL limit:

Identifier Items	Limit
External java routine name	1305 bytes
Name of an alias, auxiliary table, collection, constraint, clone table, correlation, cursor (except for DECLARE CURSOR WITH RETURN or EXEC SQL utility), distinct type (both parts of a two-part name), function (both parts of a two-part name), host identifier, index, JARs, package (created as a result of a CREATE TRIGGER statement), parameter, procedure, schema, sequence, specific statement, storage group, savepoint, SQL condition, SQL label, SQL parameter, SQL variable, synonym, table, table trigger, view, XML attribute-name, XML element-name.	128 bytes
Name of authorization ID or security label	8 bytes
Routine version identifier	64 bytes
Name of column	30 bytes
Name of cursor (created with DECLARE CURSOR WITH RETURN)	8 bytes
Name of location	16 bytes
Name of buffer pool, catalog, cursor (created with EXEC SQL utility), database, package (created with BIND PACKAGE command), plan, program, table space	8 bytes
Name of package when used with CREATE FUNCTION, CREATE PROCEDURE, CREATE TRIGGER, or BIND specifying Xfs file as DBRMLIB	128 bytes
Name of profile created with CREATE TRUSTED CONTEXT or ALTER TRUSTED CONTEXT	127 bytes

The following table shows the numeric values and SQL limit:

Numeric Values		Limit
INTEGER Largest/Smallest		2147483647 -2147483648
SMALLINT Largest/Smallest		32767 -32768
BIGINT Largest/Smallest		9223372036854775808 -9223372036854775807
DECIMAL precision		31
DECIMAL Largest/Smallest		$10^{31} - 1$ $1 - 10^{31}$
FLOAT (approximate)		
	Largest	7.2×10^{75}
	Smallest	-7.2×10^{75}
	Smallest positive	5.4×10^{-79}
	Largest negative	-5.4×10^{-79}
REAL		
	Largest	7.2×10^{75}
	Smallest	-7.2×10^{75}
	Smallest positive	5.4×10^{79}
	Largest negative	-5.4×10^{79}

The following table shows the DECFLOAT numeric values and SQL limits:

Numeric Values DECFLOAT	Limit
Smallest DECFLOAT(16)	$-9.999999999999999 \times 10^{384}$
Largest DECFLOAT(16)	$9.9999999999999 \times 10^{384}$
Smallest positive DECFLOAT(16)	$1.000000000000000 \times 10^{-383}$
Largest negative DECFLOAT(16)	$-1.000000000000000 \times 10^{-383}$
Smallest DECFLOAT(34)	$-9.999999999999999999999999999999999 \times 10^{6144}$
Largest DECFLOAT(34)	

Numeric Values DECFLOAT	Limit
	$9.999999999999999999999999999999999 \times 10^{6144}$
Smallest positive DECFLOAT(34)	
	$1.000000000000000000000000000000000 \times 10^{-6143}$
Largest negative DECFLOAT(34)	
	$-1.000000000000000000000000000000000 \times 10^{-6143}$

The following table shows the string lengths and limits:

String Lengths		Limit
CHAR		255 bytes
GRAPHIC		127 double-byte characters
VARCHAR	For 4K pages:	4046 bytes
	For 8K pages:	8128 bytes
	For 16K pages:	16320 bytes
	For 32K pages:	32704 bytes
VARGRAPHIC	For 4K pages:	2023 doube-bytes
	For 8K pages:	4064 doube-bytes
	For 16K pages:	8160 double-bytes
	For 32K pages:	16352 double-bytes
BINARY		255 bytes
VARBINARY		32704 bytes
CLOB		2 147 483 647 (2GB-1byte)
DBCLOB		1 073 741 823 DBCS chars
BLOB		2 147 483 647 (2GB-1byte)
Character constant		32704 UTF-8 bytes
Hexadecimal constant		32704 hex digits
Graphic constant		16352 double-byte chars (32704 UTF-8 bytes)
Hexadecimal graphic constant		32704 hex digits
Concatenated character string		2 147 483 647 (2GB-1byte)
Concatenated graphic string		1 073 741 823 DBCS chars
Concatenated binary string		2 147 483 647 (2GB-1byte)
SCALAR EXPRESSION text string		4000 UTF-8
XML pattern text		4000 UTF-8
XML element or attribute name		1000 bytes
Length of namespace uri		1000 bytes
Length of namespace prefix		998 bytes
Largest depth of an internal XML tree		128 levels

The following table shows the date/time items and limits:

Date/Time Items		Limit
DATE value (shown in ISO format)		
	Largest	9999-12-31
	Smallest	0001-01-01
TIME value (shown in ISO format)		
	Largest	24.00.00
	Smallest	00.00.00
TIMESTAMP value	Largest	9999-12-31-24.00.00.000000000000
	Smallest	0001-01-01-00.00.00.000000000000
TIMESTAMP precision range		
	Largest	12
	Smallest	0
TIME ZONE hour range		
	Largest	24-24
	Smallest	
TIME ZONE minute range		
	Largest	59
	Smallest	0

Db2 Limits

The following table shows the Db2 SQL statements and their limits:

SQL Statements		Limit	
Columns in a table, view (dependent on complexity of CREATE VIEW statement), or table function		750 or fewer 749 if the table is a dependent	
Base table names in a view, SELECT, INSERT, UPDATE, MERGE, or DELETE		1024	
Maximum record size in bytes:		EDITPROC =	
Page Size		YES	NO
4KB	Non-hash	4046	4056
4KB	Hash(hash home page)	3807	3817
8KB	Non-hash	8128	8138
8KB	Hash(hash home page)	7889	7899
16KB	Non-hash	16320	16330
16KB	Hash(hash home page)	16081	16091
32KB	Non-hash	32704	32714
32KB	Hash(hash home page)	32465	32475
Number of rows that can be inserted with a single INSERT or MERGE		32767	
Volume IDs in a storage group		133	
Partitions in a partitioned table space or partitioned index:			
	Not (LARGE or DSSIZE > 2GB)	64	
	LARGE or DSSIZE > 2GB	4096	
Maximum sum of limit key values lengths of a partition boundary		765 UTF-8 bytes	
Size of partitions:			
Not (LARGE or DSSIZE > 2GB)			
	1 – 16 partitions	4 gigabytes	
	17 – 32 partitions	2 gigabytes	
	33 – 64 partitions	1 gigabytes	
(LARGE)			
	1 – 4096 partitions	4 gigabytes	
(DSSIZE > 4GB)			
	4K pages: 1 – 64 partitions	256 gigabytes	
	8K pages: 1 – 128 partitions	256 gigabytes	
	16K pages: 1 – 256 partitions	256 gigabytes	
	32K pages: 1 – 512 partitions	256 gigabytes	
For range partitioned table spaces with relative numbering.		1 terabyte	
Maximum size of a non-partitioned index for a partitioned table space			
• For 5-byte EA table spaces:		16 TB for 4 KB pages 32 TB for 8 KB pages 64 TB for 16 KB pages 128 TB for 32 KB pages	
• For table spaces that are defined with LARGE:		16 TB	
Table or table space size		128 terabytes	
Size of a DBRM		131072 bytes	
Maximum length of index key:			
• Partitioning index		255 – *n*	
• Non-partitioning index (padded)		2000 – *n*	
• Non-partitioning index (not padded)		2000 – *n* – 2*m*	
Where *n* is the number of key columns that allow nulls and *m* is the number of varying length columns in the key.			
Maximum number of bytes used in the partitioning of a partitioned index		255 or less (the number of partitions * (106 + limit key size) must be less than 65394),	
Columns or EXPRESSIONS in an index key		64	

SQL Statements	Limit
Tables in a FROM clause	255 or less, depending on the complexity of the statement
Number of subqueries in a statement	224
Host and indicator variable length pointed to in a SQLDA: • Non-LOB • LOB	 32767 bytes 2 147 483 647 (2GB – 1byte)
Maximum size of application SQLDA for any statement that references host variables or parameter markers	99016 bytes
Host variable length used for INSERT or UPDATE: (subject to limitations imposed by the application environment and host language) • Non-LOB • LOB	 32704 bytes 2 147 483 647 (2GB – 1byte)
Maximum number of host variables or parameter markers in a statement	16,000
SQL statement length	2 097 152 bytes
Elements in a SELECT list (depending on whether the list is for the result table of static scrollable cursor)	750 or less
Predicates in a WHERE or HAVING clause	Limited by storage
Total column length for a query operation requiring a sort key (ORDER BY, SELECT DISTINCT, UNION, EXCEPT or INTERSECT clause without the ALL keyword and the DISTINCT keyword for aggregate functions)	4032 bytes
Total column length for a query operation requiring sort and evaluating column functions (MULTIPLE DISTINCT and GROUP BY)	65529 bytes
Maximum length of a sort key	32707 bytes
Total column length for sort	32707
Maximum length of a table check constraint	3800 bytes
Number of bytes passed in a single parameter of an SQL statement: (subject to limitations imposed by the application environment and host language) • Non-LOB • LOB	 32765 bytes 2 147 483 647 (2GB – 1byte)
Number of stored procedures, triggers, and user-defined functions that an SQL statement can implicitly or explicitly reference.	64 nesting levels
Length of an SQL path	2048 bytes
Maximum length of WLM name in CREATE PROCEDURE, CREATE FUNCTION, ALTER PROCEDURE, or ALTER FUNCTION.	32 bytes
Maximum number of Xpath levels in the XMLPATTERN clause of the CREATE INDEX statement	50 nesting levels
Concurrent DB2 or application agents.	Limited by EDM & buffer pool size and amount of storage used by each DB2 or application agent
Concurrently active audit policies	32
Largest non-LOB table or table space	128 terabytes (TB)
Largest simple or segmented table space	64 GB
Largest lob space: 6-byte basic 10-byte extended	2^{48} bytes 2^{80} bytes
Largest active log data set	768GB -1
Largest archive log data set	768GB -1
Maximum number of active log copies	2

SQL Statements	Limit
Maximum number of archive log copies	2
Maximum number of active log data sets (each copy)	93
Maximum number of archive log volumes (each copy)	10000
Databases accessible to an application or end user	System storage / EDM pool size
EDM pool size	The installation parameter maximum depends on available space
Total databases	65217
Total implicitly created databases	Maximum value of the sequence SYSIBM.DSNSEQ_IMPLICITDB with a default of 10000
Maximum number of internal objects for each database	32767
Total indexes on declared global temporary tables	10000
Number of rows per page: • All user table spaces • Catalog and directory table spaces	 255 127
Maximum simple or segmented data set size	2 gigabytes
Maximum LOB data set size	64 gigabytes
Table spaces defined in a work file database	500
Tables and triggers defined in a work file database	11767

RLST Table

Legend:

authid.*tablename*		**(DSNRLST.*tablespacename*)**
authid.*indexname*	**Type**	**(index columns)**
table description		

DSNRLST*xx*		**(DSNRLST.DSNRLSxx)**
DSNARL*xx*	**U**	**(RLFFUNC,AUTHID,PLANNAME, RLFCOLLN,RLFPKG,LUNAME)**
Allows the user to specify the maximum amount of processor time for dynamic DML SQL statements from a local or remote DBMS, governs the execution of packages and plans, and allows the user to restrict all BINDs and REBINDs. The name of the RLST table consists of an authorization ID that is specified during the installation of DB2 and a prefix where xx is any 2-character alphanumeric value.		

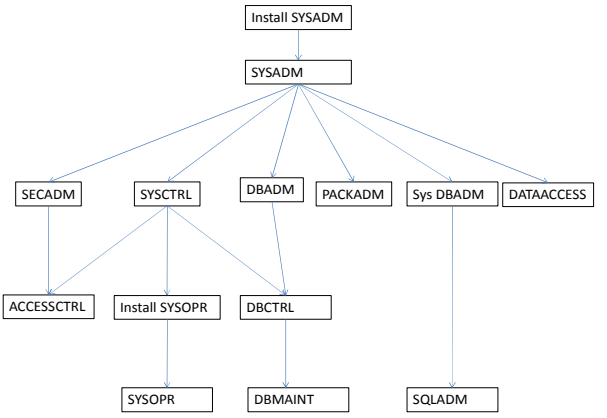
This table contains the following columns:

Name	Data Type/Description
AUTHID	CHAR(128) Primary authid for which the control info is to apply. Blank: limit applies to all authids in location specified in LUNAME. If LUNAME blank or column is omitted, location is the local location. This column is required.
PLANNAME	CHAR(8) The resource specification limits apply to this plan. If you are specifying a function that applies to plans (RLFFUNC=blank or '6'), a blank means that the limit specifications in this row apply to all plans for the location that is specified in the LUNAME. Qualify by plan name only if the dynamic statement is issued from a DBRM bound in a plan, not a package; otherwise, DB2 does not find this row. If the RLFFUNC column contains a function for packages '1,' '2,' or '7,' this column must be blank; if it is not blank, the row is ignored.
ASUTIME	INTEGER The dynamic SQL service unit time or the number of CPU service units permitted to any single dynamic DML statement used for reactive governing. A Null value indicates no limit. 0 or a negative value indicates dynamic DML statements are not allowed. Required column.
LUNAME	CHAR(8) LUNAME of the location where the request originated. Column used to specify limits for requests from remote locations. Blank for local location, *not* all locations. PUBLIC for all DBMS locations in the network; these locations do not have to be DB2 subsystems. PUBLIC only value for TCP/IP conventions.
RLFFUNC	CHAR(1) Governing function this row will serve: 1 Row will be used to govern whether bind operations are allowed 2 Row used to reactively govern dynamic DML statements by package or collection name 3 Row disables query I/O parallelism 4 Row disables query CP parallelism 5 Row disables Sysplex query parallelism 6 Row predictively governs dynamic DML statements by plan name

Name	Data Type/Description
	7 Row predictively governs dynamic DML statements by package or collection name All other values are ignored. Optional column.
RLFBIND	CHAR(1) Whether bind operations are allowed. N = bind operation is not allowed; any other value indicates bind operation is allowed. Used only if RLFFUNC = 1. This column is required if column RLFFUNC exists.
RLFCOLLN	CHAR(128) Collection name for control info. Blank means the limit applies to all collections from location specified in LUNAME. If LUNAME blank or the column is omitted, the location is the local location. If RLFFUNC = blank, 1 or 6 then RLFCOLLN must be blank.
RLFPKG	VARCHAR(128) Name of the package for control info. Blank means limit applies to all packages from location specified in LUNAME. If LUNAME is blank or column is omitted, location is the local location. If RLFFUNC = blank, 1, or 6, RLFPKG must be blank.
RLFASUERR	INTEGER Used for predictive governing (RLFFUNC=6 or 7) and only statements in cost category A. Indicates the error threshold number of system resource manager processor service units allowed for a single dynamic DML statement. If predicted processor cost is greater than threshold an SQLCODE of – 495 is returned. If set to NULL indicates no threshold. Zero or a negative value indicates all dynamic DML receive an SQLCODE of – 495.
RLFASUWARN	INTEGER Used for predictive governing (RLFFUNC=6 or 7) and only statements in cost category A. Indicates warning threshold number of system resource manager processor service units allowed for a single dynamic DML statement. If predictive processor cost is greater than warning threshold, an SQLCODE +495 is returned. If set to NULL indicates no threshold. Zero or negative value indicates all dynamic DML receive a SQLCODE +495.
RLF_CATEGORY_B	CHAR(1) Used for predictive governing (RLFUNC=6 or 7). Determines action to take when cost estimate for a statement falls in cost category B, indicating cost is indeterminate and probably too low. An EXPLAIN can be used to detect cost category B by checking the COST_CATEGORY column of the DSN_STATEMNT_TABLE. Acceptable values: • Blank—Prepare and execute statement by default. • Y—Prepare and execute statement. • N—Do not prepare or execute statement and return – 495 SQLCODE. • W—Prepare statement and return +495 SQLCODE. Application logic can control execution.

Security Authorizations

Administrative Authorities Hierarchy



Administrative Authorities and Grantable Privileges

The following table lists the administrative authorities and grantable privileges:

Authority	Included authorities (plus inherited authorities)	Additional grantable authorities
Installation SYSADM (assigned to one or two individuals when DB2 is installed)	SYSADM	Security: GRANT, REVOKE
ACCESS CTRL	None	Catalog tables: SELECT, DELETE[1], INSERT[1], UPDATE[1] [1] Except SYSIBM.SYSAUDITPOLICIES Security: GRANT, REVOKE
DATA ACCESS	None	System: DEBUGSESSION User tables, views, MQTs: DELETE, INSERT, SELECT, UPDATE Plans, packages, routines: EXECUTE: user databases: UNLOAD, LOAD, RECOVERDB, REORG, REPAIR JARs, sequences, distinct types: USAGE: catalog tables: SELECT, DELETE[1], INSERT[1], UPDATE[1] [1] Except SYSIBM.SYSAUDITPOLICIES
DBADMN	DBCTRL, DBMAINT	Tables in a database: ALTER, DELETE, INDEX, INSERT, REFERENCES, SELECT, TRIGGER, UPDATE Can be granted WITH or WITHOUT ACCESSCTRL or DATAACCESS
DBCTRL	DBMAINT	Database: DROP, UNLOAD, LOAD, RECOVERDB, REORG, REPAIR
DBMAINT	None	Database: CREATETAB, CREATETS, DISPLAYDB, IMAGCOPY, STATS, STARTDB, STOPDB
Installation SYSADM	SYSADM, SYSCTRL, DBADM, Installation SYSOPR, SYSOPR,	Security: GRANT, REVOKE

Authority	Included authorities (plus inherited authorities)	Additional grantable authorities
	PACKADM, DBCTRL, DBMAINT, SECADM, System DBADM, SQLADM, ACCESSCTRL, DATAACCESS	
Install SYSOPR	SYSOPR	Privileges: ARCHIVE, BINDAGENT, STARTDB (not access mode) System-defined packages and routines: EXECUTE Catalog tables: SELECT Updatable catalog tables: DELETE, INSERT, UPDATE
PACKADM	None	Collection: CREATEIN packages in a collection: BIND, COPY, EXECUTE
SECADM	ACCESSCTRL	Catalog tables: SELECT Updatable catalog tables: DELETE, INSERT, UPDATE Security: GRANT, REVOKE Security-related objects: ALTER, CREATE, DROP, TRANSFER OWNERSHIP
SQLADM	None	System: EXPLAIN, MONITOR1, MONITOR2 Catalog tables: SELECT, DELETE, INSERT, UPDATE Except SYSIBM.SYSAUDITPOLICIES
SYSADM	SYSCTRL, DBADM, Installation SYSOPR, SYSOPR, PACKADM, DBCTRL, DBMAINT, SECADM, System DBADM, SQLADM, ACCESSCTRL, DATAACCESS	Plans and routines: EXECUTE Packages: all Distinct types and sequences: USAGE System: DEBUGSESSION, EXPLAIN
SYSCTRL	Install SYSOPR, DBCTRL, (ACCESSCTRL without grant)	System: BINDADD, BINDAGENT, DBDS, CREATEALIAS, CREATEDBA, CREATEDDBC, CREATEDSG, CREATETMTAB, MONITOR[1], MONITOR[2], STOSPACE Tables: DELETE, INSERT, SELECT, UPDATE Plans: BIND Packages: BIND, COPY Collections: CREATEIN Schemas: ALTERIN, CREATEIN, DROPIN Use: BUFFERPOOLS, STOGROUP, TABLESPACE [1] Except SYSIBM.SYSAUDITPOLICIES.
SYSOPR	None	Privileges: DISPLAY, RECOVER, STOPALL, TRACE Routines: DISPLAY, START, STOP

Authority	Included authorities (plus inherited authorities)	Additional grantable authorities
System DBADM	SQLADM	System: BINDADD, BINDAGENT, CREATEALIAS, CREATEDBA, CREATEDBC, CREATETMTAB, DISPLAY, EXPLAIN, MONITOR[1], MONITOR[1], SQLADM, STOPALL, TRACE; Collections: CREATEIN; User databases: CREATETAB, CREATETS, DISPLAYDB, DROP, IMAGCOPY, RECOVERDB, STARTDB, STOPDB; User tables (w/o row permissions or column mask): ALTER, INDEX, REFERENCES, TRIGGER; Packages: BIND, COPY; Plans: BIND, Schemas: ALTERIN:CREATEIN, DROPIN; Sequences: ALTER; Distinct types: USAGE; Use: TABLESPACE; Catalog tables: SELECT, DELETE[1], INSERT[1], UPDATE[1] [1] Except SYSIBM.SYSAUDITPOLICIES.

Table and View Privileges

The following table lists the table and view privileges:

Privilege	Allows these SQL statements on a table or view:
ALTER	ALTER TABLE
DELETE	DELETE to delete rows
INDEX	CREATE INDEX
INSERT	INSERT to insert rows
REFERENCES	ALTER or CREATE TABLE, to add or remove a referential constraint referring to the named table or a list of columns in the table
SELECT	SELECT
TRIGGER	CREATE TRIGGER on a table
UPDATE	UPDATE any column or specific list of columns[1]
UNLOAD	UNLOAD
TRANSFER OWNERSHIP	TRANSFERs the OWNER of the table
GRANT ALL	Grant all table privileges listed above.

Plan Privileges

The following table lists the plan privileges:

Privilege	Allows these subcommands on a plan:
BIND	BIND, REBIND, and FREE PLAN to bind or free the plan
EXECUTE	RUN, to Execute a named application plan

Package Privileges

The following table lists the package privileges:

Privilege	Allows these functions on a package:
BIND	Use the BIND/REBIND and FREE PACKAGE subcommands and DROP PACKAGE statement
COPY	Create a copy of a package
EXECUTE	Include a named package/collection in PKLIST
GRANT ALL	Grant all package privileges listed above

Collection Privileges

The following table lists the collection privileges:

Privilege	Allows these functions on a collection:
CREATE IN/ON	BIND PACKAGE, which creates a package inside a collection

Database Privileges

The following table lists the database privileges:

Privilege	Allows these functions on a database:
CREATETAB	Create tables in the database
CREATETS	Create table spaces in the database
DISPLAYDB	Issue the DISPLAY DATABASE command
DROP	Drop or alter the database
IMAGCOPY	Run the QUIESCE, COPY, MERGECOPY, MODIFY, RECOVERY, and QUIESCE utilities for copies of table spaces in the database
UNLOAD	Use UNLOAD utility for tables in the database
LOAD	Use LOAD utility for tables in the database
RECOVER-DB	Use RECOVER, REBUILD INDEX, and REPORT for objects in the database
REORG	Use REORG utility for objects in database
REPAIR	Use REPAIR and DIAGNOSE utilities for objects (except REPAIR DBD and DIAGNOSE WAIT)
STARTDB	Issue the START DATABASE command
STATS	Use RUNSTATS, CHECK, LOAD, REBUILD INDEX, REORG INDEX, REORG TABLESPACE, and MODIFY STATISTICS utilities for objects
STOPDB	Issues the STOP DATABASE command

Use Privileges

The following table lists the use privileges:

Privilege	Allows use of these objects:
USE OF BUFFERPOOL	One or more buffer pools
USE OF ALL BUFFERPOOLS	Create objects referencing all buffer pools
USE OF STOGROUP	One or more storage groups
USE OF TABLESPACE	One or more table spaces

System Privileges

The following table lists the system privileges:

Privilege	Allows these functions:
ARCHIVE	Issue the ARCHIVE LOG, SET LOG, and SET ARCHIVE command
BINDADD	Use BIND subcommand with ADD option
BINDAGENT	BIND, REBIND, FREE, or DROP package, or COPY a package on behalf of the grantor
BSDS	Issue the RECOVER BSDS command
CREATE-ALIAS	Create alias for table or view
CREATE-DBA	Create database and have DBADM authority
CREATE-DBC	Create database and have DBCTRL authority
CREATESG	Create a storage group
CREATE_SECURE_ OBJECT	Create secure objects (like secure trigger or secure user defined function)
CREATETMTAB	Create a GLOBAL temporary table
DEBUGSESSION	DEBUGINFO connection attribute for native SQL and JAVA stored procedures
DISPLAY	DISPLAY ARCHIVE, DISPLAY BUFFERPOOL, DISPLAY DATABASE, DISPLAY LOCATION, DISPLAY LOG, DISPLAY THREAD, DISPLAY TRACE commands
EXPLAIN	Issue SQL EXPLAIN PLAN, EXPLAIN ALL, SQL PREPARE, DESCRIBE TABLE, use EXPLAIN(ONLY)

Privilege	Allows these functions:
	and SQLERROR(CHECK) options in BIND, and allow dynamic SQL statements when CURRENT EXPLAIN MODE is set to EXPLAIN
MONITOR1	Receive trace data, except sensitive data
MONITOR2	Receive all trace data
RECOVER	RECOVER INDOUBT to recover threads
STOPALL	STOP DB2
STOSPACE	Uses the STOSPACE utility
TRACE	START TRACE, STOP TRACE, and MODIFY TRACE commands

Schema Privileges

The following table lists the schema privileges:

Privilege	Allows use of these operations in designated schemas:
CREATEIN	CREATE distinct types, user-defined functions, triggers and stored procedures
ALTERIN	ALTER user-defined functions or stored procedures, or COMMENT ON distinct types, user-defined functions, triggers, and stored procedures
DROPIN	DROP distinct types, user-defined functions, triggers, and stored procedures

Sequence Privileges

The following table lists the sequence privileges:

Privilege	Allows use of these operations on sequence objects:
ALTER	Alter sequence object and modify comment
USAGE	Allows retrieval of next and previous value

Distinct Type Usage Privileges

The following table lists the distinct type privileges:

Privilege	Allows use of these objects:
USAGE ON DISTINCT TYPE	A distinct type
USAGE ON SEQUENCE	A sequence
USAGE ON JAR (JAVA class) for a routine	A JAVA class
ALL ON VARIABLE	Read and write of global variable
READ ON VARIABLE	Access global variable content
WRITE ON VARIABLE	Modify content of global variable

Routine Privileges

The following table lists the routine privileges:

Privilege	Allows use of these objects:
EXECUTE ON FUNCTION	A user-defined function or specific function
EXECUTE ON PROCEDURE	A stored procedure

TRANSFER OWNERSHIP Privileges

The following table lists the TRANSFER OWNERSHIP privileges:

Privilege	Allows these functions on a table space:
TRANSFER OWNERSHIP	Transfers ownership of DATABASE, INDEX, STOGROUP, TABLE, TABLESPACE, VIEW to ROLE, USER or SESSION_USER

Tablespace Privileges

The following table lists the tablespace privileges:

Privilege	Allows these functions on a table space:
TRANSFER OWNERSHIP	Transfers the OWNER of the table space

Index Privileges

The following table lists the index privileges:

Privilege	Allows these functions on a table space:
TRANSFER OWNERSHIP	Transfers the OWNER of the index.

Storage Group Privileges

The following table lists the storage group privileges:

Privilege	Allows these functions on a table space:
TRANSFER OWNERSHIP	Transfers the OWNER of the storage group

DSNZPARM Parameter Information

The following table lists DSNZPARM parameter information by macro parameter, macro name, panel, acceptable values and description.

Macro Parameter	Macro Name	Panel	Acceptable Values Description
ABEXP	DSN6SPRM	DSNTIPO	YES, NO Invoke explain processing during auto bind
ABIND	DSN6SPRM	DSNTIPO	YES, NO, COEXIST Allow automatic bind for plans and packages
ACCEL	DSN6SPRM	DSNTIP82	NO, COMMAND, AUTO Accelerator server startup option
ACCELMODEL	DSN6SPRM	Not on any installation panel	NO, YES Modeling of accelerator query workloads
ACCESS_CNTL_ MODULE	DSN6SYSP	DSNTIPO3	DSNX@XAC for migration, 1-8 characters for installation Access control module name
ACCUMACC	DSN6SYSP	DSNTIPN	NO, 2-65535: 10 Rollup accounting for DDF and RRSAF
ACCUMUID	DSN6SYSP	DSNTIPN	0-17: 0 Rollup accounting aggregation fields
ADMTPROC	DSN6SPRM	DSNTIPX	1-8 characters: *ssn*ADMT Admin Scheduler JCL procedure name
AEXITLIM	DSN6SPRM	DSNTIPP	0-32767: 10 Authorization exit ABEND limit
AGCCSID	DSNHDECP	DSNTIPF	1-65534: none Defaut CCSID for ASCII (graphic)
ALTERNATE_CP	DSN6SPRM	DSNTIP62	1 - 14 characters: blank Alternate SMS copy pool
AMCCSID	DSNHDECP	DSNTIPF	1-65533: none Defaut CCSID for ASCII (mixed)
APPENSCH	DSNHDECP	DSNTIPF	ASCII, EBCDIC, UNICODE, or ccsid (between 1-65533) Default application encoding
APPLCOMPAT	DSN6SPRM	DSNTIP41	*function-level*, V11R1, V10R1: V12R1M500 for new installation APPLCOMPAT bind option default (application compatibility level)
ARCPFX1	DSN6ARVP	DSNTIPH	1-35 char valid dataset name prefix: DSNCAT.ARCHLOG1 or DSNCAT.DSN1.ARCLG1 Archive log COPY1 dataset prefix
ARCPFX2	DSN6ARVP	DSNTIPH	1-35 char valid dataset name prefix: DSNCAT.ARCHLOG2 or DSNCAT.DSN1.ARCLG2 Archive log COPY2 dataset prefix
ARCRETN	DSN6ARVP	DSNTIPA	0-9999: 9999 Archive log retention period (days)
ARCWRTC	DSN6ARVP	DSNTIPA	1 to 16 route codes: 1,3,4 Archive log WTOR route codes
ARCWTOR	DSN6ARVP	DSNTIPA	NO, YES Archive log WTOR option
ARC2FRST	DSN6LOGP	DSNTIPO	NO, YES Use COPY2 archive logs first
ASCCSID	DSNHDECP	DSNTIPF	0-65533: none Defaut CCSID for ASCII (single-byte)
AUDITST	DSN6SYSP	DSNTIPN	NO, YES (class 1), list of audit classes (1-11), or *(all classes) Audit traces when DB2 started
AUTH	DSN6SPRM	DSNTIPP	YES, NO Enable DB2 authorization checking
AUTHCACH	DSN6SPRM	DSNTIPP	0-4096 (in multiples of 256): 3072 Default plan authorization cache size
AUTHEXIT_CACHE REFRESH	DSN6SPRM	DSNTIPP	ALL, NONE Refresh authorization cache entries after RACF access change
AUTH_ COMPATIBILITY	DSN6SPRM	DSNTIPP	Null (blank), SELECT_FOR_UNLOAD Authorization compatibility

Macro Parameter	Macro Name	Panel	Acceptable Values Description
AUTHEXIT_ CHECK	DSN6SPRM	DSNTIPP	DB2, PRIMARY Authorization ID for checking with access control authorization exit
BACKODUR	DSN6SYSP	DSNTIP11	0-255: 5 Restart backout limit
BIF_ COMPATIBILITY	DSN6SPRM	DSNTIPX	V9_DECIMAL_VARCHAR, V9_TRIM, CURRENT (V9_DECIMAL_VARCHAR for migration) CHAR built-in function compatibility
BINDNV	DSN6SPRM	DSNTIPP1	BINDADD, BIND Bind authorization required for new package bind
BLKSIZE	DSN6ARVP	DSNTIPA	8192-28672: 24576 Blksize to be used for archive log datasets
BMPTOUT	DSN6SPRM	DSNTIPI	1-254: 4 Timeout multiplier for IMS BMP connections
CACHEDYN	DSN6SPRM	DSNTIP8	YES, NO Dynamic SQL statement cache support option
CACHEDYN_ STABILIZATION	DSN6SPRM	DSNTIP8	BOTH, CAPTURE, LOAD, NONE Stabilize cached dynamic SQL statements
CACHEPAC	DSN6SPRM	DSNTIPP	0-10M: 5M Max storage for package authorization cache
CACHERAC	DSN6SPRM	DSNTIPP	0-10M: 5M Max storage for routine authorization cache
CATALOG	DSN6SPRM	DSNTIPA2	1-8 chars: DSNCAT DB2 VSAM catalog high-level qualifier
CATDDACL & CATDMGCL & CATDSTCL	DSN6SPRM	DSNTIPA2	Blank or any valid SMS dataclass name Directory and catalog SMS data classes
CATXDACL & CATXMGCL & CATXSTCL	DSN6SPRM	DSNTIPA2	Blank or any valid SMS dataclass name Directory and catalog indexes SMS data classes
CDSSRDEF	DSN6SPRM	DSNTIP81	1, ANY CURRENT DEGREE special register default value
CHECK_ FASTREPLICATION	DSN6SPRM	DSNTIP61	PREFERRED, REQUIRED, NONE DB2 CHECK utility fast replication
CHGDC	DSN6SPRM	DSNTIPO	1, 2, 3 Data Propagator propagate change option
CHKFREQ & CHKLOGR	DSN6SYSP	DSNTIPL1	If CHKTYPE='LOGRECS' → 1000-16000000, 'MINUTES' → NOTUSED, 'BOTH' → 1000-99999999 Checkpoint frequency
CHKFREQ & CHKMINS	DSN6SYSP	DSNTIPL1	If CHKTYPE='LOGRECS' → NOTUSED, 'MINUTES' → 1-60, 'BOTH' → 1-1439: 5 Checkpoint frequency
CHKTYPE	DSN6SYSP	DSNTIPL1	LOGRECS, MINUTES, BOTH Checkpoint type
CMTSTAT	DSN6FAC	DSNTIPR	ACTIVE, INACTIVE DDF threads status after commit/abort
COMPACT	DSN6ARVP	DSNTIPA	NO, YES Archive log compression option
COMPRESS_ DIRLOB	DSN6SPRM	DSNTIPA2	NO, YES Compress LOB table spaces in DB2 directory
COMPRESS_ SPT01	DSN6SPRM	DSNTIPA2	NO, YES SPT01 directory table spaces compression is enabled
CONDBAT	DSN6SYSP	DSNTIPE	0-150000: 10000 Maximum number of concurrent inbound DDF connection
COPY_ FASTREPLICATION	DSN6SPRM	Not on any installation panel	PREFERRED, REQUIRED, NONE DB2 COPY utility FlashCopy fast replication
CTHREAD	DSN6SYSP	DSNTIPE	1-20000: 200 Maximum number of allied threads
DBACRVW	DSN6SPRM	DSNTIPP1	NO, YES

Macro Parameter	Macro Name	Panel	Acceptable Values Description
			Allow DBADM to create view/alias/MQT for another auth ID
DATE	DSNHDECP	DSNTIP4	ISO, USA, EUR, JIS, LOCAL
			Date format
DATELEN	DSNHDECP	DSNTIP4	0, 10-254
			Local date length
DDLTOX	DSN6SPRM	DSNTIP1	1-254: 1
			DDL wait
DDF	DSN6FAC	DSNTIPR	NO, AUTO, COMMAND
			DDF startup option
DDF_ COMPATIBILITY	DSN6FAC	Not on any installation panel	Null, DISABLE_IMPCAST_JV, IDNTFY_V12_PRIOR_VER, IGNORE_TZ, SP_PARMS_NJV
			DDF compatibility characteristics
DDL_ MATERIALIZATION	DSN6SPRM	DSNTIP71	1 (ALWAYS_IMMEDIATE), 2 (ALWAYS_PENDING)
			Materialize changes of an object
DEALLCT	DSN6LOGP	DSNTIPA	Minutes, seconds, 1440, NOLIMIT: 0
			Archive read tape unit deallocation period
DECARTH	DSNHDECP	DSNTIP4	DEC15, DEC31, 15, 31, *DPP.S*
			Decimal arithmetic option
DECDIV3	DSN6SPRM	DSNTIP4	NO, YES
			Minimum divide scale
DECIMAL	DSNHDECP	DSNTIPF	.(period) or ,(comma)
			Decimal point is
DEF_DECFLOAT_ ROUND_MODE	DSNHDECP	DSNTIPF	ROUND_: CEILING, DOWN, FLOOR, HALF_DOWN, HALF_EVEN, HALF_UP, UP ROUND_HALF_EVEN
			DECFLOAT Rounding mode
DEFAULT_INSERT _ALGORITHM	DSN6SPRM	DSNTIP71	1, 2: 2
			Default algorithm for inserting data into table spaces
DEFLANG	DSNHDECP	DSNTIPF	ASM, C, CPP, IBMCOB, FORTRAN, PL1
			Default programming language
DEFLTID	DSN6SPRM	DSNTIPP1	1-8 chars: IBMUSER
			Default authorization identifier
DEL_CFSTRUCTS _ON_RESTART	DSN6GRP	DSNTIPK	NO, YES
			Delete CF structures on restart
DELIM	DSNHDECP	DSNTIPF	DEFAULT, " (quotation mark), ' (apostrophe)
			Default string delimiter
DESCSTAT	DSN6SPRM	DSNTIP4	NO, YES
			Build DESCRIBE SQLDA for static SQL
DISABLE_EDMRTS	DSN6SPRM	DSNTIP8	NO, YES
			Disable the collection of real-time statistics (package LASTUSED)
DISALLOW_SEL_ INTO_UN	DSN6SPRM	Not on any installation panel	YES, NO
			Disallow UNION as the outermost from-clause of a SELECT INTO
DLDFREQ	DSN6SYSP	DSNTIPL1	ON, OFF
			Update Level ID at checkpoint intervals
DLITOUT	DSN6SPRM	DSNTIPI	1-254: 6
			Timeout for IMS/DLI connections
DPSEGSZ	DSN6SYSP	DSNTIP71	0, 4,... 64 (multiples of 4): 32
			Default SEGSIZE for explicit tablespaces
DSHARE	DSN6GRP	DSNTIPA1	NO, YES, or blank for update
			Data sharing enablement status
DSMAX	DSN6SPRM	DSNTIPC	1-200000: (default based on calculations)
			Maximum concurrent data sets opened
DSQLDELI	DSNHDECP	DSNTIPF	' (apostrophe) or " (quotes)
			Distributed SQL string delimiter
DSSTIME	DSN6SYSP	DSNTIPN	1-60: 5
			Time between resetting of dataset statistics
DSVCI	DSN6SYSP	DSNTIP7	YES, NO
			Variable CSI for DB2-managed data sets (NO = 4K only)
DYNRULES	DSNHDECP	DSNTIP4	YES, NO
			Use for DYNAMICRULES
	DSN6SPRM	DSNTIPC	5120-4194304: 51200 (in KB)

Macro Parameter	Macro Name	Panel	Acceptable Values Description
EDM_SKELETON_POOL			Maximum EDM skeleton pool size
EDMDBDC	DSN6SPRM	DSNTIPC	5000-4194304: 23400 (in KB) EDM pool DBD cache size
EDMSTMTC	DSN6SPRM	DSNTIPC	5000-4194304: 113386 (in KB) EDM manager statement cache size
EDPROP	DSN6SPRM	DSNTIPO	1, 2, 3 Data Propagator non-relational propagate option
EN_PJSJ	DSN6SPRM	Not on any installation panel	OFF, ON Enable dynamic index ANDing
ENSCHEME	DSNHDECP	DSNTIPF	EBCDIC, ASCII, UNICODE Default encoding scheme
EVALUNC	DSN6SPRM	DSNTIP8	NO, YES Allow predicate evaluation on uncommitted data
EXTRAREQ	DSN6SYSP	DSNTIP5	0-100: 100 Upper limit of extra DRDA query blocks from remote server
EXTRASRV	DSN6SYSP	DSNTIP5	0-100: 100 Upper limit of DRDA query blocks to a DRDA client
EXTSEC	DSN6SYSP	DSNTIPR	NO, YES DDF/DRDA extended security option
FCCOPYDDN	DSN6SPRM	DSNTIP61	Valid DB2 utilities template: HLQ.&DB..&SN..N&DSNUM..&UQ Default for FLASHCOPY(FCOPYDDN)
FLASHCOPY_COPY	DSN6SPRM	DSNTIP61	NO, YES Default FLASHCOPY for COPY utility
FLASHCOPY_LOAD	DSN6SPRM	DSNTIP61	NO, YES Default FLASHCOPY for LOAD utility
FLASHCOPY_PPRC	DSN6SPRM	DSNTIP61	NONT, PREFERRED, REQUIRED, blank Flashcopy peer-to-peer remote copy
FLASHCOPY_REBUILD_INDEX	DSN6SPRM	DSNTIP61	NO, YES Default FLASHCOPY for REBUILD INDEX utility
FLASHCOPY_REORG_INDEX	DSN6SPRM	DSNTIP61	NO, YES Default FLASHCOPY for REORG INDEX utility
FLASHCOPY_REORG_TS	DSN6SPRM	DSNTIP61	NO, YES Default FLASHCOPY for REORG TABLESPACE utility
GCCSID	DSNHDECP	DSNTIPF	1-65533: none EBCDIC CCSID (graphic)
GET_ACCEL_ARCHIVE	DSN6SPRM	DSNTIP82	NO, YES Default for GET_ACCEL_ARCHIVE special register
GRPNAME	DSN6GRP	DSNTIPK	1-8 characters: DSNCAT Data sharing group name
HONOR_KEEPDICTIONARY	DSN6SPRM	Not on any installation panel	NO, YES Honor KEEPDICTIONARY on LOAD and REORG when tables are converted
IDAUTH_MODULE	DSN6SYSP	DSNTIPO3	1-8 char: DSN@ATH for migration Authorization ID exit module name
IDBACK	DSN6SYSP	DSNTIPE	1-20000: 50 Maximum number of background connections
IDFORE	DSN6SYSP	DSNTIPE	1-20000: 50 Maximum number of foreground connections
IDTHTOIN	DSN6FAC	DSNTIPR	0-9999: 120 (in seconds) DDF timeout for idle DB access threads
IDXBPOOL	DSN6SYSP	DSNTIP1	Any 4K, 8K, 16K or 32K buffer pool name: BP0 Default BP for user indexes
IGNSORTN	DSN6SPRM	DSNTIP6	YES, NO Ignore utility SORTNUM keyword
IMMEDWRI	DSN6GRP	DSNTIP8	YES, NO Write updated GBP-dependent pages immediately

Macro Parameter	Macro Name	Panel	Acceptable Values Description
IMPDSDEF	DSN6SYSP	DSNTIP72	YES, NO Define data sets for implicitly created table spaces and indexes
IMPDSSIZE	DSN6SYSP	Not on any installation panel	1, 2, 4, 8, 16, 32, 64, 128, 256 (64 for SAP) Maximum size in GB for implicitly created table spaces
IMPLICIT_ TIMEZONE	DSNHDECP	DSNTIP4	CURRENT, SESSION, -12:59 to +14:00 Implicit time zone
IMPTKMOD	DSN6SYSP	Not on any installation panel	YES, NO Track page modifications of implicitly created table spaces
IMPTSCMP	DSN6SYSP	DSNTIP72	YES, NO Use data compression for implicit table spaces
INDEX_CLEANUP_ THREADS	DSN6SPRM	DSNTIPE1	0-128: 10 Max number of threads to clean up pseudo deleted index entries
INDEX_MEMORY_ CONTROL	DSN6SYP	DSNTIP71	AUTO, DISABLE, 10-200000 Memory for fast index traversing
INLISTP	DSN6SPRM	Not on any installation panel	1-5000: 50 IN list predicates pushdown degree
IRLMAUT	DSN6SPRM	DSNTIPI	YES, NO DB2 automatically starts/stops IRLM
IRLMPRC	DSN6SPRM	DSNTIPI	1-8 characters: IRLMPROC IRLM started task name
IRLMRWT	DSN6SPRM	DSNTIPI	1-3600: 30 (in seconds) IRLM timeout detection time
IRLMSID	DSN6SPRM	DSNTIPI	1 to 4 characters: IRLM IRLM subsystem name
IRLMSWT	DSN6SPRM	DSNTIPI	1-3600: 120 (in seconds) Time to IRLM Auto Start
IX_TB_PART_ CONV_EXCLUDE	DSN6SPRM	DSNTIP71	NO, YES Exclude end columns in index- to table-controlled partitioning keys conversion
LIKE_BLANK_ INSIGNIFICANT	DSN6SPRM	Not on any installation panel	NO, YES Ignore trailing blanks in LIKE predicate
IXQTY	DSN6SYSP	DSNTIP7	0-4194304: 0 (in KB) Default space for index spaces (0 = 1 cylinder)
LBACKOUT	DSN6SYSP	DSNTIP11	AUTO, YES, NO, LIGHT, LIGHTAUTO Backward log processing postponed
LC_CTYPE	DSNHDECP	DSNTIPF	Locale of 0-50 characters: blank Locale LC_CTYPE
LOB_INLINE_ LENGTH	DSN6SYSP	DSNTIPD	0-32680: 0 LOB Inline Length
LRDRTHLD	DSN6SSYSP	DSNTIPE1	0-1439: 10 (in minutes) Long-running reader threshold warning
MAINTYPE	DSN6SPRM	DSNTIP81	NONE, SYSTEM, USER, ALL Default for CURRENT MAINTAINED TABLE TYPES FOR OPTIMIZATION special register
MATERIALIZE_ NODET_SQLTUDF	DSN6SPRM	Not on any installation panel	NO, YES Materialize result of NOT DETERMINISTIC SQL table UDF
MAX_ CONCURRENT_ PKG_OPS	DSN6SPRM	Not on any installation panel	10 Maximum number of simultaneous package bind requests
MAXCONQN	DSN6FAC	DSNTIP5	OFF, ON, 1-19999 Maximum connection queue depth
MAXCONQW	DSN6FAC	DSNTIP5	OFF, ON, 5-3600 Maximum connection queue wait
MAX_NUM_CUR	DSN6SPRM	DSNTIPX	0-99999: 500 Maximum open cursors per thread
MAX_ST_PROC	DSN6SPRM	DSNTIPX	0-99999: 2000 Max stored procedures per thread
MAXARCH	DSN6LOGP	DSNTIPA	10-10000: 10000 Max archive logs recorded in BSDS
MAXDBAT	DSN6SYSP	DSNTIPE	0-19999: 200 Maximum active DBATs allowed

Macro Parameter	Macro Name	Panel	Acceptable Values Description
MAXKEEPD	DSN6SPRM	DSNTIPE	0-204800: 5000 Maximum number of prepared SQL statements to save past commit
MAXOFILR	DSN6SYSP	DSNTIPE	0-CTHREAD value: 100 Maximum open datasets for LOB file references
MAXRBLK	DSN6SPRM	DSNTIPC	0, 128-10000000: 400000 (in KB) RID pool size
MAXRTU	DSN6LOGP	DSNTIPA	1-99: 2 Maximum concurrent drives for archive log read activity
MAXSORT_IN_MEMORY	DSN6SPRM	DSNTIPC	1000-SRTPOOL value: 1000 (in KB) Max storage for in-memory sort
MAXTEMPS	DSN6SPRM	DSNTIP91	0-2147483647: 0 Maximum storage in workfile database per agent
MAXTEMPS_RID	DSN6SPRM	DSNTIP91	NONE, NOLIMIT, 1-329166 Maximum storage in workfile database for a single RID list
MAXTYPE1	DSN6FAC	DSNTIPR	0- CONDBAT value: 0 Maximum inactive DBATs
MCCSID	DSNHDECP	DSNTIPF	1-65533: none EBCDIC CCSID (mixed)
MEMBNAME	DSN6GRP	DSNTIPK	1-8 characters: DSN1 Data sharing member name
MGEXTSZ	DSN6SYSP	DSNTIP7	YES, NO Size SECQTY allocations according to a sliding scale
MINDVSCL	DSN6SPRM	Not on any installation panel	NONE, 3, 6 Min scale for decimal division result
MIXED	DSNHDECP	DSNTIPF	YES, NO Mixed data
MON	DSN6SYSP	DSNTIPN	NO, YES (class 1), list of classes (1-8), or * Monitor trace status when DB2 started
MONSIZE	DSN6SYSP	DSNTIPN	1048576-67108864: 1048576 Monitor trace default buffer size
MOVE_TO_ARCHIVE_DEFAULT	DSN6SPRM	Not on any installation panel	Y, N, E Default for build-in global variable SYSIBMADM.MOVE_TO_ARCHIVE
MXDTCACH	DSN6SPRM	DSNTIP81	0-512: 20 (in MB) Maximum memory for data caching per thread
NPGTHRSH	DSN6SPRM	Not on any installation panel	0-NPAGE: 0 NPAGE threshold (access path adjustment)
NUMLKTS	DSN6SPRM	DSNTIPJ	0-104857600: 2000 Maximum number of row/page locks per table/TS before escallation
NUMLKUS	DSN6SPRM	DSNTIPJ	0-104857600: 10000 Maximim number of page/row locks per user
OBJECT_CREATE_FORMAT	DSN6SPRM	DSNTIP7	BASIC, EXTENDED: EXTENDED RBA/LRSN format for new objects
OPTHINTS	DSN6SPRM	DSNTIP8	NO, YES Allow hints from user to optimizer
OPT1ROWBLOCKSORT	DSN6SPRM	Not on any installation panel	ENABLE, DISABLE OPTIMIZE FOR 1 ROW
OTC_LICENSE	DSN6SYSP	DSNTIP7	YES: none License terms accepted
OUTBUFF	DSN6LOGP	DSNTIPL	409600-409600000, 400K-400000K, 1M-390M (for field) or 400-400000 (for parameter) Output buffer size for active log write activity
PADIX	DSN6SPRM	DSNTIP71	NO, YES New indexes pad option

Macro Parameter	Macro Name	Panel	Acceptable Values Description
PADNTSTR	DSNHDECP	DSNTIP4	YES, NO Pad null-terminated strings
PAGESET_ PAGENUM	DSN6SPRM	DSNTIP71	ABSOLUTE, RELATIVE Paging of new range-partitioned table spaces
PARA_EFF	DSN6SPRM	DSNTIP81	0-100: 50 Parallelism efficiency factor
PARAMDEG	DSN6SPRM	DSNTIP81	0-254: 0 Maximum degree of parallelism
PARAMDEG_DPSI	DSN6SPRM	DSNTIP81	DISABLE, 0-254: 0 Maximum degree of parallelism with DPSI
PARAMDEG_UTIL	DSN6SPRM	DSNTIP6	0-32767: 99 Maximum number of parallel utility subtasks
PCLOSEN	DSN6SYSP	DSNTIPL1	1-32767: 10 Read-only switch checkpoints
PCLOSET	DSN6SYSP	DSNTIPL1	1-32767: 10 Read-only switch time
PCTFREE_UPD	DSN6SPRM	DSNTIP71	AUTO, 0-99: 0 Reserve freespace for update
PEER_RECOVERY	DSN6SPRM	DSNTIP71	NONE, RECOVER, ASSIST, BOTH: NONE Peer recovery
PKGREL_COMMIT	DSN6SPRM	DSNTIP8	NO, YES Package release commit (allows break into persistent threads)
PLANMGMT	DSN6SPRM	DSNTIP8	OFF, BASIC, EXTENDED Plan management
PLANMGMTSCOPE	DSN6SPRM	DSNTIP8	STATIC Plan management scope
POOLINAC	DSN6FAC	DSNTIP5	0-9999: 120 (in seconds) Time a DBAT remains idle in pool before it is terminated
PRIQTY	DSN6ARVP	DSNTIPA	1-4369: 125 (in cylinders) Primary space quantity for archive log dataset
PREVENT_ ALTERTB_ LIMITKEY	DSN6SPRM	DSNTIP72	NO, YES Forbid altering limit key for index-controlled partitioned table spaces
PREVENT_NEW_ IXCTRL_PART	DSN6SPRM	Not on any installation panel	YES, NO Forbid creation of new index-controlled partitioned tables
PRIVATE_ PROTOCOL	DSN6FAC	Not on any installation panel	NO, YES Private protocol processing
PROFILE_ AUTOSTART	DSN6SYSP	DSNTIPO	NO, YES Profile auto start
PROTECT	DSN6ARVP	DSNTIPP	NO, YES Use external security to protect archive log datasets
PTASKROL	DSN6SYSP	Not on any installation panel	YES, NO Roll up parallel tasks accounting trace
QUERY_ACCEL_ OPTIONS	DSN6SPRM	DSNTIP82	NONE, YES Query options enabled for accelerator
QUERY_ ACCELERATION	DSN6SPRM	DSNTIP82	1 (NONE), 2 (ENABLE), 3 (ENABLE_WITH_FAILBACK), 4 (ELIGIBLE), 5 (ALL): 1 Default for special register CURRENT_QUERY_ACCELERATION
QUIESCE	DSN6ARVP	DSNTIPA	1-999: 5 (in seconds) Maximum time to achieve full system quiesce
RANDOMATT	DSN6GRP	DSNTIPK	YES, NO Random group attach
REALSTORAGE_ MANAGEMENT	DSN6SPRM	DSNTIPE1	ON, OFF, AUTO DB2 manages real storage consumption
REALSTORAGE_ MAX	DSN6SPRM	Not on any installation panel	NOLIMIT, 1-65535 (in GB) Maximum real and auxiliary DB2 storage limit
REC_ FASTREPLICATION	DSN6SPRM	DSNTIP61	NONE, PREFERRED, REQUIRED Use FLASHCOPY for recovery from FlashCopy image copy

Macro Parameter	Macro Name	Panel	Acceptable Values Description
RECALL	DSN6SPRM	DSNTIPO	YES, NO Allow DFSMShsm automatic recall
RECALLD	DSN6SPRM	DSNTIPO	0-32767: 120 (in seconds) Maximum DFSMShsm recall delay
REFSHAGE	DSN6SPRM	DSNTIP81	0, ANY Default for special register CURRENT REFRESH AGE
REMOTE_COPY_ SW_ACCEL	DSN6LOGP	DSNTIPL	DISABLE, ENABLE Use software for peer-to-peer remote log copy
RENAMETABLE	DSN6SPRM	DSNTIP72	DISALLOW_DEP_VIEW_SQLTUDF, ALLOW_DEP_VIEW_SQLTUD Allow RENAME TABLE for views or SQL table functions
REORG_DROP_ PBG_PARTS	DSN6SPRM	DSNTIP63	ENABLE, DISABLE REORG utility to remove empty trailing partitions
REORG_MAPPING _DATABASE	DSN6SPRM	DSNTIP6	Database name: Blanks Default database for mapping table in online reorg
REORG_PART_ SORT_NPSI	DSN6SPRM	DSNTIP63	AUTO, NO, YES Sort all NPSI keys in REORG TABLESPACE PART utility
REORG_LIST_ PROCESSING	DSN6SPRM	DSNTIP63	PARALLEL, SERIAL PARALLEL option default for REORG TABLESPACE utility
RESTART	DSN6SPRM	DSNTIPS	RESTART, DEFER Restart or defer processing for specified objects at DB2 startup
RESTORE_ RECOVER_ FROMDUMP	DSN6SPRM	DSNTIP62	YES, NO System-level backup from dump on tape
RESTORE_ TAPEUNITS	DSN6SPRM	DSNTIP62	NOLIMIT, 1-255 Maximum number of tape units for system-level restore
RESTRICT_ALT_ COL_FOR_DCC	DSN6SPRM	Not on any installation panel	NO, YES ALTER COLUMN restrictions for tables with data capture active
RESYNC	DSN6FAC	DSNTIPR	1-99: 2 DDF resynchronization time period
RETLWAIT	DSN6SPRM	DSNTIPI	0-254: 0 Retained lock timeout
RETRY_STOPPED_ OBJECTS	DSN6SPRM	DSNTIP72	NO, YES Retry a claim against a stopped object
REVOKE_DEP_ PRIVILEGES	DSN6SPRM	DSNTIPP1	NO, YES, SQLSTMT Revoke dependent privileges
RGFCOLID	DSN6SPRM	DSNTIPZ	1-8 chars: DSNRGCOL Owner of registration tables
RGFDBNAM	DSN6SPRM	DSNTIPZ	1-8 chars: DSNRGFDB Registration tables database
RGFDEDPL	DSN6SPRM	DSNTIPZ	NO, YES Registration control all applications option
RGFDEFLT	DSN6SPRM	DSNTIPZ	ACCEPT, REJECT, APPL Registration action for DDL with unregistered objects
RGFESCP	DSN6SPRM	DSNTIPZ	Any non-alphanumeric char: none Registration tables (ART/ORT) escape character
RGFFULLQ	DSN6SPRM	DSNTIPZ	YES, NO Registration fully qualified names required
RGFINSTL	DSN6SPRM	DSNTIPZ	NO, YES Install data definition control support
RGFNMORT	DSN6SPRM	DSNTIPZ	1-17 chars: DSN_REGISTER_OBJT Object registration table (ORT)
RGFNMPRT	DSN6SPRM	DSNTIPZ	1-17 chars: DSN_REGISTER_APPL Application registration table (ART)
RLF	DSN6SYSP	DSNTIPO4	NO, YES Resource limit facility auto start
RLFENABLE	DSN6SYSP	DSNTIPO4	DYNAMIC, STATIC, ALL RLF governing

Macro Parameter	Macro Name	Panel	Acceptable Values Description
RLFAUTH	DSN6SYSP	DSNTIPP1	1-8 chars: SYSIBM RLF authorization ID
RLFERR	DSN6SYSP	DSNTIPO4	NOLIMIT, NORUN, 1-5000000 RLF error option for dynamic SQLs
RLFERRD	DSN6FAC	DSNTIPO4	NOLIMIT, NORUN, 1-5000000 DDF RLF error option for dynamic SQLs
RLFERRSTC	DSN6SYSP	DSNTIPO4	NOLIMIT, NORUN, 1-5000000 RLF error option for static SQLs
RLFERRDSTC	DSN6FAC	DSNTIPO4	NOLIMIT, NORUN, 1-5000000 DDF RLF error option for static SQLs
RLFTBL	DSN6SYSP	DSNTIPO4	2 alphanumeric chars (not national characters): 01 RLF table suffix default
ROUTCDE	DSN6SYSP	DSNTIPO	1-16 route codes separated by commas: 1 WTO route codes for DB2 messages
RRULOCK	DSN6SPRM	DSNTIPI	NO, YES User U (update) lock for RR/RS
SCCSID	DSNHDECP	DSNTIPF	1-65533: none EBCDIC CCSID (single-byte)
SECADM1	DSN6SPRM	DSNTIPP1	1-8 chars for authid (1-128 for role): SECADM Security admin 1
SECADM1_INPUT_STYLE	DSN6SPRM	Not on any installation panel	HEX, CHAR SECADM1 input style
SECADM1_TYPE	DSN6SPRM	DSNTIPP1	AUTHID, ROLE Security admin 1 type
SECADM2	DSN6SPRM	DSNTIPP1	1-8 chars for authid (1-128 for role): SECADM Security admin 2
SECADM2_INPUT_STYLE	DSN6SPRM	Not on any installation panel	HEX, CHAR SECADM2 input style
SECADM2_TYPE	DSN6SPRM	DSNTIPP1	AUTHID, ROLE Security admin 2 type
SECQTY	DSN6ARVP	DSNTIPA	1-4369: 15 (in cylinders) Secondary space quatity for archive log dataset
SEPARATE_SECURITY	DSN6SPRM	DSNTIPP1	NO, YES Separate DB2 administrator security duties
SIGNON_MODULE	DSN6SYSP	DSNTIPO3	1-8 chars: DSN3@SGN Signon module
SIMULATED_CPU_COUNT	DSN6SPRM	Not on any installation panel	OFF, 1-255 Simulated CPU count
SIMULATED_CPU_SPEED	DSN6SPRM	Not on any installation panel	OFF, 1 to 2147483647 Simulated CPU speed
SITETYP	DSN6SPRM	DSNTIPO	LOCALSITE, RECOVERYSITE Site type for recovery purposes
SJTABLES	DSN6SPRM	Not on any installation panel	1-3 (always consider), 4-254 (only consider if number of tables is higher), 255+ (don't consider): 10 Threshold value for tables in a Star Join
SKIPUNCI	DSN6SPRM	DSNTIP8	NO, YES Skip uncommitted inserts
SMF89	DSN6SYSP	Not on any installation panel	NO, YES Detailed measured usage tracking enabled
SMFACCT	DSN6SYSP	DSNTIPN	NO, YES (class 1), *list of classes*, or * (all): 1,2,3,7,8 Accounting traces to SMF at start up
SMFCOMP	DSN6SYSP	DSNTIPN	OFF, ON Compress DEST(SMF) trace records
SMFSTAT	DSN6SYSP	DSNTIPN	NO, YES (classes 1, 3-6), *list of classes*, or * (all) Statistics traces to SMF at start up
SPT01_INLINE_LENGTH	DSN6SPRM	DSNTIPA2	NOINLINE, 1-32138: 32138 Maximum length of LOB column data in SPT01 directory table space

Macro Parameter	Macro Name	Panel	Acceptable Values Description
SQLDELI	DSNHDECP	DSNTIPF	DEFAULT, " (quotation mark), ' (apostrophe) Default SQL string delimiter
SQLLEVEL	DSNHDECP	DSNTIPF	V10R1, V11R1, *function-level*: V12R1M100 (for migration) SQL level
SRTPOOL	DSN6SPRM	DSNTIPC	240-128000: 10000 (in KB) Sort pool size
SSID	DSNHDECP	DSNTIPM	1-4 chars: DSN1 Subsystem name
STARJOIN	DSN6SPRM	DSNTIP81	DISABLE, ENABLE, 1-32768 Star join availability
STATFDBK_ PROFILE	DSN6SPRM	DSNTIP8	YES, NO Modifications to statistics profiles
STATFDBK_SCOPE	DSN6SPRM	DSNTIP8	ALL, DYNAMIC, NONE, STATIC SQL types to collect statistics recommendations for
STATHIST	DSN6SPRM	DSNTIP6	SPACE, NONE, ALL, ACCESSPATH Statistics history collection default
STATIME	DSN6SYSP	DSNTIPN	1-60: 5 (in minutes) Statistics trace collection interval
STATROLL	DSN6SPRM	DSNTIP6	YES, NO RUNSTATS aggregates partition level statistics
STATSINT	DSN6SPRM	DSNTIPO	1-1440: 30 (in minutes) Real-time statisctics externalization interval
STDSQL	DSNHDECP	DSNTIP4	YES, NO Standard SQL language
STORMXAB	DSN6SYSP	DSNTIPX	0-225: 0 Maximum number of stored procedure or UDF abends allowed
STORTIME	DSN6SYSP	DSNTIPX	5-1800, NOLIMIT: 180 (in seconds) Stored procedure dispatch timeout
SUBQ_MIDX	DSN6SPRM	Not on any installation panel	ENABLE, DISABLE Subquery multiple index access
SUPERRS	DSN6SPRM	DSNTIPM	YES, NO Suppress soft errors
SUPPRESS_HINT_ SQLCODE_DYN	DSN6SPRM	Not on any installation panel	YES, NO Suppress SQLCODEs +394 and +395
SVOLARC	DSN6ARVP	DSNTIPA	YES, NO Allocate archive with volume count of 1
SYNCVAL	DSN6SYSP	DSNTIPN	NO, 0-59 Synchronize statatistics on minutes after the hour
SYSADM	DSN6SPRM	DSNTIPP1	1-8 chars: SYSADM 1st of 2 authids with installation SYSADM authority
SYSADM2	DSN6SPRM	DSNTIPP1	1-8 chars: SYSADM 2nd of 2 authids with installation SYSADM authority
SYSOPR1	DSN6SPRM	DSNTIPP1	1-8 chars: SYSOPR 1st of 2 authids with installation SYSOPR authority
SYSOPR2	DSN6SPRM	DSNTIPP1	1-8 chars: SYSOPR 2nd of 2 authids with installation SYSOPR authority
SYSTEM_LEVEL_ BACKUPS	DSN6SPRM	DSNTIP62	NO, YES RECOVERY uses system-level backups for object recoveries
TBSBP8K	DSN6SYSP	DSNTIP1	Any 8K bufferpool name: BP8K0 Default bufferpool for 8K page table spaces created implicitly
TBSBP16K	DSN6SYSP	DSNTIP1	Any 16K bufferpool name: BP16K0 Default bufferpool for 16K page table spaces created implicitly
TBSBP32K	DSN6SYSP	DSNTIP1	Any 32K bufferpool name: BP32K Default bufferpool for 32K page table spaces created implicitly

Macro Parameter	Macro Name	Panel	Acceptable Values Description
TBSBPLOB	DSN6SYSP	DSNTIP1	Any 4K/8K/16K/32K bufferpool: BP0 Default bufferpool for LOB table spaces created implicitly
TBSBPOOL	DSN6SYSP	DSNTIP1	Any 4K bufferpool name: BP0 Default bufferpool for 4K page table spaces created implicitly
TBSBPXML	DSN6SYSP	DSNTIP1	Any 4K/8K/16K/32K bufferpool name: BP16K0 Default bufferpool for XML table spaces created implicitly
TCPALVER	DSN6FAC	DSNTIP5	NO, YES, CLIENT, SERVER, SERVER_ENCRYPT DDF TCP/IP connect security form
TCPKPALV	DSN6FAC	DSNTIP5	ENABLE, DISABLE, 1-65534: 120 DDF TCP/IP KeepAlive override specification
TEMPLATE_TIME	DSN6SPRM	DSNTIP6	LOCAL, UTC Default time for TEMPLATE
TIME	DSNHDECP	DSNTIP4	ISO, USA, EUR, JIS, LOCAL Time format
TIMELEN	DSNHDECP	DSNTIP4	0, 8-254 Local time length
TRACSTR	DSN6SYSP	DSNTIPN	NO, YES (classes 1-3), list of global classes (1-9), * Global trace auto start
TRACTBL	DSN6SYSP	DSNTIPN	4K-396K: 64K Global trace table size for DEST(RES)
TRKRSITE	DSN6SPRM	DSNTIPO	NO, YES This DB is a remote tracker site
TSQTY	DSN6SYSP	DSNTIP7	0-4194304: 0 (1 cyl for non-LOB and 10 cyls for LOB table spaces) Default space (KB) for table spaces
TSTAMP	DSN6ARVP	DSNTIPH	NO, YES, EXT Use date and time within archive log data set name
TWOACTV	DSN6LOGP	DSNTIPH	1, 2 Number of active log copies
TWOARCH	DSN6LOGP	DSNTIPH	1, 2 Number of archive log copies
TWOBSDS	DSN6LOGP	Not on any installation panel	YES, NO Maintain two copies of the BSDS
UGCCSID	DSNHDECP	DSNTIPF	1208 UNICODE CCSID (1200 for graphic)
UIFCIDS	DSN6SYSP	DSNTIPN	YES, NO Include UNICODE data when writing IFCIDs
UMCCSID	DSNHDECP	DSNTIPF	1208 UNICODE CCSID (1208 for mixed)
UNION_ COLNAME_7	DSN6SPRM	Not on any installation panel	NO, YES DB2 V8/V7 UNION column name behavior
UNIT	DSN6ARVP	DSNTIPA	Device type or unit name: TAPE Allocation unit for COPY1 archive log data sets
UNIT2	DSN6ARVP	DSNTIPA	Device type or unit name: none Allocation unit for COPY1 archive log data sets
URCHKTH	DSN6SYSP	DSNTIP11	0-255: 5 Number of checkpoint cycles before inflight UR message issued
URLGWTH	DSN6SYSP	DSNTIP11	0-1000K: 10K Maximum number of log records from uncomitted UR before warning
USCCSID	DSNHDECP	DSNTIPF	1208 UNICODE CCSID
UTIL_DBBSG	DSN6SPRM	DSNTIP62	blank or valid SMS management name DB backup storage group
UTIL_LGBSG	DSN6SPRM	DSNTIP62	blank or valid SMS management name. LOG backup storage group

Macro Parameter	Macro Name	Panel	Acceptable Values Description
UTILS_HSM_ MSGDS_HLQ	DSN6SPRM	DSNTIP62	1-6 characters (valid dsn qualifier): blank High-level qualifier for HSM messages data sets
UTIL_TEMP_ STORCLAS	DSN6SPRM	DSNTIP6	blank or valid SMS management name. Utilities temporary storage class
UTILS_DUMP_ CLASS_NAME	DSN6SPRM	DSNTIP62	blank or valid DFSMS dump class name (1-8 chars) DFSMShsm dump class name
UTILITY_OBJECT_ CONVERSION	DSN6SPRM	DSNTIP7	BASIC, EXTENDED, NOBASIC, NONE Utility object conversion
UTIMOUT	DSN6SPRM	DSNTIP6	1-254: 6 Lock timeout multiplier for utility timeouts
VOLTDEVT	DSN6SPRM	DSNTIP6	Valid device or unit name: SYSDA Device type for temporary data sets
WFDBSEP	DSN6SPRM	DSNTIP91	YES, NO Separate table spaces in workfile
WFSTGUSE_ AGENT_ THRESHOLD	DSN6SPRM	DSNTIP91	0-100: 0 Percent of available space in workfile for a single agent
WFSTGUSE_ SYSTEM_ THRESHOLD	DSN6SPRM	DSNTIP91	0-100: 90 Percent of available space in workfile for all agents
WLMENV	DSN6SYSP	DSNTIPX	1-32 chars: blank Default WLM environment for stored procedures and UDFs
XLKUPDLT	DSN6SPRM	DSNTIPI	NO, YES, TARGET Locking method for searched U/D
XML_RANDOMIZE _DOCID	DSN6SYSP	DSNTIP8	NO, YES Generate random XML DOCIDs
XML_RESTRICT_ EMPTY_TAG	DSN6SPRM	Not on any installation panel	NO, YES Serialize an empty XML element to <X></X>
ZOSMETRICS	DSN6SPRM	Not on any installation panel	NO, YES Provide z/OS metrics

IFCIDS

Accounting Trace Records

IFCID	Event
0003	Plan-level details (requires at least class 1 active)
0239	Package-level details (requires at least class 7 active)

Class 1: Accounting Data (activated IFCIDs)

IFCID	Event
0003	Plan accounting
0106	System parameters in effect
0200	UDF entry/exit signal
0239	Package accounting

Class 2: In DB2 Time (activated IFCIDs)

IFCID	Event
0232	DB2 thread entry/exit signal

Class 3: Wait Time (activated IFCIDs)

IFCID	Event
0006/0007	Begin/End read I/O operation
0008	Beginning of synchronous write I/O
0009	End of synchronous or asynchronous write I/O
0032/0033	Begin/End wait for log manager
0044	Lock suspend or IDENTIFY call IRLM
0045	Lock resume
0051/0052	Resume/Wait shared latch
0056/0057	Wait/Resume exclusive latch
0117/0118	Begin/End thread wait time for log I/O
0127	Agent ready to suspend page wait
0128	Page requestor resumed by I/O limit
0170	Suspend for synchronous execution unit switch
0171	Resume agent waiting DB2 service task
0174/0175	Begin/End ARCHIVE LOG MODE(QUIESCE)
0213/0214	Begin/End of wait for claim request
0215/0216	Begin/End of wait for drain request
0226/0227	Begin/End of suspend for page latch
0242/0243	Begin/End wait for scheduling stored procedure
0313	Messages for long-running URs
0321/0322	Begin/End Force-at-Commit operation
0329	Asynchronous group buffer pool requests
0351/0352	Start/End of TCP/IP LOB materialization
0378/0379	Start/End of accelerator call event
0382/0382	Begin/End suspend for parallel task task synchronization
0413/0414	Begin/End of a wait for a pipe suspend

Class 4: Installation-Defined Accounting Record

IFCID	Event
0151	User-defined accounting trace

Class 5: Time Spent Processing IFI Requests (activated IFCIDs)

IFCID	Event
0187	Entry to and exit from IFI

Class 7: Package Level Accounting in-DB2 Time (activated IFCIDs)

IFCID	Event
0232	DB2 thread entry/exit signal for package/DBRM level accounting
0239	Package accounting
0240	Event signal for package accounting

Class 8: Package Level Accounting Wait Time (activated IFCIDs)

IFCID	Event
0006/0007	Begin/End of a read I/O operation
0008	Beginning of a synchronous write I/O
0009	End of synchronous or asynchronous write I/O
0032/0033	Begin/End of wait for log manager
0044	Lock suspend or IDENTIFY call IRLM
0045	Lock resume
0051/0052	Resume/Wait shared latch
0056/0057	Wait/Resume exclusive latch
0117/0118	Begin/End thread wait time for log I/O
0127	Agent ready to suspend page wait
0128	Page requestor resumed by I/O init
0170	Suspend for synchronous execution unit switch
0171	Resume agent waiting DB2 service task
0174/0175	Begin/End ARCHIVE LOG MODE(QUIESCE)
0213/0214	Begin/End of wait for claim request
0215/0216	Begin/End of wait for drain request
0226/0227	Begin/End of suspend for page latch
0241	Begin/end suspension of package/DBRM
0242/0243	Begin/end wait for scheduling stored procedure
0321/0322	Begin/End force-at-commit operation
0329	Asynchronous group buffer pool requests
0351/0352	Start/end of TCP/IP LOB materialization
0378/0379	Start/end of accelerator call event
0382/0383	Begin/end suspend for parallel task task synchronization
0413/0414	Begin/end of a wait for a pipe suspend

Class 10: Package SQL Detail (activated IFCIDs)

IFCID	Event
0239	Package details

AUDIT Trace Records

Class 1: Authorization Failures

IFCID	Event
0140	Authorization failures

Class 2: Explicit Grant and Revoke

IFCID	Event
0141	Explicit grant and revokes

Class 3: Create, Drop, and Alter Operations against Audit Tables

IFCID	Event
0142	Audit of CREATEs, ALTERs, DROPs

Class 4: First Change of Audited Object

IFCID	Event
0143	First attempted write to audited object

Class 5: First Read of Audited Object

IFCID	Event
0144	First attempted read of audited object

Class 6: SQL Statement at Bind

IFCID	Event
0145	Audit log record of some SQL statements

Class 7: Change in Authorization for Audited Object

IFCID	Event
0055	Issuance of SET CURRENT SQLID
0083	End IDENTIFY request
0087	Ending of SIGNON request
0169	Distributed authid translation
0319	Audit trail for security processing

Class 8: Utility Access to Any Object

IFCID	Event
0023	Utility start information
0024	Utility object or phase change
0025	Utility end information
0219	LISTDEF data set information
0220	Utility output data set information

Class 9: Installation-Defined Audit Record

IFCID	Event
0146	User-defined audit trace

Class 10: Trusted Context Audit

IFCID	Event
0269	Trusted context enable and reuse
0270	Trusted context CREATE and ALTER

Class 11: Audit Administrative Authorities

IFCID	Event
0271	Row and column access control
0361	Administrative authorities

Global Trace Records

Class 1: IBM Service

IFCID	Event
0106	System parameters in effect
0132	IBM Service
0134	EDM pool full condition diagnosis
0138	IBM Service

Class 2: IBM Service

IFCID	Event
0106	System parameters in effect
0131	IBM Service
0133	DBD problem diagnosis
0139	IBM Service

Class 3: IBM Service

IFCID	Event
0000	Module entry exit trace
0038	Log buffer write
0046	Begin exit unit switch

IFCID	Event
0047/0048	Begin/End new SRB
0049/0050	Begin/End new TCB
0051/0052	Shared latch resume/wait
0056/0057	Exclusive latch wait/resume
0068/0069	Begin/End abend
0070/0071	Begin/End commit phase 2
0072/0073	Begin/End create thread
0074/0075	Begin/End terminate thread
0076/0077	Begin/End EOM (end of memory) request
0080/0081	Begin/End establish exit
0082/0083	Begin/End IDENTIFY request
0084/0085	Begin/End prepare commit
0086/0087	Begin/End SIGNON request
0088/0089	Begin/End sync
0093/0094	Entry/Exit suspend
0106	System parameters in effect
0114	Begin archive read I/O wait
0115	End archive read I/O wait on DASD
0116	End archive read I/O wait on tape
0117	Begin archive read
0174/0175	Begin/End ARCHIVE LOG MODE(QUIESCE)
0228/0229	Begin/End archive allocation wait
0252/0260	Begin/End CP request
0265/0266	Begin/End request for SCA
0267/0268	Begin/End CF structure rebuild
0364	Create, Terminate allied address space

Class 4: IBM Service

IFCID	Event
0106	System parameters in effect
0130	IBM Service

Class 5: IBM Service (Overflow Hybrid Join, Host Variable Tracing, DBD Invalidation)

IFCID	Event
0135	IBM Service
0136	IBM Service
0137	IBM Service
0190	Hybrid join overflow
0247	Input host variable
0248	Output host variable
0249	DBD invalidations

Class 6: User Defined Serviceability Trace

IFCID	Event
0156	User-defined serviceability trace

Class 7: IBM Service (Distributed Data)

IFCID	Event
0164	Distributed data
0165	Distributed data
0166	Distributed data

Class 8: IBM Service (Distributed SQL)

IFCID	Event
0168	Distributed SQL

Class 9: IBM Service (DB2 DRDA Protocol)

IFCID	Event
0180	Distributed communication flow
0181	Distributed communication flow
0182	Distributed communication flow

Class 10: Storage Manager Pool Statistics

IFCID	Event
0217	Storage manager pool statistics

Class 11: DB2-supplied Stored Procedure and UDF Trace

IFCID	Event
0344	DB2-supplied SP, UDF Entry/Exit
0345	DB2-supplied SP, UDF data

Monitor Trace Records

Class 1: Entry To or Exit From a Routine or Trigger

IFCID	Event
0200	Entry to and exit from a routine or trigger

Class 2: Time In-DB2 (CPU and Elapsed)

IFCID	Event
0232	DB2 thread entry exit signal

Class 3: Wait Time In-DB2

IFCID	Event
0006/0007	Begin/End read I/O operation
0008	Beginning of synchronous write I/O
0009	End of synchronous or asynchronous write I/O
0032/0033	Begin/End wait for log manager
0044	Lock suspend or IDENTIFY call IRLM
0045	Lock resume
0051/0052	Resume/Wait shared latch
0056/0057	Wait/Resume exclusive latch
0117/0118	Begin/End thread wait time for log I/O
0127	Agent ready to suspend page wait
0128	Page requestor resumed by I/O limit
0170	Suspend for synchronous execution unit switch
0171	Resume agent waiting DB2 service task
0174/0175	Begin/End ARCHIVE LOG MODE(QUIESCE)
0213/0214	Begin/End of wait for claim request
0215/0216	Begin/End of wait for drain request
0226/0227	Begin/End of suspend for page latch
0242/0243	Begin/End wait for scheduling stored procedure
0321/0322	Begin/End Force-at-Commit operation
0329	Asynchronous group buffer pool requests
0351/0352	Start/End of TCP/IP LOB materialization
0378/0379	Start/End of accelerator call event
0382/0382	Begin/End suspend for parallel task task synchronization
0413/0414	Begin/End of a wait for a pipe suspend

Class 4: Installation-Defined Monitor Record

IFCID	Event
0155	User-defined monitor trace

Class 5: Time Spent Processing

IFCID	Event
0187	Entry to and exit from IFI

Class 6: Data Capture Data

IFCID	Event
0185	Data capture information

Class 7: Package Level Accounting In-DB2 Time

IFCID	Event
0232	DB2 thread entry/exit signal for package/DBRM level accounting
0239	Package accounting
0240	Event signal for package accounting

Class 8: Package Level Accounting Wait Time (activated IFCIDs)

IFCID	Event
0006/0007	Begin/End of a read I/O operation
0008	Beginning of a synchronous write I/O
0009	End of synchronous or asynchronous write I/O
0032/0033	Begin/End of wait for log manager
0044	Lock suspend or IDENTIFY call IRLM
0045	Lock resume
0051/0052	Resume/Wait shared latch
0056/0057	Wait/Resume exclusive latch
0117/0118	Begin/End thread wait time for log I/O
0127	Agent ready to suspend page wait
0128	Page requestor resumed by I/O init
0170	Suspend for synchronous execution unit switch
0171	Resume agent waiting DB2 service task
0174/0175	Begin/End ARCHIVE LOG MODE(QUIESCE)
0213/0214	Begin/End of wait for claim request
0215/0216	Begin/End of wait for drain request
0226/0227	Begin/End of suspend for page latch
0241	Begin/end suspension of package/DBRM
0242/0243	Begin/end wait for scheduling stored procedure
0321/0322	Begin/End force-at-commit operation
0329	Asynchronous group buffer pool requests
0351/0352	Start/end of TCP/IP LOB materialization
0378/0379	Start/end of accelerator call event
0382/0383	Begin/end suspend for parallel task task synchronization
0413/0414	Begin/end of a wait for a pipe suspend

Class 9: SQL Statement-Level Accounting

IFCID	Event
0124	Activate/Deactivate statement-level accounting

Class 10: Package Detail

IFCID	Event
0239	Package details

Class 11: Plan Level Accounting

IFCID	Event
0003	Plan level accounting
0200	Entry to or Exit from a routine or trigger

Class 29: Subsystem wide collection of SQL statistics for Statements

IFCID	Event
0316	Dynamic statement detail
0318	Activate dynamic statement statistics collection
0400	Activate static statement statistics collection
0401	Static statement detail

Performance Trace Records

Class 1: Background Events

IFCID	Event
0001	System services statistics
0002	Database services statistics
0031	EDM pool full
0042/0043	Begin/End checkpoint
0076/0077	Begin/End EOM (End of Memory) request
0078/0079	Begin/End EOT (End of Task) request
0102/0103	On/Off for short on storage
0105	Buffer manager mapping of DBIDs and OBIDs
0106	System parameters in effect
0107	Data set OPEN/CLOSE information
0153	User-defined exception condition trace

Class 2: Subsystem-Related Events

IFCID	Event
0003	Accounting
0068/0069	Begin/End ROLLBACK
0070/0071	Begin/End COMMIT phase 2
0072/0073	Begin/End CREATE THREAD
0074/0075	Begin/End TERMINATE THREAD
0080/0081	Begin/End establish exit
0082/0083	Begin IDENTIFY request
0084/0085	Begin/Edit prepare to COMMIT (IMS/CICS/RRSAF)
0086/0087	Begin SIGNON (IMS/CICS/RRSAF)
0088/0089	Begin/End SYNC request
0106	Systems parameters in effect at trace invocation
0174/0175	Begin/End ARCHIVE LOG MODE(QUIESCE) command processing
0321/0322	Begin/End Force-at-Commit operation

Class 3: SQL-Related Events

IFCID	Event
0022	Mini plans generated
0053	End Describe, COMMIT, RLBK, or ERROR
0055	Issuance of SET CURRENT SQLID
0058	End SQL statement
0059	Begin FETCH SQL
0060	Begin SELECT SQL
0061	Begin INSERT, UPDATE, or DELETE
0062	Begin DDL
0063	Identify SQL statement to be parsed
0064	Begin SQL PREPARE
0065	Begin OPEN CURSOR
0066	Begin CLOSE CURSOR
0092	Access method services command start
0095/0096	Begin/End of sort
0097	Access method services command end
0106	System parameters in effect
0112	Plan attributes for successful allied thread allocation
0173	Written when -904 is issued
0177	Successful package allocation
0233	Start or end of a call to a stored procedure at a DB2 server
0237	Written when SET CURRENT DEGREE statements are executed
0250	Connect/disconnect from coupling facility
0272	ASSOCIATE LOCATORS information
0273	Execution of ALLOCATE CURSOR statement
0311	ALLOCATE CURSOR information
0324	Function resolution information
0325	Start/End trigger activation

IFCID	Event
0343	MAXTEMPZPARM limit exceeded
0360	Incremental rebind: pre-DB2 10 parallelism

Class 4: Buffer Manager I/O and EDM Pool Requests

IFCID	Event
0006/0007	Begin/End read I/O
0008	Begin synchronous write I/O
0009	End synchronous or asynchronous write I/O
0010	Begin asynchronous write I/O
0029/0030	Begin/End EDM I/O request
0105	Buffer manager mapping of DBIDs and OBIDs
0106	System parameters in effect
0107	Buffer manager OPEN/CLOSE activity
0127/0128	Begin/End waiting for asynchronous I/O
0226/0227	Begin/End of an agent suspend to wait for a page latch
0321/0322	Begin/End FORCE at COMMIT
0357/0358	Begin/End of index insert with parallelism
0359	Index page split
0477	Allocation/Deallocation of structures for fast index traversal

Class 5: Log Manager

IFCID	Event
0032/0033	Begin/End log manager wait
0034/0035	Begin/End read I/O
0036/0037	Begin/End non-I/O wait
0038/0039	Begin/End write I/O
0040/0041	Begin/End archive write
0104	Log manager data set mapping
0106	System parameters in effect
0114	Begin read I/O archive
0115	End read I/O archive on DASD
0116	End read I/O archive on tape
0117/0118	Begin/End log wait archive
0119/0120	Begin/End BSDS write I/O
0228/0229	Begin/End archive allocation wait

Class 6: Summary Lock Information

IFCID	Event
0020	Page lock and table space lock summary
0044/0045	Suspend/Resume lock
0105	Buffer manager mapping of DBIDs and OBIDs
0106	Systems parameter in effect
0107	Buffer manager OPEN/CLOSE activity
0172	Units of work involved in deadlock
0196	Lock timeout details
0213/0214	Begin/End wait for a drain lock
0218	Summary of lock avoidance technique
0337	Lock escalation occurred

Class 7: Detailed Lock Information

IFCID	Event
0021	Every lock acquired, changed, or released on return from IRLM
0105	Buffer manager mapping of DBIDs and OBIDs
0106	System parameters in effect
0107	Buffer manager OPEN/CLOSE activity
0223	Detailed information about each successful use of lock avoidance

Class 8: Data Manager Detail

IFCID	Event
0013/0014	Begin/End hash scan
0015	Input to matching/nonmatching index scan
0016	Input to the first insert
0017	Begin sequential scan
0018	End sequential scan, tablespace scans
0105	Buffer manager mapping of DBIDs and OBIDs
0106	System parameters in effect
0107	Buffer manager OPEN/CLOSE activity
0125	RID list processing usage
0221	Degree of parallel I/O processing for a parallel group
0222	Elapsed time of a parallel group
0231	Parallel group completion
0305	Table check constraints
0311	Information about operations on temporary tables
0363	Straw model use

Class 9: Sort Detail

IFCID	Event
0026	Work file obtained
0027	Number of ordered records in sort run
0028	Detailed sort information
0095/0096	Begin/End sort
0106	Systems parameter in effect at trace invocation

Class 10: Bind, Commands, and Utilities

IFCID	Event
0023	Utility start
0024	Utility phase change
0025	Utility end
0090	DB2 command text
0091	DB2 command completion
0105	Buffer manager mapping of DBIDs and OBIDs
0106	Systems parameter in effect
0107	Buffer manager OPEN/CLOSE activity
0108/0109	Begin/End BIND/REBIND
0110/0111	Begin/End FREE
0201	Status of a buffer pool before/after ALTER BUFFERPOOL
0219	Information about the use of a utility LISTDEF list
0220	Utility output data set information
0256	Attributes of group buffer pool before/after ALTER GROUP BUFFERPOOL
0360	Incremental rebind: pre-DB2 10 parallelism

Class 11: Dispatching

IFCID	Event
0046	Begin execution unit switch
0047/0048	Begin/End new SRB
0049/0050	Begin/End new TCB
0051/0052	Resume/Wait shared latch
0056/0057	Wait/Resume exclusive latch
0093	Suspend called
0094	Event resumed
0106	Systems parameters in effect at trace invocation
0113	Plan attributes for successful system agent allocation

Class 12: Storage Manager

IFCID	Event
0098/0099	Begin/End GETMAIN/FREEMAIN (nonpool)
0100/0101	Begin/End GETMAIN/FREEMAIN (pool related)
0106	Systems parameter in effect

Class 13: Edit and Validation Exits

IFCID	Event
0011	Results of a validation exit call
0012	Results edit exit call – Encode
0019	Results edit exit call – Decode
0105	Buffer manager mapping of DBIDs and OBIDs
0106	Systems parameter in effect at trace invocation
0107	Buffer manager OPEN/CLOSE activity

Class 14: In and Out of DB2

IFCID	Event
0067	Start of accounting collection
0106	Systems parameter in effect
0121/0122	Entry/Exit allocating DB2 connection

Class 15: Installation-Defined Performance Record

IFCID	Event
0154	User-defined routine condition performance trace

Class 16: Distributed Processing

IFCID	Event
0157	Type of request received at the requesting location
0158	Type of request received at the serving location
0159	Create conversation and wait requests information at requesting location
0160	Requesting agent data as measured at the interface between DB2 and network
0161	Serving agent data as measured at the interface between DB2 and network
0162	Actions taken by the requesting location in support of distributed processing
0163	Creation and termination of database access threads information
0167	Information about conversation requests that are queued
0183	Type of request being processed at the requester

Class 17: Claim and Drain Detail

IFCID	Event
0211	Information about claims
0212	Information about drains
0213/0214	Begin/End of a wait for a drain lock
0215/0216	Begin/End of a wait for the number of claims to go to zero

Class 18: Event-based Console Messages

IFCID	Event
0197	DB2 message monitoring

Class 19: Data Set Open and Close Activity

IFCID	Event
0370	Open TRACE data sets
0371	Close TRACE data sets

Class 20: Data Sharing Coherency Summary

IFCID	Event
0249	EDM pool DBD invalidation
0250	Group buffer pool connect/disconnect
0251	P-Lock operations
0256	Attributes of a group buffer pool before/after ALTER GROUP BUFFERPOOL
0257	Details of an IRLM notify request

IFCID	Event
0261	Information about group buffer pool checkpoints
0262	Information about castout threshold processing for group bufferpool
0267/0268	Begin/End of coupling facility structure rebuild/explanation/contraction

Class 21: Data Sharing Coherency Detail

IFCID	Event
0255	Buffer refresh caused by cross-invalidation of a data page
0259	P-Lock request or P-Lock negotiation request
0263	Page set and partition castout detail
0329	Asynchronous group buffer request

Class 22: Authorization Exit Parameters

IFCID	Event
0314	Authorization exit parameters

Class 23: Language Environment Run-time Diagnostics

IFCID	Event
0327	Language environment run-time information

Class 24: Stored Procedure Detail

IFCID	Event
0380	Stored procedure detail information
0499	Information about unique statement IDs in stored procedures

Class 25-29: Reserved

Class 30-32: Available for Local Use

Statistics Trace Trace Records

Class 1: Statistical Data

IFCID	Event
0001	System services
0002	Database services
0105	Buffer manager mapping of DBIDs and OBID
0106	System parameters in effect
0202	Buffer pool attributes
0225	System storage usage

Class 2: Installation Defined Statistics Record

IFCID	Event
0152	User-defined statistics trace

Class 3: Deadlock and Lock Timeout Information

IFCID	Event
0172	Units of work involved in deadlock
0196	Lock timeout details
0250	Connect/Disconnect from group buffer pool
0258	Data set extend information
0261	Group buffer pool checkpoint
0262	Group buffer pool castout
0313	Inflight URs and indoubt URs
0330	Active log shortage
0335	Stalled system event notification
0337	Lock escalation

Class 4: DB2 Exception Conditions

IFCID	Event
0173	Dynamic SQL exceeding ASUTIME
0191	Data capture for DDIS errors
0192	DDM level 6A header errors
0193	UOW disposition/sqlcode mismatch
0194	Invalid SNA FMH-5 received
0195	First failure data capture for DRDS
0203	Heuristic decision occurred
0204	Partner cold start detected
0205	Incorrect logname/sync point parameters
0206	SNA compare states protocol error
0207	Heuristic damage occurred
0208	SNA sync point protocol error
0209	Sync point communication failure
0210	Log name changed on warm start
0235	Conditional restart data loss
0236	Exchange log names protocol error
0238	DB2 restart error
0267/0268	Start/End of CF structure rebuild
0343	MAXTEMPs zparm limit exceeded
0402	Profile thresholds exceeded

Class 5: Data Sharing Global Statistics

IFCID	Event
0230	Data sharing global statistics

Class 6: Storage Statistics

IFCID	Event
0225	System storage usage

Class 7: DRDA Location Statistics

IFCID	Event
0365	Location statistics

Class 8: Buffer Pool Data Set Statistics

IFCID	Event
0199	Buffer pool data set statistics
0389	Indexes with fast index access structures

Class 9: Aggregated CPU and Wait time Statistics By Connection Type

IFCID	Event
0369	Aggregated accounting by connection type

7. RETURN CODES

SQL Codes

SQL Code	Explanation	SQL State
+000	SUCCESSFUL EXECUTION	00000 / 01ddd
–007	STATEMENT CONTAINS THE ILLEGAL CHARACTER *invalid-character*	42601
–010	STRING CONSTANT BEGINNING *string* IS NOT TERMINATED	42603
–011	COMMENT NOT CLOSED	42601
+020	BIND, REBIND, OR PRECOMPILE OPTION *option-name* IS NOT SUPPORTED BY THE TARGET SERVER AND WILL BE IGNORED	01615
–029	INTO CLAUSE REQUIRED	42601
–051	*identifier-name* **(***sqltype***)** WAS PREVIOUSLY DECLARED OR REFERENCED	3C000
–056	AN SQLSTATE OR SQLCODE VARIABLE DECLARATION IS IN A NESTED COMPOUND STATEMENT	42630
–057	THE RETURN STATEMENT IN AN SQL FUNCTION MUST RETURN A VALUE	42631
–058	VALUE SPECIFIED ON RETURN STATEMENT MUST BE AN INTEGER	428F2
–060	INVALID *specification-type* SPECIFICATION: *specification-value*	42815
–078	PARAMETER NAMES MUST BE SPECIFIED FOR ROUTINE *routine-name*	42629
–079	QUALIFIER FOR OBJECT name WAS SPECIFIED AS *qualifier 1* BUT *qualifier 2* IS REQUIRED	428EK
–084	UNACCEPTABLE SQL STATEMENT	42612
–087	A NULL VALUE WAS SPECIFIED IN A CONTEXT WHERE A NULL IS NOT ALLOWED	22004
–096	VARIABLE *variable-name* DOES NOT EXIST OR IS NOT SUPPORTED BY THE SERVER AND A DEFAULT VALUE WAS NOT PROVIDED	42704
–097	THE USE OF LONG VARCHAR OR LONG VARGRAPHIC IS NOT ALLOWED IN THIS CONTEXT	42601
+098	A DYNAMIC SQL STATEMENT ENDS WITH A SEMICOLON	01568
+100	ROW NOT FOUND FOR FETCH, UPDATE OR DELETE, OR THE RESULT OF A QUERY IS AN EMPTY TABLE	02000
–101	THE STATEMENT IS TOO LONG OR TOO COMPLEX	54001
–102	STRING CONSTANT IS TOO LONG. STRING BEGINS *string*	54002
–103	*constant* IS AN INVALID NUMERIC CONSTANT	42604
–104	ILLEGAL SYMBOL "*token*". SOME SYMBOLS THAT MIGHT BE LEGAL ARE: *token-list*	42601
–105	INVALID STRING	42604
–107	THE NAME *name-value* IS TOO LONG. MAXIMUM ALLOWABLE SIZE IS *size*	46002 or 42622
–108	THE NAME *name* is QUALIFIED INCORRECTLY	42601
–109	*clause-type* CLAUSE IS NOT PERMITTED	42601
–110	INVALID HEXADECIMAL CONSTANT BEGINNING *constant*	42606
+111	THE SUBPAGES OPTION IS NOT SUPPORTED FOR TYPE 2 INDEXES	01590
–112	THE OPERAND OF AN AGGREGATE FUNCTION INCLUDES AN AGGREGATE FUNCTION, AN OLAP SPECIFICATION, OR A SCALAR FULLSELECT	42607
–113	INVALID CHARACTER FOUND IN: *string*, REASON CODE *nnn*	42602
–114	THE LOCATION NAME *location* DOES NOT MATCH THE CURRENT SERVER	42961
–115	A PREDICATE IS INVALID BECAUSE THE COMPARISON OPERATOR *operator* IS FOLLOWED BY A PARENTHESIZED LIST OR BY ANY OR ALL WITHOUT A SUBQUERY	42601
+117	THE NUMBER OF INSERT VALUES IS NOT THE SAME AS THE NUMBER OF OBJECT COLUMNS	01525
–117	THE NUMBER OF VALUES ASSIGNED IS NOT THE SAME AS THE NUMBER OF SPECIFIED OR IMPLIED COLUMNS	42802
–118	THE OBJECT TABLE OR VIEW OF THE DELETE OR UPDATE STATEMENT IS ALSO IDENTIFIED IN A FROM CLAUSE	42902
–119	A COLUMN OR EXPRESSION IN A HAVING CLAUSE IS NOT VALID	42803

SQL Code	Explanation	SQL State
–120	AN AGGREGATE FUNCTION OR OLAP SPECIFICATION IS NOT VALID IN THE CONTEXT IN WHICH IT WAS INVOKED	42903
–121	THE TARGET *name* IS IDENTIFIED MORE THAN ONCE FOR ASSIGNMENT IN THE SAME SQL STATEMENT	42701
–122	COLUMN OR EXPRESSION IN THE SELECT LIST IS NOT VALID	42803
–123	THE PARAMETER IN POSITION *n* IN THE FUNCTION *name* MUST BE A CONSTANT OR KEYWORD	42601
–125	AN INTEGER IN THE ORDER BY CLAUSE DOES NOT IDENTIFY A COLUMN OF THE RESULT	42805
–126	THE SELECT STATEMENT CONTAINS BOTH AN UPDATE CLAUSE AND AN ORDER BY CLAUSE	42829
–127	DISTINCT IS SPECIFIED MORE THAN ONCE IN A SUBSELECT	42905
–128	INVALID USE OF NULL IN A PREDICATE	42601
–129	THE STATEMENT CONTAINS TOO MANY TABLE NAMES	54004
–130	THE ESCAPE CLAUSE CONSISTS OF MORE THAN ONE CHARACTER, OR THE STRING PATTERN CONTAINS AN INVALID OCCURRENCE OF THE ESCAPE CHARACTER	22019 or 22025
–131	STATEMENT WITH LIKE PREDICATE HAS INCOMPATIBLE DATA TYPES	42818
–132	AN OPERAND OF value IS NOT VALID	42824
–133	AN AGGREGATE FUNCTION IN A SUBQUERY OF A HAVING CLAUSE IS INVALID BECAUSE ALL COLUMN REFERENCES IN ITS ARGUMENT ARE NOT CORRELATED TO THE GROUP BY RESULT THAT THE HAVING CLAUSE IS APPLIED TO	42906
–134	IMPROPER USE OF STRING, LOB, OR XML VALUE	42907
–136	SORT CANNOT BE EXECUTED BECAUSE THE SORT KEY LENGTH IS TOO LONG	54005
–137	THE LENGTH RESULTING FROM operation IS GREATER THAN maximum-length	54006
–138	THE SECOND OR THIRD ARGUMENT OF THE SUBSTR OR SUBSTRING FUNCTION IS OUT OF RANGE	22011
–142	THE SQL STATEMENT IS NOT SUPPORTED	42612
–144	INVALID SECTION NUMBER *number*	58003
–147	ALTER FUNCTION *function-name* FAILED BECAUSE SOURCE FUNCTIONS OR SPATIAL FUNCTION CANNOT BE ALTERED	42917
–148	THE SOURCE TABLE OR TABLESPACE *source-name* CANNOT BE ALTERED, REASON *reason-code*	42809
–150	THE OBJECT OF THE INSERT, DELETE, UPDATE, MERGE OR TRUNCATE STATEMENT IS A VIEW, SYSTEM-MAINTAINED MATERIALIZED QUERY TABLE, OR TRANSITION TABLE FOR WHICH THE REQUESTED OPERATION IS NOT PERMITTED	42807
–151	THE UPDATE STATEMENT IS INVALID BECAUSE THE CATALOG DESCRIPTION OF COLUMN *column-name* INDICATES THAT IT CANNOT BE UPDATED	42808
–152	THE DROP *clause* CLAUSE IN THE ALTER STATEMENT IS INVALID BECAUSE *constraint-name* IS A *constraint-type*	42809
–153	THE STATEMENT IS INVALID BECAUSE THE VIEW OR TABLE DEFINITION DOES NOT INCLUDE A UNIQUE NAME FOR EACH COLUMN	42908
–154	THE STATEMENT FAILED BECAUSE VIEW OR TABLE DEFINITION IS NOT VALID	42909
–156	THE STATEMENT DOES NOT IDENTIFY A TABLE	42809
–157	ONLY A TABLE NAME CAN BE SPECIFIED IN A FOREIGN KEY CLAUSE. *object-name* IS NOT THE NAME OF A TABLE.	42810
–158	THE NUMBER OF COLUMNS SPECIFIED FOR *name* IS NOT THE SAME AS THE NUMBER OF COLUMNS IN THE RESULT TABLE	42811
–159	THE STATEMENT REFERENCES *object-name* WHICH IDENTIFIES AN *object-type* RATHER THAN AN *expected-type*	42809
–160	THE WITH CHECK OPTION CLAUSE IS NOT VALID FOR THE SPECIFIED VIEW	42813
–161	THE INSERT OR UPDATE IS NOT ALLOWED BECAUSE A RESULTING ROW DOES NOT SATISFY THE VIEW DEFINITION	44000
–164	*authorization-id* DOES NOT HAVE THE PRIVILEGE TO CREATE A VIEW WITH QUALIFICATION *qualifier-name*	42502
–170	THE NUMBER OF ARGUMENTS SPECIFIED FOR *function-name* IS INVALID	42605
–171	THE DATA TYPE, LENGTH, OR VALUE OF ARGUMENT *argument-position* OF *function-name* IS INVALID	22546 or 42815
–173	UR IS SPECIFIED ON THE WITH CLAUSE BUT THE CURSOR IS NOT READ-ONLY	42801
–180	THE DATE, TIME, OR TIMESTAMP VALUE *value* IS INVALID	22007
–181	THE STRING REPRESENTATION OF A DATETIME VALUE IS NOT A VALID DATETIME VALUE	22007
–182	AN ARITHMETIC EXPRESSION WITH A DATETIME VALUE IS	42816

SQL Code	Explanation	SQL State
	INVALID	
–183	AN ARITHMETIC OPERATION ON A DATE OR TIMESTAMP HAS A RESULT THAT IS NOT WITHIN THE VALID RANGE OF DATES	22008
–184	AN ARITHMETIC EXPRESSION WITH A DATETIME VALUE CONTAINS A PARAMETER MARKER	42610
–185	THE LOCAL FORMAT OPTION HAS BEEN USED WITH A DATE OR TIME AND NO LOCAL EXIT HAS BEEN INSTALLED	57008
–186	THE LOCAL DATE LENGTH OR LOCAL TIME LENGTH HAS BEEN INCREASED AND EXECUTING PROGRAM RELIES ON THE OLD LENGTH	22505
–187	A REFERENCE TO A CURRENT DATETIME SPECIAL REGISTER IS INVALID BECAUSE THE MVS TOD CLOCK IS BAD OR THE MVS PARMTZ IS OUT OF RANGE	22506
–188	THE STRING REPRESENTATION OF A NAME IS INVALID	22503
–189	CCSID *ccsid I*S INVALID	22522
–190	THE ATTRIBUTES SPECIFIED FOR THE COLUMN *table-name.column-name* ARE NOT COMPATIBLE WITH THE EXISTING COLUMN DEFINITION	42837
–191	A STRING CANNOT BE USED BECAUSE IT IS INVALID MIXED DATA	22504
–195	LAST COLUMN OF *table-name* CANNOT BE DROPPED	42814
–196	COLUMN *table-name.column-name* CANNOT BE DROPPED. REASON = *reason-code*	42817
–197	A QUALIFIED COLUMN NAME IS NOT ALLOWED IN THE ORDER BY CLAUSE WHEN A SET OPERATOR IS ALSO SPECIFIED	42877
–198	THE OPERAND OF THE PREPARE OR EXECUTE IMMEDIATE STATEMENT IS BLANK OR EMPTY	42617
–199	ILLEGAL USE OF KEYWORD *keyword*. TOKEN *token-list* WAS EXPECTED	42601
+203	THE QUALIFIED COLUMN NAME *column-name* WAS RESOLVED USING A NON-UNIQUE OR UNEXPOSED NAME	01552
–203	A REFERENCE TO COLUMN *column-name* IS AMBIGUOUS	42702
+204	*name* IS AN UNDEFINED NAME	01532
–204	*name* IS AN UNDEFINED NAME	42704
+205	*column-name* IS NOT A COLUMN OF TABLE *table-name*	01533
–205	column-name IS NOT A COLUMN OF TABLE *table-name*	42703
+206	column-name IS NOT A COLUMN OF AN INSERTED TABLE, UPDATED TABLE, MERGED TABLE, OR ANY TABLE IDENTIFIED IN A FROM CLAUSE	01533
–206	object-name IS NOT VALID IN THE CONTEXT WHERE IT IS USED	42703
–208	THE ORDER BY CLAUSE IS INVALID BECAUSE COLUMN *name* IS NOT PART OF THE RESULT TABLE	42707
–212	*name* IS SPECIFIED MORE THAN ONCE IN THE REFERENCING CLAUSE OF A TRIGGER DEFINITION	42712
–214	AN EXPRESSION IN THE FOLLOWING POSITION, OR STARTING WITH *position-or-expression-start* IN THE *clause-type* CLAUSE IS NOT VALID. REASON CODE = *reason code*	42822
–216	THE NUMBER OF ELEMENTS ON EACH SIDE OF A PREDICATE OPERATOR DOES NOT MATCH. PREDICATE OPERATOR IS *operator*	428C4
+217	THE STATEMENT WAS NOT EXECUTED AS ONLY EXPLAIN INFORMATION REQUESTS ARE BEING PROCESSED	01604
+218	THE SQL STATEMENT REFERENCING A REMOTE OBJECT CANNOT BE EXPLAINED	01537
+219	THE REQUIRED EXPLANATION TABLE *table-name* DOES NOT EXIST	01532
–219	THE REQUIRED EXPLANATION TABLE *table-name* DOES NOT EXIST	42704
+220	THE COLUMN *column-name* IN EXPLANATION TABLE *table-name* IS NOT DEFINED PROPERLY	01546
–220	THE COLUMN *column-name* IN EXPLANATION TABLE *table-name* IS NOT DEFINED PROPERLY	55002
–221	"SET OF OPTIONAL COLUMNS" IN EXPLANATION TABLE *table-name* IS INCOMPLETE. OPTIONAL COLUMN *column-name* IS MISSING	55002
+222	HOLE DETECTED USING *cursor-name*	02502
–222	AN UPDATE OR DELETE OPERATION WAS ATTEMPTED AGAINST A HOLE USING *cursor-name*	24510
–224	THE RESULT TABLE DOES NOT AGREE WITH THE BASE TABLE USING *cursor-name*	24512
–225	FETCH STATEMENT FOR *cursor-name* IS NOT VALID FOR THE DECLARATION OF THE CURSOR	42872

SQL Code	Explanation	SQL State
–227	FETCH *fetch-orientation* IS NOT ALLOWED, BECAUSE CURSOR *cursor-name* HAS AN UNKNOWN POSITION (*sqlcode,sqlstate*)	24513
–228	FOR UPDATE CLAUSE SPECIFIED FOR READ-ONLY CURSOR *cursor-name*	42620
–229	THE LOCALE *locale* SPECIFIED IN A SETLC_CTYPE OR OTHER STATEMENT THAT IS LOCALE SENSITIVE WAS NOT FOUND	42708
+231	CURRENT POSITION OF CURSOR *cursor-name* IS NOT VALID FOR THE SPECIFIED FETCH ORIENTATION OF THE CURRENT ROW OR ROWSET	02000
+236	SQLDA INCLUDES *integer1* SQLVAR ENTRIES, BUT *integer2* ARE REQUIRED FOR *integer3* COLUMNS	01005
+237	SQLDA INCLUDES *integer1* SQLVAR ENTRIES, BUT *integer2* ARE REQUIRED BECAUSE AT LEAST ONE OF THE COLUMNS BEING DESCRIBED IS A DISTINCT TYPE	01594
+238	SQLDA INCLUDES *integer1* SQLVAR ENTRIES, BUT *integer2* SQLVAR ENTRIES ARE REQUIRED FOR *integer3* COLUMNS BECAUSE AT LEAST ONE OF THE COLUMNS BEING DESCRIBED IS A LOB	01005
+239	SQLDA INCLUDES *integer1* SQLVAR ENTRIES, BUT *integer2* ARE REQUIRED FOR *integer3* COLUMNS BECAUSE AT LEAST ONE OF THE COLUMNS BEING DESCRIBED IS A DISTINCT TYPE	01005
–240	THE PARTITION CLAUSE OF A LOCK TABLE STATEMENT IS INVALID	428B4
–242	THE OBJECT NAMED *object-name* OF TYPE *object-type* WAS SPECIFIED MORE THAN ONCE IN THE LIST OF OBJECTS, OR THE NAME IS THE SAME AS AN EXISTING OBJECT	42713
–243	SENSITIVE CURSOR *cursor-name* CANNOT BE DEFINED FOR THE SPECIFIED SELECT STATEMENT	36001
–244	SENSITIVITY *sensitivity* SPECIFIED ON THE FETCH IS NOT VALID FOR CURSOR *cursor-name*	428F4
–245	THE INVOCATION OF FUNCTION *routine-name* IS AMBIGUOUS	428F5
–246	STATEMENT USING CURSOR *cursor-name* SPECIFIED NUMBER OF ROWS *num-rows* WHICH IS NOT VALID WITH *dimension*	42873
–247	A HOLE WAS DETECTED ON A MULTIPLE ROW FETCH STATEMENT USING CURSOR *cursor-name*, BUT INDICATOR VARIABLES WERE NOT PROVIDED TO DETECT THE CONDITION	24519
–248	A POSITIONED DELETE OR UPDATE STATEMENT FOR CURSOR *cursor-name* SPECIFIED ROW *n* OF A ROWSET, BUT THE ROW IS NOT CONTAINED WITHIN THE CURRENT ROWSET	24521
–249	DEFINITION OF ROWSET ACCESS FOR CURSOR *cursor-name* IS INCONSISTENT WITH THE FETCH ORIENTATION CLAUSE *clause* SPECIFIED	24523
–250	THE LOCAL LOCATION NAME IS NOT DEFINED WHEN PROCESSING A THREE-PART OBJECT NAME	42718
–251	TOKEN *name* IS NOT VALID	42602
+252	A NON-ATOMIC *statement* STATEMENT SUCCESSFULLY PROCESSED ALL REQUESTED ROWS, WITH ONE OR MORE WARNING CONDITIONS	01659
–253	A NON-ATOMIC *statement* STATEMENT SUCCESSFULLY COMPLETED FOR SOME OF THE REQUESTED ROWS, POSSIBLY WITH WARNINGS, AND ONE OR MORE ERRORS	22529
–254	A NON-ATOMIC *statement* STATEMENT ATTEMPTED TO PROCESS MULTIPLE ROWS OF DATA, BUT ERRORS OCCURRED	22530
–270	FUNCTION NOT SUPPORTED	42997
–300	THE STRING CONTAINED IN HOST VARIABLE OR PARAMETER *position-number* IS NOT NUL-TERMINATED	22024
–301	THE VALUE OF INPUT HOST VARIABLE OR PARAMETER NUMBER *position-number* CANNOT BE USED AS SPECIFIED BECAUSE OF ITS DATA TYPE	42895
–303	A VALUE CANNOT BE ASSIGNED TO VARIABLE NUMBER *position-number* BECAUSE THE DATA TYPES ARE NOT COMPARABLE	42806
+304	A VALUE WITH DATA TYPE *data-type1* CANNOT BE ASSIGNED TO A HOST VARIABLE BECAUSE THE VALUE IS NOT WITHIN THE RANGE OF THE HOST VARIABLE IN POSITION *position-number* WITH DATA TYPE *data-type2*	01515
–304	A VALUE WITH DATA TYPE *data-type1* CANNOT BE ASSIGNED TO A HOST VARIABLE BECAUSE THE VALUE IS NOT WITHIN THE RANGE OF THE HOST VARIABLE IN	22003

SQL Code	Explanation	SQL State
	POSITION *position-number* WITH DATA TYPE *data-type2*	
–305	THE NULL VALUE CANNOT BE ASSIGNED TO OUTPUT HOST VARIABLE NUMBER *position-number* BECAUSE NO INDICATOR VARIABLE IS SPECIFIED	22002
–309	A PREDICATE IS INVALID BECAUSE A REFERENCED HOST VARIABLE HAS THE NULL VALUE	22512
–311	THE LENGTH OF INPUT HOST VARIABLE NUMBER *position-number* IS NEGATIVE OR GREATER THAN THE MAXIMUM	22501
–312	VARIABLE *variable-name* IS NOT DEFINED OR NOT USABLE	42618
–313	THE NUMBER OF HOST VARIABLES SPECIFIED IS NOT EQUAL TO THE NUMBER OF PARAMETER MARKERS	07001
–314	THE STATEMENT CONTAINS AN AMBIGUOUS HOST VARIABLE REFERENCE	42714
–327	THE ROW CANNOT BE INSERTED BECAUSE IT IS OUTSIDE THE BOUND OF THE PARTITION RANGE FOR THE LAST PARTITION	22525
–330	A STRING CANNOT BE USED BECAUSE IT CANNOT BE CONVERTED. REASON *reason-code*, CHARACTER *code-point*, HOST VARIABLE *position-number*	22021
+331	THE NULL VALUE HAS BEEN ASSIGNED TO A HOST VARIABLE OR PARAMETER BECAUSE THE STRING CANNOT BE CONVERTED FROM *source-ccsid* TO *target-ccsid.* REASON *reason-code*, POSITION *position-number*	01520
–331	CHARACTER CONVERSION CANNOT BE PERFORMED BECAUSE A STRING, POSITION *position-number* CANNOT BE CONVERTED FROM *source-ccsid* TO *target-ccsid*, REASON *reason-code*	22021
–332	CHARACTER CONVERSION BETWEEN CCSID *from-ccsid* TO *to-ccsid* REQUESTED BY *reason-code* IS NOT SUPPORTED	57017
–333	THE SUBTYPE OF A STRING VARIABLE IS NOT THE SAME AS THE SUBTYPE KNOWN AT BIND TIME AND THE DIFFERENCE CANNOT BE RESOLVED BY CHARACTER CONVERSION	56010
+335	DB2 CONVERTED A HOST VARIABLE, PARAMETER, OR COLUMN NUMBER *var-num var-name-or-num* TO COLUMN NAME, HOST VARIABLE, OR EXPRESSION NUMBER *col-name-or-num* FROM *from-ccsid* TO *to-ccsid*, AND RESULTING IN SUBSTITUTION CHARACTERS.	01517
–336	THE SCALE OF THE DECIMAL NUMBER MUST BE ZERO	428FA
–338	AN ON CLAUSE IS INVALID	42972
–340	THE COMMON TABLE EXPRESSION *name* HAS THE SAME IDENTIFIER AS ANOTHER OCCURRENCE OF A COMMON TABLE EXPRESSION DEFINITION WITHIN THE SAME STATEMENT	42726
–341	A CYCLIC REFERENCE EXISTS BETWEEN THE COMMON TABLE EXPRESSIONS *name1* AND *name2*	42835
–342	THE COMMON TABLE EXPRESSION *name* MUST NOT USE SELECT DISTINCT AND MUST USE UNION ALL BECAUSE IT IS RECURSIVE	42925
–343	THE COLUMN NAMES ARE REQUIRED FOR THE RECURSIVE COMMON TABLE EXPRESSION *name*	42908
–344	THE RECURSIVE COMMON TABLE EXPRESSION *name* HAS MISMATCHED DATA TYPES OR LENGTHS OR CODE PAGE FOR COLUMN *column-name*	42825
–345	THE FULLSELECT OF THE RECURSIVE COMMON TABLE EXPRESSION *name* MUST BE A UNION ALL AND MUST NOT INCLUDE AGGREGATE FUNCTIONS, GROUP BY, HAVING, ORDER BY, OFFSET, FETCH FIRST, OR AN EXPLICIT JOIN INCLUDING AN ON CLAUSE	42836
–346	AN INVALID REFERENCE TO COMMON TABLE EXPRESSION *name* OCCURS IN THE FIRST FULLSELECT, AS A SECOND OCCURRENCE IN THE SAME FROM CLAUSE, OR IN THE FROM CLAUSE OF A SUBQUERY	42836
+347	THE RECURSIVE COMMON TABLE EXPRESSION *name* MAY CONTAIN AN INFINITE LOOP	01605
–348	*sequence-expression* CANNOT BE SPECIFIED IN THIS CONTEXT	428F9
–350	*column-name* WAS IMPLICITLY OR EXPLICITLY REFERENCED IN A CONTEXT IN WHICH IT CANNOT BE USED	42962
–351	AN UNSUPPORTED SQLTYPE WAS ENCOUNTERED IN POSITION *position-number* OF THE SELECT-LIST	56084
–352	AN UNSUPPORTED SQLTYPE WAS ENCOUNTERED IN POSITION *position-number* OF THE INPUT-LIST	56084
–353	FETCH IS NOT ALLOWED, BECAUSE CURSOR *cursor-name* HAS AN UNKNOWN POSITION	24513
+354	A ROWSET FETCH STATEMENT MAY HAVE RETURNED ONE OR MORE ROWS OF DATA. HOWEVER, ONE OR MORE	01668

SQL Code	Explanation	SQL State
	WARNING CONDITIONS WERE ALSO ENCOUNTERED. USE THE GET DIAGNOSTICS STATEMENT FOR MORE INFORMATION REGARDING THE CONDITIONS THAT WERE ENCOUNTERED.	
–354	A ROWSET FETCH STATEMENT MAY HAVE RETURNED ONE OR MORE ROWS OF DATA. HOWEVER, ONE OR MORE NON-TERMINATING ERROR CONDITIONS WERE ENCOUNTERED. USE THE GET DIAGNOSTICS STATEMENT FOR MORE INFORMATION REGARDING THE CONDITIONS THAT WERE ENCOUNTERED.	22537
–355	A LOB COLUMN IS TOO LARGE TO BE LOGGED	42993
–356	KEY EXPRESSION *expression-number* IS NOT VALID, REASON CODE = *reason-code*	429BX
–359	THE RANGE OF VALUES FOR THE IDENTITY COLUMN OR SEQUENCE IS EXHAUSTED	23522
+361	COMMAND WAS SUCCESSFUL BUT RESULTED IN THE FOLLOWING: *msg-token*	0168B
–363	THE EXTENDED INDICATOR VARIABLE VALUE FOR PARAMETER *position-number* IS OUT OF RANGE	22010
+364	DECFLOAT EXCEPTION *exception-type* HAS OCCURRED DURING *operation-type* OPERATION, POSITION *position-number*	0168C 0168D 0168E 0168F 0168G
–365	USE OF THE VALUE OF EXTENDED INDICATOR VARIABLE IN POSITION *value-position* IS NOT VALID	22539
–372	ONLY ONE ROWID, IDENTITY, ROW CHANGE TIMESTAMP, ROW BEGIN, ROW END, TRANSACTION START ID, SECURITY LABEL OR DATA CHANGE OPERATION COLUMN IS ALLOWED IN A TABLE	428C1
–373	DEFAULT CANNOT BE SPECIFIED FOR IDENTITY COLUMN OR SQL VARIABLE *name*	42623
–374	THE CLAUSE *clause* HAS NOT BEEN SPECIFIED IN THE CREATE OR ALTER FUNCTION STATEMENT FOR LANGUAGE SQL FUNCTION *function-name* BUT AN EXAMINATION OF THE FUNCTION BODY REVEALS THAT IT SHOULD BE SPECIFIED	428C2
+385	ASSIGNMENT TO AN SQLSTATE OR SQLCODE VARIABLE IN AN SQL ROUTINE *routine-name* MAY BE OVERWRITTEN AND DOES NOT ACTIVATE ANY HANDLER	01643
–390	THE OBJECT *object-name*, SPECIFIC NAME *specific-name*, IS NOT VALID IN THE CONTEXT WHERE IT IS USED	42887
–392	SQLDA PROVIDED FOR CURSOR *cursor-name* HAS BEEN CHANGED FROM THE PREVIOUS FETCH (*reason-code*)	42855
–393	THE CONDITION OR CONNECTION NUMBER IS INVALID	35000
+394	ALL USER-SPECIFIED OPTIMIZATION HINTS USED DURING ACCESS PATH SELECTION	01629
+395	A USER SPECIFIED OPTIMIZATION HINTS ARE INVALID (REASON CODE = '*reason-code*').	01628
–396	*object-type object-name* ATTEMPTED TO EXECUTE AN SQL STATEMENT DURING FINAL CALL PROCESSING	38505
–397	GENERATED IS SPECIFIED AS PART OF A COLUMN DEFINITION, BUT IT IS NOT VALID FOR THE DEFINITION OF THE COLUMN	428D3
–398	A LOCATOR WAS REQUESTED FOR HOST VARIABLE NUMBER *position-number* BUT THE VARIABLE IS NOT A LOB	428D2
–399	INVALID VALUE ROWID WAS SPECIFIED	22511
–401	THE DATA TYPES OF THE OPERANDS OF AN OPERATION ARE NOT COMPARABLE	42818
+402	LOCATION *location* IS UNKNOWN	01521
–402	AN ARITHMETIC FUNCTION OR OPERATOR *function-operator* IS APPLIED TO CHARACTER OR DATETIME DATA	42819
+403	THE LOCAL OBJECT REFERENCED BY THE CREATE ALIAS STATEMENT DOES NOT EXIST	01522
–404	THE SQL STATEMENT SPECIFIES A STRING THAT IS TOO LONG	22001
–405	THE NUMERIC CONSTANT *constant* CANNOT BE USED AS SPECIFIED BECAUSE IT IS OUT OF RANGE	42820
–406	A CALCULATED OR DERIVED NUMERIC VALUE IS NOT WITHIN THE RANGE OF ITS OBJECT COLUMN	22003
–407	AN UPDATE, INSERT, OR SET VALUE IS NULL, BUT THE OBJECT COLUMN *column-name* CANNOT CONTAIN NULL VALUES	23502
–408	THE VALUE IS NOT COMPATIBLE WITH THE DATA TYPE OF ITS TARGET. TARGET NAME IS *name*	42821
–409	INVALID OPERAND OF A COUNT FUNCTION	42607

SQL Code	Explanation	SQL State
–410	A NUMERIC VALUE *value* IS TOO LONG, OR IT HAS A VALUE THAT IS NOT WITHIN THE RANGE OF ITS DATA TYPE	42820
–412	THE SELECT CLAUSE OF A SUBQUERY SPECIFIES MULTIPLE COLUMNS	42823
–413	OVERFLOW OR UNDERFLOW OCCURRED DURING NUMERIC DATA TYPE CONVERSION	22003
–414	A LIKE PREDICATE IS INVALID BECAUSE THE FIRST OPERAND IS NOT A STRING	42824
–415	THE CORRESPONDING COLUMNS, *column-number*, OF THE OPERANDS OF A SET OPERATOR ARE NOT COMPATIBLE	42825
–416	AN OPERAND OF A SET OPERATOR CONTAINS A LONG STRING COLUMN	42907
–417	A STATEMENT STRING TO BE PREPARED INCLUDES PARAMETER MARKERS AS THE OPERANDS OF THE SAME OPERATOR	42609
–418	A STATEMENT STRING TO BE PREPARED CONTAINS AN INVALID USE OF PARAMETER MARKERS	42610
–419	THE DECIMAL DIVIDE OPERATION IS INVALID BECAUSE THE RESULT WOULD HAVE A NEGATIVE SCALE	42911
–420	THE VALUE OF A STRING ARGUMENT WAS NOT ACCEPTABLE TO THE *function-name* FUNCTION	22018
–421	THE OPERANDS OF A SET OPERATOR DO NOT HAVE THE SAME NUMBER OF COLUMNS	42826
–423	INVALID VALUE FOR LOCATOR IN POSITION *position-#*	0F001
–426	DYNAMIC COMMIT NOT VALID AT AN APPLICATION SERVER WHERE UPDATES ARE NOT ALLOWED	2D528
–427	DYNAMIC ROLLBACK NOT VALID AT AN APPLICATION SERVER WHERE UPDATES ARE NOT ALLOWED	2D529
–430	*routine-type routine-name* (SPECIFIC NAME *specific-name*) HAS ABNORMALLY TERMINATED	38503
–431	ROUTINE *routine-name* (SPECIFIC NAME *specific-name*) OF TYPE *routine-type* HAS BEEN INTERRUPTED BY THE USER	38504
–433	VALUE *value* IS TOO LONG	22001
+434	*clause* IS A DEPRECATED CLAUSE	01608
–435	AN INVALID SQLSTATE *sqlstate* IS SPECIFIED IN THE RAISE_ERROR FUNCTION, RESIGNAL STATEMENT, OR SIGNAL STATEMENT	428B3
+438	APPLICATION RAISED WARNING WITH DIAGNOSTIC TEXT: *text*	Application defined
–438	APPLICATION RAISED ERROR WITH DIAGNOSTIC TEXT: *text*	Application defined
+440	NO *routine-type* BY THE NAME *routine-name* HAVING COMPATIBLE ARGUMENTS WAS FOUND	01681
–440	NO AUTHORIZED *routine-type* BY THE NAME *routine-name* HAVING COMPATIBLE ARGUMENTS WAS FOUND	42884
–441	INVALID USE OF 'DISTINCT' OR 'ALL' WITH FUNCTION *function-name*	42601
–443	ROUTINE *routine-name* (SPECIFIC NAME *specific-name*) HAS RETURNED AN ERROR SQLSTATE WITH DIAGNOSTIC TEXT *msg-text*	xxx
–444	USER PROGRAM *name* COULD NOT BE FOUND	42724
+445	VALUE *value* HAS BEEN TRUNCATED	01004
–449	CREATE OR ALTER STATEMENT FOR FUNCTION OR PROCEDURE *routine-name* CONTAINS AN INVALID FORMAT OF THE EXTERNAL NAME CLAUSE OR IS MISSING THE EXTERNAL NAME CLAUSE	42878
–450	USER-DEFINED FUNCTION OR STORED PROCEDURE *name*, PARAMETER NUMBER *parmnum*, OVERLAYED STORAGE BEYOND ITS DECLARED LENGTH	39501
–451	THE *data-item* DEFINITION, IN THE CREATE OR ALTER STATEMENT FOR *routine-name* CONTAINS DATA TYPE *type* WHICH IS NOT SUPPORTED FOR THE TYPE AND LANGUAGE OF THE ROUTINE	42815
–452	UNABLE TO ACCESS THE FILE REFERENCED BY HOST VARIABLE *variable-position*. REASON CODE: *reason-code*	428A1
–453	THERE IS A PROBLEM WITH THE RETURNS CLAUSE IN THE CREATE FUNCTION STATEMENT FOR *function-name*	42880
–454	THE SIGNATURE PROVIDED IN THE CREATE FUNCTION STATEMENT FOR *function-name* MATCHES THE SIGNATURE OF SOME OTHER FUNCTION ALREADY EXISTING IN THE SCHEMA	42723
–455	IN CREATE FUNCTION FOR *function-name*, THE SCHEMA NAME *schema-name1* PROVIDED FOR THE SPECIFIC NAME DOES NOT MATCH THE SCHEMA NAME *schema-name2* OF THE FUNCTION	42882
–456	IN CREATE FUNCTION FOR *function-name*, THE SPECIFIC	42710

SQL Code	Explanation	SQL State
	NAME *specific-name* ALREADY EXISTS IN THE SCHEMA	
–457	A USER-DEFINED FUNCTION OR USER-DEFINED TYPE CANNOT BE CALLED *name* SINCE IT IS RESERVED FOR SYSTEM USE	42939
–458	IN A REFERENCE TO FUNCTION *function-name* BY SIGNATURE, A MATCHING FUNCTION COULD NOT BE FOUND	42883
–461	A VALUE WITH DATA TYPE *source-data-type* CANNOT BE CAST TO TYPE *target-data-type*	42846
+462	EXTERNAL FUNCTION OR PROCEDURE *name* (SPECIFIC NAME *specific-name*) HAS RETURNED A WARNING SQLSTATE, WITH DIAGNOSTIC TEXT *text*	01Hxx
+464	PROCEDURE *proc* RETURNED *num* QUERY RESULT SETS, WHICH EXCEEDS THE DEFINED LIMIT *integer*	0100E
+466	PROCEDURE *proc* RETURNED *num* QUERY RESULTS SETS	0100C
–469	SQL CALL STATEMENT MUST SPECIFY AN OUTPUT HOST VARIABLE FOR PARAMETER *number*	42886
–470	SQL CALL STATEMENT SPECIFIED A NULL VALUE FOR INPUT PARAMETER *number*, BUT THE STORED PROCEDURE DOES NOT SUPPORT NULL VALUES	39004
–471	INVOCATION OF FUNCTION OR PROCEDURE *name* FAILED DUE TO REASON *rc*	55023
–472	CURSOR *cursor-name* WAS LEFT OPEN BY EXTERNAL FUNCTION *function-name* (SPECIFIC NAME *specific-name*)	24517
–473	A USER DEFINED DATA TYPE CANNOT BE CALLED THE SAME NAME AS A SYSTEM PREDEFINED TYPE (BUILT-IN TYPE)	42918
–475	THE RESULT TYPE *type-1* OF THE SOURCE FUNCTION CANNOT BE CAST TO THE RETURNS TYPE *type-2* OF THE USER-DEFINED FUNCTION *function-name*	42866
–476	REFERENCE TO FUNCTION *function-name* WAS NAMED WITHOUT A SIGNATURE, BUT THE FUNCTION IS NOT UNIQUE WITHIN ITS SCHEMA	42725
–478	ALTER, DROP OR REVOKE AFFECTING OBJECT TYPE *object-type* CANNOT BE PROCESSED BECAUSE OBJECT *dependent-object* OF TYPE *dependent-type* IS DEPENDENT ON IT	46003, 4600C or 42893
–480	THE PROCEDURE *procedure-name* HAS NOT YET BEEN CALLED	51030
-481	THE GROUP BY CLAUSE CONTAINS *element-1* NESTED WITHIN *element-2*	428B0
–482	THE PROCEDURE procedure-name RETURNED NO LOCATORS	51030
–483	IN CREATE FUNCTION FOR *function-name* STATEMENT, THE NUMBER OF PARAMETERS DOES NOT MATCH THE NUMBER OF PARAMETERS OF THE SOURCE FUNCTION	42885
–487	*object-type object-name* ATTEMPTED TO EXECUTE AN SQL STATEMENT WHEN THE DEFINITION OF THE FUNCTION OR PROCEDURE DID NOT SPECIFY THIS ACTION	38001
–490	NUMBER *number* DIRECTLY SPECIFIED IN AN SQL STATEMENT IS OUTSIDE THE RANGE OF ALLOWABLE VALUES IN THIS CONTEXT (*minval, maxval*)	428B7
–491	CREATE STATEMENT FOR USER-DEFINED FUNCTION *function-name* MUST HAVE A RETURNS CLAUSE AND: THE EXTERNAL CLAUSE WITH OTHER REQUIRED KEYWORDS; THE PARAMETER NAMES; OR THE SOURCE CLAUSE	42601
–492	THE CREATE FUNCTION FOR *function-name* HAS A PROBLEM WITH PARAMETER NUMBER number. IT MAY INVOLVE A MISMATCH WITH A SOURCE FUNCTION	42879
+494	NUMBER OF RESULT SETS GREATER THAN NUMBER OF LOCATORS	01614
–496	THE SQL STATEMENT CANNOT BE EXECUTED BECAUSE IT REFERENCES A RESULT SET THAT WAS NOT CREATED BY THE CURRENT SERVER	51033
–497	THE MAXIMUM LIMIT OF INTERNAL IDENTIFIERS HAS BEEN EXCEEDED FOR DATABASE *database-name*	54041
–499	CURSOR *cursor-name* HAS ALREADY BEEN ASSIGNED TO THIS OR ANOTHER RESULT SET FROM PROCEDURE *procedure-name*	24516
–500	THE IDENTIFIED CURSOR WAS CLOSED WHEN THE CONNECTION WAS DESTROYED	24501
–501	THE CURSOR IDENTIFIED IN A FETCH OR CLOSE STATEMENT IS NOT OPEN	24501
–502	THE CURSOR IDENTIFIED IN AN OPEN STATEMENT IS ALREADY OPEN	24502
–503	A COLUMN CANNOT BE UPDATED BECAUSE IT IS NOT IDENTIFIED IN THE UPDATE CLAUSE OF THE SELECT STATEMENT OF THE CURSOR	42912

SQL Code	Explanation	SQL State
–504	CURSOR NAME *cursor-name* IS NOT DECLARED	34000
–507	THE CURSOR IDENTIFIED IN THE UPDATE OR DELETE STATEMENT IS NOT OPEN	24501
–508	THE CURSOR IDENTIFIED IN THE UPDATE OR DELETE STATEMENT IS NOT POSITIONED ON A ROW OR ROWSET THAT CAN BE UPDATED OR DELETED	24504
–509	THE TABLE IDENTIFIED IN THE UPDATE OR DELETE STATEMENT IS NOT THE SAME TABLE DESIGNATED BY THE CURSOR	42827
–510	THE TABLE DESIGNATED BY THE CURSOR OF THE UPDATE OR DELETE STATEMENT CANNOT BE MODIFIED	42828
–511	THE FOR UPDATE CLAUSE CANNOT BE SPECIFIED BECAUSE THE RESULT TABLE DESIGNATED BY THE SELECT STATEMENT CANNOT BE MODIFIED	42829
–512	STATEMENT REFERENCE TO REMOTE OBJECT IS INVALID	56023
–513	THE ALIAS *alias-name* MUST NOT BE DEFINED ON ANOTHER LOCAL OR REMOTE ALIAS	42924
–514	THE CURSOR *cursor-name* IS NOT IN A PREPARED STATE	26501
–516	THE DESCRIBE STATEMENT DOES NOT SPECIFY A PREPARED STATEMENT	26501
–517	CURSOR *cursor-name* CANNOT BE USED BECAUSE ITS STATEMENT NAME DOES NOT IDENTIFY A PREPARED SELECT STATEMENT	07005
–518	THE EXECUTE STATEMENT DOES NOT IDENTIFY A VALID PREPARED STATEMENT	07003
–519	THE PREPARE STATEMENT IDENTIFIES THE SELECT STATEMENT OF THE OPENED CURSOR *cursor-name*	24506
–526	THE REQUESTED OPERATION OR USAGE DOES NOT APPLY TO *table-type* TEMPORARY TABLE *table-name*	42995
–530	THE INSERT OR UPDATE VALUE OF FOREIGN KEY *constraint-name* IS INVALID	23503
–531	PARENT KEY IN A PARENT ROW CANNOT BE UPDATED BECAUSE IT HAS ONE OR MORE DEPENDENT ROWS IN RELATIONSHIP *constraint-name*	23504
–532	THE RELATIONSHIP *constraint-name* RESTRICTS THE DELETION OF ROW WITH RID X *rid-number*	23504
–533	INVALID MULTIPLE-ROW INSERT	21501
–534	THE PRIMARY KEY CANNOT BE UPDATED BECAUSE OF MULTIPLE-ROW UPDATE	21502
+535	THE RESULT OF THE POSITIONED UPDATE OR DELETE MAY DEPEND ON THE ORDER OF THE ROWS.	01591
–536	THE DELETE STATEMENT IS INVALID BECAUSE TABLE *table-name* CAN BE AFFECTED BY THE OPERATION	42914
–537	THE PRIMARY KEY, FOREIGN KEY, UNIQUE, OR PARTITIONING KEY CLAUSE IDENTIFIES COLUMN *column-name* MORE THAN ONCE	42709
–538	FOREIGN KEY *name* DOES NOT CONFORM TO THE DESCRIPTION OF A PARENT KEY OF TABLE *table-name*	42830
–539	TABLE *table-name* DOES NOT HAVE A PRIMARY KEY	42888
–540	THE DEFINITION OF TABLE *table-name* IS INCOMPLETE BECAUSE IT LACKS A PRIMARY INDEX OR A REQUIRED UNIQUE INDEX	57001
+541	THE REFERENTIAL OR UNIQUE CONSTRAINT *name* HAS BEEN IGNORED BECAUSE IT IS A DUPLICATE	01543
-541	DUPLICATE TEMPORAL FOREIGN KEY CONSTRAINT EXISTS FOR TABLE *table-name* (*constraint name*)	42891
–542	*column-name* CANNOT BE A COLUMN OF A HASH KEY, PRIMARY KEY, A UNIQUE CONSTRAINT, OR A PARENT KEY BECAUSE IT CAN CONTAIN NULL VALUES	42831
–543	A ROW IN A PARENT TABLE CANNOT BE DELETED BECAUSE THE CHECK CONSTRAINT *check-constraint* RESTRICTS THE DELETION	23511
–544	THE CHECK CONSTRAINT SPECIFIED IN THE ALTER TABLE STATEMENT CANNOT BE ADDED BECAUSE AN EXISTING ROW VIOLATES THE CHECK CONSTRAINT	23512
–545	THE REQUESTED OPERATION IS NOT ALLOWED BECAUSE A ROW DOES NOT SATISFY THE CHECK CONSTRAINT *check-constraint*	23513
–546	THE CHECK CONSTRAINT *constraint-name* IS INVALID	42621
–548	A CHECK CONSTRAINT THAT IS DEFINED WITH *column-name* IS INVALID	42621
–549	THE *statement* STATEMENT IS NOT ALLOWED FOR *object-type1 object-name* BECAUSE THE BIND OPTION DYNAMIC-RULES(RUN) IS NOT IN EFFECT FOR *object-type2*.	42509
+551	*auth-id* DOES NOT HAVE THE PRIVILEGE TO PERFORM OPERATION *operation* ON OBJECT *object-name*	01548

SQL Code	Explanation	SQL State
–551	*auth-id* DOES NOT HAVE THE PRIVILEGE TO PERFORM OPERATION *operation* ON OBJECT *object-name*	42501
+552	*auth-id* DOES NOT HAVE THE PRIVILEGE TO PERFORM OPERATION *operation*	01542
–552	*auth-id* DOES NOT HAVE THE PRIVILEGE TO PERFORM OPERATION *operation*	42502
–553	AUTHORIZATION ID OR SCHEMA NAME *name* SPECIFIED IS NOT VALID FOR REQUESTED OPERATION	42503
–554	AN AUTHORIZATION ID OR ROLE CANNOT GRANT A PRIVILEGE TO ITSELF	42502
–555	AN AUTHORIZATION ID OR ROLE CANNOT REVOKE A PRIVILEGE FROM ITSELF	42502
–556	*revoke-target* CANNOT HAVE THE *privilege* PRIVILEGE *object-name* REVOKED BY *revoker-id* BECAUSE THE REVOKEE DOES NOT POSSESS THE PRIVILEGE OR THE REVOKER DID NOT MAKE THE GRANT	42504
–557	INCONSISTENT GRANT/REVOKE KEYWORD *keyword*. PERMITTED KEYWORDS ARE *keyword-list*	42852
+558	THE WITH GRANT OPTION IS IGNORED	01516
–559	ALL AUTHORIZATION FUNCTIONS HAVE BEEN DISABLED	57002
+562	A GRANT OF A PRIVILEGE WAS IGNORED BECAUSE THE GRANTEE ALREADY HAS THE PRIVILEGE FROM THE GRANTOR	01560
-562	THE SPECIFIED PRIVILEGES CANNOT BE GRANTED TO PUBLIC	42508
–567	*bind-type* AUTHORIZATION ERROR USING *auth-id* AUTHORITY PACKAGE = *package-name* PRIVILEGE = *privilege*	42501
–571	THE STATEMENT WOULD RESULT IN A MULTIPLE SITE UPDATE	25000
–573	TABLE *table-name* DOES NOT HAVE A UNIQUE KEY WITH THE SPECIFIED COLUMN NAMES	42890
–574	THE SPECIFIED DEFAULT VALUE OR IDENTITY ATTRIBUTE VALUE CONFLICTS WITH THE DEFINITION OF COLUMN *column-name*	42894
–575	OBJECT *object name* (OBJECT-TYPE *object-type*) CANNOT BE REFERENCED EXPLICITLY OR IMPLICITLY	51024
–577	*Object-type object-name* ATTEMPTED TO MODIFY DATA WHEN THE DEFINITION OF THE FUNCTION OR PROCEDURE DID NOT SPECIFY THIS ACTION	38002
-578	THE RETURN STATEMENT WAS NOT EXECUTED FOR SQL FUNCTION *function-name*	2F005
–579	*Object-type object-name* ATTEMPTED TO READ DATA WHEN THE DEFINITION OF THE FUNCTION OR PROCEDURE DID NOT SPECIFY THIS ACTION	38004
–580	THE RESULT-EXPRESSIONS OF A CASE EXPRESSION CANNOT ALL BE NULL	42625
–581	THE DATA TYPES OF THE RESULT-EXPRESSIONS OF A CASE EXPRESSION ARE NOT COMPATIBLE	42804
–582	THE SEARCH-CONDITION IN A SEARCHED-WHEN-CLAUSE OF A CASE IS NOT VALID IN THE CONTEXT IN WHICH IT WAS SPECIFIED. THE SEARCH CONDITION CONTAINS A QUANTIFIED PREDICATE OR AN IN PREDICATE THAT INCLUDES A FULLSELECT, AND THESE ARE NOT ALLOWED IN THE SPECIFIED CONTEXT	42625
–583	THE USE OF FUNCTION OR EXPRESSION *name* IS INVALID BECAUSE IT IS NOT DETERMINISTIC OR HAS AN EXTERNAL ACTION	42845
–584	INVALID USE OF NULL OR DEFAULT	42608
+585	THE COLLECTION *collection-id* APPEARS MORE THAN ONCE WHEN SETTING THE *special- register* SPECIAL REGISTER	01625
–585	THE COLLECTION *collection-id* APPEARS MORE THAN ONCE IN THE SET *special-register* STATEMENT	42732
–586	THE TOTAL LENGTH OF THE CURRENT PATH SPECIAL REGISTER CANNOT EXCEED 2048 CHARACTERS	42907
–589	A POSITIONED DELETE OR UPDATE STATEMENT FOR CURSOR *cursor-name* SPECIFIED A ROW OF A ROWSET, BUT THE CURSOR IS NOT POSITIONED ON A ROWSET	24520
–590	NAME *name* IS NOT UNIQUE IN THE CREATE OR ALTER FOR ROUTINE OR TRIGGER *object-name*	42734
–592	NOT AUTHORIZED TO CREATE FUNCTIONS OR PROCEDURES IN WLM ENVIRONMENT *env-name*	42510
–593	NOT NULL MUST BE SPECIFIED FOR *column-name* BECAUSE IT IS DEFINED AS A ROWID (OR DISTINCT TYPE FOR ROWID), ROW CHANGE TIMESTAMP COLUMN, ROW BEGIN COLUMN, ROW END COLUMN, OR COLUMN OF A PERIOD	42831

SQL Code	Explanation	SQL State
	column-name	
–594	ATTEMPT TO CREATE A NULLABLE ROWID OR DISTINCT TYPE COLUMN *column-name*	42831
+599	COMPARISON FUNCTIONS ARE NOT CREATED FOR A DISTINCT TYPE BASED ON A LONG STRING DATA TYPE	01596
–601	THE NAME (VERSION OR VOLUME SERIAL NUMBER) OF THE OBJECT TO BE DEFINED OR THE TARGET OF A RENAME STATEMENT IS IDENTICAL TO THE EXISTING NAME (VERSION OR VOLUME SERIAL NUMBER) *object-name* OF THE OBJECT TYPE *object-type*	42710 or 46002
–602	TOO MANY COLUMNS, PERIODS OR KEY-EXPRESSIONS SPECIFIED IN A CREATE INDEX OR ALTER INDEX STATEMENT	54008
–603	A UNIQUE INDEX CANNOT BE CREATED BECAUSE THE TABLE CONTAINS ROWS WHICH ARE DUPLICATES WITH RESPECT TO THE VALUES OF THE IDENTIFIED COLUMNS AND PERIODS	23515
–604	A DATA TYPE DEFINITION SPECIFIES AN INVALID ATTRIBUTE SUCH AS LENGTH, PRECISION, OR SCALE ATTRIBUTE	42611
–607	OPERATION OR OPTION *operation* IS NOT DEFINED FOR THIS OBJECT	42832
+610	A CREATE/ALTER ON OBJECT *object-name* HAS PLACED OBJECT IN *utility-name* PENDING	01566
–611	ONLY LOCKMAX 0 CAN BE SPECIFIED WHEN THE LOCK SIZE OF THE TABLESPACE IS TABLESPACE OR TABLE	53088
–612	*identifier* IS A DUPLICATE NAME	42711
–613	THE PRIMARY KEY OR A HASH KEY OR A UNIQUE CONSTRAINT IS TOO LONG OR HAS TOO MANY COLUMNS AND PERIODS	54008
–614	THE INDEX CANNOT BE CREATED OR ALTERED, OR THE LENGTH OF A COLUMN CANNOT BE CHANGED BECAUSE THE SUM OF THE INTERNAL LENGTHS OF THE COLUMNS FOR THE INDEX IS GREATER THAN THE ALLOWABLE MAXIMUM	54008
–615	*operation-type* IS NOT ALLOWED ON A PACKAGE IN USE	55006
–616	*obj-type1 obj-name1* CANNOT BE DROPPED BECAUSE IT IS REFERENCED BY *obj-type2 obj-name2*	42893
–618	OPERATION *operation* IS NOT ALLOWED ON SYSTEM DATABASES	42832
–619	OPERATION DISALLOWED BECAUSE THE DATABASE IS NOT STOPPED	55011
–620	KEYWORD keyword IN *stmt-type* STATEMENT IS NOT PERMITTED FOR A *space-type* SPACE IN THE *database-type* DATABASE	53001
–621	DUPLICATE DBID *dbid* WAS DETECTED AND PREVIOUSLY ASSIGNED TO *database-name*	58001
–622	FOR MIXED DATA IS INVALID BECAUSE THE MIXED DATA INSTALL OPTION IS NO	56031
–623	CLUSTER IS NOT VALID FOR *table-name*	55012
–624	TABLE *table-name* ALREADY HAS A PRIMARY KEY OR UNIQUE CONSTRAINT WITH SPECIFIED COLUMNS AND PERIODS	42889
–625	TABLE *table-name* DOES NOT HAVE AN INDEX TO ENFORCE THE UNIQUENESS OF THE PRIMARY OR UNIQUE KEY	55014
–626	THE ALTER STATEMENT IS NOT EXECUTABLE BECAUSE THE PAGE SET IS NOT STOPPED	55015
–627	THE ALTER STATEMENT IS INVALID BECAUSE THE TABLE SPACE OR INDEX HAS USER-MANAGED DATA SETS	55016
–629	SET NULL CANNOT BE SPECIFIED BECAUSE FOREIGN KEY *name* CANNOT CONTAIN NULL VALUES	42834
–631	FOREIGN KEY *name* IS TOO LONG OR HAS TOO MANY COLUMNS	54008
–632	THE TABLE CANNOT BE DEFINED AS A DEPENDENT OF *table-name* BECAUSE OF DELETE RULE RESTRICTIONS	42915
–633	THE DELETE RULE MUST BE *delete-rule*	42915
–634	THE DELETE RULE MUST NOT BE CASCADE	42915
–635	THE DELETE RULES CANNOT BE DIFFERENT OR CANNOT BE SET NULL	42915
–636	RANGES SPECIFIED FOR PARTITION *part-num* ARE NOT VALID	56016
–637	DUPLICATE *keyword* KEYWORD OR CLAUSE	42614
–638	TABLE *table-name* CANNOT BE CREATED BECAUSE COLUMN DEFINITION IS MISSING	42601
–639	A NULLABLE COLUMN OF A FOREIGN KEY WITH A DELETE RULE OF SET NULL CANNOT BE A COLUMN OF THE KEY OF A PARTITIONED INDEX	56027
–640	LOCKSIZE ROW CANNOT BE SPECIFIED BECAUSE TABLE IN	56089

SQL Code	Explanation	SQL State
	THIS TABLESPACE HAS TYPE 1 INDEX	
–643	A CHECK CONSTRAINT OR THE VALUE OF AN EXPRESSION FOR A COLUMN OF AN INDEX EXCEEDS THE MAXIMUM ALLOWABLE LENGTH KEY EXPRESSION	54024
–644	INVALID VALUE SPECIFIED FOR KEYWORD OR CLAUSE *keyword-or-clause* IN STATEMENT *stmt-type*	42615
+645	WHERE NOT NULL IS IGNORED BECAUSE THE INDEX KEY CANNOT CONTAIN NULL VALUES OR THE INDEX IS AN XML INDEX	01528
–646	TABLE *table-name* CANNOT BE CREATED IN SPECIFIED TABLE SPACE *table-space-name* BECAUSE IT ALREADY CONTAINS A TABLE	55017
–647	BUFFERPOOL *bp-name* FOR IMPLICIT OR EXPLICIT TABLESPACE OR INDEXSPACE *name* HAS NOT BEEN ACTIVATED	57003
+650	THE TABLE BEING CREATED OR ALTERED CANNOT BECOME A DEPENDENT TABLE	01538
–650	THE ALTER STATEMENT CANNOT BE EXECUTED, REASON *reason-code*	56090
–651	TABLE DESCRIPTION EXCEEDS MAXIMUM SIZE OF OBJECT DESCRIPTOR	54025
–652	VIOLATION OF INSTALLATION DEFINED EDIT OR VALIDATION PROCEDURE *proc-name*	23506
+653	TABLE *table-name* IN PARTITIONED TABLESPACE *tspace-name* IS NOT AVAILABLE BECAUSE ITS PARTITIONED INDEX HAS NOT BEEN CREATED	01551
–653	TABLE *table-name* IN PARTITIONED TABLE SPACE *tspace-name* IS NOT AVAILABLE BECAUSE ITS PARTITIONED INDEX HAS NOT BEEN CREATED	57004
+655	STOGROUP *stogroup_name* HAS BOTH SPECIFIC AND NON-SPECIFIC VOLUME IDS. IT WILL NOT BE ALLOWED IN FUTURE RELEASES	01597
–655	THE CREATE OR ALTER STOGROUP IS INVALID BECAUSE THE STORAGE GROUP WOULD HAVE BOTH SPECIFIC AND NON-SPECIFIC VOLUME IDS	56036
+658	THE SUBPAGES VALUE IS IGNORED FOR THE CATALOG INDEX *index-name*	01600
–658	AN *object-type* CANNOT BE DROPPED USING THE *statement* STATEMENT	42917
–660	INDEX *index-name* CANNOT BE CREATED OR ALTERED ON PARTITIONED TABLE SPACE *tspace-name* BECAUSE KEY LIMITS ARE NOT SPECIFIED	53035
–661	*object-type object-name* CANNOT BE CREATED ON PARTITIONED TABLE SPACE *tspace-name* BECAUSE THE NUMBER OF PARTITION SPECIFICATIONS IS NOT EQUAL TO THE NUMBER OF PARTITIONS OF THE TABLE SPACE	53036
–662	A PARTITIONED INDEX CANNOT BE CREATED ON A TABLE SPACE, OR A TABLE SPACE CANNOT BE INDEX-CONTROLLED. TABLE SPACE *tspace-name*, REASON *reason-code*	53037
–663	THE NUMBER OF KEY LIMIT VALUES IS EITHER ZERO, OR GREATER THAN THE NUMBER OF COLUMNS IN THE KEY OF INDEX *index-name*	53038
+664	THE INTERNAL LENGTH OF THE LIMIT-KEY FIELDS SPECIFIED IN THE PARTITION CLAUSE OF THE *statement-name* STATEMENT EXCEEDS THE EXISTING INTERNAL LIMIT KEY LENGTH STORED IN CATALOG TABLE *tablename*	01540
–665	THE PARTITION CLAUSE OF AN ALTER STATEMENT IS OMITTED OR INVALID	53039
–666	*stmt-verb object* CANNOT BE EXECUTED BECAUSE *function* IS IN PROGRESS	57005
–667	THE CLUSTERING INDEX FOR A PARTITIONED TABLE SPACE CANNOT BE EXPLICITLY DROPPED	42917
–668	THE COLUMN CANNOT BE ADDED TO THE TABLE BECAUSE THE TABLE HAS AN EDIT PROCEDURE DEFINED WITH ROW ATTRIBUTE SENSITIVITY	56018
–669	THE OBJECT CANNOT BE EXPLICITLY DROPPED. REASON *reason-code*	42917
–670	THE RECORD LENGTH OF THE TABLE EXCEEDS THE PAGE SIZE LIMIT	54010
–671	THE BUFFERPOOL ATTRIBUTE OF THE TABLE SPACE CANNOT BE ALTERED AS SPECIFIED BECAUSE IT WOULD CHANGE THE PAGE SIZE OF THE TABLE SPACE	53040
–672	OPERATION DROP NOT ALLOWED ON TABLE *table-name*	55035
–677	INSUFFICIENT VIRTUAL STORAGE FOR BUFFERPOOL EXPANSION	57011

SQL Code	Explanation	SQL State
–678	THE CONSTANT *constant* SPECIFIED FOR THE INDEX LIMIT KEY MUST CONFORM TO THE DATA TYPE *data-type* OF THE CORRESPONDING COLUMN *column-name*	53045
–679	THE OBJECT *name* CANNOT BE CREATED BECAUSE A DROP IS PENDING ON THE OBJECT	57006
–680	TOO MANY COLUMNS SPECIFIED FOR A TABLE, VIEW, OR TABLE FUNCTION	54011
–681	COLUMN *column-name* IN VIOLATION OF INSTALLATION DEFINED FIELD PROCEDURE. RT: *return-code*, RS: *reason-code*, MSG: *message-token*	23507
–682	FIELD PROCEDURE *procedure-name* COULD NOT BE LOADED	57010
–683	THE SPECIFICATION FOR COLUMN, DISTINCT TYPE, FUNCTION, OR PROCEDURE *data-item* CONTAINS INCOMPATIBLE CLAUSES	42842
–684	THE LENGTH OF CONSTANT LIST BEGINNING *string* IS TOO LONG	54012
–685	INVALID FIELD TYPE, *column-name*	58002
–686	COLUMN DEFINED WITH A FIELD PROCEDURE CAN NOT COMPARE WITH ANOTHER COLUMN WITH DIFFERENT FIELD PROCEDURE	53043
–687	FIELD TYPES INCOMPARABLE	53044
–688	INCORRECT DATA RETURNED FROM FIELD PROCEDURE, *column-name, msgno*	58002
–689	TOO MANY COLUMNS DEFINED FOR A DEPENDENT TABLE	54011
–690	THE STATEMENT IS REJECTED BY DATA DEFINITION CONTROL SUPPORT. REASON *reason-code*	23508
–691	THE REQUIRED REGISTRATION TABLE *table-name* DOES NOT EXIST	57018
–692	THE REQUIRED UNIQUE INDEX *index-name* FOR DDL REGISTRATION TABLE *table-name* DOES NOT EXIST	57018
–693	THE COLUMN *column-name* IN DDL REGISTRATION TABLE OR INDEX *name* IS NOT DEFINED PROPERLY	55003
–694	THE SCHEMA STATEMENT CANNOT BE EXECUTED BECAUSE A DROP IS PENDING ON THE DDL REGISTRATION TABLE *table-name*	57023
–695	INVALID VALUE *seclabel* SPECIFIED FOR SECURITY LABEL COLUMN OF TABLE *table-name*	23523
–696	THE DEFINITION OF TRIGGER *trigger-name* INCLUDES AN INVALID USE OF CORRELATION NAME OR TRANSITION TABLE NAME *name*. REASON CODE = *reason-code*	42898
–697	OLD OR NEW CORRELATION NAMES ARE NOT ALLOWED IN A TRIGGER DEFINED WITH THE FOR EACH STATEMENT CLAUSE. OLD_TABLE OR NEW_TABLE NAMES ARE NOT ALLOWED IN A TRIGGER WITH BEFORE CLAUSE	42899
–713	THE REPLACEMENT VALUE *value* FOR *special-register* IS INVALID	42815 or 3F000
–715	PROGRAM *program-name* WITH MARK *release-dependency-mark* FAILED BECAUSE IT DEPENDS ON FUNCTIONS OF THE RELEASE FROM WHICH FALLBACK HAS OCCURRED	56064
–716	PROGRAM *program-name* PRECOMPILED WITH INCORRECT LEVEL FOR THIS RELEASE	56065
–717	*bind type* FOR *object-type object-name* WITH MARK *release-dependency-mark* FAILED BECAUSE *object-type* DEPENDS ON FUNCTIONS OF THE RELEASE FROM WHICH FALLBACK HAS OCCURRED	56066
–718	REBIND FOR PACKAGE *package-name* FAILED BECAUSE IBMREQD OF *ibmreqd* IS INVALID	56067
–719	BIND ADD ERROR USING *auth-id* AUTHORITY PACKAGE *package-name* ALREADY EXISTS	42710
–720	BIND ERROR, ATTEMPTING TO REPLACE PACKAGE = *package-name* WITH VERSION = *version2* BUT THIS VERSION ALREADY EXISTS	42710
–721	BIND ERROR FOR PACKAGE = *pkg-id* CONTOKEN = *'contoken'X* IS NOT UNIQUE SO IT CANNOT BE CREATED	42710
–722	*bind-type* ERROR USING *auth-id* AUTHORITY PACKAGE *package-name* DOES NOT EXIST	42704
–723	AN ERROR OCCURRED IN A TRIGGERED SQL STATEMENT IN *trigger-name*. INFORMATION RETURNED: SQLCODE *sql-code,* SQLSTATE *sql-state,* MESSAGE TOKENS *token-list,* SECTION NUMBER *section-number*	09000
–724	THE ACTIVATION OF THE *object-type* OBJECT *object-name* WOULD EXCEED THE MAXIMUM LEVEL OF INDIRECT SQL CASCADING	54038
–725	THE SPECIAL REGISTER *register* AT LOCATION *location* WAS SUPPLIED AN INVALID VALUE	42721

SQL Code	Explanation	SQL State
–726	BIND ERROR ATTEMPTING TO REPLACE PACKAGE = *package-name*. THERE ARE ENABLE OR DISABLE ENTRIES CURRENTLY ASSOCIATED WITH THE PACKAGE	55030
–729	A STORED PROCEDURE SPECIFYING COMMIT ON RETURN CANNOT BE THE TARGET OF A NESTED CALL STATEMENT	429B1
–730	THE PARENT OF A TABLE IN A READ-ONLY SHARED DATABASE MUST ALSO BE A TABLE IN A READ-ONLY SHARED DATABASE	56053
–731	USER DEFINED DATASET *dsname* MUST BE DEFINED WITH SHAREOPTIONS(1,3)	56054
–732	THE DATABASE IS DEFINED ON THIS SUBSYSTEM WITH THE ROSHARE READ ATTRIBUTE BUT THE TABLE SPACE OR INDEX SPACE HAS NOT BEEN DEFINED ON THE OWNING SUBSYSTEM	56055
–733	THE DESCRIPTION OF A TABLE SPACE, INDEX SPACE, OR TABLE IN A ROSHARE READ DATABASE MUST BE CONSISTENT WITH ITS DESCRIPTION IN THE OWNER SYSTEM	56056
–734	THE ROSHARE ATTRIBUTE OF A DATABASE CANNOT BE ALTERED FROM ROSHARE READ	56057
–735	DATABASE *dbid* CANNOT BE ACCESSED BECAUSE IT IS NO LONGER A SHARED DATABASE	55004
–736	INVALID OBID *obid* SPECIFIED	53014
–737	IMPLICIT TABLE SPACE NOT ALLOWED	56056
+738	DEFINITION CHANGE OF *object object-name* MAY REQUIRE SIMILAR CHANGE ON READ-ONLY SYSTEMS	01530
–739	CREATE OR ALTER FUNCTION *function-name* FAILED BECAUSE FUNCTIONS CANNOT MODIFY DATA WHEN THEY ARE PROCESSED IN PARALLEL	56088
–740	FUNCTION *name* IS DEFINED WITH THE OPTION MODIFIES SQL DATA WHICH IS NOT VALID IN THE CONTEXT IN WHICH IT WAS INVOKED	51034
–741	A *database-type* DATABASE IS ALREADY DEFINED FOR MEMBER *member-name*	55020
–742	DSNDB07 IS THE IMPLICIT WORK FILE DATABASE	53004
–746	THE SQL STATEMENT IN FUNCTION, TRIGGER, OR IN STORED PROCEDURE *name* VIOLATES THE NESTING SQL RESTRICTION	57053
–747	TABLE *table-name* IS NOT AVAILABLE UNTIL THE AUXILIARY TABLES AND INDEXES FOR ITS EXTERNALLY STORED COLUMNS HAVE BEEN CREATED	57054
–748	AN INDEX *index-name* ALREADY EXISTS ON AUXILIARY TABLE *table-name*	54042
–750	THE SOURCE TABLE *source-name* CANNOT BE RENAMED OR ALTERED AS SPECIFIED	42986
–751	*object-type object-name* (SPECIFIC NAME *specific name*) ATTEMPTED TO EXECUTE AN SQL STATEMENT sql-*statement* THAT IS NOT ALLOWED	38003
–752	THE CONNECT STATEMENT IS INVALID BECAUSE THE PROCESS IS NOT IN THE CONNECTABLE STATE	0A001
–763	INVALID TABLE SPACE NAME *space-name*	560A1
–764	A LOB TABLE SPACE AND ITS ASSOCIATED BASE TABLE SPACE MUST BE IN THE SAME DATABASE	560A2
–765	TABLE IS NOT COMPATIBLE WITH DATABASE	560A3
–766	THE OBJECT OF A STATEMNT IS A TABLE FOR WHICH THE REQUESTED OPERATION IS NOT PERMITTED	560A4
–767	MISSING OR INVALID COLUMN SPECIFICATION FOR INDEX *index-name*	42626
–768	AN AUXILIARY TABLE ALREADY EXISTS FOR THE SPECIFIED COLUMN OR PARTITION	560A5
–769	SPECIFICATION OF CREATE AUX TABLE DOES NOT MATCH THE CHARACTERISTICS OF THE BASE TABLE	53096
–770	TABLE *table-name* CANNOT HAVE A LOB COLUMN UNLESS IT ALSO HAS A ROWID, OR AN XML COLUMN UNLESS IT ALSO HAS A DOCID COLUMN	560A6
–771	INVALID SPECIFICATION OF A ROWID COLUMN	428C7
–773	CASE NOT FOUND FOR CASE STATEMENT	20000
–776	USE OF CURSOR *cursor-name* IS NOT VALID	428D4
–778	ENDING LABEL *label-name* DOES NOT MATCH THE BEGINNING LABEL	428D5
–779	LABEL *label* SPECIFIED ON A GOTO, ITERATE, OR LEAVE STATEMENT IS NOT VALID	42736
–780	UNDO SPECIFIED FOR A HANDLER	428D6
–781	CONDITION *condition-name* IS NOT DEFINED OR THE DEFINITION IS NOT IN SCOPE	42737

SQL Code	Explanation	SQL State
–782	A CONDITION OR SQLSTATE *condition-value* SPECIFIED IS NOT VALID	428D7
–783	SELECT LIST FOR CURSOR *cursor-name* IN FOR STATEMENT IS NOT VALID. COLUMN *column-name* IS NOT UNIQUE	42738
-784	CONSTRAINT *constraint-name* CANNOT BE DROPPED	42860
–785	USE OF SQLCODE OR SQLSTATE IS NOT VALID	428D8
–787	RESIGNAL STATEMENT ISSUED OUTSIDE OF A HANDLER	0K000
-788	THE SAME ROW OF TARGET TABLE *table-name* WAS IDENTIFIED MORE THAN ONCE FOR AN UPDATE, DELETE, OR CHANGE OPERATION OF THE MERGE STATEMENT	21506
-789	THE DATA TYPE OR OTHER ATTRIBUTES FOR PARAMETER OR SQL VARIABLE *name* ARE NOT SUPPORTED IN THE ROUTINE	429BB
–797	THE TRIGGER *trigger-name* IS DEFINED WITH AN UNSUPPORTED TRIGGER SQL STATEMENT	42987
–798	A VALUE CANNOT BE SPECIFIED FOR COLUMN *column-name* WHICH IS DEFINED AS GENERATED ALWAYS	428C9
+799	A SET STATEMENT REFERENCES A SPECIAL REGISTER THAT DOES NOT EXIST AT THE SERVER SITE	01527
+802	EXCEPTION ERROR *exception-type* HAS OCCURRED DURING *operation-type* OPERATION ON *data-type* DATA, POSITION *position-number*	01519
–802	EXCEPTION ERROR *exception-type* HAS OCCURRED DURING *operation-type* OPERATION ON *data-type* DATA, POSITION *position-number*	22003 or 22012
–803	AN INSERTED OR UPDATED VALUE IS INVALID BECAUSE THE INDEX IN INDEX SPACE *indexspace-name* CONSTRAINS COLUMNS OF THE TABLE SO NO TWO ROWS CAN CONTAIN DUPLICATE VALUES IN THOSE COLUMNS. RID OF EXISTING ROW IS X *record-id*	23505
–804	AN ERROR WAS FOUND IN THE APPLICATION PROGRAM INPUT PARAMETERS FOR THE SQL STATEMENT. REASON *reason*	07002
–805	DBRM OR PACKAGE NAME *location-name.collection-id.dbrm-name.consistency-token* NOT FOUND IN PLAN *plan-name*. REASON *reason*	51002
+806	BIND ISOLATION LEVEL RR CONFLICTS WITH TABLESPACE LOCKSIZE PAGE OR LOCKSIZE ROW AND LOCKMAX 0	01553
+807	THE RESULT OF DECIMAL MULTIPLICATION MAY CAUSE OVERFLOW	01554
–807	ACCESS DENIED: PACKAGE *package-name* IS NOT ENABLED FOR ACCESS FROM *connection-type connection-name*	23509
–808	THE CONNECT STATEMENT IS NOT CONSISTENT WITH THE FIRST CONNECT STATEMENT	08001
–811	THE RESULT OF AN EMBEDDED SELECT STATEMENT OR A SUBSELECT IN THE SET CLAUSE OF AN UPDATE STATEMENT IS A TABLE OF MORE THAN ONE ROW, OR THE RESULT OF A SUBQUERY OF A BASIC PREDICATE IS MORE THAN ONE VALUE	21000
–812	THE SQL STATEMENT CANNOT BE PROCESSED BECAUSE A BLANK COLLECTION-ID WAS FOUND IN THE CURRENT PACKAGESET SPECIAL REGISTER WHILE TRYING TO FORM A QUALIFIED PACKAGE NAME FOR PROGRAM *program-name.consistency-token* USING PLAN *plan-name*	22508
–817	THE SQL STATEMENT CANNOT BE EXECUTED BECAUSE THE STATEMENT WILL RESULT IN A PROHIBITED UPDATE OPERATION	25000
–818	THE PRECOMPILER-GENERATED TIMESTAMP *x* IN THE LOAD MODULE IS DIFFERENT FROM THE BIND TIMESTAMP *y* BUILT FROM THE DBRM *z*	51003
–819	THE VIEW CANNOT BE PROCESSED BECAUSE THE LENGTH OF ITS PARSE TREE IN THE CATALOG IS ZERO	58004
–820	THE SQL STATEMENT CANNOT BE PROCESSED BECAUSE *catalog-table* CONTAINS A VALUE THAT IS NOT VALID IN THIS RELEASE	58004
–822	THE SQLDA CONTAINS AN INVALID DATA ADDRESS OR INDICATOR VARIABLE ADDRESS	51004
–840	TOO MANY ITEMS RETURNED IN A SELECT, INSERT LIST, OR FROM UNNEST	54004
–842	A CONNECTION TO *location-name* ALREADY EXISTS	08002
–843	THE SET CONNECTION OR RELEASE STATEMENT MUST SPECIFY AN EXISTING CONNECTION	08003
–845	A PREVIOUS VALUE EXPRESSION CANNOT BE USED BEFORE THE NEXT VALUE EXPRESSION GENERATES A VALUE IN THE CURRENT APPLICATION PROCESS FOR SEQUENCE *sequence-name*	51035

SQL Code	Explanation	SQL State
–846	INVALID SPECIFICATION OF AN IDENTITY COLUMN OR SEQUENCE OBJECT *object-type object-name*. REASON CODE = *reason_code*	42815
+863	THE CONNECTION WAS SUCCESSFUL BUT ONLY SBCS WILL BE SUPPORTED	01539
–867	INVALID SPECIFICATION OF A ROWID COLUMN	428C7
–870	THE NUMBER OF HOST VARIABLES IN THE STATEMENT IS NOT EQUAL TO THE NUMBER OF DESCRIPTORS	58026
–872	A VALID CCSID HAS NOT YET BEEN SPECIFIED FOR THIS SUBSYSTEM	51032
–873	THE STATEMENT REFERENCED DATA ENCODED WITH DIFFERENT ENCODING SCHEMES OR CCSIDS IN AN INVALID CONTEXT	53090
–874	THE ENCODING SCHEME SPECIFIED FOR THE *object-type* MUST BE THE SAME AS THE CONTAINING TABLE SPACE OR OTHER PARAMETERS	53091
–875	*operand* CANNOT BE USED WITH THE ASCII DATA REFERENCED	42988
–876	*object-type* CANNOT BE CREATED OR ALTERED REASON, reason-code	53092
–877	CCSID ASCII OR CCSID UNICODE IS NOT ALLOWED FOR THIS DATABASE OR TABLE SPACE	53093
–878	THE *explain-object* USED FOR EXPLAIN MUST BE ENCODED IN UNICODE. IT CANNOT BE IN ASCII OR EBCDIC.	53094
–879	CREATE OR ALTER STATEMENT FOR *object-name* CANNOT DEFINE A COLUMN, TYPE, VARIABLE, FUNCTION OR STORED PROCEDURE PARAMETER AS MIXED OR GRAPHIC WITH ENCODING SCHEME *encoding-scheme*	53095
–880	SAVEPOINT *savepoint-name* DOES NOT EXIST OR IS INVALID IN THIS CONTEXT	3B001
–881	A SAVEPOINT WITH NAME *savepoint-name* ALREADY EXISTS, BUT THIS SAVEPOINT NAME CANNOT BE REUSED	3B501
–882	SAVEPOINT DOES NOT EXIST	3B502
+883	ROLLBACK TO SAVEPOINT OCCURRED WHEN THERE WERE OPERATIONS THAT CANNOT BE UNDONE, OR AN OPERATION THAT CANNOT BE UNDONE OCCURRED WHEN THERE WAS A SAVEPOINT OUTSTANDING	01640
–900	THE SQL STATEMENT CANNOT BE EXECUTED BECAUSE THE APPLICATION PROCESS IS NOT CONNECTED TO A SERVER	08003
–901	UNSUCCESSFUL EXECUTION CAUSED BY A SYSTEM ERROR THAT DOES NOT PRECLUDE THE SUCCESSFUL EXECUTION OF SUBSEQUENT SQL STATEMENTS	58004
-902	POINTER TO THE ESSENTIAL CONTROL BLOCK (CT/RDA) HAS VALUE 0, REBIND REQUIRED	58005
–904	UNSUCCESSFUL EXECUTION CAUSED BY AN UNAVAILABLE RESOURCE. REASON *reason-code*, TYPE OF RESOURCE *resource-type*, AND RESOURCE NAME *resource-name*	57011
–905	UNSUCCESSFUL EXECUTION DUE TO RESOURCE LIMIT BEING EXCEEDED, RESOURCE NAME = *resource-name* LIMIT = *limit-amount1* CPU SECONDS (*limit-amount2* SERVICE UNITS) DERIVED FROM *limit-source*	57014
–906	THE SQL STATEMENT CANNOT BE EXECUTED BECAUSE THIS FUNCTION IS DISABLED DUE TO A PRIOR ERROR	51005
-907	AN ATTEMPT WAS MADE TO MODIFY THE TARGET TABLE, *table-name*, OF THE MERGE STATEMENT BY CONSTRAINT OR TRIGGER *trigger-name*	27000
–908	*bind-type* ERROR USING *auth-id* AUTHORITY. BIND, REBIND OR AUTO-REBIND OPERATION IS NOT ALLOWED	23510
–909	THE OBJECT HAS BEEN DELETED OR ALTERED	57007
–910	THE SQL STATEMENT CANNOT ACCESS AN OBJECT ON WHICH UNCOMMITTED CHANGES ARE PENDING	57007
–911	THE CURRENT UNIT OF WORK HAS BEEN ROLLED BACK DUE TO DEADLOCK OR TIMEOUT. REASON *reason-code*, TYPE OF RESOURCE *resource-type*, AND RESOURCE NAME *resource-name*	40001
–913	UNSUCCESSFUL EXECUTION CAUSED BY DEADLOCK OR TIMEOUT. REASON CODE *reason-code*, TYPE OF RESOURCE *resource-type*, AND RESOURCE NAME *resource-name*	57033
–917	BIND PACKAGE FAILED	42969
–918	THE SQL STATEMENT CANNOT BE EXECUTED BECAUSE A CONNECTION HAS BEEN LOST	51021
–919	A ROLLBACK OPERATION IS REQUIRED	56045
–922	AUTHORIZATION FAILURE: *error-type* ERROR. REASON *reason-code*	42505
–923	CONNECTION NOT ESTABLISHED: DB2 *condition* REASON	57015

SQL Code	Explanation	SQL State
	reason-code, TYPE *resource-type*, NAME *resource-name*	
–924	DB2 CONNECTION INTERNAL ERROR, *function-code, return-code, reason-code*	58006
–925	COMMIT NOT VALID IN IMS, CICS, OR RRSAF ENVIRONMENT	2D521
–926	ROLLBACK NOT VALID IN IMS, CICS, OR RRSAF ENVIRONMENT	2D521
–927	THE LANGUAGE INTERFACE (LI) WAS CALLED WHEN THE CONNECTING ENVIRONMENT WAS NOT ESTABLISHED. THE PROGRAM SHOULD BE INVOKED UNDER THE DSN COMMAND	51006
–929	FAILURE IN A DATA CAPTURE EXIT: *token*	58002
–939	ROLLBACK REQUIRED DUE TO UNREQUESTED ROLLBACK OF A REMOTE SERVER	51021
–947	THE SQL STATEMENT FAILED BECAUSE IT WILL CHANGE A TABLE DEFINED WITH DATA CAPTURE CHANGES, BUT THE DATA CANNOT BE PROPAGATED	56038
–948	DISTRIBUTED OPERATION IS INVALID	56062
–950	THE LOCATION NAME SPECIFIED IN THE CONNECT STATEMENT IS INVALID OR NOT LISTED IN THE COMMUNICATIONS DATABASE	42705
–951	OBJECT *object-name* OBJECT TYPE *object-type* IS IN USE AND CANNOT BE THE TARGET OF THE SPECIFIED ALTER STATEMENT	55007
–952	PROCESSING WAS INTERUPTED BY A CANCEL REQUEST FROM A CLIENT PROGRAM	57014
–981	THE SQL STATEMENT FAILED BECAUSE THE RRSAF CONNECTION IS NOT IN A STATE THAT ALLOWS SQL OPERATIONS, REASON *reason-code*	57015
–989	AFTER TRIGGER *trigger-name* ATTEMPTED TO MODIFY A ROW IN TABLE *table-name* THAT WAS MODIFIED BY AN SQL DATA CHANGE STATEMENT WITHIN A FROM CLAUSE	560C3
–991	CALL ATTACH WAS UNABLE TO ESTABLISH AN IMPLICIT CONNECT OR OPEN TO DB2. RC1=*rc1* RC2=*rc2*	57015
–992	PACKAGE *package-name* CANNOT BE EXECUTED OR DEPLOYED ON LOCATION *location-name*	51008
–1403	THE USERNAME AND/OR PASSWORD SUPPLIED IS INCORRECT	08004
–1760	CREATE PROCEDURE FOR *procedure-name* MUST HAVE VALID LANGUAGE AND EXTERNAL CLAUSES	42601
–2001	THE NUMBER OF HOST VARIABLE PARAMETERS FOR A STORED PROCEDURE IS NOT EQUAL TO THE NUMBER OF EXPECTED HOST VARIABLE PARAMETERS. ACTUAL NUMBER *sqldanum*, EXPECTED NUMBER *opnum*	53089
–4302	JAVA STORED PROCEDURE OR USER-DEFINED FUNCTION *routine-name* (SPECIFIC NAME *specific-name*) HAS EXITED WITH AN EXCEPTION *exception-string*	38000
–4700	ATTEMPT TO USE NEW FUNCTION BEFORE FUNCTION LEVEL IS ACTIVATED	56038
–4701	THE NUMBER OF PARTITIONS, OR THE COMBINATION OF THE NUMBER OF TABLE SPACE PARTITIONS AND THE CORRESPONDING LENGTH OF THE PARTITIONING LIMIT KEY EXCEEDS THE SYSTEM LIMIT; OR THE COMBINATION OF THE NUMBER OF TABLE SPACE PARTITIONS EXCEEDS THE MAXPARTITIONS FOR PARTITION BY GROWTH TABLE SPACE	54054
–4704	AN UNSUPPORTED DATA TYPE WAS ENCOUNTERED AS AN INCLUDE COLUMN	56084
–4705	*option* SPECIFIED ON ALTER STATEMENT FOR *object-name (object-type)* IS NOT VALID	530A3
–4706	ALTER STATEMENT FOR AN SQL ROUTINE OR ADVANCED TRIGGER CANNOT BE PROCESSED BECAUSE THE OPTIONS CURRENTLY IN EFFECT (ENVID *current-envid*) ARE NOT THE SAME AS THE ONES THAT WERE IN EFFECT (ENVID *defined-envid*) WHEN THE OBJECT OR VERSION WAS FIRST DEFINED	530A4
–4709	EXPLAIN MONITORED STMTS FAILED WITH REASON CODE = *yyyyy*	560CK
–4710	EXCHANGE DATA STATEMENT SPECIFIED *table1* AND *table2* BUT THE TABLES DO NOT HAVE A DEFINED CLONE RELATIONSHIP	530A7
+4726	THE STATEMENT WAS SUCCESSFULLY PREPARED, BUT IT CANNOT BE EXECUTED BECAUSE *auth-id* DOES NOT HAVE *privilege* PRIVILEGE ON OBJECT *object-name* BUT HAS EXPLAIN PRIVILEGE	0168Z
-4727	SYSTEM PARAMETER *system-parameter* VALUE *parameter-value* IS INCONSISTENT WITH CLAUSE *clause* SPECIFIED ON *statement-name* STATEMENT	530A8

SQL Code	Explanation	SQL State
-4728	ANOTHER VERSION OF OBJECT *object-name* EXISTS AND IS DEFINED WITH AN INCOMPATIBLE OPTION. THE OPTION IS *option-name*	428HL
-4729	PROCEDURE IS DEFINED AS AUTONOMOUS, AND CANNOT BE INVOKED BECAUSE THE NESTED ENVIRONMENT ALREADY INVOKED AN AUTONOMOUS PROCEDURE	51043
-4730	INVALID SPECIFICATION OF XML COLUMN *table-name.column-name* IS NOT DEFINED IN THE XML VERSIONING FORMAT, REASON *reason-code*	55079
-4731	THE NATIVE SQL ROUTINE STATEMENT FOR PACKAGE *location-name.collection-id.program-name.consistency-token* STATEMENT NUMBER *statement-number* CANNOT BE PROCESSED	560C5
-4732	THE MAXIMUM NUMBER OF ALTERS ALLOWED HAS BEEN EXCEEDED FOR *object-type*	54055
-4733	THE ALTER TABLE STATEMENT CANNOT BE EXECUTED BECAUSE COLUMN *column-name* IS MIXED DATA, OR THE DATA TYPE OR LENGTH SPECIFIED DOES NOT AGREE WITH THE EXISTING DATA TYPE OR LENGTH	429BQ
-4734	THE LOAD MODULE FOR THE PROCEDURE ASSUMES A PARAMETER VARCHAR OPTION THAT IS NOT CONSISTENT WITH THE OPTION SPECIFIED ON THE CREATE PROCEDURE STATEMENT FOR *procedure-name*	560CU
-4735	INVALID TABLE REFERENCE FOR TABLE LOCATOR	560CV
-4736	A PERIOD SPECIFICATION OR PERIOD CLAUSE IS NOT SUPPORTED AS SPECIFIED FOR OBJECT *object-name*. REASON CODE = *reason-code*	560CY
-4737	STATEMENT *statement* IS NOT ALLOWED WHEN USING A TRUSTED CONNECTION	429BY
-4738	TABLE *table-name* CANNOT BE DEFINED AS SPECIFIED IN THE *statement* STATEMENT IN A COMMON CRITERIA ENVIRONMENT	56038
-4739	ENVIRONMENT SETTINGS (IDENTIFIED BY *envid1*) USED BY *object-name* ARE NOT THE SAME AS THE ONES THAT WERE IN EFFECT (IDENTIFIED BY *envid2*) WHEN OTHER COLUMN MASKS AND ROW PERMISSIONS WERE DEFINED FOR TABLE *table-name*	530A4
-4743	ATTEMPT TO USE NEW FUNCTION WHEN THE APPLICATION COMPATIBILITY SETTING IS SET FOR A PREVIOUS LEVEL	56038
-4744	THE STATEMENT EXPLICITLY OR IMPLICITLY REFERENCED TEMPORAL TABLE *table-name* IN AN UNSUPPORTED CONTEXT. REASON CODE *reason-code*	530A9
+4745	A SECTION WAS BOUND SUCCESSFULLY, BUT AN ERROR OCCURRED WHEN A STATEMENT IN A RELATED EXTENDED SECTION WAS BOUND. INFORMATION RETURNED: SECTION NUMBER *section-number*, SQLCODE *sqlcode*, SQLSTATE sqlstate, AND MESSAGE TOKENS token-list	01578
-4746	THE STATEMENT CANNOT BE PROCESSED FOR AN ACCELERATED QUERY. REASON CODE *reason-code*	560D5
-4747	THE CREATE OR ALTER TABLE STATEMENT FAILED BECAUSE SOME FUNCTIONALITY WAS SPECIFIED IN THE TABLE DEFINITION THAT IS NOT SUPPORTED WITH THE TABLE TYPE. UNSUPPORTED FUNCTIONALITY :*functionality-keyword*.	429CB
+4748	THE *statement* STATEMENT FOR OBJECT *object-name* WAS SUCCESSFUL ON THE DB2 SERVER. THE OBJECT MAY NOT HAVE BEEN SUCCESSFULLY PROCESSED ON THE ACCELERATOR SERVER FOR REASON *reason-code*	0169B
-4749	PACKAGE = *package-name bind-type* ERROR WITH APREUSESOURCE(*copy-type*), THE *copy-type* COPY DOES NOT EXIST	56067
-4750	*csect-name* PACKAGE *package-name* SWITCH TO THE *copy-indicator* COPY FAILED. THIS COPY IS NOT EXECUTABLE WITHOUT AN EXPLICIT REBIND OR AUTOBIND (REASON = *reason-code*)	56067
+4751	*bind-type* WARNING FOR PACKAGE = *package-name*, THE USE OF *keyword* RESULTED IN UNSUCCESSFUL COMPLETION FOR ONE OR MORE STATEMENTS	01599
-4751	*bind-type* FOR PACKAGE = *package-name* FAILED BECAUSE OF THE USE OF *keyword*. ONE OR MORE STATEMENTS WERE NOT SUCCESSFULLY PROCESSED	56095
-4753	*function-name* FUNCTION FAILURE, RC=*return-code* REASON=*reason-code* WAS RECEIVED FROM z/OS UNICODE SERVICES	560DC
–5001	TABLE *table-name* IS NOT VALID	42703
–5012	HOST VARIABLE *host-variable* IS NOT EXACT NUMERIC	42618

SQL Code	Explanation	SQL State
	WITH SCALE ZERO	
–7008	*object-name* NOT VALID FOR OPERATION (*reason-code*)	55019
–16000	AN XQUERY EXPRESSION CANNOT BE PROCESSED BECAUSE THE *context-component* COMPONENT OF THE STATIC CONTEXT HAS NOT BEEN ASSIGNED. ERROR QNAME = *err:XPST0001*	10501
–16001	AN XQUERY EXPRESSION STARTING WITH TOKEN *token* CANNOT BE PROCESSED BECAUSE THE FOCUS COMPONENT OF THE DYNAMIC CONTEXT HAS NOT BEEN ASSIGNED. ERROR QNAME = *err:XPDY0002*	10501
–16002	AN XQUERY EXPRESSION HAS AN UNEXPECTED TOKEN *token* FOLLOWING *text*. EXPECTED TOKENS MAY INCLUDE: *token-list*. ERROR QNAME= *err:XPST0003*	10505
–16003	AN EXPRESSION OF DATA TYPE *value-type* CANNOT BE USED WHEN THE DATA TYPE *expected-type* IS EXPECTED IN THE CONTEXT. ERROR QNAME= *err:XPTY0004*	10507
–16005	AN XQUERY EXPRESSION REFERENCES AN ELEMENT NAME, ATTRIBUTE NAME, TYPE NAME, FUNCTION NAME, NAMESPACE PREFIX, OR VARIABLE NAME *undefined-name* THAT IS NOT DEFINED WITHIN THE STATIC CONTEXT. ERROR QNAME= *err:XPST0008*	10506
–16007	THE XQUERY PATH EXPRESSION REFERENCES AN AXIS *axis-type* THAT IS NOT SUPPORTED. ERROR QNAME = *err:XQST0010*	10505
–16009	AN XQUERY FUNCTION NAMED *function-name* WITH *number-of-parms* PARAMETERS IS NOT DEFINED IN THE STATIC CONTEXT. ERROR QNAME= *err:XPST0017*	10506
–16011	THE RESULT OF AN INTERMEDIATE STEP EXPRESSION IN AN XQUERY PATH EXPRESSION CONTAINS AN ATOMIC VALUE. ERROR QNAME = *err:XPTY0019*	10507
–16012	THE CONTEXT ITEM IN AN AXIS STEP MUST BE A NODE. ERROR QNAME = *err:XPTY0020*	10507
–16015	AN ELEMENT CONSTRUCTOR CONTAINS AN ATTRIBUTE NODE NAMED *attribute-name* THAT FOLLOWS AN XQUERY NODE THAT IS NOT AN ATTRIBUTE NODE. ERROR QNAME = *err:error-name*	10507
–16016	THE ATTRIBUTE NAME *attribute-name* CANNOT BE USED MORE THAN ONCE IN AN ELEMENT CONSTRUCTOR. ERROR QNAME = *err:XQTY0025*	10503
–16020	THE CONTEXT NODE IN A PATH EXPRESSION THAT BEGINS WITH AN INITIAL "/" OR "//" DOES NOT HAVE AN XQUERY DOCUMENT NODE ROOT. ERROR QNAME = *err:XPDY0050*	10507
–16022	OPERANDS OF TYPES *xquery-data-types* ARE NOT VALID FOR OPERATOR *operator-name* . ERROR QNAME = *err:XPTY0004*	10507
–16023	THE XQUERY PROLOG CANNOT CONTAIN MULTIPLE DECLARATIONS FOR THE SAME NAMESPACE PREFIX *ns-prefix*. ERROR QNAME = *err:XQST0033*	10503
–16024	THE NAMESPACE PREFIX *prefix-name* CANNOT BE REDECLARED OR CANNOT BE BOUND TO THE SPECIFIED URI. ERROR QNAME = *err:XQST0070*	10503
–16026	THE NAME *attribute-name* IS USED FOR MORE THAN ONE ATTRIBUTE IN THE CONSTRUCTOR FOR THE ELEMENT NAMED *element-name*. ERROR QNAME=*err:XQST0040*	10503
–16029	TWO OR MORE NAMESPACES WITHIN THE SAME XQUERY ELEMENT CONSTRUCTOR USE THE SAME NAMESPACE PREFIX *prefix-name*. ERROR QNAME=*err:XQST0071*	10503
–16031	XQUERY LANGUAGE FEATURE USING SYNTAX *string* IS NOT SUPPORTED	10509
–16032	THE STRING *string* IS NOT A VALID URI. ERROR QNAME = *err:XQST0046*	10504
–16033	THE TARGET DATA TYPE *type-name* OF A CASTABLE EXPRESSION IS NOT AN ATOMIC DATA TYPE DEFINED FOR THE IN-SCOPE XML SCHEMA TYPES OR IS A DATA TYPE THAT CANNOT BE USED IN A CASTABLE EXPRESSION. ERROR QNAME=*err:XPST0080*	10507
–16036	THE URI THAT IS SPECIFIED IN A NAMESPACE DECLARATION CANNOT BE A ZERO-LENGTH STRING	10504
-16038	THE ARGUMENTS OF FN:DATETIME HAVE DIFFERENT TIMEZONES. ERROR QNAME=*ERR:FORG0008*	10608
-16041	AN IMPLICIT OR EXPLICIT INVOCATION OF THE *fn:boolean* FUNCTION IN THE XQUERY EXPRESSION COULD NOT COMPUTE THE EFFECTIVE BOOLEAN VALUE OF THE SEQUENCE. ERROR QNAME=*err:FORG0006*	10608
–16046	A NUMERIC XQUERY EXPRESSION ATTEMPTED TO DIVIDE BY ZERO. ERROR QNAME = *err:FOAR0001*	10601

SQL Code	Explanation	SQL State
–16047	AN XQUERY EXPRESSION RESULTED IN ARITHMETIC OVERFLOW OR UNDERFLOW. ERROR QNAME= *err:FOAR0002*	10601
–16048	AN XQUERY PROLOG CANNOT CONTAIN MORE THAN ONE *declaration-type* DECLARATION. ERROR QNAME = *error-qname*	10502
–16049	THE LEXICAL VALUE value IS NOT VALID FOR THE *type-name* DATA TYPE IN THE FUNCTION OR CAST. ERROR QNAME = *err:FOCA0002*	10602
–16051	THE VALUE *value* OF DATA TYPE *source-type* IS OUT OF RANGE FOR AN IMPLICIT OR EXPLICIT CAST TO TARGET DATA TYPE *target-type*. ERROR QNAME = *err:error-qname*	10602
–16052	NAN CANNOT BE USED AS A FLOAT OR DOUBLE VALUE IN A DATETIME OPERATION. ERROR QNAME = *ERR:FOCA0005*	10602
–16055	AN ARITHMETIC OPERATION INVOLVING A DATETIME VALUE RESULTED IN OVERFLOW. ERROR QNAME = *ERR:FODT0001*	10605
-16056	AN ARITHMETIC OPERATION INVOLVING A DURATION VALUE RESULTED IN OVERFLOW. ERROR QNAME = *ERR:FODT0002*	10605
-16057	A TIMEZONE VALUE IS NOT VALID. ERROR QNAME = *ERR:FODT0003*	10605
–16061	THE VALUE value CANNOT BE CONSTRUCTED AS, OR CAST (USING AN IMPLICIT OR EXPLICIT CAST) TO THE DATA TYPE *data-type*. ERROR QNAME = *err:FORG0001*	10608
–16065	AN EMPTY SEQUENCE CANNOT BE CAST TO THE DATA TYPE *data-type*, ERROR QNAME = *err:FORG006*	10608
–16066	THE ARGUMENT PASSED TO THE AGGREGATE FUNCTION *function-name* IS NOT VALID. ERROR QNAME = *err:FORG0006*	10608
-16067	THE FLAGS ARGUMENT VALUE PASSED TO THE FUNCTION *function-name* IS NOT VALID. ERROR QNAME = err:FORX0001	10609
-16068	THE REGULAR EXPRESSION ARGUMENT VALUE PASSED TO THE FUNCTION *function-name* IS NOT VALID. ERROR QNAME= *err:FORX0002*	10609
-16069	A REGULAR EXPRESSION ARGUMENT *value* PASSED TO THE FUNCTION *function-name* MATCHES A ZERO-LENGTH STRING. ERROR QNAME= *err:FORX0003*	10609
–16075	THE SEQUENCE TO BE SERIALIZED CONTAINS AN ITEM THAT IS AN ATTRIBUTE NODE. ERROR QNAME = *err:SENR0001*	2200W
-16080	AN XQUERY *expression-type* UPDATING EXPRESSION IS USED IN AN INVALID CONTEXT. ERROR QNAME = *err:XUST0001*	10701
-16081	THE XQUERY-UPDATE-CONSTANT IN THE XMLMODIFY FUNCTION IS NOT AN UPDATING EXPRESSION OR AN EMPTY SEQUENCE EXPRESSION. ERROR QNAME = *err:XUST0002*	10702
-16083	INCOMPATIBLE EXPRESSION-TYPE EXPRESSIONS EXIST IN THE XQUERY-UPDATE-CONSTANT IN THE XMLMODIFY FUNCTION. QNAME = *err:error-name*	10704
-16085	THE TARGET NODE OF AN XQUERY *expression-type* EXPRESSION IS NOT VALID. ERROR QNAME = *err:error-name*	10703
-16086	THE REPLACEMENT SEQUENCE OF A REPLACE EXPRESSION CONTAINS INVALID NODES FOR THE SPECIFIED TARGET NODE. ERROR QNAME=*err:error-name*	10706
-16087	THE RESULT OF APPLYING THE UPDATING EXPRESSIONS IN THE XMLMODIFY FUNCTION IS NOT A VALID INSTANCE OF THE XQUERY AND XPATH DATA MODEL. ADDITIONAL INFORMATION: *information-1*, *information-2*. ERROR QNAME = *err:XUDY0021*	10707
-16088	AN *expression-type* EXPRESSION HAS A BINDING OF A NAMESPACE PREFIX *prefix-string* TO NAMESPACE URI *uri-string*, INTRODUCED TO AN ELEMENT NAMED *element-name*, THAT CONFLICTS WITH AN EXISTING NAMESPACE BINDING OF THE SAME PREFIX TO A DIFFERENT URI IN THE IN-SCOPE NAMESPACES OF THAT ELEMENT NODE. ERROR QNAME = *err:XUDY0023*	10708
-16089	AN *expression-type* EXPRESSION AND POSSIBLY OTHER UPDATING EXPRESSIONS IN AN XMLMODIFY FUNCTION INTRODUCE CONFLICTING NAMESPACE BINDINGS INTO AN ELEMENT NAMED *element-name*. THE PREFIX *prefix-string* IS BOUND TO *uri-string* WHILE ANOTHER BINDING OF THE SAME PREFIX USES A DIFFERENT NAMESPACE URI. ERROR QNAME = *err:XUDY0024*	10708
–16246	INCOMPLETE ANNOTATION MAPPING AT OR NEAR LINE	225DE

SQL Code	Explanation	SQL State
	lineno IN XML SCHEMA DOCUMENT *uri*. REASON CODE = *reason-code*	
–16247	SOURCE XML TYPE *source-data-type* CANNOT BE MAPPED TO TARGET SQL TYPE *target-data-type* IN THE ANNOTATION AT OR NEAR LINE *lineno* IN XML SCHEMA DOCUMENT *uri*	225DE
–16248	UNKNOWN ANNOTATION *annotation-name* AT OR NEAR LINE *lineno* IN XML SCHEMA DOCUMENT *uri*	225DE
–16249	THE *db2-xdb:expression* ANNOTATION *expression* AT OR NEAR LINE *lineno* IN XML SCHEMA DOCUMENT *uri* IS TOO LONG	225DE
–16250	THE *db2-xdb:defaultSQLSchema* WITH VALUE *schema-name* AT OR NEAR LINE *lineno* IN XML SCHEMA DOCUMENT *uri* CONFLICTS WITH ANOTHER *db2-xdb:defaultSQLSchema* SPECIFIED IN ONE OF THE XML SCHEMA DOCUMENTS WITHIN THE SAME XML SCHEMA	225DE
–16251	DUPLICATE ANNOTATION DEFINED FOR *object-name* AT OR NEAR *location* IN XML SCHEMA DOCUMENT *uri*	225DE
–16252	THE *db2-xdb:rowSet* NAME *rowset-name* SPECIFIED AT OR NEAR LINE *lineno* IN THE XML SCHEMA DOCUMENT *uri* IS ALREADY ASSOCIATED WITH ANOTHER TABLE	225DE
–16253	THE *db2-xdb:condition* ANNOTATION *condition* AT OR NEAR LINE *lineno* IN XML SCHEMA DOCUMENT *uri* IS TOO LONG	225DE
–16254	A *db2-xdb:locationPath locationpath* AT OR NEAR LINE *lineno* IN XML SCHEMA DOCUMENT *uri* IS NOT VALID WITH REASON CODE *reason-code*	225DE
–16255	A *db2-xdb:rowSet* VALUE *rowset-name* USED AT OR NEAR LINE *lineno* IN XML SCHEMA DOCUMENT *uri* CONFLICTS WITH A *db2-xdb:table* ANNOTATION WITH THE SAME NAME	225DE
–16257	XML SCHEMA FEATURE *feature* SPECIFIED IS NOT SUPPORTED FOR DECOMPOSITION	225DE
–16258	THE XML SCHEMA CONTAINS A RECURSIVE ELEMENT WHICH IS AN UNSUPPORTED FEATURE FOR DECOMPOSITION. THE RECURSIVE ELEMENT IS IDENTIFIED AS *elementnamespace : elementname* OF TYPE *typenamespace : typename*	225DE
–16259	INVALID MANY-TO-MANY MAPPINGS DETECTED IN XML SCHEMA DOCUMENT *uri1* NEAR LINE *lineno1* AND IN XML SCHEMA DOCUMENT *uri2* NEAR LINE *lineno2*	225DE
–16260	XML SCHEMA ANNOTATIONS INCLUDE NO MAPPINGS TO ANY COLUMN OF ANY TABLE	225DE
–16262	THE ANNOTATED XML SCHEMA HAS NO COLUMNS MAPPED FOR ROWSET *rowsetname*	225DE
–16265	THE XML DOCUMENT CANNOT BE DECOMPOSED USING XML SCHEMA *xsrobject-name* WHICH IS NOT ENABLED OR IS INOPERATIVE FOR DECOMPOSITION	225DE
–16266	AN SQL ERROR OCCURRED DURING DECOMPOSITION OF DOCUMENT *docid* WHILE ATTEMPTING TO INSERT DATA. INFORMATION RETURNED FOR THE ERROR INCLUDES SQLCODE *sqlcode*, SQLSTATE *sqlstate*, AND MESSAGE TOKENS *token-list*	225DE
+20002	THE *clause* SPECIFICATION IS IGNORED FOR OBJECT *object-name*	01624
–20003	GBCACHE NONE CANNOT BE SPECIFIED FOR TABLESPACE OR INDEX IN GRECP	560A7
–20004	8K or 16K BUFFERPOOL PAGESIZE INVALID FOR A WORKFILE OBJECT	560A8
–20005	THE INTERNAL ID LIMIT OF *limit* HAS BEEN EXCEEDED FOR OBJECT TYPE *object-type*	54035
+20007	USE OF OPTIMIZATION HINTS IS DISALLOWED BY A DB2 SUBSYSTEM PARAMETER. THE SPECIAL REGISTER 'OPTIMIZATION HINT' IS SET TO AN EMPTY STRING.	01602
–20008	ATTEMPT TO USE AN UNSUPPORTED FEATURE ON OBJECT *object-name*. REASON CODE: *reason-code*	560A9
-20016	THE VALUE OF THE INLINE LENGTH ASSOCIATED WITH *object-name* IS TOO BIG OR THE INLINE LENGTH CLAUSE IS NOT ALLOWED IN THE CONTEXT	429B2
-20019	THE RESULT TYPE RETURNED FROM THE FUNCTION BODY CANNOT BE ASSIGNED TO THE DATA TYPE DEFINED IN THE RETURNS CLAUSE	42866
-20038	THE STATEMENT COULD NOT BE PROCESSED BECAUSE THE STATEMENT CONTAINS THE FOLLOWING INCOMPATIBLE CLAUSES OR ELEMENTS: *keywords1* AND *keywords2*	42613
-20046	THE SELECTIVITY CLAUSE FOLLOWING *predicate-string* CAN ONLY SPECIFIED FOR A SPATIAL PREDICATE FUNCTION	428E5

SQL Code	Explanation	SQL State
–20058	THE FULLSELECT SPECIFIED FOR MATERIALIZED QUERY TABLE *table-name* IS NOT VALID	428EC
–20060	UNSUPPORTED DATA TYPE *data-type* ENCOUNTERED IN SQL *object-type object-name*	560AB
–20070	AUXILIARY TABLE *table-name* CANNOT BE CREATED BECAUSE COLUMN *column-name* IS NOT A LOB COLUMN	53098
–20071	WLM ENVIRONMENT NAME MUST BE SPECIFIED *function-name*	53099
–20072	*csect-name bind-type bind-subtype* ERROR USING *authorization-id* AUTHORITY. OPERATION IS NOT ALLOWED ON A *package-type* PACKAGE *package-name*	56052
–20073	THE FUNCTION function-name CANNOT BE ALTERED BECAUSE IT IS REFERENCED IN EXISTING VIEW OR MATERIALIZED QUERY TABLE DEFINITIONS	42927
–20074	THE OBJECT object-name CANNOT BE CREATED BECAUSE THE FIRST THREE CHARACTERS ARE RESERVED FOR SYSTEM OBJECTS	42939
–20091	A VIEW NAME WAS SPECIFIED AFTER LIKE IN ADDITION TO THE INCLUDING IDENTITY COLUMN ATTRIBUTES CLAUSE	560AD
–20092	A TABLE OR VIEW WAS SPECIFIED IN THE LIKE CLAUSE BUT THE OBJECT CANNOT BE USED IN THIS CONTEXT	560AE
–20093	THE TABLE *table-name* CANNOT BE CONVERTED TO OR FROM A MATERIALIZED QUERY TABLE, OR THE MATERIALIZED QUERY TABLE PROPERTY CANNOT BE ALTERED. REASON CODE = *reason-code*	428EW
-20094	THE COLUMN *column-name* IS A GENERATED COLUMN AND CANNOT BE USED IN THE BEFORE TRIGGER *trigger-name*	42989
–20100	AN ERROR OCCURRED WHEN BINDING A TRIGGERED SQL STATEMENT. INFORMATION RETURNED: SECTION NUMBER: *section-number* SQLCODE *sqlerror*, SQLSTATE *sqlstate*, AND MESSAGE TOKENS *token-list*	56059
–20101	THE FUNCTION *function-name* FAILED WITH REASON *reason-code*	56060
–20102	CREATE OR ALTER STATEMENT FOR ROUTINE *routine-name* SPECIFIED THE *option* OPTION WHICH IS NOT ALLOWED FOR THE TYPE OF ROUTINE	42849
–20104	AN ATTEMPT TO ALTER A CCSID FROM *from-ccsid* TO *to-ccsid* FAILED	42856
–20106	THE CCSID FOR TABLE SPACE OR DATABASE CANNOT BE CHANGED BECAUSE THE TABLE SPACE OR DATABASE ALREADY CONTAINS A TABLE THAT IS REFERENCED IN EXISTING VIEW OR MATERIALIZED QUERY TABLE DEFINITIONS OR AN EXTENDED INDEX	42945
–20107	HOST VARIABLE OR PARAMETER NUMBER *position-number* CANNOT BE USED AS SPECIFIED BECAUSE REASON *reason*	53022
–20108	A RESULT SET CONTAINS AN UNSUPPORTED DATA TYPE IN POSITION NUMBER *position-number* FOR CURSOR *cursor-name* OPENED BY STORED PROCEDURE *procedure-name*	56084
–20110	CANNOT IMPLICITLY CONNECT TO A REMOTE SITE WITH A SAVEPOINT OUTSTANDING	51036
–20111	CANNOT ISSUE SAVEPOINT, RELEASE SAVEPOINT, ROLLBACK TO SAVEPOINT FROM A TRIGGER, FROM A USER-DEFINED FUNCTION, OR FROM A GLOBAL TRANSACTION	3B503
-20017	A WINDOW SPECIFICATION FOR AN OLAP SPECIFICATION IS NOT VALID. REASON CODE = *reason-code*	428EZ
-20120	AN SQL TABLE FUNCTION MUST RETURN A TABLE RESULT	428F1
+20122	DEFINE NO OPTION IS NOT APPLICABLE IN THE CONTEXT SPECIFIED	01644
–20123	CALL TO STORED PROCEDURE *procedure* FAILED BECAUSE THE RESULT SET RETURNED FOR CURSOR *cursor-name* IS SCROLLABLE, BUT THE CURSOR IS NOT POSITIONED BEFORE THE FIRST ROW	560B1
–20124	OPEN CURSOR *cursor* FAILED BECAUSE THE CURSOR IS SCROLLABLE, BUT THE CLIENT DOES NOT SUPPORT THIS	560B2
–20125	CALL TO STORED PROCEDURE *procedure-name* FAILED BECAUSE THE RESULT SET FOR CURSOR *cursor-name* IS SCROLLABLE, BUT THE CLIENT DOES NOT SUPPORT THIS, REASON *reason-code*	560B3
–20127	VALUE SPECIFIED ON FETCH STATEMENT FOR ABSOLUTE OR RELATIVE IS TOO LARGE FOR DRDA	54051
+20141	TRUNCATION OF VALUE WITH LENGTH *length* OCCURRED FOR *hv-or-parm-number*	01004
–20142	SEQUENCE *sequence-name* CANNOT BE USED AS SPECIFIED	428FB
–20143	THE ENCRYPTION OR DECRYPTION FUNCTION FAILED, BECAUSE THE ENCRYPTION PASSWORD VALUE IS NOT SET	51039

SQL Code	Explanation	SQL State
–20144	THE ENCRYPTION IS INVALID BECAUSE THE LENGTH OF THE PASSWORD WAS LESS THAN 6 BYTES OR GREATER THAN 127 BYTES	428FC
–20146	THE DECRYPTION FAILED, THE DATA IS NOT ENCRYPTED	428FE
–20147	THE ENCRYPTION FUNCTION FAILED. MULTIPLE PASS ENCRYPTION IS NOT SUPPORTED	55048
-20148	A RETURN STATEMENT DOES NOT EXIST OR WAS NOT INVOKED DURING THE EXECUTION OF ROUTINE *routine-name* WITH SPECIFIC NAME *specific-name*	429BD
–20163	HEXADECIMAL CONSTANT GX IS NOT ALLOWED	560B9
–20165	AN SQL DATA CHANGE STATEMENT WITHIN A FROM CLAUSE IS NOT ALLOWED IN THE CONTEXT IN WHICH IT WAS SPECIFIED	428FL
–20166	AN SQL DATA CHANGE STATEMENT WITHIN A SELECT SPECIFIED A VIEW *view-name* WHICH IS NOT A SYMMETRIC VIEW OR COULD NOT HAVE BEEN DEFINED AS A SYMMETRIC VIEW	428FM
–20177	SET DATA TYPE CLAUSE ON ALTER TABLE SPECIFIED FLOATING POINT, BUT THIS CHANGE IS DISALLOWED	530A1
–20178	VIEW *view-name* ALREADY HAS AN INSTEAD OF *operation* TRIGGER DEFINED	428FP
–20179	THE INSTEAD OF TRIGGER CANNOT BE CREATED BECAUSE THE VIEW *view-name* IS DEFINED USING THE WITH CHECK OPTION	428FQ
–20180	COLUMN column-name IN TABLE table-name CANNOT BE ALTERED AS SPECIFIED	428FR
–20181	COLUMN CANNOT BE ADDED TO INDEX index-name	428FS
–20182	PARTITIONING CLAUSE *clause* ON *stmt-type* STATEMENT FOR *index-name* IS NOT VALID	530A2
–20183	THE PARTITIONING, ADD PARTITION, ADD PARTITIONING KEY, ALTER PARTITION, ROTATE PARTITION OR PARTITION BY RANGE CLAUSE SPECIFIED ON CREATE OR ALTER FOR table-name IS NOT VALID	428FT
–20185	CURSOR cursor-name IS NOT DEFINED TO ACCESS ROWSETS, BUT A CLAUSE WAS SPECIFIED THAT IS VALID ONLY WITH ROWSET ACCESS	24518
–20186	A CLAUSE SPECIFIED FOR THE DYNAMIC SQL STATEMENT BEING PROCESSED IS NOT VALID	07501
+20187	ROLLBACK TO SAVEPOINT CAUSED A NOT LOGGED TABLE SPACE TO BE PLACED IN THE LPL	01656
–20200	THE INSTALL OR REPLACE OF *jar-id* WITH URL *url* FAILED DUE TO REASON *reason-code (reason-string)*	46001
–20201	THE INSTALL, REPLACE, REMOVE OR ALTER OF *jar-name* FAILED DUE TO REASON CODE *reason-code (reason-string)*	46002
–20202	THE REMOVE OF *jar-name* FAILED AS *class* IS IN USE	46003
–20203	USER-DEFINED FUNCTION OR PROCEDURE *name* HAS A JAVA METHOD WITH AN INVALID SIGNATURE. THE ERROR IS AT OR NEAR PARAMETER *number*. THE SIGNATURE IS *signature*	46007
–20204	THE USER-DEFINED FUNCTION OR PROCEDURE *routine-name* WAS UNABLE TO MAP TO A SINGLE JAVA METHOD	46008
–20207	THE INSTALL OR REMOVE OF *jar-name* SPECIFIED THE USE OF A DEPLOYMENT DESCRIPTOR	46501
–20210	THE SQL STATEMENT CANNOT BE EXECUTED BECAUSE IT WAS PRECOMPILED AT A LEVEL THAT IS INCOMPATIBLE WITH THE CURRENT VALUE OF THE ENCODING BIND OPTION OR SPECIAL REGISTER	560B8
–20211	THE SPECIFICATION ORDER BY, OFFSET, OR FETCH FIRST N ROWS ONLY IS INVALID	428FJ
–20212	USER-DEFINED ROUTINE *name* ENCOUNTERED AN EXCEPTION ATTEMPTING TO LOAD JAVA CLASS *class-name* FROM JAR *jar-name*. ORIGINAL EXCEPTION: *exception-string*	46103
–20213	STORED PROCEDURE *procedure-name* HAS RETURNED A DYNAMIC RESULT SET, PARAMETER *number*,THAT IS NOT VALID	46502
–20223	THE OPERATION FAILED. ENCRYPTION FACILITY NOT AVAILABLE return-code, reason-code	560BF
+20224	ENCRYPTED DATA THAT WAS ORIGINALLY A BINARY STRING CANNOT BE DECRYPTED TO A CHARACTER STRING	01658
–20224	ENCRYPTED DATA THAT WAS ORIGINALLY A BINARY STRING CANNOT BE DECRYPTED TO A CHARACTER STRING	22528
–20227	REQUIRED CLAUSE IS MISSING FOR ARGUMENT *number* OF *expression*	42633
-20228	A STACKED DIAGNOSTICS AREA IS NOT AVAILABLE	0Z002
–20232	CHARACTER CONVERSION FROM CCSID *from-ccsid* TO *to-*	57017

SQL Code	Explanation	SQL State
	ccsid FAILED WITH ERROR CODE *error-code* FOR TABLE *dbid.obid* COLUMN *column-number* REQUESTED BY *csect-name*	
–20235	THE COLUMN *column-name* CANNOT BE ADDED OR ALTERED BECAUSE *table-name* IS A MATERIALIZED QUERY TABLE	428FY
+20237	FETCH PRIOR ROWSET FOR CURSOR *cursor-name* RETURNED A PARTIAL ROWSET	02504
–20240	INVALID SPECIFICATION OF A SECURITY LABEL COLUMN *table-name.column-name* REASON CODE *reason-code*	42963
+20245	NOT PADDED CLAUSE IS IGNORED FOR INDEXES CREATED ON AUXILIARY TABLES	01663
–20248	ATTEMPTED TO EXPLAIN STATEMENT WITH STMTID OR STMTTOKEN *ID-token* BUT THE REQUIRED EXPLAIN INFORMATION IS NOT ACCESSIBLE. REASON *reason-code*	26501
–20249	THE PACKAGE *package-name* NEEDS TO BE REBOUND IN ORDER TO BE SUCCESSFULLY EXECUTED (*required-maintenance)*	560C5
–20252	DIAGNOSTICS AREA FULL. NO MORE ERRORS CAN BE RECORDED FOR THE NOT ATOMIC STATEMENT	429BI
–20257	FINAL TABLE IS NOT VALID WHEN THE TARGET VIEW *view-name* OF THE SQL DATA CHANGE STATEMENT IN A FULLSELECT HAS AN INSTEAD OF TRIGGER DEFINED	428G3
–20258	INVALID USE OF INPUT SEQUENCE ORDERING	428G4
–20260	THE ASSIGNMENT CLAUSE OF THE UPDATE OPERATION AND THE VALUES CLAUSE OF THE INSERT OPERATION MUST SPECIFY AT LEAST ONE COLUMN THAT IS NOT AN INCLUDE COLUMN	428G5
–20264	FOR TABLE *table-name, primary-auth-id* WITH SECURITY LABEL *primary-auth-id-seclabel*, IS NOT AUTHORIZED TO PERFORM *operation* ON A ROW WITH SECURITY LABEL *row-seclabel.* THE RECORD IDENTIFIER(RID) OF THIS ROW IS *rid-number*	42512
–20265	SECURITY LABEL IS *reason* FOR *primary-auth-id*	42501
–20266	ALTER VIEW FOR *view-name* FAILED	560C7
-20267	THE FUNCTION *function-name* (SPECIFIC *specific-name*) MODIFIES SQL DATA AND IS INVOKED IN AN ILLEGAL CONTEXT. REASON CODE *reason-code*	429BL
+20270	OPTION NOT SPECIFIED FOLLOWING ALTER PARTITION CLAUSE	01664
+20271	THE NAME AT ORDINAL POSITION *position-number* IN THE STATEMENT, WITH NAME *object-name*, WAS TRUNCATED	01665
+20272	TABLE SPACE *table-space-name* HAS BEEN CONVERTED TO USE TABLE-CONTROLLED PARTITIONING INSTEAD OF INDEX-CONTROLLED PARTITIONING, ADDITIONAL INFORMATION: *old-limit-key-value*	01666
–20275	The XML NAME *xml-name* IS NOT VALID. REASON CODE = *reason-code*	42634
-20276	The XML namespace prefix *xml-namespace-prefix* is not valid. Reason code = *reason-code*	42635
+20278	THE VIEW *view-name* MAY NOT BE USED TO OPTIMIZE THE PROCESSING OF QUERIES	01667
-20279	THE VIEW *view-name* CANNOT BE ENABLED FOR QUERY OPTIMIZATION. REASON CODE = *reason-code*	428G8
–20281	*primary-auth-id* DOES NOT HAVE THE MLS WRITE_DOWN PRIVILEGE	42513
–20283	A DYNAMIC CREATE STATEMENT CANNOT BE PROCESSED WHEN THE VALUE OF CURRENT SCHEMA DIFFERS FROM CURRENT SQLID	429BN
–20286	DB2 CONVERTED STRING *token-type token* FROM *from-ccsid* TO *to-ccsid,* AND RESULTED IN SUBSTITUTION CHARACTERS	428GB
–20289	INVALID STRING UNIT *unit* SPECIFIED FOR FUNCTION *function-name*	428GC
–20295	THE EXECUTION OF A BUILT IN FUNCTION *function* RESULTED IN AN ERROR REASON CODE *reason-code*	22531
-20300	THE LIST OF COLUMNS SPECIFIED FOR THE *clause* CLAUSE IS NOT ALLOWED IN COMBINATION WITH THE LIST OF COLUMNS FOR THE PARTITIONING KEY FOR THE TABLE	428GD
–20304	INVALID INDEX DEFINITION INVOLVING AN XMLPATTERN CLAUSE OR A COLUMN OF DATA TYPE XML. REASON CODE = *reason-code*	429BS
–20305	AN XML VALUE CANNOT BE INSERTED OR UPDATED BECAUSE OF AN ERROR DETECTED WHEN INSERTING OR UPDATING THE INDEX IDENTIFIED BY *index-id* ON TABLE *table-name*. REASON CODE = *reason-code*	23525

SQL Code	Explanation	SQL State
–20306	AN INDEX ON AN XML COLUMN CANNOT BE CREATED BECAUSE OF AN ERROR DETECTED WHEN INSERTING THE XML VALUES INTO THE INDEX. REASON CODE = *reason-code*	23526
–20310	THE REMOVE OF *jar-name1* FAILED, AS IT IS IN USE BY *jar-name2*	4600C
–20311	THE VALUE PROVIDED FOR THE NEW JAVA PATH IS ILLEGAL	4600D
–20312	THE ALTER OF JAR *jar-id* FAILED BECAUSE THE SPECIFIED PATH REFERENCES ITSELF	4600E
–20313	DEBUG MODE OPTION FOR ROUTINE *routine-name* CANNOT BE CHANGED	55058
–20314	THE PARAMETER LIST (OR OPTION) DOES NOT MATCH THE PARAMETER LIST (OR OPTION) FOR ALL OTHER VERSIONS OF ROUTINE *routine-name*	428GH
–20315	THE CURRENTLY ACTIVE VERSION FOR OBJECT object-name (object-type) CANNOT BE DROPPED	55059
-20326	AN XML ELEMENT NAME, ATTRIBUTE NAME, NAMESPACE PREFIX, OR URI ENDING WITH *string-end* EXCEEDS THE LIMIT OF 1000 BYTES	54057
–20327	THE DEPTH OF AN XML DOCUMENT EXCEEDS THE LIMIT OF 128 LEVELS	54058
–20328	THE DOCUMENT WITH TARGET NAMESPACE *namespace* AND SCHEMA LOCATION *location* HAS ALREADY BEEN ADDED FOR THE XML SCHEMA IDENTIFIED BY *schema name*	42749
–20329	THE COMPLETION CHECK FOR THE XML SCHEMA FAILED BECAUSE ONE OR MORE XML SCHEMA DOCUMENTS IS MISSING. ONE MISSING XML SCHEMA DOCUMENT IS IDENTIFIED BY *uri-type* AS *uri*	428GI
–20330	THE *object-type* IDENTIFIED BY XML *uri-type1 uri1* AND XML *uri-type2 uri2* IS NOT FOUND IN THE XML SCHEMA REPOSITORY	4274A
–20331	THE XML COMMENT VALUE *string* IS NOT VALID	2200S
–20332	THE XML PROCESSING INSTRUCTION VALUE *string* IS NOT VALID	2200T
–20335	MORE THAN ONE *xsrobject-type* EXISTS IDENTIFIED BY XML *uri-type1 uri1* AND *uri-type2 uri2* EXISTS IN THE XML SCHEMA REPOSITORY	22533
-20338	THE DATA TYPE OF EITHER THE SOURCE OR TARGET OPERAND OF AN XMLCAST SPECIFICATION MUST BE XML	42815
–20339	XML SCHEMA *name* IS NOT IN THE CORRECT STATE TO PERFORM OPERATION *operation*	55063
–20340	XML SCHEMA *xmlschema-name* INCLUDES AT LEAST ONE XML SCHEMA DOCUMENT IN NAMESPACE *namespace* THAT IS NOT CONNECTED TO THE OTHER XML SCHEMA DOCUMENTS	22534
+20341	OWNERSHIP TRANSFER WAS IGNORED BECAUSE *auth-id* IS ALREADY THE OWNER OF THE OBJECT.	01676
-20342	AUTHORIZATION ID *auth-ID* DOES NOT HAVE THE REQUIRED PRIVILEGE *privilege-name* ON OBJECT *object-name* OF TYPE *type-name* FOR OWNERSHIP TRANSFER	42514
–20345	THE XML VALUE IS NOT A WELL-FORMED DOCUMENT WITH A SINGLE ROOT ELEMENT	2200L
+20348	THE PATH VALUE HAS BEEN TRUNCATED	01011
–20353	AN OPERATION INVOLVING COMPARISON CANNOT USE OPERAND *name* DEFINED AS DATA TYPE *type-name*	42818
–20354	INVALID SPECIFICATION OF A ROW CHANGE TIMESTAMP COLUMN FOR TABLE *table-name*	429BV
–20356	THE TABLE WITH DBID = *dbid* AND OBID = *obid* CANNOT BE TRUNCATED BECAUSE DELETE TRIGGERS EXIST FOR THE TABLE, OR THE TABLE IS THE PARENT TABLE IN A REFERENTIAL CONSTRAINT	428GJ
+20360	TRUSTED CONNECTION CAN NOT BE ESTABLISHED FOR SYSTEM AUTHID *authorization-name*	01679
–20361	AUTHORIZATION ID *authorization-name* IS NOT DEFINED FOR THE TRUSTED CONTEXT *context-name*	42517
–20362	ATTRIBUTE *attribute-name* WITH VALUE *value* CANNOT BE DROPPED BECAUSE IT IS NOT PART OF THE DEFINITION OF TRUSTED CONTEXT *context-name*	4274C
–20363	ATTRIBUTE *attribute-name* WITH VALUE *value* IS NOT A UNIQUE SPECIFICATION FOR TRUSTED CONTEXT *context-name*	4274D
+20365	A SIGNALING NAN WAS ENCOUNTERED, OR AN EXCEPTION OCCURRED IN AN ARITHMETIC OPERATION OR FUNCTION INVOLVING A DECFLOAT	01565
–20365	A SIGNALING NAN WAS ENCOUNTERED, OR AN EXCEPTION OCCURRED IN AN ARITHMETIC OPERATION OR FUNCTION INVOLVING A DECFLOAT	22502

SQL Code	Explanation	SQL State
–20366	TABLE WITH DBID=database-id AND OBID=object-id CANNOT BE TRUNCATED BECAUSE UNCOMMITTED UPDATES EXIST ON THE TABLE WITH 'IMMEDIATE' OPTION SPECIFIED IN THE STATEMENT	57007
+20367	OPTION *clause* IS NOT SUPPORTED IN THE CONTEXT IN WHICH IT WAS SPECIFIED	01680
+20368	TRUSTED CONTEXT *context-name* IS NO LONGER DEFINED TO BE USED BY SPECIFIC VALUES FOR ATTRIBUTE attribute-*name*	01681
–20369	AN ALTER TRUSTED CONTEXT STATEMENT FOR context-name ATTEMPTED TO REMOVE THE LAST CONNECTION TRUST ATTRIBUTE ASSOCIATED WITH THE TRUSTED CONTEXT	428GK
+20371	THE ABILITY TO USE TRUSTED CONTEXT *context-name* WAS REMOVED FROM SOME, BUT NOT ALL AUTHORIZATION IDS SPECIFIED IN THE STATEMENT	01682
–20372	THE SYSTEM AUTHID CLAUSE OF A CREATE OR ALTER TRUSTED CONTEXT STATEMENT FOR *context-name* SPECIFIED *authorization-name*, BUT ANOTHER TRUSTED CONTEXT IS ALREADY DEFINED FOR THAT AUTHORIZATION ID	428GL
–20373	A CREATE OR ALTER TRUSTED CONTEXT STATEMENT SPECIFIED *authorization-name* MORE THAN ONCE OR THE TRUSTED CONTEXT IS ALREADY DEFINED TO BE USED BY THIS AUTHORIZATION ID, PROFILE NAME, ID OR PUBLIC	428GM
–20374	AN ALTER TRUSTED CONTEXT STATEMENT FOR *context-name* SPECIFIED *authorization-name* BUT THE TRUSTED CONTEXT IS NOT CURRENTLY DEFINED TO BE USED BY THIS AUTHORIZATION ID, PROFILE NAME, OR PUBLIC	428GN
–20377	AN ILLEGAL XML CHARACTER *hex-char* WAS FOUND IN AN SQL/XML EXPRESSION OR FUNCTION ARGUMENT THAT BEGINS WITH STRING *start-string*	0N002
+20378	A NON-ATOMIC *statement* STATEMENT SUCCESSFULLY COMPLETED FOR SOME OF THE REQUESTED ROWS, POSSIBLY WITH WARNINGS, AND ONE OR MORE ERRORS, AND THE CURSOR CAN BE USED	01683
-20379	AN AUTHORIZATION ID OR A ROLE CANNOT USE ITS SECADM AUTHORITY TO TRANSFER THE OWNERSHIP OF AN OBJECT TO ITSELF	42502
–20380	ALTER INDEX WITH REGENERATE OPTION FOR *index-name* FAILED. INFORMATION RETURNED: SQLCODE *sqlcode*, SQLSTATE *sqlstate,* MESSAGE TOKENS *token-list*	560CC
–20381	ALTER INDEX WITH REGENERATE OPTION IS NOT VALID FOR *index-name*	530A5
–20382	CONTEXT ITEM CANNOT BE A SEQUENCE WITH MORE THAN ONE ITEM	2200V
-20385	THE STATEMENT CANNOT BE PROCESSED BECAUSE THERE ARE PENDING DEFINITION CHANGES FOR OBJECT *object-name* OF TYPE *object- type* (REASON *reason-code*)	57007
–20398	ERROR ENCOUNTERED DURING XML PARSING AT LOCATION *n text*	2200M
–20399	ERROR ENCOUNTERED DURING XML VALIDATION: LOCATION *n*; TEXT: *text*; XSRID *schema-ID*	2201R
–20400	XML SCHEMA ERROR *n text*	2200M
–20409	AN XML DOCUMENT OR CONSTRUCTED XML VALUE CONTAINS A COMBINATION OF XML NODES THAT CAUSES AN INTERNAL IDENTIFIER LIMIT TO BE EXCEEDED	560CG
–20410	THE NUMBER OF CHILDREN NODES OF AN XML NODE IN AN XML VALUE HAS EXCEEDED THE LIMIT NUMBER OF CHILDREN NODES	560CH
–20411	A FETCH CURRENT CONTINUE OPERATION WAS REQUESTED FOR *cursor-name* BUT THERE IS NO PRESERVED, TRUNCATED DATA TO RETURN	24524
–20412	SERIALIZATION OF AN XML VALUE RESULTED IN CHARACTERS THAT COULD NOT BE REPRESENTED IN THE TARGET ENCODING	2200W
–20422	A CREATE TABLE, OR DECLARE GLOBAL TEMPORARY TABLE, OR ALTER TABLE STATEMENT FOR *table-name* ATTEMPTED TO CREATE A TABLE WITH ALL THE COLUMNS DEFINED AS HIDDEN	428GU
-20423	ERROR OCCURRED DURING TEXT SEARCH PROCESSING (*server*, *index-name*, *text*)	38H10
-20424	TEXT SEARCH SUPPORT IS NOT AVAILABLE *reason-code*	38H11
-20425	*column-name* (IN *table-name*) WAS SPECIFIED AS AN ARGUMENT TO A TEXT SEARCH FUNCTION, BUT A TEXT INDEX DOES NOT EXIST FOR THE COLUMN	38H12

SQL Code	Explanation	SQL State
-20426	CONFLICTING TEXT SEARCH ADMINISTRATION STORED PROCEDURE RUNNING ON THE SAME INDEX	38H13
-20427	ERROR OCCURRED DURING TEXT SEARCH ADMINISTRATION STORED PROCEDURE *error*	38H14
-20428	URI SPECIFIED IN THE XMLSCHEMA CLAUSE IS AN EMPTY STRING	428GV
-20430	GLOBAL VARIABLE *variable-name* CANNOT BE SET IN THIS CONTEXT	428GX
-20433	AN UNTYPED PARAMETER MARKER WAS SPECIFIED, BUT AN ASSUMED DATA TYPE CANNOT BE DETERMINED FROM ITS USE	429C1
-20434	AN UPDATE OPERATION HAS SET ALL OF ITS TARGET COLUMNS TO UNASSIGNED	22540
-20435	THE SELECT CLAUSE INCLUDES MULTIPLE INVOCATIONS OF THE ARRAY_AGG FUNCTION. ALL INVOCATIONS THAT EXPLICITLY SPECIFY AN ORDER BY CLAUSE MUST SPECIFY THE SAME ORDER	428GZ
-20436	THE DATA TYPE SPECIFIED FOR AN ARRAY TYPE IS NOT VALID	429C2
-20437	AN ARRAY INDEX CANNOT BE APPLIED TO AN OBJECT THAT IS NOT AN ARRAY	428H0
-20438	THE DATA TYPE OF THE EXPRESSION FOR AN ARRAY INDEX VALUE IS NOT CASTABLE TO THE DATA TYPE OF THE ARRAY INDEX	428H1
-20439	AN ARRAY INDEX WITH VALUE *value* IS NULL OR OUT OF RANGE, OR AN ARRAY ELEMENT WITH THAT INDEX DOES NOT EXIST	2202E
-20440	THE ARRAY VALUE WITH CARDINALITY *cardinality* HAS TOO MANY ELEMENTS FOR THE REQUESTED OPERATION. THE MAXIMUM NUMBER OF ELEMENTS ALLOWED FOR THE REQUESTED OPERATION IS *value*	2202F
-20441	*type-name* TYPE IS NOT VALID WHERE SPECIFIED. REASON CODE *reason-code*	428H2
-20442	THERE IS NOT ENOUGH STORAGE TO REPRESENT THE ARRAY VALUE	57011
-20444	AN ERROR OCCURRED IN A KEY-EXPRESSION EVALUATION IN *index-name* INFORMATION RETURNED: SQLCODE: *sqlcode*, SQLSTATE: *sqlstate*, MESSAGE TOKEN *token-list* AND RID X *rid*	560CM
-20447	FORMAT STRING *format-string* IS NOT VALID FOR THE *function-name* FUNCTION	22007
-20448	*string-expression* CANNOT BE INTERPRETED USING FORMAT STRING *format-string* FOR THE TIMESTAMP_FORMAT FUNCTION	22007
-20457	THE PROCEDURE *procedure-name* HAS ENCOUNTERED AN UNSUPPORTED VERSION, *version*, FOR PARAMETER *number*	38554
+20458	THE PROCEDURE *procedure-name* HAS ENCOUNTERED AN INTERNAL PARAMETER PROCESSING ERROR IN PARAMETER *number1*. THE VALUE FOR PARAMETER *number2* CONTAINS FURTHER INFORMATION ABOUT THE ERROR	01H54
+20459	THE PROCEDURE *procedure-name* HAD ENCOUNTERED AN INTERNAL PROCESSING ERROR. THE VALUE FOR PARAMETER *number* CONTAINS FURTHER INFORMATION ABOUT THE ERROR	01H55
+20460	THE PROCEDURE *procedure-name* SUPPORTS A HIGHER VERSION, *version1*. THAN THE SPECIFIED VERSION, *version2*, FOR PARAMETER *number*	01H56
+20461	THE PROCEDURE *procedure-name* RETURNED OUTPUT IN THE ALTERNATE LOCALE, *locale1*, INSTEAD OF THE LOCALE *locale2*, SPECIFIED IN PARAMETER *number*	01H57
-20465	THE BINARY XML VALUE IS INCOMPLETE OR CONTAINS UNRECOGNIZED DATA AT LOCATION *position* WITH THE HEX DATA *text*	22541
-20467	THE STATEMENT WAS NOT EXECUTED BECAUSE AN EXPRESSION IS NOT A CONSTANT OR VARIABLE. THE INVALID EXPRESSION IS IN THE STATEMENT NEAR THE SYNTAX ELEMENT *syntax-element*	428H7
+20468	THE COMBINATION OF TARGET NAMESPACE *target-namespace* AND SCHEMA LOCATION HINT *location-hint* IS NOT UNIQUE IN THE DB2 XML SCHEMA REPOSITORY	0168X
-20469	ROW OR COLUMN ACCESS CONTROL CANNOT BE ACTIVATED FOR TABLE *table-name* FOR REASON *reason-code*. *object-type object-name* IS NOT IN A VALID STATE FOR ACTIVATING ACCESS CONTROL FOR THIS TABLE	55019
-20470	*object-type1 object-name1* MUST BE DEFINED AS SECURE BECAUSE *object-type2 object-name2* IS DEPENDENT ON IT	248H8

SQL Code	Explanation	SQL State
-20471	THE INSERT OR UPDATE IS NOT ALLOWED BECAUSE A RESULTING ROW DOES NOT SATISFY ROW PERMISSIONS	22542
-20472	PERMISSION OR MASK *object-name* CANNOT BE ALTERED AS SPECIFIED. REASON CODE *reason-code*	428H9
-20473	THE INPUT ARGUMENT OF FUNCTION *function-name* THAT IS DEFINED WITH THE NOT-SECURED OPTION MUST NOT REFERENCE COLUMN *column-name* FOR WHICH A COLUMN MASK IS ENABLED AND THE COLUMN ACCESS CONTROL IS ACTIVATED FOR THE TABLE	428HA
-20474	PERMISSION OR MASK CANNOT BE CREATED FOR THE *object-name* OBJECT OF THE *object-type* TYPE. REASON CODE *reason-code*	428HB
-20475	A COLUMN MASK IS ALREADY DEFINED FOR THE COLUMN *column-name* IN TABLE *table-name* (EXISTING MASK NAME *mask-name*)	428HC
-20476	THE function-name FUNCTION WAS INVOKED WITH AN INVALID FORMAT STRING format-string	22018
-20477	THE function-name FUNCTION IS NOT ABLE TO USE FORMAT STRING format-string TO INTERPRET THE ARGUMENT string-expression	22018
-20478	THE STATEMENT CANNOT BE PROCESSED BECAUSE COLUMN MASK *mask-name* (DEFINED FOR COLUMN *column-name*) EXISTS AND THE COLUMN MASK CANNOT BE APPLIED OR THE DEFINITION OF THE MASK CONFLICTS WITH THE REQUESTED STATEMENT. REASON CODE *reason-code*	428HD
-20479	THE SOURCE TABLE *table-name* CANNOT BE ALTERED AS SPECIFIED BECAUSE THE TABLE IS INVOLVED IN ROW OR COLUMN ACCESS CONTROLS. REASON CODE *reason-code*	42917
-20487	HASH ORGANIZATION CLAUSE IS NOT VALID FOR *table-name*	428HJ
-20488	SPECIFIED HASH SPACE IS TOO LARGE FOR THE IMPLICITLY CREATED TABLE SPACE. REASON *reason-code*. (PARTITION *partition-number*)	428HK
-20490	A VERSIONING CLAUSE WAS SPECIFIED FOR TABLE *table-name*, BUT THE TABLE CANNOT BE USED AS A SYSTEM PERIOD TEMPORAL TABLE. REASON CODE = *reason-code*	428HM
-20491	INVALID SPECIFICATION OF PERIOD *period-name*. REASON CODE = *reason-code*	428HN
-20493	A TIMESTAMP WITHOUT TIME ZONE VALUE CANNOT BE ASSIGNED TO A TIMESTAMP WITH TIME ZONE TARGET	22007
-20494	A PUBLIC ALIAS NAME, *name*, CAN ONLY BE QUALIFIED WITH SYSPUBLIC AND NOT THE SCHEMA NAME *schema-name*	428EK
-20497	A STRING REPRESENTATION OF A DATETIME VALUE THAT CONTAINS A TIME ZONE CANNOT BE IMPLICITLY OR EXPLICITLY CAST TO A TARGET DEFINED AS DATETIME WITHOUT TIME ZONE	22007
-20505	THE WITH ORDINALITY CLAUSE IS NOT VALID WITH UNNEST OF AN ASSOCIATIVE ARRAY	428HT
-20517	XMLMODIFY ATTEMPTED TO UPDATE A COLUMN WHICH WAS NOT SPECIFIED IN THE UPDATE SET CLAUSE	42811
+20520	ATTEMPT TO USE A DEPRECATED FEATURE ON OBJECT *object-name*. REASON CODE *reason-code*	01694
-20522	INVALID SPECIFICATION OF WITHOUT OVERLAPS CLAUSE. REASON CODE *reason-code*	428HW
-20523	TABLE *table-name* WAS SPECIFIED AS A HISTORY TABLE, BUT THE TABLE DEFINITION IS NOT VALID FOR A HISTORY TABLE. REASON CODE = *reason-code*	428HX
-20524	INVALID PERIOD SPECIFICATION OR PERIOD CLAUSE FOR PERIOD *period-name*. REASON CODE = *reason-code*	428HY
-20525	THE REQUESTED ACTION IS NOT VALID FOR TABLE *table-name* BECAUSE THE TABLE IS THE WRONG TYPE OF TABLE. REASON CODE = *reason-code*	428HZ
-20527	*period-name* IS NOT A PERIOD IN TABLE *table-name*	4274M
-20528	THE TARGET OF THE DATA CHANGE OPERATION IS A TABLE *table-name*, WHICH INCLUDES A PERIOD *period-name*. A ROW THAT THIS DATA CHANGE OPERATION ATTEMPTED TO MODIFY WAS ALSO MODIFIED BY ANOTHER TRANSACTION.	57062
-20529	THE ARGUMENT OF THE WRAP FUNCTION OR CREATE_WRAPPED PROCEDURE IS NOT VALID	5UA0O
-20530	AN OBFUSCATED STATEMENT IS NOT VALID. REASON CODE= *reason-code*	42638
-20531	THE VERSION NUMBER *actual-version* SPECIFIED IN A BINARY XML VALUE IS NOT SUPPORTED. THE HIGHEST	22544

SQL Code	Explanation	SQL State
	SUPPORTED VERSION IS *supported-version*	
-20535	THE DATA CHANGE OPERATION *operation* IS NOT SUPPORTED FOR THE TARGET OBJECT *object-name* BECAUSE OF AN IMPLICIT OR EXPLICIT SYSTEM PERIOD SPECIFICATION INVOLVING *period-name*. REASON CODE: *reason-code*	51046
-20539	THE QUERY FAILED BECAUSE A NEGATIVE VALUE OR THE NULL VALUE IS USED IN THE *clause* CLAUSE	2201W or 2201X
+20543	A SYSTEM PARAMETER WAS OVERRIDDEN FOR *object-name* WHEN PROCESSING THE *statement-name* STATEMENT. REASON *reason-code*	0169A
-20547	THE STATEMENT FAILED BECAUSE THE TARGET OF AN ASSIGNMENT IS A READ-ONLY VARIABLE. VARIABLE NAME: *variable-name*	42813
-20550	AN ARGUMENT, OR COMBINATION OF ARGUMENTS, SPECIFIED FOR THE *operator-name* OPERATOR ARE NOT VALID	428I4
-20551	CONSTRUCTING AN ASSOCIATIVE ARRAY FAILED BECAUSE THE INPUT DATA INCLUDES AT LEAST ONE DUPLICATE ARRAY INDEX VALUE. DUPLICATED INDEX VALUE: *value*	22545
-20553	AN ENABLE ARCHIVE CLAUSE WAS SPECIFIED FOR TABLE *table-name*, BUT THE TABLE CANNOT BE USED AS AN ARCHIVE-ENABLED TABLE. REASON CODE = *reason-code*	428HM
-20554	TABLE *table-name* WAS SPECIFIED AS AN ARCHIVE TABLE, BUT THE TABLE DEFINITION IS NOT VALID FOR AN ARCHIVE TABLE. REASON CODE = *reason-code*	428HX
-20555	AN ARCHIVE-ENABLED TABLE IS NOT ALLOWED IN THE SPECIFIED CONTEXT. REASON CODE *reason-code*	428I6
-20556	THE OPERATION FAILED BECAUSE MULTIPLE RESULT VALUES CANNOT BE RETURNED FROM A SCALAR FUNCTION *function-name*	22547
-20565	THE REPLACEMENT VALUE FOR *built-in-global-var* IS INVALID	42815
-20567	TABLE *table-name* CANNOT BE DEFINED AS A DEPENDENT OF REFERENTIAL CONSTRAINT *constraint-name*. REASON *reason-code*	42915
–30000	EXECUTION FAILED DUE TO A DISTRIBUTION PROTOCOL ERROR THAT WILL NOT AFFECT THE SUCCESSFUL EXECUTION OF SUBSEQUENT COMMANDS OR SQL STATEMENTS: REASON *reason-code (sub-code)*	58008
–30002	THE SQL STATEMENT CANNOT BE EXECUTED DUE TO A PRIOR CONDITION IN A CHAIN OF STATEMENTS	57057
–30005	EXECUTION FAILED BECAUSE FUNCTION NOT SUPPORTED BY THE SERVER: LOCATION *location-name* PRODUCT ID *product-identifier* REASON *reason-code (sub-code)*	56072
–30020	EXECUTION FAILED DUE TO A DISTRIBUTION PROTOCOL ERROR THAT CAUSED DEALLOCATION OF THE CONVERSATION: REASON *reason-code (sub-code)*	58009
–30021	EXECUTION FAILED DUE TO A DISTRIBUTION PROTOCOL ERROR THAT WILL AFFECT THE SUCCESSFUL EXECUTION OF SUBSEQUENT COMMANDS OR SQL STATEMENTS: MANAGER *manager* AT LEVEL *level* NOT SUPPORTED ERROR	58010
–30025	EXECUTION FAILED BECAUSE FUNCTION IS NOT SUPPORTED BY THE SERVER WHICH CAUSED TERMINATION OF THE CONNECTION: LOCATION *location* PRODUCT ID *product-identifier* REASON *reason-code (sub-code)*	56073
–30030	COMMIT REQUEST WAS UNSUCCESSFUL, A DISTRIBUTION PROTOCOL VIOLATION HAS BEEN DETECTED, THE CONVERSATION HAS BEEN DEALLOCATED. ORIGINAL SQLCODE = *original-sqlcode* AND ORIGINAL SQLSTATE = *original-sqlstate*	58013
–30040	EXECUTION FAILED DUE TO UNAVAILABLE RESOURCES THAT WILL NOT AFFECT THE SUCCESSFUL EXECUTION OF SUBSEQUENT COMMANDS OR SQL STATEMENTS. REASON *reason-code* TYPE OF RESOURCE *resource-type* RESOURCE NAME *resource-name* PRODUCT ID *product-identifier* RDBNAME *rdbname*	57012
–30041	EXECUTION FAILED DUE TO UNAVAILABLE RESOURCES THAT WILL AFFECT THE SUCCESSFUL EXECUTION OF SUBSEQUENT COMMANDS AND SQL STATEMENTS. REASON *reason-code* TYPE OF RESOURCE *resource-type* RESOURCE NAME *resource-name* PRODUCT ID *product-identifier* RDBNAME *rdbname*	57013
-30045	EXECUTION FAILED BECAUSE THE DEFINITION OF OBJECT *object-name* OF TYPE *object-type* BEING ACCESSED AT	428I5

SQL Code	Explanation	SQL State
	server-name-1 DIFFERS FROM THE DEFINITION OF THE OBJECT AT *server-name-2*	
-30047	STATEMENT FAILED BECAUSE OBJECT OF TYPE *object-type* CANNOT BE ACCESSED USING DIFFERENT DISTRIBUTED PROTOCOLS ON A CONNECTION FROM *server-name-1* TO *server-name-2*	58008
–30050	*command-or-SQL-statement-type* COMMAND OR SQL STATEMENT INVALID WHILE BIND PROCESS IN PROGRESS	58011
–30051	BIND PROCESS WITH SPECIFIED PACKAGE NAME AND CONSISTENCY TOKEN NOT ACTIVE	58012
–30052	PROGRAM PREPARATION ASSUMPTIONS ARE INCORRECT	42932
–30053	OWNER AUTHORIZATION FAILURE	42506
–30060	RDB AUTHORIZATION FAILURE	08004
–30061	RDB NOT FOUND	08004
-30062	RDB ACCESS FAILURE	08004
–30070	command COMMAND NOT SUPPORTED ERROR	58014
–30071	object-type OBJECT NOT SUPPORTED ERROR	58015
–30072	Parameter subcode PARAMETER NOT SUPPORTED ERROR	58016
–30073	Parameter subcode PARAMETER VALUE NOT SUPPORTED ERROR	58017
–30074	REPLY MESSAGE WITH codepoint (svrcod) NOT SUPPORTED ERROR	58018
-30080	*prot* COMMUNICATIONS ERROR DETECTED. API=*api*, LOCATION=*loc*, FUNCTION=*func*, ERROR CODES=*rc1 rc2 rc3*	08001
–30082	CONNECTION FAILED FOR SECURITY REASON reason-code (reason string)	08001
–30090	REMOTE OPERATION INVALID FOR APPLICATION EXECUTION ENVIRONMENT	25000
+30100	OPERATION COMPLETED SUCCESSFULLY BUT A DISTRIBUTION PROTOCOL VIOLATION HAS BEEN DETECTED. ORIGINAL SQLCODE=original-sqlcode AND ORIGINAL SQLSTATE=original-sqlstate	01558
–30104	ERROR IN BIND OPTION option AND BIND VALUE option-value	56095
–30105	BIND OPTION option1 IS NOT ALLOWED WITH BIND OPTION option2	56096
-30106	INVALID INPUT DATA DETECTED FOR A MULTIPLE ROW INSERT OPERATION. INSERT PROCESSING IS TERMINATED	22527

SQL State / SQL Code Cross-Reference

SQL State	SQL Code
00000	+000
01004	+20141
01004	+445
01005	+236
01005	+238
01005	+239
0100C	+466
01011	+20348
01515	+304
01516	+558
01517	+335
01519	+802
01520	+331
01521	+402
01522	+403
01525	+117
01527	+799
01528	+645
01530	+738
01532	+204
01532	+219
01533	+205
01533	+206
01537	+218
01538	+650
01539	+863
01540	+664
01542	+552
01543	+541
01546	+220
01548	+551
01551	+653
01552	+203
01553	+806
01554	+807
01558	+30100
01560	+562
01565	+20365
01566	+610
01568	+098
01578	+4745
01590	+111
01591	+535
01594	+237
01596	+599
01597	+655
01599	+4751
01600	+658
01602	+20007
01604	+217
01605	+347
01608	+434
01614	+494
01615	+020
01624	+20002
01625	+585
01628	+395
01629	+394
01640	+883
01643	+385
01644	+20122
01656	+20187
01658	+20224
01659	+252
01663	+20245
01664	+20270
01665	+20271
01666	+20272
01667	+20278
01668	+354
01676	+20341
01679	+20360
01680	+20367
01681	+20368
01681	+440
01682	+20371
01683	+20378
01688	+361
0168C	+364
0168D	+364
0168E	+364
0168F	+364
0168G	+364
0168X	+20468
0168Z	+4726
01694	+20520
0169A	+20543
0169B	+4748
01H54	+20458
01H55	+20459
01H56	+20460
01H57	+20461
01HXX	+462
02000	+100
02000	+231
02502	+222
02504	+20237
07001	-313
07002	-804
07003	-518
07005	-517
07501	-20186
08001	-30081
08001	-30082
08001	-808
08002	-842
08003	-843
08003	-900
08004	-1403
08004	-30060
08004	-30061
08004	-30062
09000	-723
0A001	-752
0F001	-423
0K000	-787
0N002	-20377
0Z002	-20228
10501	-16000
10501	-16001
10502	-16048
10503	-16016
10503	-16023
10503	-16024
10503	-16026
10503	-16029
10504	-16032
10504	-16036
10505	-16002
10505	-16007
10506	-16005
10506	-16009
10507	-16003
10507	-16011
10507	-16012
10507	-16015
10507	-16020
10507	-16022
10507	-16031
10507	-16033
10509	-16031
10601	-16046
10601	-16047
10602	-16049
10602	-16051
10602	-16052
10605	-16055
10605	-16056
10605	-16057
10608	-16038
10608	-16041
10608	-16061
10608	-16065
10608	-16066
10609	-16067
10609	-16068
10609	-16069
10701	-16080
10702	-16081
10703	-16085
10704	-16083
10706	-16086
10707	-16087
10708	-16088
10708	-16089
20000	-773
20521	-925
21000	-811
21501	-533
21502	-534
21506	-788
22001	-404
22001	-404
22001	-433
22002	-305
22003	-304
22003	-406
22004	-087
22007	-180
22007	-181
22007	-20447
22007	-20448
22007	-20493
22007	-20497
22008	-183
2200L	-20345

SQL State	SQL Code
2200M	-20398
2200M	-20400
2200S	-20331
2200T	-20332
2200V	-20382
2200W	-16075
2200W	-20412
22010	-363
22011	-138
22018	-20476
22018	-20477
22018	-420
22019	-130
2201R	-20399
2201W	-20539
2201X	-20539
22021	-330
22021	-331
22024	-300
22025	-130
2202E	-20439
2202F	-20440
22501	-311
22502	-20365
22503	-188
22504	-191
22505	-186
22506	-187
22508	-812
22511	-399
22512	-309
22522	-189
22525	-327
22527	-30106
22528	-20224
22529	-253
22530	-254
22531	-20295
22533	-20335
22534	-20340
22537	-354
22539	-365
22540	-20434
22541	-20465
22542	-20471
22544	-20531
22545	-20551
22546	-171
22547	-20556
225DE	-16246
225DE	-16247
225DE	-16248
225DE	-16249
225DE	-16250
225DE	-16251
225DE	-16252
225DE	-16253
225DE	-16254
225DE	-16255
225DE	-16257
225DE	-16258
225DE	-16259
225DE	-16260
225DE	-16262
225DE	-16265

SQL State	SQL Code
225DE	-16266
23502	-407
23503	-530
23504	-531
23504	-532
23505	-803
23506	-652
23507	-681
23508	-690
23509	-807
23510	-908
23511	-543
23512	-544
23513	-545
23515	-603
23522	-359
23523	-695
23525	-20305
23526	-20306
24501	-500
24501	-501
24501	-507
24502	-502
24504	-508
24506	-519
24510	-222
24512	-224
24513	-227
24513	-353
24516	-499
24517	-472
24518	-20185
24519	-247
24520	-589
24521	-248
24523	-249
24524	-20411
248H8	-20470
25000	-30090
25000	-571
25000	-817
26501	-20248
26501	-514
26501	-516
27000	-907
2D521	-926
2D528	-426
2D529	-427
2F005	-578
34000	-504
35000	-393
36001	-243
38000	-4302
38001	-487
38002	-577
38003	-751
38004	-579
38503	-430
38504	-431
38505	-396
38554	-20457
38H10	-20423
38H11	-20424
38H12	-20425
38H13	-20426

SQL State	SQL Code
38H14	-20427
39004	-470
39501	-450
3B001	-880
3B501	-881
3B502	-882
3B503	-20111
3C000	-051
3F000	-713
40001	-911
42501	-20265
42501	-551
42501	-567
42502	-164
42502	-20379
42502	-552
42502	-554
42502	-555
42503	-553
42504	-556
42505	-922
42506	-30053
42508	-562
42509	-549
42510	-592
42512	-20264
42512	-20264
42513	-20281
42514	-20342
42517	-20361
42546	-171
42601	-007
42601	-011
42601	-029
42601	-097
42601	-104
42601	-108
42601	-109
42601	-115
42601	-123
42601	-128
42601	-1760
42601	-199
42601	-441
42601	-491
42601	-638
42602	-113
42602	-251
42603	-010
42604	-103
42604	-105
42605	-170
42606	-110
42607	-112
42607	-409
42608	-584
42609	-417
42610	-184
42610	-418
42611	-604
42612	-084
42612	-142
42613	-628
42614	-637
42615	-644

SQL State	SQL Code
42617	-198
42618	-312
42618	-5012
42620	-228
42621	-546
42621	-548
42623	-373
42625	-580
42625	-582
42626	-767
42629	-078
42630	-056
42631	-057
42633	-20227
42634	-20275
42635	-20276
42638	-20530
42701	-121
42702	-203
42703	-205
42703	-206
42703	-5001
42704	-096
42704	-204
42704	-219
42704	-722
42705	-950
42707	-208
42708	-229
42709	-537
42710	-456
42710	-601
42710	-719
42710	-720
42710	-721
42711	-612
42712	-212
42713	-242
42714	-314
42718	-250
42721	-725
42723	-454
42724	-444
42725	-476
42726	-340
42732	-585
42734	-590
42736	-779
42737	-781
42738	-783
42749	-20328
4274A	-20330
4274C	-20362
4274D	-20363
4274M	-20527
42801	-173
42802	-117
42803	-119
42803	-122
42804	-581
42805	-125
42806	-303
42807	-150
42808	-151
42809	-147

SQL State	SQL Code
42809	-148
42809	-152
42809	-156
42809	-159
42810	-157
42811	-158
42813	-160
42814	-195
42815	-060
42815	-171
42815	-171
42815	-20338
42815	-20565
42815	-451
42815	-713
42815	-846
42816	-182
42817	-196
42818	-131
42818	-20353
42819	-402
42820	-405
42820	-410
42821	-408
42822	-214
42823	-412
42824	-132
42824	-414
42825	-344
42825	-415
42826	-421
42827	-509
42828	-510
42829	-126
42829	-511
42830	-538
42831	-542
42831	-593
42831	-594
42832	-607
42832	-618
42834	-629
42835	-341
42836	-345
42836	-346
42837	-190
42842	-683
42845	-583
42846	-461
42849	-20102
42852	-557
42855	-392
42856	-20104
42860	-784
42866	-20019
42866	-475
42872	-225
42873	-246
42877	-197
42878	-449
42879	-492
42880	-453
42882	-455
42883	-458
42884	-240

SQL State	SQL Code
42884	-440
42885	-483
42886	-469
42887	-390
42888	-539
42889	-624
42890	-573
42891	-541
42893	-478
42893	-616
42894	-574
42895	-301
42898	-696
42899	-697
428A1	-452
428B0	-481
428B3	-435
428B4	-240
428B7	-490
428C1	-372
428C1	-373
428C2	-374
428C4	-216
428C7	-771
428C7	-867
428C9	-798
428D2	-398
428D3	-397
428D4	-776
428D5	-778
428D6	-780
428D7	-782
428D8	-785
428E5	-20046
428EC	-20058
428EK	-079
428EK	-20494
428EW	-20093
428EZ	-20017
428F1	-20120
428F2	-058
428F4	-244
428F5	-245
428F9	-348
428FA	-336
428FB	-20142
428FC	-20144
428FE	-20146
428FJ	-20211
428FL	-20165
428FM	-20166
428FP	-20178
428FQ	-20179
428FR	-20180
428FS	-20181
428FT	-20183
428FY	-20235
428G2	-20435
428G3	-20257
428G4	-20258
428G5	-20260
428G8	-20279
428GB	-20286
428GC	-20289
428GD	-20300

SQL State	SQL Code
428GD	-20300
428GH	-20314
428GI	-20329
428GJ	-20356
428GK	-20369
428GL	-20372
428GM	-20373
428GN	-20374
428GU	-20422
428GV	-20428
428GX	-20430
428H0	-20437
428H1	-20438
428H2	-20441
428H7	-20467
428H8	-20470
428H9	-20472
428HA	-20473
428HB	-20474
428HC	-20475
428HD	-20478
428HJ	-20487
428HK	-20488
428HL	-4728
428HM	-20490
428HM	-20553
428HN	-20491
428HT	-20505
428HW	-20522
428HX	-20523
428HX	-20554
428HY	-20524
428HZ	-20525
428I1	-20517
428I1	-20517
428I3	-20547
428I4	-20550
428I5	-30045
428I6	-20555
42901	-111
42902	-118
42903	-120
42905	-127
42906	-133
42907	-134
42907	-416
42907	-586
42908	-153
42908	-343
42909	-154
42911	-419
42912	-503
42914	-536
42915	-20567
42915	-632
42915	-633
42915	-634
42915	-635
42917	-147
42917	-20479
42917	-658
42917	-667
42917	-669
42918	-473
42924	-513

SQL State	SQL Code
42925	-342
42927	-20073
42932	-30052
42939	-20074
42939	-457
42945	-20106
42961	-114
42962	-350
42963	-20240
42969	-917
42972	-338
42986	-750
42987	-797
42988	-875
42989	-20094
42993	-355
42995	-526
42997	-270
429B1	-729
429B2	-20016
429BB	-789
429BD	-20148
429BI	-20252
429BL	-20267
429BN	-20283
429BQ	-4733
429BS	-20304
429BV	-20354
429BX	-356
429BY	-4737
429C1	-20433
429C2	-20436
429CB	-4747
44000	-161
46001	-20200
46002	-107
46002	-107
46002	-20201
46003	-20202
46003	-478
46007	-20203
46008	-20204
4600C	-20310
4600C	-478
4600D	-20311
4600E	-20312
46103	-20212
46501	-20207
46502	-20213
51002	-805
51003	-818
51004	-822
51005	-906
51006	-927
51008	-992
51021	-918
51021	-939
51024	-575
51030	-480
51030	-482
51032	–872
51033	–496
51034	–740
51035	–845
51036	–20110

SQL State	SQL Code
51039	–20143
51043	-4729
51046	-20535
53001	–620
53004	–742
53014	–736
53022	–20107
53035	–660
53036	–661
53037	–662
53038	–663
53039	–665
53040	–671
53041	–676
53043	–686
53044	–687
53045	–678
53088	–611
53089	–2001
53090	–873
53091	–874
53092	–876
53093	–877
53094	–878
53095	–879
53096	–769
53098	–20070
53099	–20071
530A1	–20177
530A2	–20182
530A3	–4705
530A4	-4739
530A4	–4706
530A5	–20381
530A7	–4710
530A8	-4727
530A9	-4744
54001	–101
54002	–102
54004	–129
54004	–840
54005	–136
54006	–137
54008	–602
54008	–613
54008	–614
54008	–631
54010	–670
54011	–680
54011	–689
54012	–684
54024	–643
54025	–651
54027	–400
54035	–20005
54038	–724
54041	–497
54042	–748
54051	–20127
54054	–4701
54055	-4732
54057	-20326
54058	–20327
55002	–220
55002	–221

SQL State	SQL Code
55003	–693
55004	–735
55006	–615
55007	-951
55011	-619
55012	-623
55012	-623
55014	-625
55015	-625
55015	-626
55016	–627
55017	–646
55019	-20469
55019	–7008
55020	–741
55023	–471
55030	–726
55035	–672
55048	–20147
55058	–20313
55059	–20315
55063	–20339
55079	-4730
56010	–333
56016	–636
56018	–668
56023	–512
56027	–639
56031	–622
56036	–655
56038	-4738
56038	-4743
56038	–4700
56038	–947
56040	–411
56045	–919
56052	–20072
56053	–730
56054	–731
56055	–732
56056	–733
56056	–737
56057	–734
56059	–20100
56060	–20101
56062	–948
56064	–715
56065	–716
56066	–717
56067	–718
56067	-4749
56067	-4750
56072	–30005
56073	–30025
56080	–728
56084	–20108
56084	–351

SQL State	SQL Code
56084	–352
56084	–4704
56088	–739
56089	–640
56089	-20163
56090	–650
56095	–30104
56096	–30105
56095	-4751
560A1	–763
560A2	–764
560A3	–765
560A4	–766
560A5	–768
560A6	–770
560A7	–20003
560A8	–20004
560A9	–20008
560AB	–20060
560AD	–20091
560AE	–20092
560B1	–20123
560B2	–20124
560B3	–20125
560B5	-20129
560B8	-20210
560BF	-20223
560C3	-989
560C3	–989
560C5	-4731
560C5	–20249
560C7	–20266
560CC	–20380
560CG	–20409
560CH	–20410
560CK	–4709
560CM	-20444
560CU	-4734
560CV	-4735
560CY	-4736
560D5	-4746
560DC	-4753
57001	–540
57002	–559
57003	–647
57004	–653
57005	–666
57006	–679
57007	-20385
57007	–20366
57007	–909
57007	–910
57008	–185
57010	–682
57011	–677
57011	–904
57011	-20442

SQL State	SQL Code
57012	–30040
57013	–30041
57014	–905
57014	–952
57015	–923
57015	–981
57015	–991
57017	–20232
57017	–332
57018	–691
57018	–692
57023	–694
57033	–913
57051	–495
57053	–746
57054	–747
57057	–30002
57062	-20528
58001	–621
58002	–685
58002	–688
58002	–929
58003	–144
58004	–819
58004	–820
58004	–901
58005	-902
58006	–924
58008	–30000
58008	-30047
58009	–30020
58010	–30021
58011	–30050
58012	–30051
58013	–30030
58014	–30070
58015	–30071
58016	–30072
58017	–30073
58018	–30074
58026	–870
5UA0O	-20529
*	–438
*	+438
xxx	–443

(* Application Defined)

SQLSTATE Class Codes

The first two characters of the SQLSTATE contain the class code followed by a three-character subcode. You can use the class code to test for classes or errors. You can use the entire SQLSTATE to test for specific errors. The first character of the SQLSTATE represents the overall success of the statement execution. A zero indicates successful execution; a nonzero indicates an unsuccessful execution.

Class Code	Explanation
00	Unqualified Successful Completion
01	Warning
02	No Data
07	Dynamic SQL Error
08	Connection Exception
09	Trigger Action Exception
0A	Feature Not Supported
0E	Invalid Schema Name List Specification
0F	Invalid Token
0K	Resignal When Handler Not Active
0N	SQL/XML Mapping Error
0W	Prohibited Statement Encountered During Trigger
0Z	Diagnostics Exception
10	XQuery Error
20	Case Not Found for Case Statement
21	Cardinality Violation
22	Data Exception
23	Constraint Violation
24	Invalid Cursor State
25	Invalid Transaction State
26	Invalid SQL Statement Identifier
27	Triggered Data Capture Violation
28	Invalid Authorization Specification
2D	Invalid Transaction Termination
2E	Invalid Connection Name
2F	SQL Function Exception
34	Invalid Condition Name
35	Invalid Condition Number
36	Cursor Sensitivity Exception
38	External Function Exception
39	External Function Call Exception
3B	Savepoint Exception
3C	Ambiguous Cursor Name
3F	Invalid Schema (Collection) Name
40	Transaction Rollback
42	Syntax Error or Access Rule Violation
44	WITH CHECK OPTION Violation
46	Java Errors
51	Invalid Application State
53	Invalid Operand or Inconsistent Specification
54	SQL or Product Limit Exceeded
55	Object Not in Prerequisite State
56	Miscellaneous SQL or Product Error
57	Resource Not Available or Operator Intervention
58	System Error
5U	Common Utilities and Tools

CAF Return Codes

The following table explains some of the return and reason codes that are returned in:

- The variable named in the return and reason code parameters of the CAF call
- Registers 15 and 0 (if parameters are not used)

Return Code	Reason Code	Explanation
+000	00000000	Successful completion.
+004	00C10824	CAF reset complete. Ready to make a new connection.
+008	00C10831	Release level mismatch between DB2 and CAF code.
+200	00C10201	Received a second CONNECT from the same TCB. The first CONNECT could have been implicit or explicit.
+200	00C10202	Received a second OPEN from the same TCB. The first OPEN could have been implicit or explicit.
+200	00C10203	CLOSE issued when there was no active OPEN.
+200	00C10204	DISCONNECT issued when there was no active CONNECT.
+200	00C10205	TRANSLATE issued when there was no connection to DB2.
+200	00C10206	Wrong number of parameters or the end-of-list bit was off.
+200	00C10207	Unrecognized function parameter.
+200	00C10208	Received requests to access two different DB2 subsystems from the same TCB.
+204	Various	CAF system error. Probable error in the attach or DB2.

Resource Type Codes

TYPE Code	Unavailable Resource	Name, Content, Format
00000100	Database	DB
00000200	Table space	DB.SP
00000201	Index space	DB.SP
00000202	Table space	RD.DB.TS
00000205	Compression Dictionary	DB.SP
00000210	Partition	DB.SP.PT
00000220	Data set	DSN
00000230	Temporary file	SZ
00000240	Database procedure	DBP
00000300	Page	DB.SP.PG
00000301	Index minipage	DB.SP.PG.MP
00000302	Table space page	DB.SP.PG
00000303	Index space page	DB.SP.PG
00000304	Table space RID	DB.SP.RID
00000305	Index access/table space RID	DB.SP.RID
00000306	Index access/table space page	DB.SP.PG
00000307	Index space EOF	DB.SP.01
00000308	Table space page	DB.SP.PT.PG
00000309	Index space page	DB.SP.PT.PG
0000030A	Table space RID	DB.SP.PT.RID
00000400	ICF catalog	IC
00000401	Authorization function	
00000402	Security Server	SAF/RACF return/reason code
00000500	Storage group	SG
00000602	EDM DBD Space	
00000603	EDM DYNAMIC STATEMENT Space	
00000604	EDM skeleton storage	
00000605	EDM above-the-bar storage	
00000606	EDM below-the-bar storage	
00000700	Buffer pool space	BP
00000701	Group buffer pool	GBP
00000800	Plan	PL
00000801	Package	COLLECTION.PACKAGE. CONTOKEN
00000802	BINDLOCK01 through BINDLOCK20	BINDLOCK01 through BINDLOCK20
00000900	32KB data area	
00000901	Sort storage	
00000903	Hash anchor	DB.SP.PG.AI
00000904	RIDLIST storage	
00000905	IRLM storage	
00000906	DB2	MEMBER
00000907	LOB storage	
00000908	Basic Floating Point Extensions Facility	

TYPE Code	Unavailable Resource	Name, Content, Format
00000909	Extended Time-of-Day (TOD) Clock	
0000090A	XML Storage	
00000A00	Table	RD.CR.TB
00000A10	Alias	RELDEP.OWNER.ALIAS RD.CR.AL
00000A11	Distinct type	SC.DT
00000A12	User-defined function	SC.SN
00000A13	Stored procedure	SC.SN
00000A14	Sequence	
00000A16	Role	
00000A17	Trigger	
00000B00	View	RD.CR.VW
00000C00	Index	RD.CR.IX
00000C01	Index	CR.IX
00000D00	DBID/OBID	RD.DI.OI
00000D01	DBID/OBIG	DI.OI
00000D02	OBID	OI
00000E00	SU limit exceeded	CN
00000F00	Auxiliary column	DI.OI.ROWID.COLN or DI.OI.DOCID.COLN
00000F01	LOB lock	DIX.PIX.ROWID.VRSN
00000F81	XML lock	DIX.PIX.DOCID
00001000	DDF	LOCATION or SUBSYSTEM ID
00001001	System conversation	LU.MODE. RTNCD.FDBK2. RCPRI.RCSEC. SENSE
00001002	Agent conversation	LU.MODE. RTNCD.FDBK2. RCPRI.RCSEC. SENSE
00001003	CNOS processing	LU.MODE. RTNCD.FDBK2. RCPRI.RCSEC. SENSE
00001004	CDB (Communication Database)	LOCATION.AUTHORIZATION-ID. PL
00001005	DB access agent	LOCATION
00001006	DRDA related error	
00001007	TCP/IP domain name	LINKNAME.DOMAIN.ERRNO
00001008	TCP/IP service name	LOCATION.SERVICE.ERRNO
00001080	ACCEL	SERVER DOMAIN
00001102	Bootstrap data set (BSDS)	MEMBER
00001202	Dynamic statement cache	MEMBER.STMTID
00002000	Table space CS-claim class	DB.SP
00002001	Table space RR-claim class	DB.SP
00002002	Table space write-claim class	DB.SP
00002003	Index space CS-claim case	DB.SP
00002004	Index space RR-claim class	DB.SP
00002005	Index space write-claim class	DB.SP

TYPE Code	Unavailable Resource	Name, Content, Format
00002006	Table space partition CS-claim class	DB.SP.PT
00002007	Table space partition RR-claim class	DB.SP.PT
00002008	Table space partition write-claim class	DB.SP.PT
00002009	Index space partition CS-claim	DB.SP.PT
00002010	Index space partition RR-claim class	DB.SP.PT
00002011	Index space partition Write-claim class	DB.SP.PT
00002100	Table space DBET entry	DB.SP
00002101	Index space DBET entry	DB.SP
00002102	Table space partition DBET entry	DB.SP.PT
00002103	Index space partition DBET entry	DB.SP.PT
00002104	DBET hash chain lock timeout	INTERNAL LOCK NN
00002105	Logical partition DBET entry	DB.SP.PT
00002200	Routine Parameter Storage	DBP
00002201	Debug Agent Storage	DBP
00002300	ICSF encryption and decryption facilities	
00003000	Code (release maintenance_level or system parameter)	REL, APAR, ZPARM
00003002	Number of Stored Procedures	
00003072	Index	
00003073	Index	
00003328	Release dependency	
00003329	DBID/OBID	DI.OI
00003330	OBID limit exceeded	
00003840	LOB column	
00004000	Profile exception threshold exceeded	PID.PTYPE.PNAME

Where/Stands for:

- AI (hash anchor ID)
- ALIAS (alias owner)
- APAR (APAR number)
- AUTHORIZATION-ID (DB2 authorization identifier)
- BP (buffer pool identifier)
- CN (column name)
- COLLECTION (collection-ID of the package

- COLN (column number within the base table of the LOB column that has been marked invalid)
- CONTOKEN (consistency token of the packages)
- CR (creator of the object
- DB (database name)
- DBP (database procedure name)
- DI (DBID in decimal of resource)
- DIX (DBID in hexadecimal)
- DOCID (document ID)
- DOMAIN (TCP/IP domain name or IP address)
- DT (distinct type)
- DSN (data set name)
- FDBK2 (VTAM secondary return code)
- GBP (group buffer pool name)
- IC (ICF catalog alias name)
- IX (index name)
- LOCATION (location in which the specified resource is not available)
- LU (logical unit name)
- MEMBER (group member name)
- MODE (logical unit mode name)
- MP (hexadecimal mini-page number)
- NN (decimal number of the hash chain)
- OI (OBID in decimal of resource)
- OWNER (alias owner)
- PACKAGE (package identifier)
- PG (hexadecimal page number)
- PID (profile ID)
- PIX (PSID in hexadecimal)
- PNAME (name of profile filtering scope)
- PTYPE (type of profile filtering scope)
- PL (plan identifier)
- PT (decimal partition number)
- RCPRI (APPC primary return code)
- RCSEC (APPC secondary return code)
- RD (DB2 release dependency mark)
- REL (DB2 release name)
- RELDEP (DB2 release dependency mark)
- RID (record identifier)
- RTNCD (VTAM primary return code)
- ROWID (Row ID)
- SC (schema)
- SENSE (SNA sense codename)
- SERVER (accelerator server name)
- ST (space allocation type in work file database)
- SG (storage group name)
- SN (routine name)
- SP (space name)
- SZ (temporary file page size
- TB (table name)
- TS (table space name)
- VW (view name)

Accounting Trace Records, 6-34
ADMTPROC Commands
  MODIFY SHUTDOWN, 3-16
  MODIFY TRACE, 3-16
  START, 3-16
  STOP, 3-16
Aggregate Functions, 1-52
ALLOCATE CURSOR, 1-8
ALTER DATABASE, 1-8
ALTER FUNCTION, 1-8, 1-9, 1-10
ALTER INDEX, 1-11
ALTER MASK, 1-11
ALTER PERMISSION, 1-11
ALTER PROCEDURE – SQL native, 1-12
ALTER PROCEDURE -- external, 1-11
ALTER SEQUENCE, 1-13
ALTER STOGROUP, 1-13
ALTER TABLE, 1-14
ALTER TABLESPACE, 1-17
ALTER TRIGGER (advanced), 1-17
ALTER TRIGGER (basic), 1-18
ALTER TRUSTED CONTEXT, 1-18
ALTER VIEW statement, 1-19
ASSOCIATE LOCATORS statement, 1-19
AUDIT Trace Records, 6-35
BEGIN DECLARE statement, 1-19
CAF Return Codes, 7-42
CALL statement, 1-19
Catalog Tables
  IPLIST, 4-1, 6-13
  IPNAMES, 4-2
  LOCATIONS, 4-3
  LULIST, 4-4
  LUMODES, 4-5
  LUNAMES, 4-5
  MODESELECT, 4-7
  SYSAUDITPOLICIES, 4-7
  SYSAUTOALERTS, 4-9
  SYSAUTOALERTS_OUT, 4-10
  SYSAUTORUNS_HIST, 4-11
  SYSAUTORUNS_HISTOU, 4-11
  SYSAUTOTIMEWINDOWS, 4-11
  SYSAUXRELS, 4-12
  SYSCHECKDEP, 4-13
  SYSCHECKS, 4-13
  SYSCHECKS2, 4-14
  SYSCOLAUTH, 4-14
  SYSCOLDIST, 4-16
  SYSCOLDIST_HIST, 4-19
  SYSCOLDISTSTATS, 4-17
  SYSCOLSTATS, 4-20
  SYSCOLUMNS, 4-21
  SYSCOLUMNS_HIST, 4-27
  SYSCONSTDEP, 4-29
  SYSCONTEXT, 4-30
  SYSCONTEXTAUTHIDS, 4-32
  SYSCONTROLS, 4-33
  SYSCONTROLS_DESC, 4-34
  SYSCONTROLS_RTXT, 4-35
  SYSCOPY, 4-35
  SYSCTXTTRUSTATTRS, 4-41
  SYSDATABASE, 4-42
  SYSDATATYPES, 4-44
  SYSDBAUTH, 4-45
  SYSDBD_DATA, 4-228
  SYSDBRM, 4-47
  SYSDEPENDENCIES, 4-48
  SYSDUMMY1, 4-54
  SYSDUMMYA, 4-54
  SYSDUMMYE, 4-54
  SYSDUMMYU, 4-55
  SYSDYNQRY, 4-50
  SYSDYNQRY_EXPL, 4-53
  SYSDYNQRY_OPL, 4-53
  SYSDYNQRY_SHTEL, 4-53
  SYSDYNQRY_SPAL, 4-54
  SYSDYNQRY_TXTL, 4-54
  SYSDYNQRYDEP, 4-51
  SYSENVIRONMENT, 4-55
  SYSFIELDS, 4-57
  SYSFOREIGNKEYS, 4-58
  SYSINDEXCLEANUP, 4-59
  SYSINDEXCONTROL, 4-59
  SYSINDEXES, 4-60
  SYSINDEXES_HIST, 4-66
  SYSINDEXES_RTSECT, 4-67
  SYSINDEXES_TREE, 4-67
  SYSINDEXPART, 4-68
  SYSINDEXPART_HIST, 4-71
  SYSINDEXSPACESTATS, 4-73
  SYSINDEXSTATS, 4-78
  SYSINDEXSTATS_HIST, 4-80
  SYSIXSPACESTATS_H, 4-73
  SYSJARCLASS_SOURCE, 4-81
  SYSJARCONTENTS, 4-81
  SYSJARDATA, 4-81
  SYSJAROBJECTS, 4-82
  SYSJAVAOPTS, 4-82
  SYSJAVAPATHS, 4-83
  SYSKEYCOLUSE, 4-83
  SYSKEYS, 4-84
  SYSKEYTARGETS, 4-85
  SYSKEYTARGETS_HIST, 4-88
  SYSKEYTARGETSTATS, 4-87
  SYSKEYTGTDIST, 4-89
  SYSKEYTGTDIST_HIST, 4-92
  SYSKEYTGTDISTSTATS, 4-91
  SYSLEVELUPDATESS, 4-94
  SYSLOBSTATS, 4-94
  SYSLOBSTATS_HIST, 4-95
  SYSOBD_AUX, 4-96
  SYSOBDS, 4-95
  SYSOBJROLEDEP, 4-96
  SYSPACKAGE, 4-97
  SYSPACKAUTH, 4-105
  SYSPACKCOPY, 4-107

SYSPACKDEP, 4-114
SYSPACKLIST, 4-116
SYSPACKSTMT, 4-116
SYSPACKSTMT_STMB, 4-120
SYSPACKSTMT_STMT, 4-120
SYSPARMS, 4-120
SYSPENDINGDDL, 4-122
SYSPENDINGDDLTEXT, 4-124
SYSPENDINGOBJECTS, 4-125
SYSPKSYSTEM, 4-125
SYSPLAN, 4-126
SYSPLANAUTH, 4-131
SYSPLANDEP, 4-132
SYSPLSYSTEM, 4-132
SYSPROFILE_TEXT, 4-133, 4-207
SYSQUERY, 4-133
SYSQUERY_AUX, 4-142
SYSQUERYOPTS, 4-135
SYSQUERYPLAN, 4-136
SYSQUERYPREDICATE, 4-142
SYSQUERYSEL, 4-145
SYSRESAUTH, 4-148
SYSROLES, 4-149
SYSROUTINEAUTH, 4-150
SYSROUTINES, 4-152
SYSROUTINES_OPTS, 4-160
SYSROUTINES_SRC, 4-161
SYSROUTINES_TREE, 4-162
SYSROUTINESTEXT, 4-160
SYSSCHEMAAUTH, 4-162
SYSSEQUENCEAUTH, 4-163
SYSSEQUENCES, 4-164
SYSSEQUENCESDEP, 4-167
SYSSESSION, 4-168
SYSSESSION_DATA, 4-169
SYSSESSION_EX, 4-168
SYSSESSION_GV, 4-168
SYSSESSION_STATUS, 4-169
SYSSPTSEC_DATA, 4-229
SYSSPTSEC_EXPL, 4-229
SYSSTATFEEDBACK, 4-170
SYSSTMT, 4-171
SYSSTOGROUP, 4-173
SYSSTRINGS, 4-174
SYSSYNONYMS, 4-175
SYSTABAUTH, 4-176
SYSTABCONST, 4-178
SYSTABLEPART, 4-179
SYSTABLEPART_HIST, 4-185
SYSTABLES, 4-187
SYSTABLES_HIST, 4-206
SYSTABLES_PROFILES, 4-207
SYSTABLESPACE, 4-194
SYSTABLESPACESTATS, 4-200
SYSTABLESTATS_HIST, 4-209
SYSTABSPACESTATS_H, 4-200
SYSTABSTATS, 4-208
SYSTELS, 4-147
SYSTRIGGERS, 4-209
SYSTRIGGERS_STMT, 4-212
SYSUSERAUTH, 4-212
SYSVARIABLEAUTH, 4-215
SYSVARIABLES, 4-214
SYSVARIABLES_DESC, 4-216
SYSVARIABLES_TEXT, 4-216
SYSVIEWDEP, 4-216
SYSVIEWS, 4-217
SYSVIEWS_STMT, 4-219
SYSVIEWS_TREE, 4-219
SYSVOLUMES, 4-219
SYSXMLRELS, 4-220
SYSXMLSTRING, 4-220
SYSXMLTYPMOD, 4-221
SYSXMLTYPMSCHEMA, 4-221
USERNAMES, 4-222
XSRANNOTATIONINFO, 4-223
XSRCOMPONENT, 4-223
XSROBJECTCOMPONENTS, 4-223
XSROBJECTGRAMMAR, 4-224
XSROBJECTHIERARCHIES, 4-224
XSROBJECTPROPERTY, 4-225
XSROBJECTS, 4-225
XSRPROPERTY, 4-226
CHANGE IMS command, 3-16
CICS Attachment Facility
  DSNC, 3-14
  DSNC DISCONNECT, 3-14
  DSNC DISPLAY, 3-14
  DSNC MODIFY, 3-14
  DSNC STOP, 3-14
  DSNC STRT, 3-14
CLOSE statement, 1-19
Collection privileges, 6-19
Commands, 3-1
  ADMTPROC Commands, 3-16
  CICS Attachment Facility, 3-14
  DB2 Commands, 3-1
  DSN Commands, 3-8
  IMS Commands, 3-16
  TSO Commands, 3-14
  z/OS IRLM Commands, 3-15
COMMENT statement, 1-19
COMMIT statement, 1-20
CONNECT statement, 1-20
CREATE ALIAS statement, 1-20
CREATE AUXILIARY TABLE statement, 1-20
CREATE DATABASE statement, 1-20
CREATE DESCRIBE PROCEDURE statement, 1-37
CREATE FUNCTION (compiled SQL scalar) statement, 1-20

CREATE FUNCTION (External scalar) statement, 1-22
CREATE FUNCTION (External table) statement, 1-23
CREATE FUNCTION (inline SQL scalar) statement, 1-24
CREATE FUNCTION (sourced) statement, 1-23
CREATE FUNCTION (SQL table) statement, 1-24
CREATE FUNCTION statement, 1-20
CREATE GLOBAL TEMPORARY TABLE statement, 1-24
CREATE INDEX statement, 1-25
CREATE MASK statement, 1-26
CREATE PERMISSION statement, 1-26
CREATE PROCEDURE (external) statement, 1-26
CREATE PROCEDURE statement, 1-27
CREATE ROLE statement, 1-29
CREATE SEQUENCE statement, 1-30
CREATE STOGROUP statement, 1-30
CREATE SYNONYM (deprecated) statement, 1-30
CREATE TABLE statement, 1-30
CREATE TABLESPACE statement, 1-32
CREATE TRIGGER (advanced) statement, 1-33
CREATE TRIGGER (basic) statement, 1-33
CREATE TRUSTED CONTEXT statement, 1-34
CREATE TYPE (array) statement, 1-34
CREATE TYPE (distinct) statement, 1-34
CREATE VARIABLE statement, 1-35
CREATE VIEW statement, 1-35
Data Types, 1-79
Database privileges, 6-19
DB2 Commands
  ACCESS DATABASE, 3-1
  ACTIVATE FUNCTION LEVEL, 3-1
  ALTER BUFFERPOOL, 3-1
  ALTER GROUPBUFFERPOOL, 3-1
  ALTER UTILITY, 3-1
  ARCHIVE LOG, 3-1
  CANCEL, 3-1
  DISPLAY ACCEL, 3-1
  DISPLAY ARCHIVE, 3-2
  DISPLAY BUFFERPOOL, 3-2
  DISPLAY DATABASE, 3-2
  DISPLAY DDF, 3-2
  DISPLAY FUNCTION SPECIFIC, 3-2
  DISPLAY GROUP, 3-2
  DISPLAY GROUPBUFFERPOOL, 3-2
  DISPLAY LOCATION, 3-2
  DISPLAY LOG, 3-2
  DISPLAY PROCEDURE, 3-2
  DISPLAY PROFILE, 3-2
  DISPLAY RLIMIT, 3-3
  DISPLAY THREAD, 3-3
  DISPLAY TRACE, 3-3
  DISPLAY UTILITY, 3-4
  MODIFY DDF, 3-4
  MODIFY TRACE, 3-4
  RECOVER BSDS, 3-4
  RECOVER INDOUBT, 3-4
  RECOVER POSTPONED, 3-4
  REFRESH DB2, 3-4
  RESET GENERICLU, 3-4
  RESET INDOUBT, 3-4
  SET ARCHIVE, 3-4
  SET LOG, 3-5
  SET SYSPARM, 3-5
  START ACCEL, 3-5
  START DATABASE, 3-5
  START DB2, 3-5
  START DDF, 3-5
  START FUNCTION SPECIFIC, 3-5, 3-6
  START PROCEDURE, 3-5
  START PROFILE, 3-5
  START RLIMIT, 3-6
  START TRACE, 3-6
  STOP ACCEL, 3-6
  STOP DATABASE, 3-6
  STOP DB2, 3-6
  STOP DDF, 3-6
  STOP FUNCTION SPECIFIC, 3-7
  STOP PROCEDURE, 3-7
  STOP PROFILE, 3-7
  STOP RLIMIT, 3-7
  STOP TRACE, 3-7
  TERM UTILITY, 3-7
DB2 Limits, 6-10
DECLARE CURSOR statement, 1-35
DECLARE GLOBAL TEMPORARY TABLE statement, 1-35
DECLARE STATEMENT statement, 1-36
DECLARE TABLE statement, 1-36
DECLARE VARIABLE statement, 1-36
DELETE (cursor-positioned) statement, 1-36
DELETE (searched) statement, 1-36

DESCRIBE CURSOR statement, 1-36
DESCRIBE INPUT statement, 1-36
DESCRIBE OUTPUT statement, 1-36
DESCRIBE TABLE statement, 1-37
Directory Tables, 4-227
  DBDR, 4-227
  SCTR, 4-227
  SPTR, 4-227
  SYSLGRNX, 4-228
  SYSSPTSEC_EXPL, 4-229
  SYSUTIL, 4-229
  SYSUTILX, 4-231
DISPLAY IMS command, 3-16
Distinct type usage privileges, 6-20
DROP statement, 1-37
DSN Commands
  BIND PACKAGE, 3-8
  BIND PLAN, 3-9
  BIND QUERY, 3-9
  DCLGEN TABLE, 3-10
  END, 3-10
  FREE PACKAGE, 3-10
  FREE PLAN, 3-10
  FREE QUERY, 3-10
  FREE STABILIZED DYNAMIC QUERY, 3-10
  REBIND PACKAGE, 3-11
  REBIND PLAN, 3-12
  REBIND TRIGGER PACKAGE, 3-13
  RUN, 3-13
  SPUFI, 3-13
DSN TSO command, 3-14
DSN_COLDIST_TABLE, 5-8
DSN_DETCOST_TABLE, 5-9
DSN_FILTER_TABLE, 5-13
DSN_FUNCTION_TABLE, 5-14
DSN_KEYTGTDIST_TABLE, 5-15
DSN_PGRANGE_TABLE, 5-16
DSN_PGROUP_TABLE, 5-17
DSN_PREDICAT_SELECTIVITY_TABLE, 5-21
DSN_PREDICAT_TABLE, 5-19
DSN_PTASK_TABLE, 5-22
DSN_QUERY_TABLE, 5-23
DSN_QUERYINFO_TABLE, 5-24
DSN_SORT_TABLE, 5-25
DSN_SORTKEY_TABLE, 5-26
DSN_STAT_FEEDBACK_TABLE, 5-34
DSN_STATEMENT_CACHE_TABLE, 5-27
DSN_STATEMNT_TABLE, 5-31
DSN_STRUCTT_TABLE, 5-33
DSN_VIEWREF_TABLE, 5-35
DSNC CICS attachment facility command, 3-14
DSNC DISCONNECT CICS attachment facility command, 3-14
DSNC DISPLAY CICS attachment facility command, 3-14
DSNC MODIFY CICS attachment facility command, 3-14
DSNC STOP CICS attachment facility command, 3-14
DSNC STRT CICS attachment facility command, 3-14
DSNH TSO command, 3-14
DSNZPARM Parameter Information, 6-22
Durations, 1-76
  Computed, 1-76
  Labeled, 1-76
  Labeled Durations, 1-76
embedded SELECT INTO statement queries, 1-7
END DECLARE SECTION statement, 1-37
EXCHANGE statement, 1-37
EXECUTE IMMEDIATE statement, 1-37
EXECUTE statement, 1-37
EXPLAIN statement, 1-37
Expressions, 1-1
FETCH statement, 1-38
FREE LOCATOR statement, 1-38
Fullselect query, 1-7
Functions, 1-52
GET DIAGNOSTICS statement, 1-38
Global Trace Records, 6-36
GRANT (authorization) statement, 1-39
GRANT (collection privileges) statement, 1-39
GRANT (database privileges), 1-39
GRANT (function or procedure privileges) statement, 1-39
GRANT (package privileges) statement, 1-39
GRANT (plan privileges) statement, 1-40
GRANT (schema privileges) statement, 1-40
GRANT (sequence privileges) statement, 1-40
GRANT (system privileges) statement, 1-40
GRANT (table or view privileges) statement, 1-40
GRANT (type or JAR file privileges) statement, 1-40
GRANT (use privileges) statement, 1-40
GRANT (variable privileges) statement, 1-40

HOLD LOCATOR statement, 1-40
IFCIDs
Class 1 Accounting Data, 6-34
Class 1 Authorization Failures, 6-35
Class 1 Background Events, 6-40
Class 1 Entry To or Exit from a Routine or Trigger, 6-38
Class 1 IBM Service, 6-36
Class 1 Statistical Data, 6-44
Class 10 Bind, Commands, and Utilities, 6-42
Class 10 Package Detail, 6-39
Class 10 Package SQL Detail, 6-35
Class 10 Storage Manager Pool Statistics, 6-38
Class 10 Trusted Context Audit, 6-36
Class 11 Audit Administrative Authorities, 6-36
Class 11 DB2-supplied stored procedure and UDF trace, 6-38
Class 11 Dispatching, 6-42
Class 11 Plan Level Accounting, 6-39
Class 12 Storage Manager, 6-42
Class 13 Edit and Validation Exits, 6-43
Class 14 In and Out of DB2, 6-43
Class 16 Distributed Processing, 6-43
Class 17 Claim and Drain Detail, 6-43
Class 18 Event-based Console Messages, 6-43
Class 19 Data Set Open and Close Activity, 6-43
Class 2 Explicit Grant or Revoke, 6-35
Class 2 IBM Service, 6-36
Class 2 In DB2 Time, 6-34
Class 2 Installation Defined Statistics Record, 6-44
Class 2 Subsystem-Related Events, 6-40
Class 2 Time In-DB2 (CPU and Elapsed), 6-38
Class 20 Data Sharing Coherency Summary, 6-43
Class 21 Data Sharing Coherency Detail, 6-44
Class 22 Authorization Exit Parameters, 6-44
Class 23 Language Environment Run-time Diagnostics, 6-44
Class 24 Stored Procedure Detail, 6-44
Class 29 Subsystem wide collection of SQL statistics for statements, 6-39
Class 3 Create, Drop, and Alter Operations against Audit Tables, 6-35
Class 3 Deadlock and Lock Timeout Information, 6-44
Class 3 IBM Service, 6-36
Class 3 SQL-Related Events, 6-40
Class 3 Wait Time, 6-34
Class 3 Wait Time In-DB2, 6-38
Class 30-32 Available for Local Use, 6-44
Class 4 Buffer Manager I/O and EDM Pool Requests, 6-41
Class 4 DB2 Exception Conditions, 6-45
Class 4 First Change of Audited Object, 6-35
Class 4 IBM Service, 6-37
Class 4 Installation-Defined Accounting Record, 6-34
Class 4 Installation-Defined Monitor Record, 6-38
Class 5 Data Sharing Global Statistics, 6-45
Class 5 First Read of Audited Object, 6-36
Class 5 IBM Service (Overflow Hybrid Join, Host Variable Tracing, DB2 invalidation, 6-37
Class 5 Log Manager, 6-41
Class 5 Time Spent Processing, 6-38
Class 5 Time Spent Processing IFI Requests, 6-34, 6-45
Class 6 Data Capture Data, 6-39
Class 6 DBM1 Storage Summary, 6-45
Class 6 SQL Statement at Bind Statement, 6-36
Class 6 Summary Lock Information, 6-41
Class 6 User-Defined Serviceability Trace, 6-37

Class 7 Change in Authorization for Audited Object, 6-36
Class 7 Detailed Lock Information, 6-41
Class 7 IBM Service (Distributed Data), 6-37
Class 7 Location Statistics, 6-45
Class 7 Package Level Accounting in-DB2 Time, 6-35
Class 7 Package Level Accounting In-DB2 Time, 6-39
Class 8 Buffer Pool Data Set Statistics, 6-45
Class 8 Data Manager Detail, 6-42
Class 8 IBM Service (Distributed SQL), 6-37
Class 8 Package Level Accounting Wait Time, 6-35
Class 8 Package Level Accounting Wait Time in DB2, 6-39
Class 8 Utility Access to Any Object, 6-36
Class 9 Connection Type Statistics, 6-45
Class 9 IBM Service (DB2 DRDA Protocol, 6-38
Class 9 Installation-Defined Audit Record, 6-36
Class 9 Sort Detail, 6-42
Class 9 SQL Statement-Level Accounting, 6-39
IFCIDS, 6-34
IMS Commands
CHANGE, 3-16
DISPLAY, 3-16
SSR, 3-16
START, 3-16
STOP, 3-16
TRACE, 3-16
INCLUDE statement, 1-40
Index privileges, 6-21
INSERT statement, 1-41
LABEL statement, 1-41
LOCK TABLE statement, 1-41
MERGE statement, 1-41
Miscellaneous System Information, 6-1
DB2 Limits, 6-10
DSNZPARM Parameter Information, 6-22
IFCIDS, 6-34
Security Authorizations, 6-16
SQL Limits, 6-8
SQLCA, 6-1
SQLDA, 6-4
MODIFY ABEND z/OS IRLM command, 3-15
MODIFY DIAG z/OS IRLM command, 3-15
MODIFY PURGE z/OS IRLM command, 3-15
MODIFY SET z/OS IRLM command, 3-15
MODIFY STATUS z/OS IRLM command, 3-15
Monitor Trace Records, 6-38
MQ Series Table Functions, 1-74
MQ Table Functions
MQREADALL, 1-74
MQREADALLCLOB, 1-74
MQRECEIVEALL, 1-74
MQRECEIVEALLCLOB, 1-75
MQRECEIVEALLXML, 1-75
Naming Definitions, 1-47
On-line DB2 Utilities
BACKUP SYSTEM, 2-1, 2-20
CATENFM, 2-1, 2-20
CATMAINT, 2-1, 2-20
CHECK, 2-20
CHECK DATA, 2-1
CHECK INDEX, 2-2, 2-20
CHECK LOB, 2-2, 2-20
COPY, 2-20
COPYTOCOPY, 2-3, 2-20
DIAGNOSE, 2-3, 2-20
DSNU, 2-1
EXEC SQL, 2-3, 2-20
LISTDEF, 2-4, 2-20
LOAD, 2-4, 2-20
MERGECOPY, 2-6, 2-20
MODIFY RECOVERY, 2-6, 2-20
MODIFY STATISTICS, 2-7, 2-20
OPTIONS, 2-7, 2-20
QUIESCE, 2-7, 2-20
REBUILD, 2-20
REBUILD INDEX, 2-7
RECOVER, 2-8, 2-20
REORG, 2-20
REORG INDEX, 2-8
REORG TABLESPACE, 2-9
REPAIR, 2-11, 2-20
REPORT, 2-12, 2-20
RESTORE SYSTEM, 2-12, 2-20
RUNSTATS, 2-20
RUNSTATS INDEX, 2-13
RUNSTATS TABLESPACE, 2-12
STOSPACE, 2-14, 2-20
TEMPLATE, 2-14, 2-20
UNLOAD, 2-15, 2-20
Other System Commands

CICS Attachment Facility, 3-14
Other System Tables, 4-232
SYSACCELERATEDTABLES, 4-232
SYSACCELERATORS, 4-232
SYSIBM.DSNPROGAUTH, 4-233
Package privileges, 6-18
Performance Trace Records, 6-40
Plan privileges, 6-18
PLAN_TABLE, 5-1
Predicate Processing Summary, 1-80
Predicates, 1-3
Queries, 1-5
Embedded SELECT, 1-7
Fullselect, 1-7
SELECT Statement, 1-7
Reserved Words, 1-77
Resource Type Codes, 7-43
Return Codes, 7-1
CAF Return Codes, 7-42
Resource Type Codes, 7-43
SQL Codes, 7-1
SQL State, SQL Code Cross Reference, 7-36
SQLSTATE Class Codes, 7-41
Routine privileges, 6-20
Scalar Functions, 1-53
ABS, 1-53
ACOS, 1-53
ADD_MONTHS, 1-53
ARRAY_DELETE, 1-53
ARRAY_FIRST, 1-53
ARRAY_LAST, 1-53
ARRAY_NEXT, 1-53
ARRAY_PRIOR, 1-53
ASCII, 1-53
ASCII_CHR, 1-53
ASCII_STR, 1-54
ASIN, 1-54
ATAN, 1-54
ATAN2, 1-54
ATANH, 1-54
BIGINT, 1-54
BINARY, 1-54
BITAND, 1-54
BITANDNOT, 1-54
BITNOT, 1-54
BITOR, 1-54
BITORNOT, 1-54
BLOB, 1-55
CARDINALITY, 1-55
CCSID_ENCODING, 1-55
CEILING, 1-55
CHAR, 1-55
CHAR9, 1-55
CHARACTER_LENGTH, 1-56
CLOB, 1-56
COALESCE, 1-56
COLLATION_KEY, 1-56
COMPARE_DECFLOAT, 1-56
CONCAT, 1-56
CONTAINS, 1-56
COS, 1-56
COSH, 1-56
DATE, 1-57
DAYOFMONTH, 1-57
DAYOFWEEK, 1-57
DAYOFWEEK_ISO, 1-57
DAYOFYEAR, 1-57
DAYS, 1-57
DBCLOB, 1-57
DEC, 1-58
DECFLOAT, 1-57, 1-58
DECFLOAT_FORMAT, 1-57
DECFLOAT_SORTKEY, 1-57
DECRYPT_CHAR, 1-58
DEGREES, 1-58
DIFFERENCE, 1-58
DIGITS, 1-58
DOUBLE, 1-58
DSN_XMLVALIDATE, 1-58
EBCDIC_CHR, 1-58
EBCDIC_STR, 1-59
ENCRYPT, 1-59
EXP, 1-59
EXTRACT date, 1-59
EXTRACT time, 1-59
EXTRACT timezone, 1-59
FLOAT, 1-59
FLOOR, 1-59
GENERATE_UNIQUE, 1-59
GETHINT, 1-59
GETVARIABLE, 1-60
GRAPHIC, 1-60
HEX, 1-60
HOUR, 1-60
IDENTITY_VAL_LOCAL, 1-60
IFNULL, 1-60
INSERT, 1-60
INTEGER, 1-60
JULIAN_DAY, 1-60
LAST_DAY, 1-60
LEFT, 1-61
LENGTH, 1-61
LOCATE, 1-61
LOCATE_IN_STRING, 1-61
LOG, 1-61
LOG10, 1-61
LOWER, 1-60, 1-61
LPAD, 1-61
LTRIM, 1-61
MAX, 1-62
MICROSECOND, 1-62
MIDNIGHT_SECONDS, 1-62
MIN, 1-62
MINUTE, 1-62
MOD, 1-62
MONTH, 1-62
MONTHS_BETWEEN, 1-62
MQREAD, 1-62

MQREADCLOB, 1-62
MQRECEIVE, 1-63
MQRECEIVECLOB, 1-63
MQSEND, 1-63
MULTIPLY_ALT, 1-63
NEXT_DAY, 1-63
NORMALIZE_DECFLOAT, 1-63
NORMALIZE_STRING, 1-63
NULL, 1-64
OVERLAY, 1-64
PACK, 1-64
POSITION, 1-64
POSSTR, 1-64
POWER, 1-64
QUANTIZE, 1-64
QUARTER, 1-64
RADIANS, 1-64
RAISE_ERROR, 1-65
RAND, 1-65
REAL, 1-65
REPEAT, 1-65
REPLACE, 1-65
RID, 1-65
RIGHT, 1-65
ROUND, 1-65
ROUND_TIMESTAMP, 1-65
ROWID, 1-66
RPAD, 1-66
RTRIM, 1-66
SCORE, 1-66
SECOND, 1-66
SIGN, 1-66
SIN, 1-66
SINH, 1-66
SMALLINT, 1-66
SOAPHTPNC, 1-66
SOAPHTTPC, 1-66
SOUNDEX, 1-66
SPACE, 1-67
SQRT, 1-67
STRIP, 1-67
SUBSTR, 1-67
SUBSTRING, 1-67
TAN, 1-67
TANH, 1-67
TIME, 1-67
TIMESTAMP, 1-67
TIMESTAMP_FORMAT, 1-68
TIMESTAMP_ISO, 1-68
TIMESTAMP_TZ, 1-68
TIMESTAMPADD, 1-68
TIMESTAMPDIFF, 1-68
TO_CHAR, 1-68
TO_DATE, 1-68
TO_NUMBER, 1-68
TOTALORDER, 1-68
TRANSLATE, 1-69
TRIM, 1-69
TRIM_ARRAY, 1-69
TRUNC_TIMESTAMP, 1-69
TRUNCATE, 1-69
UNICODE, 1-69
UNICODE_STR, 1-69
UPPER, 1-69
VALUE, 1-69
VARBINARY, 1-69
VARCHAR char, 1-70
VARCHAR_FORMAT, 1-71
VARCHAR_FORMAT timestamp, 1-71
VARGRAPHIC, 1-71
VERIFY_GROUP_FOR_USER, 1-71
VERIFY_ROLE_FOR_USER, 1-71
VERIFY_TRUSTED_CONTEXT_ROLE_FOR_USER, 1-71
WEEK, 1-71, 1-72
WEEK_ISO, 1-72
XMLATTRIBUTES, 1-72
XMLCOMMENT, 1-72
XMLCONCAT, 1-72
XMLDOCUMENT, 1-72
XMLELEMENT, 1-72
XMLFOREST, 1-73
XMLMODIFY, 1-73
XMLNAMESPACES, 1-73
XMLPARSE, 1-73
XMLPI, 1-73
XMLQUERY, 1-73, 1-74
XMLSERIALIZE, 1-73
XMLTEXT, 1-74
XMLXSROBJECTID, 1-74
YEAR, 1-74
Schema privileges, 6-20
Search Conditions, 1-4
Security Authorizations, 6-16
  administrative authorities, 6-16
  collection privileges, 6-19
  database privileges, 6-19
  distinct type usage privileges, 6-20
  index privileges, 6-21
  package privileges, 6-18
  plan privileges, 6-18
  routine privileges, 6-20
  schema privileges, 6-20
  sequence privileges, 6-20
  storage group privileges, 6-21
  system privileges, 6-19
  table and view privileges, 6-18
  tablespace privileges, 6-21
  TRANSFER OWNERSHIP privileges, 6-20
  use privileges, 6-19
SELECT statement query, 1-7
Sequence privileges, 6-20
SQL
  Data types, 1-79
  Durations, 1-76

Functions, 1-52
Predicate Processing Summary, 1-80
Queries, 1-5
Reserved Words, 1-77
Statements, 1-8
Syntax Elements, 1-1
SQL Functions
Aggregate, 1-52
MQ Series, 1-74
Scalar, 1-53
Table, 1-74
SQL Limits, 6-8
SQLCA, 6-1
SQLDA, 6-4
SQLSTATE Class Codes, 7-41
SSR IMS command, 3-16
Stand-alone Utilities
DSN1COMP, 2-20
DSN1COPY, 2-20
DSN1LOGP, 2-20
DSN1PRNT, 2-20
DSN1SDMP, 2-20
DSNJCNVB, 2-20
DSNJU003, 2-20
DSNJU004, 2-20
Stand-alone Utilities
DSN1COPY, 2-18
DSN1LOGP, 2-18
DSN1PRNT, 2-19
DSN1SDMP, 2-19
DSNJCNVB, 2-17
DSNJCNVT, 2-17
DSNJLOGF, 2-17
DSNJU003, 2-17
DSNJU004, 2-18
START IMS command, 3-16
START z/OS IRLM command, 3-15
Statements
ALLOCATE CURSOR, 1-8
ALTER DATABASE, 1-8
ALTER FUNCTION, 1-8, 1-9, 1-10
ALTER INDEX, 1-11
ALTER MASK, 1-11
ALTER PERMISSION, 1-11
ALTER PROCEDURE – SQL native, 1-12
ALTER PROCEDURE -external, 1-11
ALTER PROCEDURE –SQL external, 1-12
ALTER PROCEWDURE -- external, 1-11
ALTER SEQUENCE, 1-13
ALTER STOGROUP, 1-13
ALTER TABLE, 1-14
ALTER TABLESPACE, 1-17
ALTER TRIGGER (advanced), 1-17
ALTER TRIGGER (basic), 1-18
ALTER TRUSTED CONTEXT, 1-18
ALTER VIEW, 1-19
ASSOCIATE LOCATORS, 1-19
BEGIN DECLARE, 1-19
CALL, 1-19
CLOSE, 1-19
COMMENT, 1-19
COMMIT, 1-20
CONNECT, 1-20
CREATE ALIAS, 1-20
CREATE AUXILIARY TABLE, 1-20
CREATE DATABASE, 1-20
CREATE FUNCTION, 1-20
CREATE FUNCTION (compiled SQL scalar), 1-20
CREATE FUNCTION (External scalar), 1-22
CREATE FUNCTION (External table), 1-23
CREATE FUNCTION (inline SQL scalar), 1-24
CREATE FUNCTION (sourced), 1-23
CREATE FUNCTION (SQL table), 1-24
CREATE GLOBAL TEMPORARY TABLE, 1-24
CREATE INDEX, 1-25
CREATE MASK, 1-26
CREATE PERMISSION, 1-26
CREATE PROCEDURE, 1-27
CREATE PROCEDURE – SQL external, 1-27
CREATE PROCEDURE (external), 1-26
CREATE ROLE, 1-29
CREATE SEQUENCE, 1-30
CREATE STOGROUP, 1-30
CREATE SYNONYM (deprecated), 1-30
CREATE TABLE, 1-30
CREATE TABLESPACE, 1-32
CREATE TRIGGER (advanced), 1-33
CREATE TRIGGER (basic), 1-33
CREATE TRUSTED CONTEXT, 1-34
CREATE TYPE (array), 1-34
CREATE TYPE (distinct), 1-34
CREATE VARIABLE, 1-35
CREATE VIEW, 1-35
DECLARE CURSOR, 1-35
DECLARE GLOBAL TEMPORARY TABLE, 1-35
DECLARE STATEMENT, 1-36

DECLARE TABLE, 1-36
DECLARE VARIABLE, 1-36
DELETE (cursor-postioned), 1-36
DELETE (searched), 1-36
DESCRIBE CURSOR, 1-36
DESCRIBE INPUT, 1-36
DESCRIBE OUTPUT, 1-36
DESCRIBE PROCEDURE, 1-37
DESCRIBE TABLE, 1-37
DROP, 1-37
END DECLARE SECTION, 1-37
EXCHANGE, 1-37
EXECUTE, 1-37
EXECUTE IMMEDIATE, 1-37
EXPLAIN, 1-37
FETCH, 1-38
FREE LOCATOR, 1-38
GET DIAGNOSTICS, 1-38
GRANT (authorization)N, 1-39
GRANT (collection privileges), 1-39
GRANT (database privileges), 1-39
GRANT (function or procedure privileges, 1-39
GRANT (package privileges), 1-39
GRANT (plan privileges), 1-40
GRANT (schema privileges), 1-40
GRANT (sequence privileges), 1-40
GRANT (system privileges), 1-40
GRANT (table or view privileges), 1-40
GRANT (type or JAR file privileges), 1-40
GRANT (use privileges), 1-40
GRANT (variable privileges), 1-40
HOLD LOCATOR, 1-40
INCLUDE, 1-40
INSERT, 1-41
LABEL, 1-41
LOCK TABLE, 1-41
MERGE, 1-41
OPEN, 1-42
PREPARE, 1-42
REFRESH TABLE, 1-42
RELEASE (connection), 1-42
RELEASE SAVEPOINT, 1-42
RENAME, 1-42
REVOKE (database privileges), 1-43
REVOKE (function or procedure privileges), 1-43
REVOKE (package privileges), 1-43
REVOKE (plan privileges), 1-43
REVOKE (schema privileges), 1-43
REVOKE (sequence privileges), 1-43
REVOKE (system privileges), 1-44
REVOKE (table or view privileges), 1-44
REVOKE (type or JAR file privileges), 1-44
REVOKE (use privileges), 1-44
REVOKE (variable privileges), 1-44
REVOKE authorization privileges, 1-43
REVOKE collection privileges, 1-43
ROLLBACK, 1-44
SAVEPOINT, 1-44
SELECT, 1-44
SELECT INTO, 1-44
SET assignment-statement, 1-45
SET CONNECTION, 1-45
SET CURRENT APPLICATION COMPATIBILITY, 1-45
SET CURRENT APPLICATION ENCODING SCHEME, 1-45
SET CURRENT DEBUG MODE, 1-45
SET CURRENT DECFLOAT ROUNDING MODE, 1-45
SET CURRENT DEGREE, 1-45
SET CURRENT EXPLAIN MODE, 1-45
SET CURRENT GET_ACCEL_ARCHIVE, 1-45
SET CURRENT LOCALE LC_CTYPE, 1-45
SET CURRENT MAINTAINED TABLE TYPES FOR OPTIMIZATION, 1-45
SET CURRENT OPTIMIZATION HINT, 1-45
SET CURRENT PACKAGE PATH, 1-46
SET CURRENT PACKAGESET, 1-46
SET CURRENT PRECISION, 1-46
SET CURRENT QUERY ACCELERATION, 1-46
SET CURRENT REFRESH AGE, 1-46

SET CURRENT ROUTINE VERSION, 1-46
SET CURRENT RULES, 1-46
SET CURRENT SQLID, 1-46
SET CURRENT TEMPORAL BUSINESS_TIME, 1-46
SET CURRENT TEMPORAL SYSTEM_TIME, 1-46
SET ENCRYPTION PASSWORD, 1-46
SET PATH, 1-46
SET SCHEMA, 1-46
SET SESSION TIME ZONE, 1-46
SIGNAL, 1-46
TRANSFER OWNERSHIP, 1-47
TRUNCATE, 1-47
UPDATE, 1-47
VALUES, 1-47
VALUES INTO, 1-47
WHENEVER, 1-47
Statistics Trace Records, 6-44
STOP IMS command, 3-16
STOP z/OS IRLM command, 3-15
Storage group privileges, 6-21
Subselect query, 1-5
Syntax Elements, 1-1
Expressions, 1-1
Predicates, 1-3
Search Conditions, 1-4
System privileges, 6-19
System Tables, 4-1
Catalog Tables, 4-1
Directory Tables, 4-227
Other System Tables, 4-232
Table and View privileges, 6-18
Table Functions, 1-74
Tablespace privileges, 6-21
TRACE CT z/OS IRLM command, 3-15
TRACE IMS command, 3-16
TRANSFER OWNERSHIP privileges, 6-20
TSO Commands
DSN, 3-14
DSNH, 3-14
Use privileges, 6-19
User Tables, 5-1
DSN_COLDIST_TABLE, 5-8
DSN_DETCOST_TABLE, 5-9
DSN_FILTER_TABLE, 5-13
DSN_FUNCTION_TABLE, 5-14
DSN_KEYTGTDIST_TABLE, 5-15
DSN_PGRANGE_TABLE, 5-16
DSN_PGROUP_TABLE, 5-17
DSN_PREDICAT_SELECTIVITY_TABLE, 5-21
DSN_PREDICAT_TABLE, 5-19
DSN_PTASK_TABLE, 5-22
DSN_QUERY_TABLE, 5-23
DSN_QUERYINFO_TABLE, 5-24
DSN_SORT_TABLE, 5-25
DSN_SORTKEY_TABLE, 5-26
DSN_STAT_FEEDBACK_TABLE, 5-34
DSN_STATEMENT_CACHE_TABLE, 5-27
DSN_STATEMNT_TABLE, 5-31
DSN_STRUCT_TABLE, 5-33
DSN_VIEWREF_TABLE, 5-35
Plan Table, 5-1
Utilities
Authorities Required, 2-20
On-line DB2, 2-1
Stand-Alone, 2-17
VALUES INTO statement, 1-47
VALUES statement, 1-47
z/OS IRLM Commands
MODIFY ABEND, 3-15
MODIFY DIAG, 3-15
MODIFY PURGE, 3-15
MODIFY STATUS, 3-15
MODIFY STOP, 3-15
SET, 3-15
START, 3-15
TRACE CT, 3-15

A Broadcom Company

technologies

Reference Guide
for Db2 12 for z/OS

Instant Access to Essential Information

Broadcom Inc.
1320 Ridder Park Drive
San Jose, California 95131
United States
+1 408.433.8000

ca.com/db2