

APPENDIX A

DB2 Subsystem Parameters

The following table shows all the available subsystem parameters (DSNZPARMs). The details include the parameter name, a short description, acceptable values with defaults, and whether the parameter is changeable online. For more information about DSNZPARMs, refer to *IBM DB2 11 for z/OS Installation and Migration*—GC19-4056-02.

Note: the table does not show hidden DSNZPARMs that might exist.

DSNZPARMs

Parameter	Description	Acceptable Values (defaults appear in bold)	Online
ABEXP	EXPLAIN processing	YES , NO	Yes
ABIND	Auto BIND	YES , NO	Yes
ACCEL	Acceleration startup	NO , AUTO, COMMAND	Yes
ACCESS_CNTL_MODULE	DB2 access control routine load module	1–8 Char (install), DSNX@XAC (migration)	No
ACCUMACC	DDF/RRSAF accumulation data	NO , 2–65535	Yes
ACCUMUID	Aggregation fields	0 –10	Yes
ADMTPROC	JCL procedure used to start administrative scheduler	1–8, ssnADMT	No
AEXITLIM	Authorization exit limit	0–32676; 10	Yes

Parameter	Description	Acceptable Values (defaults appear in bold)	Online
AGCCSID	ASCII coded character set (graphic)	0 –65533	—
ALCUNIT	Allocation units	BLK , TRK, CYL	Yes
ALL/dbname	Start names	ALL , space names	—
AMCCSID	ASCII coded character set (mixed)	0 –65533	—
APPENSCH	Application encoding	ASCII, EBCDIC , UNICODE, ccsid	—
APPLCOMPAT	Application compatibility	V10R1, V11R1	Yes
ARCPFX1	Copy 1 prefix	1–34 char	Yes
ARCPFX2	Copy 2 prefix	1–34 char	Yes
ARCRETN	Retention period	0 – 9999	Yes
ARCWRTC	WTOR route code	1–16; 1,3,4	Yes
ARCWTOR	Write to operator	NO, YES	Yes
ARC2FRST	Read copy 2 archive	NO , YES	Yes
ASCCSID	ASCII coded character set (single-byte)	0 –65533	—
AUDITST	Audit trace	NO , YES, list, *	No
AUTH	Use protection	YES , NO	No
AUTHCACH	Plan authorization cache	0–4096; 1024	Yes
AUTHEXIT_@ CACHEREFRESH	Authorization cache refresh	ALL , NONE	Yes
AUTHEXIT_CHECK	ID used for authorization checks	PRIMARY , DB2	Yes
BACKODUR	Backout duration	0–255; 5	No
BINDNV	Bind new package	BINDADD , BIND	Yes
BLKSIZE	Block size	8192– 28672	Yes
BMPTOUT	IMS BMP timeout	1–254; 4	Yes
CACHEDYN	Cache dynamic SQL	NO, YES	Yes
CACHEDYN_FREELOCAL	Free cached dynamic statements to relieve DBM1 below-the-bar-storage	0,1	Yes
CACHEPAC	Package authorization cache	0–2 MB; 5 MB	No
CACHERAC	Routine authorization cache	0–2 MB; 5 MB	No
CATALOG	Catalog alias	1–8 char; DSNCAT	Yes
CATDDACL	SMS class for catalog and directory table spaces	Valid SMS class, blank	Yes
CATDMGCL	SMS class for catalog and directory table spaces	Valid SMS class, blank	Yes

Parameter	Description	Acceptable Values (defaults appear in bold)	Online
CATDSTCL	SMS class for catalog and directory table spaces	Valid SMS class, blank	Yes
CATXDACL	SMS class for catalog and directory indexes	Valid SMS class, blank	Yes
CATXMGCL	SMS class for catalog and directory indexes	Valid SMS class, blank	Yes
CATXSTCL	SMS class for catalog and directory indexes	Valid SMS class, blank	Yes
CDSSRDEF	Current degree	1 , ANY	Yes
CHECK_ FASTREPLICATION	Type of replication used to copy objects	PREFERRED, REQUIRED	Yes
CHECK_SETCHKP	CHECK DATA and CHECK LOB place inconsistent objects in CHECK PENDING	YES, NO	Yes
CHGDC	DROP support	1 , 2, 3	Yes
CHKFREQ	Number of log records created between checkpoints	1000–16000000 (CHKTYPE= LOGRECS), NOTUSED, @ (CHKTYPE= MINUTES), 5, @ 99999999 (CHKTYPE=BOTH)	Yes
CHKLOGR	Number of log records created between checkpoints	1000–16000000 (CHKTYPE= LOGRECS), NOTUSED @ (CHKTYPE= MINUTES), @ 99999999 (CHKTYPE=BOTH)	Yes
CHKMINS	Number of minutes between log checkpoints	NOTUSED (CHKTYPE=LOGRECS), @ 1-60, 5 @ (CHKTYPE= MINUTES), @ 1-439 (CHKTYPE=BOTH)	Yes
CHKTYPE	Interval between checkpoints	LOGRECS, MINUTES , BOTH	Yes
CMTSTAT	DDF threads	ACTIVE , INACTIVE	No
COMPACT	Compact data	NO , YES	Yes
COMPAT	IBM service	OFF	—
COMPRESS_SPT01	Compress SPT01 directory table space	YES , NO	Yes
CONDBAT	Max remote connected	0–25000; 64	Yes
CONTSTOR	Contract thread storage	NO, YES	Yes
CTHREAD	Max users	1–2000; 70	Yes
DATE	Date format	ISO , USA, EUR, JIS, LOCAL	—
DATELEN	Local date length	0 , 10–254	—

Parameter	Description	Acceptable Values (defaults appear in bold)	Online
DB2SORT	DB2 sort or DFSORT	ENABLE , DISABLE	Yes
DB2SUPLD	Serviceability parameter	—	—
DBACRVW	DBADM can create a view for other authid	YES, NO	Yes
DDF	DDF startup option	NO , AUTO, COMMAND	No
DDLTOX	DDL timeout	1–254	Yes
DEALLCT	Deallocate period	0–1439 min, 0–59 sec, NOLIMIT	Yes
DECARTH	Decimal arithmetic	DEC15 , DEC31, 15, 31	—
DECDIV3	Minimum divide scale	NO , YES	No
DECIMAL	Decimal point	, .	—
DEF_DECFLOAT_ROUND_MODE	Decfloat rounding mode	ROUND_CELING, ROUND_DOWN, ROUND_FLOOR, ROUND_HALF_DOWN, ROUND_HALF_EVEN , ROUND_HALF_UP, ROUND_HALF_EVEN	—
DEFLANG	Language default	ASM, C, CPP, COBOL, COB2, IBMCOB , FORTRAN, PL1	—
DEFLTID	Unknown authid	IBMUSER , authid	No
DEL_CFSTRUCTS_ON_RESTART	Delete coupling facility structures on restart	NO , YES	No
DELIM	String delimiter	DEFAULT , " , "	—
DESCSTAT	Describe for static	NO , YES	Yes
DISABLE_EDMRTS	Disable EDM RTS	NO , YES	Yes
DLDFREQ	Level ID update frequency	0–32767; ON	Yes
DLITOUT	DL/I batch timeout	1–254; 6	Yes
DPSEGSZ	Default Partition Segsize	0,4,12,..64 (multiples of 4), 32	Yes
DSCVI	Vary DS control interval	YES , NO	Yes
DSHARE	Data sharing	Yes , No, blank	No
DSMAX	Data set maximum	1–32767; 20000	Yes
DSQDEL	Dist SQL string delimiter	' , "	—
DSSTIME	Data set stats time	1–1440; 5	Yes
DYNRULES	Use for dynamic rules	YES , NO	—
EDM_SKELETON_POOL	Minimum size of EDM skeleton pool in KB	5120–2097152, 10240 KB	Yes
EDMDBDC	EDM DBD cache	5000K –2097152 KB	Yes
EDMPOOL	Maximum size below 2GB bar for EDM	0 –2097152	Yes
EDMSTMTC	EDM statement cache size	0–1048576K; 113386 KB	Yes

Parameter	Description	Acceptable Values (defaults appear in bold)	Online
EDPROP	DROP support	1 , 2, 3	Yes
EN_PJSJ	Enable index ANDing	ON, OFF	Yes
ENSCHHEME	Default encoding scheme	EBCDIC , ASCII	—
EVALUNC	Predicate evaluation with UR and RS	YES, NO	Yes
EXTRAREQ	Extra blocks requestor	0– 100	Yes
EXTRASRV	Extra blocks server	0– 100	Yes
EXTSEC	Extended security	NO , YES	Yes
FCCOPYDDN	Default for FCCOPYDDN of FLASHCOPY clause for utilities	Valid DB2 utilities template, HLQ.&DB.&SN.N&DS.D&JU.T&TL	Yes
FLASHCOPY	Default setting for FLASHCOPY for utilities	NO , YES, CONSISTENT	Yes
FLASHCOPY_COPY	FLASHCOPY clause used by default for COPY utility	NO, YES	Yes
FLASHCOPY_LOAD	FLASHCOPY clause used by default for LOAD utility	NO, YES	Yes
FLASHCOPY_PPRC	FLASHCOPY behavior with PPRC	NO, YES	Yes
FLASHCOPY_REBUILD_INDEX	FLASHCOPY clause used by default for REBUILD INDEX utility	NO, YES	Yes
FLASHCOPY_REORG_INDEX	FLASHCOPY clause used by default for REORG INDEX utility	NO, YES	Yes
FLASHCOPY_REORG_TS	FLASHCOPY clause used by default for REORG TABLESPACE utility	NO, YES	Yes
GCCSID	EBCDIC coded character set (graphic byte)	0–65533	—
GET_ACCEL_ARCHIVE	Get Accel Archive	NO , YES	Yes
GRPNAME	Group name	1–8 char; DSNCAT	No
HONOR_KEEPPDICTIONARY	Honor KEEPPDICTIONARY on LOAD or REORG	YES, NO	Yes
IDAUTH_MODULE	DB2 connection authorization exit routine	1–8 char; DSN3@ATH	No
IDBACK	Max batch connect	1–2000; 40	Yes
IDFORE	Max TSO connect	1–2000; 40	Yes
IDTHTOIN	Idle thread timeout	0–9999	Yes
IDXBPOOL	Default buffer pool for user indexes	BP0 –BPx	Yes
IGNSORTN	Ignore SORTNUM in utilities	YES, NO	Yes

Parameter	Description	Acceptable Values (defaults appear in bold)	Online
IMMEDWRI	Immediate write	NO , YES, PH1	Yes
IMPDSDEF	Define data sets	YES, NO	Yes
IMPDSIZE	Max DSSIZE for implicit	1,2, 4 ,8,16,32,64	Yes
IMPLICIT_TIMEZONE	Default value for time zone	CURRENT , SESSION, 12:59 to +14:00	—
IMPTKMOD	Trackmod for implicit	YES , NO	Yes
IMPTSCMP	Use data compression	YES, NO	Yes
INDEX_CLEANUP_THREADS	Index cleanup threads	0–128, 10	Yes
INDEX_IO_PARALLELISM	I/O parallelism enable for index insertion I/O	YES , NO	Yes
INLISTP	IN list elements	1–5000; 50	Yes
IRLMAUT	Auto start	YES , NO	No
IRLMPROC	Proc name	IRLMPROC , IRLM procedure name	No
IRLMRWT	Resource timeout	1–3600; 30	No
IRLMSID	Subsystem name	IRLM , IRLM name	No
IRLMSWT	Time to auto start	1–3600; 120	Yes
IX_TB_PART_@ CONV_EXCLUDE	Exclude trailing columns	NO , YES	Yes
IXQTY	Index space default size	0 –4194304	Yes
LBACKOUT	Postpone backward log processing	AUTO , YES, NO	No
LC_CTYPE	Locale LC_CTYPE	Valid locale, 0 –50 char	—
LEMAX	Maximum LE tokens	0–50; 20	No
LIKE_BLANK_INSIGNIFICANT	Blank significant in LIKE	NO , YES	
LOB_INLINE_LENGTH	Default length for inline LOBs	0 –32680	Yes
LOBVALA	User LOB value storage	1–2097152; 2048	Yes
LOBVALS	User LOB value storage	1–510002; 2048	Yes
LRDRTHLD	Long-running reader threshold	0–1439 minutes, 10	Yes
MAINTYPE	Current maintenance types for MQTs	NONE, SYSTEM , USER, ALL	Yes
MAX_CONCURRENT_PKG_OPS	Max number of automatic bind requests processed simultaneously	10	Yes
MAX_NUM_CUR	Max open cursors	0–99999; 500	Yes
MAX_ST_PROC	Max number of stored procedures	0–99999; 2000	Yes
MAXARCH	Recording max	10– 1000	No

Parameter	Description	Acceptable Values (defaults appear in bold)	Online
MAXCONQN	Max queued connections	OFF , ON, 1–19999	Yes
MAXCONQW	Max queued wait time	OFF , ON, 5–3600	Yes
MAXDBAT	Max remote active	0–1999; 64	Yes
MAXKEEPD	Max kept dynamic statements	0–65535; 5000	Yes
MAXOFILR	Maximum number of data sets concurrently open for LOB file references	0 to value of MAX USERS, 100	Yes
MAXRBLK	RID pool size	0, 16 KB–1000000 KB; 400 MB	Yes
MAXRTU	Read tape units	1–99; 2	Yes
MAXSORT_IN_MEMORY	Max in-memory sort size	1000 to value of SRTPOOL	Yes
MAXTEMPS	Max temp/stage agent	0 –214748364	Yes
MAXTEMPS_RID	Max temp storage in work files for RIDs	NONE , NOLIMIT, or 1 to 329166	Yes
MAXTYPE1	Max type 1 inactive	0 –MAX REMOTE CON value	Yes
MCCSID	EBCDIC coded character set (mixed byte)	0 –65533	—
MEMBNAME	Member name	1–8 char; DSN1	No
MGEXTSZ	Optimize extent sizing	YES , NO	Yes
MINSTOR	Thread management	YES, NO	Yes
MIXED	Mixed data	NO , YES	—
MON	Monitor trace	NO , YES	No
MONSIZE	Monitor size	256K– 1MB	No
MXDTCACH	Maximum size of memory for data caching	0–512, 20	Yes
NPGTHRSH	Use of index after table growth	0 , –1, n	Yes
NUMLKTS	Locks per table space	0–50000; 2000	Yes
NUMLKUS	Locks per user	0–100000; 10000	Yes
OBJECT_CREATE_ FORMAT	Objects created with basic or extended log	BASIC(migration), EXTENDED(install)	Yes
OPT1ROWBLOCKSORT	Block sort operations for OPT FOR 1 ROW	ENABLE, DISABLE	Yes
OPTHINTS	Optimization hints	NO , YES	Yes
OTC_LICENSE	Accept terms of one-time change license	YES, NONE	No
OUTBUFF	Output buffer	40K–400 MB; 400 KB	No
PADIX	Pad index by default	YES, NO	Yes
PADNTSTR	Pad null-terminated strings	YES, NO	Yes
PARA_EFF	Parallelism efficiency	0–100, 50	Yes

Parameter	Description	Acceptable Values (defaults appear in bold)	Online
PARAMDEG	Degree of parallelism	0 —no upper limit	Yes
PARAMDEG_DPSI	Deg of parallelism for DPSI	0 –254, DISABLE	Yes
PARAMDEG_UTIL	Deg of parallelism for utilities	0 –32767	Yes
PCLOSEN	RO switch checkpoints	1–32767; 5	Yes
PCLOSET	RO switch time	1–32767; 10	Yes
PCTFREE_UPD	Percentage free for update	AUTO, 0 –99	Yes
PKGREL_COMMIT	Package Release Commit	YES , NO	Yes
PLANMGMT	Default plan management policy	OFF, ON, BASIC, EXTENDED	Yes
PLANMGMT_SCOPE	Default plan management scope	ALL, STATIC , DYNAMIC	Yes
POOLINAC	Pool thread timeout	0–9999; 120	Yes
PREVENT_ALERTTB_LIMITKEY	Prevent Alter Limitkey	NO , YES	Yes
PREVENT_NEW_IXCTRL_PART	Prevent Index Part Create	NO , YES	Yes
PRIQTY	Primary quantity	Blank , 1–9999999	Yes
PROTECT	Archive logs protected with RACF	NO , YES	Yes
PTASKROL	Include accounting traces for parallel tasks	YES , NO	Yes
QUERY_ACCEL_OPTIONS	Acceleration options	NONE , YES	Yes
QUERY_ACCELERATION	Default for CURRENT QUERY ACCELERATION special register	NONE , ENABLE, ENABLE_WITHFAILBACK	Yes
QUIESCE	Quiesce period	0–999; 5	Yes
RANDOMATT	DB2 member can be used for randomized group attach	YES , NO	Yes
REALSTORAGE_MANAGEMENT	Whether DB2 manages real storage	ON, OFF, AUTO	Yes
REALSTORAGE_MAX	Maximum GB or real and auxiliary storage DB2 can consume	NOLIMIT , 1–65535	Yes
REC_FASTREPLICATION	Use FLASHCOPY with recover	NONE , PREFERRED , REQUIRED	Yes
RECALL	Recall database	YES , NO	No
RECALLD	Recall delay	0–32767; 120	Yes
REFSHAGE	Current refresh age	0 , ANY	Yes
REORG_DROP_PBG_PARTS	Remove trailing empty partitions during REORG	DISABLE , ENABLE	

Parameter	Description	Acceptable Values (defaults appear in bold)	Online
REORG_IGNORE_FREESPACE	Ignore free space values during REORG	NO , YES	
REORG_LIST_PROCESSING	Default for PARALLEL option on REORG	PARALLEL , SERIAL	Yes
REORG_MAPPING_DATABASE	Default database for mapping table	Blank , 8-byte character string	
REORG_PART_SORT_NPI	Default for sorting nonpartitioned index during REORG	AUTO , NO, YES	Yes
RESTART	Restart or defer	RESTART , DEFER	—
RESTORE_RECOVER_FROMDUMP	Recovery/restore	YES, NO	Yes
RESTORE_TAPEUNITS	Maximum tape units	NOLIMIT , 1–255	Yes
RESTRICT_ALT_COL_FOR_DCC	Restrictions during ALTER TABLE ALTER COLUMN if DATA CAPTURE CHANGES is used	NO , YES	Yes
RESYNC	Resync interval	1–99; 2	Yes
RETLWAIT	Retained lock timeout	0 –254	Yes
REVOKE_DEP_PRIVILEGES	Revoked privileges are also revoked from dependents	NO, YES, SQLSTMT	Yes
RGFCOLID	Registration owner	1–8 char; DSNRGCOL	No
RGFDBNAM	Registration database	1–8 char; DSNRGFDB	No
RGFDEDPL	Control all applications	NO , YES	No
RGFDEFLT	Unregistered DDL default	APPL, ACCEPT , REJECT	No
RGFESCP	ART/ORT escape character	Non-alphanumeric char	No
RGFFULLQ	Require full names	YES , NO	No
RGFINSTL	Install DD control support	NO , YES	No
RGFNMORT	OBJT registration table	1–17 char; DSN_REGISTER_OBJT	No
RGFNMPRT	APPL registration table	1–17 char ; DSN_REGISTER_APPL	No
RLF	RLF auto start	NO , YES	No
RLFAUTH	Resource authid	SYSIBM , authid	Yes
RLFERR	RLST access error	NOLIMIT , NORUN, 1–50000000	Yes
RLFERRD	RLST access error	NOLIMIT , NORUN, 1–50000000	Yes
RLFTBL	RLST name suffix	01 , 2 alphanumeric char	Yes
ROUTCDE	WTO route codes	1 , 1–14 route codes	No
RRF	Store data in reordered row format (RRF)	ENABLE , DISABLE	Yes

Parameter	Description	Acceptable Values (defaults appear in bold)	Online
RRULOCK	U lock for RR/RS	NO, YES	Yes
SCCSID	EBCDIC coded character set (single-byte)	0 –65533	—
SECADM1	Security administrator	SECADM , if type is AUTHID, then 1–8 characters; if type is ROLE, then SQL identifier	Yes
SECADM1_INPUT_STYLE	Setting passed as hex string or character	CHAR , HEX	Yes
SECADM1_TYPE	Type of security administrator	AUTHID , ROLE	Yes
SECADM2	Second security administrator	SECADM , if type is AUTHID then, 1–8 characters; if type is ROLE, then SQL identifier	Yes
SECADM2_INPUT_STYLE	Setting passed as hex string or character	CHAR , HEX	Yes
SECADM2_TYPE	Type of security administer	AUTHID , ROLE	Yes
SECQTY	Secondary quantity	Blank (clist calculated) , 1–9999999	Yes
SEPARATE_SECURITY	Separate DB2 security administration from system administration	NO , YES	Yes
SIGNON_MODULE			No
SIMULATED_CPU_COUNT	Number of CPUs being simulated	OFF , 1–255	Yes
SIMULATED_CPU_SPEED	Microseconds of execution time for CPU being simulation	OFF , 1–2147483647	Yes
SITETYP	Site type	LOCALSITE , RECOVERYSITE	No
SJTABLES	Number of tables in star join	1–255; 10	Yes
SKIPUNCI	Skip uncommitted inserts	YES, NO	Yes
SMF89	Measured usage pricing	YES, NO	Yes
SMFACCT	SMF accounting	NO, YES(1) , list (1–5,7,8), *	No
SMFCOMP	Compression SMF	OFF , ON	Yes
SMFSTAT	SMF statistics	YES (1,3,4) , NO, list(1–5) , *	No
SPT01_INLINE_LENGTH	Max inline length of LOB in SPT01	NOINLINE , 1–32138	Yes
SQLDELI	SQL string delimiter	Default , ' , "	—
SRTPOOL	Sort pool size	240K–64000 KB; 10000 KB	Yes
SSID	Subsystem name	DSN , SSID	—
STARJOIN	Enabling star join	Disable , enable, 1, 2–32768	Yes
STATFDBK_SCOPE	Statistics feedback	ALL , DYNAMIC,NONE, STATIC	Yes

Parameter	Description	Acceptable Values (defaults appear in bold)	Online
STATHIST	Collect historical statistics	SPACE, NONE , ALL, ACCESSPATH	Yes
STATIME	Statistics time	1–1440 min; 1	Yes
STATROLL	RUNSTATS aggregates partition-level statistics	YES, NO	Yes
STATSINT	Time to write RTS stats	1–1440 min; 30	Yes
STDSQL	Standard SQL language	NO , YES	—
STORMXAB	Max abend count	0 –225	Yes
STORTIME	Timeout value	5–1800 sec; 180	Yes
SUBQ_MIDX	Multiple index access	ENABLE , DISABLE	Yes
SUPERRS	Suppress logrec recording during soft errors	YES , NO	Yes
SVOLARC	Single volume	YES, NO	Yes
SYNCVAL	Statistics sync	NO , 0–59	Yes
SYSADM	System admin 1	SYSADM , authid	Yes
SYSADM2	System admin 2	SYSADM , authid	Yes
SYSOPR1	System operator 1	SYSOPR , authid	Yes
SYSOPR2	System operator 2	SYSOPR ; authid	Yes
SYSTEM_LEVEL_BACKUPS	System-level backups	YES , NO	Yes
TBSBP8K	Default 8K BP for user data	Any 8 KB buffer pool; BP8K0	Yes
TBSBP16K	Default 16K BP for user data	Any 16 KB buffer pool; BP16K0	Yes
TBSBP32K	Default 32K BP for user data	Any 32 KB buffer pool; BP32K0	Yes
TBSBPOOL	Default buffer pool for user data	BP0 –BPx	Yes
TBSBPXML	Default buffer pool for XML table spaces	Any 16 KB buffer pool, BP16K	Yes
TCPALVER	TCP/IP already verified	NO , YES	Yes
TCPKPALV	TCP/IP keep alive	ENABLE , DISABLE, 1–65524	Yes
TEMPLATE_TIME	Template time	UTC , LOCAL	Yes
TIME	Time format	ISO , JIS, USA, EUR, LOCAL	—
TIMELN	Local time length	0, 8–254	—
TRACSTR	Trace auto start	NO , YES (1–3), list (1–9)	No
TRACTBL	Trace size	4 KB–396 KB; 64 KB	No
TRKRSITE	Remote tracker site usage	NO , YES	No
TSQTY	Default allocation for table space	0 –4194304	Yes
TSTAMP	Timestamp archives	NO , YES	Yes
TWOACTV	Number of active copies	2 , 1	No

Parameter	Description	Acceptable Values (defaults appear in bold)	Online
TWOARCH	Number of archive copies	2 , 1	No
TWOBSDS	Number of BSDSs	YES , NO	No
UGCCSID	Unicode CCSID (graphic)	1208	—
UIFCIDS	Unicode IFCIDS	YES, NO	Yes
UMCCSID	Unicode CCSID (Mixed)	1208	—
UNION_COLNAME_7	Use Version 7 behavior for union results	YES, NO	Yes
UNIT	Device type 1	TAPE , any device	Yes
UNIT2	Device type 2	Device or unit name	Yes
URCHKTH	UR check frequency	0–255, 5	Yes
URLGWITH	UR log write check	0 KB– 1000 KB	Yes
USCCSID	Unicode CCSID (single-byte)	1208	—
UTIL_TEMP_STORCLAS	Storage class name for shadow data sets	Blank , valid SMS management class name	Yes
UTILITY_OBJECT_CONVERSION	Utility object conversion	BASIC, EXTENDED, NOBASIC, NONE	Yes
UTILS_DUMP_CLASS_NAME	Dump class name	Blank , valid DFSMS dump class name	Yes
UTIMOUT	Utility timeout	1–254; 6	Yes
UTSORTAL	DB2 uses RTS to determine sort work data set sizes	YES , NO	Yes
VOLTDEVT	Temporary unit name	SYSDA , valid name	Yes
WFDBSEP	Declared temporary tables will use work files with non-zero SECQTY	YES, NO	No
WFSTGUSE_AGENT_THRESHOLD	Agent level threshold in work file database	0 –100	Yes
WFSTGUSE_SYSTEM_THRESHOLD	System-level threshold in work file database	0 –100	Yes
WLMENV	WLM environment	Valid name (1–18 char)	Yes
XLKUPDT	X lock for searched U/D	YES, NO	Yes
XML_RANDOMIZE_DOCID	Generate XML DOCID values randomly	NO , YES	Yes
XMLVALA	Upper limit for storage for per user for XML values	1–2097152, 204800	Yes
XMLVALS	Upper limit for system storage for XML values	1–51200, 10240	Yes
ZOSMETRICS	Enables DB2 to gather z/OS metrics	YES, NO	No

B

APPENDIX

DB2 Directory and Catalog Tables

DB2 Directory Tables

The following table shows a list of the DB2 directory tables with a brief description of each. For columns and further details, refer to *IBM DB2 11 for z/OS SQL Reference*—SC19-4066-00.

Directory Table	Information Contents
SPT01	Referred to as the <i>skeleton package table</i> (SKPT), this table contains information about access paths and the internal form of the SQL for a package at bind time. Entries are made into this table during bind time (BIND PACKAGE) and are deleted when a package is freed (FREE PACKAGE). This table is loaded into memory at execution time (along with the SCT02 table, described next).
SCT02	Referred to as the <i>skeleton cursor table</i> (SKCT), this table contains information about access paths and the internal form of the SQL for an application plan. Entries in the table are made when a plan is bound (BIND PLAN) and are deleted when a plan is freed (FREE PLAN). Like table SPT01, this table is loaded into memory at execution time.
DBD01	This table holds information about database descriptors (DBDs), which are internal control blocks. Each database in DB2 has one DBD for its objects (table spaces, indexes, tables, referential integrity constraints, and check constraints). Updates to the table are made when a database is created or updated. DB2 accesses this information instead of continually using the DB2 catalog. This mechanism allows for faster, more efficient access to the data. The information in this directory table also resides in the DB2 catalog.

Directory Table	Information Contents
SYSLGRNX	Referred to as the <i>log range table</i> , this table contains data from the DB2 logs about the relative byte address (RBA) range for updates. This information lets DB2 efficiently find the RBAs it needs from the DB2 logs for recovery purposes. A row is inserted every time a table space or partition is opened or updated and is updated when the object is closed.
SYSUTILX	The system utilities table stores data about the execution of DB2 utilities, including status and execution steps. DB2 uses this information when it needs to restart a utility. Information in this table is added when a utility is started; the entry is removed when the execution has ended.

DB2 Catalog Tables

The following table shows a list of the DB2 catalog tables with a brief description of each. For columns and further details, refer to *IBM DB2 11 for z/OS SQL Reference*—SC19-4066-00.

Catalog Table (SYSIBM. <i>table</i>)	Information Contents
IPLIST	Allows you to specify multiple IP addresses for a given location. Insert rows into this table when you want to define a remote DB2 data sharing group. Rows can be inserted, updated, and deleted.
IPNAMES	Defines the remote DRDA servers DB2 can access using TCP/IP. Rows in this table can be inserted, updated, and deleted.
LOCATIONS	Contains a row for each accessible remote server. The row associates a LOCATION name with the TCP/IP or SNA network attributes for the remote server. Requesters are not defined in this table. Rows in this table can be inserted, updated, and deleted.
LULIST	Lets you specify multiple LU names for a given location. Insert rows into this table when you want to define a remote DB2 data sharing group. The same value for the LUNAME column cannot appear in both the SYSIBM.LUNAMES table and the SYSIBM.LULIST table. Rows in this table can be inserted, updated, and deleted.
LUMODES	Each row of the table provides VTAM with conversation limits for a specific combination of LUNAME and MODENAME. The table is accessed only during the initial conversation limit negotiation between DB2 and a remote LU. This negotiation is called <i>change-number-of-sessions</i> (CNOS) processing. Rows in this table can be inserted, updated, and deleted.
LUNAMES	Contains a row for each remote SNA client or server that communicates with DB2. Rows can be inserted, updated, or deleted.
MODESELECT	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. Rows in this table can be inserted, updated, and deleted.
SYSAUDITPOLICIES	Contains one row for each audit policy.

Catalog Table (SYSIBM.table)	Information Contents
SYSAUTOALERTS	Contains one row for each recommendation from autonomic procedures.
SYSAUTOALERTS_OUT	An auxiliary table for the OUTPUT column of the SYSIBM.SYSAUTOALERTS table.
SYSAUTORUNS_HIST	Contains one row for each time an autonomic procedure has been run.
SYSAUTORUNS_HISTOU	An auxiliary table for the OUTPUT column of the SYSIBM.SYSAUTORUNS_HIST table.
SYSAUTOTIMEWINDOWS	Contains one row for each time period during which autonomic procedures can be run.
SYSAUXRELS	Contains one row for each auxiliary table created for a LOB column. A base table space that is partitioned must have one auxiliary table for each partition of each LOB column.
SYSCHECKDEP	Contains one row for each reference to a column in a table check constraint.
SYSCHECKS	Contains one row for each table check constraint.
SYSCHECKS2	Contains one row for each table check constraint created in or after DB2 for OS/390 V7.
SYSCOLAUTH	Records the UPDATE or REFERENCES privileges that users hold on individual columns of a table or view.
SYSCOLDIST	Contains one or more rows for the first key column of an index key. Rows in this table can be inserted, updated, and deleted.
SYSCOLDIST_HIST	Contains rows from table SYSCOLDIST. Whenever you add or change rows in SYSCOLDIST, the rows are also written to the new history table. Rows in this table can be inserted, updated, and deleted.
SYSCOLDISTSTATS	Contains zero or more rows per partition for the first key column of a partitioning index or Data Partitioned Secondary Index (DPSI). Rows are inserted when RUNSTATS scans index partitions of the partitioning index. No row is inserted if the index is a nonpartitioning index. Rows in this table can be inserted, updated, and deleted.
SYSCOLDIST_HIST	Contains rows from SYSCOLDIST. Rows are added or changed in this table when RUNSTATS collects history statistics. Rows in this table can be also inserted, updated, and deleted.
SYSCOLSTATS	Contains partition statistics for selected columns. For each column, a row exists for each partition in the table. Rows are inserted when RUNSTATS collects either indexed column statistics or non-indexed column statistics for a partitioned table space. No row is inserted if the table space is nonpartitioned. Rows in this table can be inserted, updated, and deleted.
SYSCOLUMNS	Contains one row for every column of each table and view.
SYSCOLUMNS_HIST	Contains rows from table SYSCOLUMNS. Whenever you add or change rows in SYSCOLUMNS, the rows are also written to the new history table. Rows in this table can be inserted, updated, and deleted.

Catalog Table (SYSIBM.table)	Information Contents
SYSCONSTDEP	Records dependencies on check constraints or user-defined defaults for a column.
SYSCONTEXT	Contains one row for each trusted context.
SYSCONTEXTAUTHIDS	Contains one row for each authorization ID with which a trusted context can be used.
SYSCOPY	Contains information needed for recovery.
SYSCTXTTRUSTATTRS	Contains one row for each list of attributes for a given trusted context.
SYSDATABASE	Contains one row for each database except for database DSNDB01.
SYSDATATYPES	Contains one row for each distinct type defined to the system.
SYSDBAUTH	Records the privileges users hold over databases.
SYSDBRM	Contains one row for each DBRM of each application plan.
SYSDEPENDENCIES	Records the dependencies between objects.
SYSDUMMY1	Contains one row. Use this table for SQL statements in which a table reference is required, but the table's contents aren't important.
SYSDUMMYA	ASCII version of SYSDUMMY1.
SYSDUMMYE	EBCDIC version of SYSDUMMY1.
SYSDUMMYU	UNICODE version of SYSDUMMY1.
SYSENVIRONMENT	Records the environment variables when an object is created.
SYSFIELDS	Contains one row for every column that has a field procedure.
SYSFOREIGNKEYS	Contains one row for every column of every foreign key.
SYSINDEXES	Contains one row for every index.
SYSINDEXES_HIST	Contains rows from table SYSINDEXES. Whenever you add or change rows in SYSINDEXES, the rows are also written to the new history table. Rows in this table can be inserted, updated, and deleted.
SYSINDEXES_RTSECT	An auxiliary table for the RTSECTION column of the SYSIBM.SYSINDEXES table and is required to hold LOB data.
SYSINDEXES_TREE	An auxiliary table for the PARSETREE column of the SYSIBM.SYSINDEXES table and is required to hold LOB data.
SYSINDEXPART	Contains one row for each nonpartitioning index and one row for each partition of a partitioning index or a DPSI.
SYSINDEXPART_HIST	Contains rows from table SYSINDEXPART. Whenever you add or change rows in SYSINDEXPART, the rows are also written to the new history table. Rows in this table can be inserted, updated, and deleted.
SYSINDEXSPACESTATS	Contains real-time statistics for index spaces. Rows in this table can be inserted, updated, and deleted.
SYSINDEXSTATS	Contains one row for each partition of a partitioning index. Rows in this table can be inserted, updated, and deleted.

Catalog Table (SYSIBM.table)	Information Contents
SYSINDEXSTATS_HIST	Contains rows from table SYSINDEXSTATS. Whenever you add or change rows in SYSINDEXSTATS, the rows are also written to the new history table. Rows in this table can be inserted, updated, and deleted.
SYSJARCLASS_SOURCE	Serves as an auxiliary table for table SYSCONTENTS.
SYSJARCONTENTS	Contains Java class source for an installed JAR.
SYSJARDATA	Serves as an auxiliary table for table SYSOBJECTS.
SYSJAROBJECTS	Contains binary large objects representing the installed JAR.
SYSJAVAOPTS	Contains build options used during INSTALL_JAR.
SYSJAVAPATHS	Contains the complete JAR resolution path and records the dependencies one JAR has on its JAR in its Java path.
SYSKEYCOLUSE	Contains a row for every column in a unique constraint (primary key or unique key) from the SYSTABCONST table.
SYSKEYS	Contains one row for each column of an index key.
SYSKEYTARGETS	Contains one row for each key-target that is participating in an extended index definition.
SYSKEYTARGETS_HIST	Contains rows from the SYSKEYTARGETS table whenever rows are added or changed.
SYSKEYTARGETSTATS	Contains partition statistics for select key-targets.
SYSKEYTGTDIST	Contains one or more rows for the first key-target of an extended index key.
SYSKEYTGTDIST_HIST	Contains rows from table SYSKEYTGTDIST whenever rows are added or changed in SYSKEYTGTDIST.
SYSKEYTGTDISTSTATS	Contains zero or more rows per partition for the first key-target of a partitioned secondary index.
SYSLOBSTATS	Contains one row for each LOB table space.
SYSLOBSTATS_HIST	Contains rows from table SYSLOBSTATS. Whenever you add or change rows in SYSLOBSTATS, the rows are also written to the new history table. Rows in this table can be inserted, updated, and deleted.
SYSOBJROLEDEP	Lists the dependence object for each role.
SYSPACKAGE	Contains a row for every package.
SYSPACKCOPY	Contains a row for every package.
SYSPACKAUTH	Records the privileges users hold over packages.
SYSPACKDEP	Records the dependencies of packages on local tables, views, synonyms, table spaces, indexes and aliases, functions, and stored procedures.
SYSPACKLIST	Contains one or more rows for every local application plan bound with a package list. Each row represents a unique entry in the plan's package list.

Catalog Table (SYSIBM.table)	Information Contents
SYSPACKSTMT	Contains one or more rows for each statement in a package.
SYSPACKSTMT_STMB	An auxiliary table for the STMTBLOB column of the SYSIBM.SYSPACKSTMT table and is required to hold LOB data.
SYSPACKSTMT_STMT	An auxiliary table for the STATEMENT column of the SYSIBM.SYSPACKSTMT table and is required to hold LOB data.
SYSPARMS	Contains a row for each parameter of a routine or multiple rows for table parameters (one for each column of the table).
SYSPENDINGDDL	Contains information about which objects have pending definition changes. The entries exist only during the window between when the pending option is executed and when the utility applies these pending changes to the object.
SYSPENDINGOBJECTS	Contains the name of and OBID information about objects that are pending creation.
SYSPKSYSTEM	Contains zero or more rows for every package. Each row for a given package represents one or more connections to an environment in which the package could be executed.
SYSPLAN	Contains one row for each application plan.
SYSPLANAUTH	Records the privileges users hold over application plans.
SYSPLANDEP	Records the dependencies of plans on tables, views, aliases, synonyms, table spaces, indexes, functions, and stored procedures.
SYSPLSYSTEM	Contains zero or more rows for every plan. Each row for a given plan represents one or more connections to an environment in which the plan could be used.
SYSQUERY	Contains one row for each query in a set of queries.
SYSQUERY_AUX	An auxiliary table for the STMTTEXT column of the SYSIBM.SYSQUERY table.
SYSQUERYOPTS	Contains optimization parameters for the queries that are in SYSIBM.SYSQUERY.
SYSQUERYPLAN	Contains the plan hint information for the queries in the SYSIBM.SYSQUERY table. It correlates to the SYSIBM.SYSQUERY table by the QUERYID column.
SYSRELS	Contains one row for every referential constraint.
SYSRESAUTH	Records CREATE IN and PACKADM ON privileges for collections, USAGE privileges for distinct types, and USE privileges for buffer pools, storage groups, and table spaces.
SYSROLES	Contains one row for each role.
SYSROUTINEAUTH	Records the privileges users hold on routines. (A routine can be a user-defined function, cast function, or stored procedure.)
SYSROUTINES	Contains a row for every routine. (A routine can be a user-defined function, cast function, or stored procedure.)

Catalog Table (SYSIBM.table)	Information Contents
SYSROUTINES_OPTS	Contains a row for each generated routine, such as one created by the DB2 Stored Procedure Builder tool, that records the build options for the routine. Rows in this table can be inserted, updated, and deleted.
SYSROUTINES_PTREE	An auxiliary table for the PTREE column of the SYSIBM.SYSROUTINES table.
SYSROUTINES_SRC	Contains source for generated routines, such as those created by the DB2/zOS procedure process or DSNTPSMP.
SYSROUTINESTEXT	Serves as an auxiliary table for the TEXT column of table SYSROUTINES and is required to hold the LOB data.
SYSSCHEMAAUTH	Contains one or more rows for each user who is granted a privilege on a particular schema in the database.
SYSSEQUENCEAUTH	Records the privileges users hold on sequences.
SYSSEQUENCES	Contains one row for each identity column.
SYSSEQUENCESDEP	Records the dependencies of identity columns on tables.
SYSSTMT	Contains one or more rows for each SQL statement of each DBRM.
SYSSTOGROUP	Contains one row for each storage group.
SYSSTRINGS	Contains information about character conversion. Each row describes a conversion from one coded character set to another.
SYSSYNONYMS	Contains one row for each synonym of a table or view.
SYSTABAUTH	Records the privileges users hold on tables and views.
SYSTABCONST	Contains one row for each unique constraint (primary key or unique key) created in DB2 for OS/390 Version 7 or later.
SYSTABLEPART	Contains one row for each nonpartitioned table space and one row for each partition of a partitioned table space.
SYSTABLEPART_HIST	Contains rows from table SYSTABLEPART. Rows are added or changed when RUNSTATS collects history statistics. Rows in this table can be inserted, updated, and deleted.
SYSTABLES	Contains one row for each table, view, or alias.
SYSTABLES_HIST	Contains rows from table SYSTABLES. Rows are added or changed when RUNSTATS collects history statistics. Rows in this table can be inserted, updated, and deleted.
SYSTABLES_PROFILES	Contains one row for each profile that is associated with a table in SYSIBM.SYSTABLES.
SYSTABLES_PROFILES_TEXT	An auxiliary table for the PROFILE_TEXT column of the SYSIBM.SYSTABLES_PROFILES table and is required to hold LOB data.
SYSTABLESPACE	Contains one row for each table space.
SYSTABLESPACESTATS	Contains the real-time statistics for table spaces.
SYSTABSTATS	Contains one row for each partition of a partitioned table space. Rows in this table can be inserted, updated, and deleted.

Catalog Table (SYSIBM.table)	Information Contents
SYSTABSTATS_HIST	Contains rows from table SYSTABSTATS. Rows are added or changed when RUNSTATS collects history statistics. Rows in this table can be inserted, updated, and deleted.
SYSTRIGGERS	Contains one row for each trigger.
SYSTRIGGERS_STMT	An auxiliary table for the STATEMENT column of the SYSIBM.SYSTRIGGERS table and is required to hold LOB data.
SYSUSERAUTH	Records the system privileges that users hold.
SYSVIEWDEP	Records the dependencies of views on tables, functions, and other views.
SYSVIEWS	Contains one or more rows for each view.
SYSVIEWS_STMT	An auxiliary table for the STATEMENT column of the SYSIBM.SYSVIEWS table and is required to hold LOB data.
SYSVIEWS_TREE	An auxiliary table for the PARSETREE column of the SYSIBM.SYSVIEWS table and is required to hold LOB data.
SYSVOLUMES	Contains one row for each volume of each storage group.
SYSXMLRELS	Contains one row for each XML table that is created for an XML column.
SYSXMLSTRINGS	Contains a single string and its unique ID that are used to condense XML data. The string can be an element name, attribute name, name space prefix, or a name space URI.
SYSUSERNAMES	Each row in this table carries out one of the following operations: @ Outbound ID translation @ Inbound ID translation and “come from” checking @ Rows in this table can be inserted, updated, and deleted.
SYSXMLTYPMOD	Contains rows about the XML type modifiers of XML columns. Rows in this table can be inserted, updated, and deleted.
SYSXMLTYPSCHEMA	Contains the XML schema information for an XML type modifier. It holds one row per XML schema for an XML type modifier.
SYSXSRCOMPONENT	Auxiliary table for BLOB column COMPONENT in table SYSXSROBJECTCOMPONENTS.
SYSXSROBJECTCOMPONENTS	Contains one row for each component (document) in an XML schema.
SYSXSROBJECTGRAMMER	Serves as an auxiliary table for the BLOB column GRAMMER in table SYSXSROBJECTS.
SYSXSROBJECTHIERARCHIES	Contains one row for each component (document) in an XML schema to record the XML schema document hierarchy relationship.
SYSXSROBJECTPROPERTY	Serves as an auxiliary table for the BLOB column PROPERTIES in table SYSXSROBJECTS.
SYSXSROBJECTS	Contains one row for each registered XML schema.
SYSXSRPROPERTY	Serves as an auxiliary table for the BLOB column COMPONENT in table SYSXSROBJECTCOMPONENTS.

DSN and DB2 Commands

DSN Commands

The following table shows a complete list of DSN commands. For more information, refer to *IBM DB2 11 for z/OS Command Reference*—SC19-4054-00.

DSN Command (or Subcommand)	Function
BIND	Builds an application package, plan, or query
DB2	Executes a DB2 command
DCLGEN	Produces declarations for tables or views
DSN	Starts a DSN session
END	Ends a DSN session
FREE	Deletes an application package, plan, or query
REBIND	Updates an application package or plan
REBIND TRIGGER PACKAGE	Updates an application trigger package
RUN	Executes an application program
SPUFI	Executes the SQL processing by using the file input facility

DB2 Commands

The table that follows shows all available DB2 commands and their function. All DB2 commands begin with a '-'. For more information, refer to *IBM DB2 11 for z/OS Command Reference*—SC19-4054-00.

DB2 Command	Function
-ACCESS DATABASE	Forces a physical open of a table space, index space, or partition or removes group buffer pool dependent status for a table space, index space, or partition
-ALTER BUFFERPOOL	Alters attributes for the buffer pools
-ALTER GROUPBUFFERPOOL	Alters attributes for the group buffer pools
-ALTER UTILITY	Alters parameter values of the REORG utility and the REBUILD utility that use SHRLEVEL CHANGE
-ARCHIVE LOG	Enables a site to close a current active log and open the next available log data set
-CANCEL THREAD	Cancels processing for specific local or distributed threads
-DISPLAY ACCEL	Displays information about accelerator servers
-DISPLAY ARCHIVE	Displays information about archive log processing
-DISPLAY BUFFERPOOL	Displays information about the buffer pools
-DISPLAY DATABASE	Displays status information about DB2 databases
-DISPLAY DDF	Displays information about the status and configuration of the distributed data facility, as well as statistical information regarding connections or threads controlled by DDF
-DISPLAY FUNCTION SPECIFIC	Displays statistics about external user-defined functions
-DISPLAY GROUP	Displays information about the data sharing group to which a DB2 subsystem belongs, and reports the mode in which DB2 is operating
-DISPLAY GROUPBUFFERPOOL	Displays status information about DB2 group buffer pools
-DISPLAY LOCATION	Displays status information about distributed threads
-DISPLAY LOG	Displays log information and status of the offload task
-DISPLAY PROCEDURE	Displays status information about stored procedures
-DISPLAY PROFILE	Displays whether a profile is active or inactive
-DISPLAY RLIMIT	Displays status information about the resource limit facility (governor)
-DISPLAY THREAD	Displays information about DB2 threads
-DISPLAY TRACE	Displays information about DB2 traces
-DISPLAY UTILITY	Displays status information about a DB2 utility

DB2 Command	Function
-MODIFY DDF	Modifies information regarding the status and configuration of DDF, as well as statistical information regarding connections or threads controlled by DDF
-MODIFY TRACE	Changes the IFCIDs (trace events) associated with a particular active trace
-RECOVER BSDS	Reestablishes dual BSDSs
-RECOVER INDOUBT	Recovers threads left in doubt
-RECOVER POSTPONED	Completes backout processing for units of recovery left incomplete during an earlier restart
-REFRESH DB2, EARLY	Reloads the EARLY modules and rebuilds the EARLY control block
-RESET GENERICCLU	Purges information stored by VTAM in the coupling facility
-RESET INDOUBT	Purges information displayed in the indoubt thread report generated by the -DISPLAY THREAD command
-SET ARCHIVE	Controls the allocation of tape units and the deallocation time of the tape units for archive log processing
-SET LOG	Modifies the checkpoint frequency
-SET SYSPARM	Loads the subsystem parameters specified in the command
-START ACCEL	Notifies the DB2 subsystem that it should use the indicated accelerator servers
-START DATABASE	Makes the specified database available for use
-START DB2	Initializes the DB2 subsystem (can be issued only from a z/OS console)
-START DDF	Starts the distributed data facility
-START FUNCTION SPECIFIC	Activates an external function that is stopped
-START PROCEDURE	Activates the definition of stopped or cached stored procedures
-START PROFILE	Loads or reloads the profile table into a data structure in memory
-START RLIMIT	Starts the resource limit facility (governor)
-START TRACE	Initiates DB2 trace activity 300
-STOP ACCEL	Causes the DB2 subsystem to stop using the indicated accelerator servers
-STOP DATABASE	Makes specified databases unavailable for applications
-STOP DB2	Stops the DB2 subsystem
-STOP DDF	Stops the distributed data facility
-STOP FUNCTION SPECIFIC	Stops the acceptance of SQL statements for specified functions

DB2 Command	Function
-STOP PROCEDURE	Stops the acceptance of SQL CALL statements for stored procedures
-STOP PROFILE	Stops or disables the profile function
-STOP RLIMIT	Stops the resource limit facility (governor)
-STOP TRACE	Stops trace activity
-TERM UTILITY	Terminates execution of a utility

APPENDIX D

DB2 Bind Options

Bind Options for Plans, Packages, Triggers, and Queries

The following table shows all bind options for plans, packages, triggers, and queries. It also presents the various defaults, a brief description, and allowable values.

Bind Option	Valid Values	Plan	Package	Trigger	Query
ACQUIRE	USE, ALLOCATE	X			
<i>Indicates whether to acquire resources specified in the DBRM at first access or allocation</i>					
ACTION	REPLACE, ADD	X, BO	X, BO		
	REPLACE(RPLVER)		X, BO		
	REPLACE(RETAIN)	X, BO			
<i>Specifies whether object (plan or package) replaces an existing object with the same name or is new</i>					
ACOMPARE	NO, NONE, WARN, ERROR	X	X	X	
<i>Determines whether the new access paths are different from the older access paths</i>					
APPLCOMPAT	V10R1, V11R1		X	X	
<i>Specifies the package compatibility-level behavior for static SQL</i>					
APRETAINDUP	YES, NO		RO	X	
<i>Indicates whether DB2 retains an old package copy when access paths of the old copy are identical to the incoming copy; Applies to PLANMGMT(BASIC) or PLANMGMT(EXTENDED)</i>					

Bind Option	Valid Values	Plan	Package	Trigger	Query
APREUSE	<u>NO</u> , NONE, ERROR, WARN		X	X	
<i>Specifies whether DB2 tries to reuse previous access paths for SQL statements in a package</i>					
ARCHIVESENSITIVE	<u>YES</u> , NO		X	X	
<i>Determines whether references to archive-enabled tables are affected by SYSIBMADM.GET_ARCHIVE</i>					
BUSTIMESENSITIVE	<u>YES</u> , NO		X	X	
<i>Indicates whether references to application-period temporal tables are affected by CURRENT TEMPORAL BUSINESS_TIME special register</i>					
CACHESIZE	Value of PLAN AUTH CACHE, decimal	X			
<i>Specifies the size (in bytes) of the authorization cache acquired in the EDM pool for the plan</i>					
COPY	Collection-id, package-id, COPYVER		X, BO		
<i>Determines that you are copying an existing package and names the package</i>					
CONCURRENT ACCESSRESOLUTION	WAITFOROUTCOME, @ USECURRE NTLYCOMMITTED @ Default depends on SKIPUNCL setting	X	X	X	
<i>Determines which concurrent access resolution option to use for statements in a package</i>					
CURRENTDATA	YES, <u>NO</u>	X	X	X	
<i>Specifies whether to require data currency for RO and ambiguous cursors when isolation level is CS</i>					
CURRENTSERVER	Location-name	X			
<i>Determines the location to connect to before running the plan</i>					
DBPROTOCOL	<u>DRDA</u>	X	X		
<i>Indicates the protocol to use when connecting to a remote site that is identified by a three-part name</i>					
DECSTAT	NO, YES		X	X	
<i>Determines whether DB2 builds a DESCRIBE SQL descriptor when binding static SQL statements</i>					
DEFER	DEFER(PREPARE), NODEFER(PREPARE), DEFER(INHERITFROMPLAN)	X	X		
<i>Specifies whether to defer preparation of dynamic SQL statements that refer to remote objects or to prepare them immediately; DEFER(PREPARE) is assumed for REOPT(AUTO, ALWAYS, and ONCE)</i>					
DEGREE	<u>1</u> , ANY	X	X		
<i>Indicates whether to attempt to run a query by using parallel processing to maximize performance</i>					
DEPLOY	(collection-id,package-id), COPYVER(version-id)		X		
<i>Deploys a native SQL procedure</i>					
DISCONNECT	<u>EXPLICIT</u> , AUTOMATIC, CONDITIONAL	X			
<i>Determines which remote connections to destroy during commit operations</i>					

Bind Option	Valid Values	Plan	Package	Trigger	Query
DYNAMICRULES	<u>R</u> UN, BIND, DEFINEBIND(PKG ONLY), DEFINERUN (PKG ONLY), INVOKEBIND(PKG ONLY), INVOKERUN(PKG ONLY)	X	X		
	DEFINEBIND, DEFINERUN, INVOKEBIND, INVOKERUN		X		
<i>Determines which values apply at run time for dynamic SQL attributes</i>					
ENABLE/ DISABLE	B ATCH, C ICS, D B2CALL, D LIBATCH, I MS, I MSBMP, I MSMPP, R RSAF, *, _	X	X		
	R EMOTE		X		
<i>Determines which connections can use the plan or package</i>					
ENCODING	ASCII, EBCDIC, UNICODE, ccsid	X	X		
<i>Indicates application encoding for all static statements in the plan or package (defaults to installed selection)</i>					
EXPLAIN	<u>N</u> O, YES, ONLY	X	X	X	
<i>Specifies whether to populate the PLAN_TABLE with information about the SQL statements</i>					
EXTENDED INDICATOR	<u>N</u> O, YES		X		
<i>Determines whether DB2 recognizes extended indicator variables when the associated package is run</i>					
FILTER	'filter-name'	X	X		FQ
<i>Allows you to delete a set of queries in the SYSIBM.SYSQUERY table under a {tag} value specified by the SYSQUERY.USERFILTER column; also works with FREE QUERY</i>					
FLAG	I, W, E, C	X	X	X	
<i>Determines what messages to display</i>					
GENERIC	'string'	X	X		
<i>Specifies one or more bind options that are supported by the target server, but are not supported by DB2 for z/OS as options for BIND PACKAGE or REBIND PACKAGE</i>					
IMMEDIATE	<u>N</u> O, YES, INHERITFROMPLAN	X	X		
<i>Indicates whether immediate writes will be done for updates made to GBP-dependent page sets/partitions</i>					
ISOLATION	RR, RS, CS, UR, NC	X	X	X	
<i>Determines how far to isolate an application from the effects of other running applications</i>					
KEEPDYNAMIC	<u>N</u> O, YES	X	X		
<i>Determines whether DB2 keeps dynamic SQL statements after commit points</i>					
LIBRARY	dbrm-pds-name (can be multiple for PLAN)	X, BO	X, BO		
<i>Determines which partitioned data set to search for DBRMs listed in the member option</i>					
LOOKUP	<u>N</u> O, YES				BQ
<i>Determines whether a query has matching access plan hint information in the SYSIBM.SYSQUERYPLAN table</i>					

Bind Option	Valid Values	Plan	Package	Trigger	Query
MEMBER	<i>dbrm-member-name</i>	X, BO	X, BO		
<i>Determines what DBRMs to include in the plan or package</i>					
OPTHINT	<i>Hint-id</i>	X	X		
<i>Controls whether query optimization hints are used for static SQL</i>					
OPTIONS	COMPOSITE, COMMAND		X, BC		
<i>Specifies which bind options to use for the new package</i>					
OWNER	<i>Authorization-id</i>	X	X		
<i>Determines the authorization ID or the owner of the object (plan or package)</i>					
PACKAGE	<i>Location-name.collection-id.package-id (version-id)</i>		X		
	(*) – Rebind Only		X, RO		
<i>Determines which package or packages to bind or rebind</i>					
PATH	<i>Schema-name, USER, (schema-name, (USER)...)</i>	X	X		
<i>Specifies SQL path that DB2 uses to resolve unqualified UDTs, functions, and stored procedure names</i>					
PATHDEFAULT	Mutually exclusive with PATH	X	X		
<i>Resets PATH for package or plan to "SYSIBM", "SYSFUN", "SYSPROC", or plan/package qualifier</i>					
PKLIST or NOPKLIST	<i>(Location-name.collection-id.package-id...), PKLIST only</i>	X			
<i>Determines which package to include for the package list in the plan</i>					
PLAN	<i>Plan-name</i>	X			
	(*)	X, RO			
<i>Determines which plan or plans to bind or rebind</i>					
PROGAUTH	DISABLE, ENABLE	X			
<i>Specifies whether DB2 performs authorization checking to determine whether DB2 can execute a plan</i>					
PLANMGMT	OFF, BASIC, EXTENDED		X	X	
<i>Retains, during a rebind operation, all relevant package information (metadata, query text, dependencies, authorizations, access paths, and so on) in catalog tables and in the directory</i>					
QUALIFIER	<i>Qualifier-name</i>	X	X		
<i>Determines the implicit qualifier for unqualified names of objects in the plan or package</i>					
QUERYID	'number', ALL	X	X		FQ
<i>Frees entries from SYSIBM.SYSQUERY with the same value (or ALL), and corresponding entries in SYSIBM.SYSQUERYPLAN table or SYSIBM.SYSQUERYOPTS table</i>					
RECORDTEMPORAL HISTORY	YES, NO		X		
<i>Indicates whether changes to data in a system-period temporal table that are made by static or dynamic SQL statements cause changes to the corresponding history table of the system-period temporal table</i>					

Bind Option	Valid Values	Plan	Package	Trigger	Query
RELEASE	<u>COMMIT</u> , DEALLOCATE, INHERITFROMPLAN	X	X	X	
<i>Determines when to release resources that the program uses, either at commit or at termination</i>					
REOPT	ONCE, ALWAYS, AUTO, <u>NONE</u>	X	X		
<i>Indicates whether the access path is determined at run time (host variables, parameter markers, special registers)</i>					
ROUNDING	CEILING, DOWN, FLOOR, HALFDOWN, <u>HALFEVEN</u> , HALFUP, UP	X	X		
<i>Specifies the rounding mode at bind time</i>					
SQLERROR	<u>NOPACKAGE</u> , CONTINUE, CHECK		X		
<i>Specifies whether to create a package if the package contains an SQL error</i>					
SQLRULES	<u>DB2</u> , STD	X			
<i>Determines whether a Type 2 connection can be made according to DB2 rules for an existing connection</i>					
SWITCH	PREVIOUS, ORGINAL		X	X	
<i>Restores all previous or original package information in the catalog tables and directory to that of the specified package copy</i>					
SYSTIMESENSITIVE	YES, NO		X	X	
<i>Specifies whether references to system-period temporal tables are affected by the value of CURRENT TEMPORAL SYSTEM_TIME</i>					
VALIDATE	<u>RUN</u> , BIND	X	X		
<i>Specifies whether to recheck at run time "not found" and "not authorized" errors discovered at bind time</i>					

BO = BIND only, BC = BIND COPY, RO = REBIND only, FQ=FREE QUERY, BQ = BIND QUERY, BOLD/UNDERSCORE = default

APPENDIX E

Explain Tables

The following tables show descriptions of all tables populated by EXPLAIN. For additional details, refer to *IBM DB2 11 for z/OS SQL Reference*—SC19-4066-00.

PLAN_TABLE

The plan table contains information about access paths for queries that were explained or hints.

Column name	Description
QUERYNO	A number intended to identify the statement being explained. For a row produced by an EXPLAIN statement, specify the number in the QUERYNO clause. For a row produced by non-EXPLAIN statements, specify the number by using the QUERYNO clause, which is an optional part of the SELECT, INSERT, UPDATE, MERGE, and DELETE statement syntax. Otherwise, DB2 assigns a number based on the line number of the SQL statement in the source program. FETCH statements do not each have an individual QUERYNO assigned to them. Instead, DB2 uses QUERYNO of the DECLARE CURSOR statement for all corresponding FETCH statements for that cursor. When the values of QUERYNO are based on the statement number in the source program, values greater than 32,767 are reported as 0. Hence, in a very long program, the value is not guaranteed to be unique. If QUERYNO is not unique, use the value of TIMESTAMP, which is always unique. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, and if QUERYNO clause is specified, then DB2 uses its value. Otherwise, DB2 assigns a number based on the line number of the SQL statement in the nonlinear SQL function, or native SQL procedure.

Column name	Description
QBLOCKNO	A number that identifies each query block within a query. Numbers are not in any particular order, nor are they consecutive.
APPLNAME	The name of the application plan for a row. It applies only to embedded EXPLAIN statements executed from a plan or to statements explained when binding a plan. The column is blank if not applicable. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, then this column is not used.
PROGNAME	The name of program or package containing the statement being explained. For statements explained dynamically, such as with QMF or SPUFI, the associated plan/package is listed. The column is blank if not applicable. If the SQL statement is embedded in a non-inline SQL function or native SQL procedure, the column is blank.
PLANNO	The number of steps in which the query indicated in QBLOCKNO was processed. This column indicates the order in which the steps were executed.
METHOD	The join method used for the step: @ 0 = First table accessed, continuation of previous table accessed, or not used. @ 1 = Nested loop join. For each row of the present composite table, matching rows of a new table are found and joined. @ 2 = Merge scan join. The present composite table and the new tables are scanned in the order of the join columns, and matching rows are joined. @ 3 = 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. @ 4 = Hybrid join. The current composite table is scanned in the order of the join-column rows of the new table. The new table is accessed using list prefetch.
CREATOR	The creator of the new table accessed in this step; the value is blank if METHOD is 3.
TNAME	The name of a table, MQT, created or declared temporary table, materialized view, or materialized table expression. The value is blank if METHOD is 3. It can also contain the name of a table in the form DSNWFQB(<i>qblockno</i>), which is used to represent the intermediate result of a UNION ALL, an INTERSECT ALL, an EXCEPT ALL, or an outer join that is materialized. If a view is merged, the name of the view does not appear. UDSN_DIM_TBLX(<i>qblockno</i>) is used to represent the work file of a star join dimension table.
TABNO	IBM use only.

Column name	Description
ACCESSTYPE	The method of accessing the new table: @ DI = An intersection of multiple DOCID lists to return final DOCID list @ DU = Union of multiple DOCID lists to return the final DOCID list @ DX = An XML index scan of the index named in ACCESSNAME to return a DOCID list @ E = Direct row using a row change timestamp column. @ H = Hash access @ If an overflow condition occurs, the hash overflow index is identified by ACCESSCREATOR, and ACCESSNAME is used. @ HN = Hash access using an IN predicate, or an IN predicate that DB2 generates @ If a hash overflow condition occurs, the hash overflow index is identified in ACCESSCREATOR, and ACCESSNAME is used. @ IN = Index scan when matching predicate contains an IN predicate and the IN-list is accessed through an in-memory table @ I = An index (identified in ACCESSCREATOR and ACCESSNAME) @ I1 = One-fetch index scan @ M = Multiple index scan (followed by MX, MI, MH, or MU) @ MH = Hash overflow index named in ACCESSNAME @ MX = Index scan on index named in ACCESSNAME @ When the access method MX follows the access method DX, DI, or DU, the DOCID index accesses the table by using the DOCID list returned by DX, DI, or DU. @ MI = Intersection of multiple indexes @ MU = Union of multiple indexes @ N = Index scan when the matching predicate contains the IN keyword or by an index scan when DB2 rewrites a query using the IN keyword @ NR = Range list access @ O = Work file scan, as a result of a subquery @ P = Dynamic pair-wise index scan @ R = Table space scan @ RW = Work file scan of materialized user-defined table function @ V = Buffers for an INSERT statement within a SELECT @ Blank = Not applicable to the current row
MATCHCOLS	For ACCESSTYPE I, I1, N, NR, MX, or DX, it is the number of index keys used in an index scan; otherwise, the value is 0.
ACCESSCREATOR	For ACCESSTYPE I, I1, N, NR, MX, or DX, it is the creator of the index; otherwise, the value is blank.
ACCESSNAME	For ACCESSTYPE I, I1, H, MH, N, NR, MX, or DX, it is the name of the index; for ACCESSTYPE P, DSNPJW(<i>mixopseqno</i>) is the starting pair-wise join leg in MIXOPSEQNO; otherwise, the value is blank.
INDEXONLY	Indicates whether access to an index alone is enough to carry out the step, or whether data must also be accessed. Y = Yes; N = No.
SORTN_UNIQ	Specifies whether new table is sorted to remove duplicate rows. Y=Yes; N= No.
SORTN_JOIN	Specifies whether the new table is sorted for join method 2 or 4. Y = Yes; N = No.
SORTN_ORDERBY	Specifies whether the new table is sorted for ORDER BY. Y = Yes; N = No.
SORTN_GROUPBY	Indicates whether the new table is sorted for GROUP BY. Y = Yes; N = No.
SORTC_UNIQ	Indicates whether the composite table is sorted to remove duplicate rows. Y = Yes; N = No.
SORTC_JOIN	Specifies whether the composite table is sorted for join method 1, 2, or 4. Y = Yes; N = No.
SORTC_ORDERBY	Specifies whether the composite table is sorted for an ORDER BY clause or a quantified predicate. Y = Yes; N = No.
SORTC_GROUPBY	Specifies whether the composite table is sorted for a GROUP BY clause. Y = Yes; N = No.

Column name	Description
TSLOCKMODE	Indicates the lock mode to be acquired on the new table or its table space or table space partitions. If the isolation can be determined at bind time, the values are as follows: @ 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 @ If the isolation cannot be determined at bind time, the following values show the lock mode determined by the isolation at run time: @ NS = For UR isolation, no lock; for CS, RS, or RR, an S lock @ NIS = For UR isolation, no lock; for CS, RS, or RR, an IS lock @ NSS = For UR isolation, no lock; for CS or RS, an IS lock; for RR, an S lock @ SS = For UR, CS, or RS isolation, an IS lock; for RR, an S lock @ The data in this column is right-justified. For example, IX appears as a blank followed by I followed by X. If the column contains a blank, no lock is acquired. @ If the access method in the ACCESTYPE column is DX, DI, or DU, no latches are acquired on the XML index page, and no lock is acquired on the new base table data page or row, or on the XML table and the corresponding table spaces. The value of TSLOCKMODE is blank in this case.
TIMESTAMP	Deprecated, use EXPLAIN_TIME instead.
REMARKS	Can insert any character string of 254 or fewer characters.
PREFETCH	Specifies whether data pages are to be read in advance by prefetch: @ D = Optimizer expects dynamic prefetch @ S = Pure sequential prefetch @ L = Prefetch through a page list @ U = List prefetch with an unsorted RID list @ Blank = Unknown at bind time or no prefetch
COLUMN_FN_EVAL	When an 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 after any sorts
MIXOPSEQ	The sequence number of a step in a multiple-index operation: @ 1, 2, . . . n = For the steps of the multiple-index procedure (ACCESTYPE is MX, MI, MU, DX, DI, or DU), the value is the sequence number of the OR predicate in the SQL statement (ACCESTYPE is NR). @ 0 = For any other rows.
VERSION	The version identifier for the package. It applies only to an embedded EXPLAIN statement executed from a package or to a statement that is explained when binding a package. The column is blank if not applicable. @ When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
COLLID	The collection ID: @ DSNDYNAMICSQLCACHE—The row originates from the dynamic statement cache. @ DSNEXPLAINMODEYES—The row originates from an application that specifies YES for the value of the CURRENT EXPLAIN MODE special register. @ DSNEXPLAINMODEEXPLAIN—The row originates from an application that specifies EXPLAIN for the value of the CURRENT EXPLAIN MODE special register. @ When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
ACCESS_DEGREE	The number of parallel tasks or operations activated by a query. This value is determined at bind time; the actual number of parallel operations used at execution time could differ. This column contains 0 if a host variable exists. It contains the null value if the plan or package was bound using a plan table with fewer than 43 columns. Otherwise, this column can contain null if the method that it refers to does not apply.

Column name	Description
ACCESS_PGROUP_ID	The identifier of the parallel group for accessing the new table. A parallel group is a set of consecutive operations, executed in parallel, that have the same number of parallel tasks. This value is determined at bind time; it could change at execution time. This column contains the null value if the plan or package was bound using a plan table with fewer than 43 columns. Otherwise, it can contain null if the method that it refers to does not apply.
JOIN_DEGREE	The number of parallel operations or tasks used in joining the composite table with the new table. This value is determined at bind time and can be 0 if a host variable exists. The actual number of parallel operations or tasks used at execution time could differ. This column contains the null value if the plan or package was bound using a plan table with fewer than 43 columns. Otherwise, it can contain null if the method that it refers to does not apply.
JOIN_PGROUP_ID	The identifier of the parallel group for joining the composite table with the new table. This value is determined at bind time; it could change at execution time. This column contains the null value if the plan or package was bound using a plan table with fewer than 43 columns. Otherwise, it can contain null if method that it refers to does not apply.
SORTC_PGROUP_ID	The parallel group identifier for the parallel sort of the composite table. This column contains the null value if the plan or package was bound using a plan table with fewer than 43 columns. Otherwise, it can contain null if the method that it refers to does not apply.
SORTN_PGROUP_ID	The parallel group identifier for the parallel sort of the new table. This column contains the null value if the plan or package was bound using a plan table with fewer than 43 columns. Otherwise, it can contain null if the method that it refers to does not apply.
PARALLELISM_MODE	The kind of parallelism, if any, that is used at bind time: @ I = Query I/O parallelism @ C = Query CP parallelism @ X = Sysplex query parallelism @ This column contains the null value if the plan or package was bound using a plan table with fewer than 43 columns. Otherwise, it can contain null if the method that it refers to does not apply, or if the plan or package was bound before version 10.
MERGE_JOIN_COLS	The number of columns that are joined during a merge scan join (Method = 2). Contains the null value if the plan or package was bound using a plan table with fewer than 43 columns. Otherwise, it can contain null if the method that it refers to does not apply.
CORRELATION_NAME	The correlation name of a table or view that is specified in the statement. If no correlation name exists, the column is blank. This column contains the null value if the plan or package was bound using a plan table with fewer than 43 columns. Otherwise, it can contain null if the method that it refers to does not apply.
PAGE_RANGE	Specifies whether the table qualifies for page range screening so that plans scan only the partitions that are needed. Y = Yes; blank = No.
JOIN_TYPE	The type of an outer join: @ F = Full outer join @ L = Left outer join @ P = Pair-wise join @ S = Star join @ Blank = Inner join or no join @ RIGHT OUTER JOIN converts to a LEFT OUTER JOIN when you use it, so that JOIN_TYPE contains L.
GROUP_MEMBER	The member name of the DB2 that executed EXPLAIN. The column is blank if the DB2 subsystem was not in a data sharing environment when EXPLAIN was executed.

Column name	Description
IBM_SERVICE_DATA	IBM use only.
WHEN_OPTIMIZE	When the access path was determined: @ Blank = At bind time, using a default filter factor for any host variables, parameter markers, or special registers. @ B = At bind time, using a default filter factor for any host variables, parameter markers, or special registers. However, the statement is reoptimized at run time by using input variable values for input host variables, parameter markers, or special registers. You must specify the bind option REOPT(ALWYAS), REOPT(ONCE), or REOPT(AUTO) for reoptimization to occur. @ R = At runtime, using input variables for any host variables, parameter markers, or special registers. You must specify the bind option REOPT(ALWAYS), REOPT(ONCE), or REOPT(AUTO) for this to occur.
QBLOCK_TYPE	For each query block, the type of SQL operation performed. For the outermost query, the column identifies the statement type. Possible values: @ 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 subquery @ TRUNCA = TRUNCATE @ NCOSUB = Noncorrelated subquery @ TABLEX = Table expression @ TRIGGR = WHEN clause on CREATE TRIGGER @ UNION = UNION @ UNIONA = UNION ALL @ INTERSE = INTERSECT @ INTERA = INTERSECT ALL @ EXCEPT = EXCEPT @ EXCEPTA = EXCEPT ALL
OPTHINT	A string that you use to identify this row as an optimization hint for DB2. DB2 uses this row as input when choosing an access path.
HINT_USED	APREUSE—When an access path was successfully reused because the APREUSE option was specified at bind or rebind. @ 'opthint-value'— When PLAN_TABLE access path hints are used. The <i>opthint-value</i> is the value of OPTHINT column for the hint that was used. @ SYSQUERYPLAN <i>query-id</i> —When statement-level access path hints are used. The <i>query-id</i> is the value of the QUERYID column in the SYSQUERYPLAN catalog table for the hint. @ SYSQUERYSEL <i>query-id</i> —When a predicate selectivity override is used. The <i>query-id</i> is the value of the QUERYID column of the SYSQUERYSEL catalog table row for the hint. @ EXPLAIN PACKAGE: COPY <i>copy-id</i> —When the row is the result of an EXPLAIN PACKAGE statement. The <i>copy-id</i> is one of following values: @ 0 = The current copy @ 1 = The previous copy @ 2 = The original copy
PRIMARY_ACCESSTYPE	Indicates whether direct row access will be attempted first: @ D = DB2 will try to use direct row access. If it cannot use direct row access at run time, it uses the access path described in the ACCESSTYPE column of PLAN_TABLE. @ P = DB2 used data partitioned secondary index and a part-level operation to access the data. @ T = The base table or result file is materialized into a work file, and the work file is accessed via sparse index access. If a base table is involved, ACCESSTYPE indicates how the base table is accessed. @ Blank = DB2 will not try to use direct row access. The value of the ACCESSTYPE column provides information on the method of accessing the table.
PARENT_QBLOCK	The number that indicates the QBLOCKNO of the parent query.

Column name	Description
TABLE_TYPE	The type of new 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 @ = The new table is generated from an IN-LIST predicate @ If the IN-LIST predicate is selected as the matching predicate, it will be accessed as an in-memory table. @ M = Materialized query table @ Q = Temporary intermediate result table (not materialized) @ For the name of the 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 noncorrelated) @ T = Table @ W = Work file @ The value of the column is null if the query uses GROUP BY, ORDER BY, or DISTINCT, which requires an implicit sort.
TABLE_ENCODE	The encoding scheme of the table. If the table has a single CCSID set, possible values are as follows: @ A = ASCII @ E = EBCDIC @ U = Unicode @ M = The table contains multiple CCSID sets
TABLE_SCCSID	The SBCS CCSID value of the table. If TABLE_ENCODE is M, the value is 0.
TABLE_MCCSID	The mixed CCSID value of the table. If TABLE_ENCODE is M, the value is 0. If MIXED=NO in the application defaults module, the value is -2.
TABLE_DCCSID	The DBCS CCSID value of the table. If TABLE_ENCODE is M, the value is 0. If MIXED=NO in the application defaults module, the value is -2.
ROUTINE_ID	IBM use only.
CTREF	If the referenced table is a common table expression, the value is the top-level query block number.
STMTTOKEN	A user-specified statement token.
PARENT_PLANNO	Corresponds to the plan number in the parent query block where a correlated subquery is involved. For noncorrelated subqueries, the value corresponds to the plan number in the parent query block that represents the work file for the subquery.
BIND_EXPLAIN_ONLY	Identifies whether the row was inserted by the BIND command with the EXPLAIN(ONLY) option.
SECTNOI	The section number of the statement. The value is taken from the same column in SYSPACKSTMT or SYSSTMT tables and can be used to join tables to reconstruct the access path for the statement. This column is applicable only for static statements. The default value of -1 indicates that EXPLAIN information was captured in DB2 10 for z/OS or earlier versions of DB2.
EXPLAIN_TIME	The time when the EXPLAIN information was captured: @ All cached statements—When the statement entered the cache in the form of a full-precision timestamp value @ Noncached static statements—When the statement was bound in the form of a full-precision timestamp value @ Noncached dynamic statements—When EXPLAIN was executed in the form of a value equivalent to a CHAR(16) representation of the time appended by 4 zeroes
MERGC	Indicates whether the composite table is consolidated before the join. @ Y = Yes @ N = No

Column name	Description
MERGN	Indicates whether the new table is consolidated before the join, or whether access that used a data partitioned secondary index (DPSI) involved a merge operation. @ Y = Yes @ N = No @ D = Access through a DPSI involved a merge operation @ U = Access through a DPSI that did not involve a merge operation
SCAN_DIRECTION	For index access, the direction of the index scan: @ F = Forward @ R = Reverse @ Blank = Index scan not used

DSN_COLDIST_TABLE

The column distribution table contains non-uniform column group statistics that are obtained dynamically by the DB2 optimizer.

Column name	Description
QUERYNO	A number intended to identify the statement being explained. For a row produced by an EXPLAIN statement, specify the number in the QUERYNO clause. For a row produced by non-EXPLAIN statements, specify the number by using the QUERYNO clause, which is an optional part of the SELECT, INSERT, UPDATE, MERGE, and DELETE statement syntax. Otherwise, DB2 assigns a number based on the line number of the SQL statement in the source program. @ FETCH statements do not each have an individual QUERYNO assigned to them. Instead, DB2 uses the QUERYNO of the DECLARE CURSOR statement for all corresponding FETCH statements for that cursor. @ When the values of QUERYNO are based on the statement number in the source program, values greater than 32,767 are reported as 0. Hence, in a very long program, the value is not guaranteed to be unique. If QUERYNO is not unique, use the value of TIMESTAMP, which is always unique. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, and if the QUERYNO clause is specified, then DB2 uses its value. Otherwise, DB2 assigns a number based on the line number of the SQL statement in the nonline SQL function, or native SQL procedure.
APPLNAME	The application plan name. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
PROGNAME	The program name (binding an application) or the package name (binding a package). It applies only to embedded EXPLAIN statements and to statements explained as the result of binding a plan or package. A blank indicates that the column is not applicable. @ When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
COLLID	The collection ID: @ DSNDYNAMICSQLCACHE—The row originates from the dynamic statement cache. @ DSNEXPLAINMODEYES—The row originates from an application that specifies YES for the value of the CURRENT EXPLAIN MODE special register. @ DSNEXPLAINMODEEXPLAIN—The row originates from an application that specifies EXPLAIN for the value of the CURRENT EXPLAIN MODE special register. @ When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.

Column name	Description
GROUP_MEMBER	The member name of the DB2 subsystem that executed EXPLAIN. The column is blank if the DB2 subsystem was not in a data sharing environment when EXPLAIN was executed.
EXPLAIN_TIME	The EXPLAIN timestamp.
SCHEMA	The schema of the table that contains the column.
TBNAME	The name of the table that contains the column.
NAME	The name of column. If the value of NUMCOLUMNS is greater than 1, it identifies the first column name of the set of columns associated with the statistics.
COLVALUE	Contains data of a frequently occurring value in the column. Statistics are not collected for an index on a ROWID column. If the value has a noncharacter data type, data might not be printable.
TYPE	The type of statistics: @ C = Cardinality @ F = Frequent value @ H = Histogram
CARDF	For TYPE='C', the number of distinct values for the column group. For TYPE='H', the number of distinct values for the column group in a quantile indicated by the value of the QUANTILENO column.
COLGROUPCOLNO	The identity of the set of columns associated with the statistics. If the statistics are associated only with a single column, the field contains a zero length. Otherwise, the field is an array of SMALLINT column number with a dimension equal to the value in the NUMCOLUMNS column. This is an updateable column.
NUMCOLUMNS	Identifies the number of columns associated with the statistics.
FREQUENCYF	The percentage of rows in the table with the value that is specified in the COLVALUE column when the number is multiplied by 100. For example, a value of '1' indicates 100 percent. A value of '.153' indicates 15.3 percent.
QUANTILENO	The ordinary sequence number of a quantile in the whole consecutive value range, from low to high. This column is not updateable.
LOWVALUE	For TYPE='H', this is the lower bound for the quantile indicated by the value of the QUANTILENO column. It is not used if the value of the TYPE column is not 'H'. This column is not updateable.
HIGHVALUE	For TYPE='H', this is the higher bound for the quantile indicated by the value of the QUANTILENO column. This column is not used if the value of the TYPE column is not 'H'. This column is not updateable.
EXPANSION_REASON	Applies only to statements that reference archive tables or temporal tables. For other statements, this column is blank.

DSN_DETCOST_TABLE

The detailed cost table contains information about detailed cost estimation of the mini-plans in a query.

Column name	Description
QUERYNO	A number intended to identify the statement being explained. For a row produced by an EXPLAIN statement, specify the number in the QUERYNO clause. For a row produced by non-EXPLAIN statements, specify the number by using the QUERYNO clause, which is an optional part of the SELECT, INSERT, UPDATE, MERGE, and DELETE statement syntax. Otherwise, DB2 assigns a number based on the line number of the SQL statement in the source program. When the values of QUERYNO are based on the statement number in the source program, values greater than 32,767 are reported as 0. Hence, in a very long program, the value is not guaranteed to be unique. If QUERYNO is not unique, use the value of TIMESTAMP, which is always unique. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, and if the QUERYNO clause is specified, then DB2 uses its value. Otherwise, DB2 assigns a number based on the line number of the SQL statement in the nonline SQL function, or native SQL procedure.
QBLOCKNO	A number that identifies each query block within a query.
PLANNO	A number that identifies each mini-plan within a query block.
APPLNAME	The application plan name. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
PROGNAME	The program name (binding an application) or the package name (binding a package). This applies only to embedded EXPLAIN statements and to statements explained as the result of binding a plan or package. A blank indicates that the column is not applicable. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
OPENIO	The Do-at-open I/O cost for the noncorrelated subquery.
OPENCPU	The Do-at-open CPU cost for the noncorrelated subquery.
OPENCOST	The Do-at-open total cost for the noncorrelated subquery.
ONECOMPROWS	The number of rows qualified after applying local predicates.
IMFF	The filter factor of matching predicates only.
IMFFADJ	The filter factor of matching and screening predicates.
DMCOLS	The number of data manager columns.
DMROWS	The number of data manager rows returned (after all stage 1 predicates are applied).
RDSROW	The number of RDS rows returned (after all stage 1 and stage 2 predicates are applied).

Column name	Description
SNCOLS	The number of columns as sort input for a new table.
SNROWS	The number of rows as sort input for a new table.
SNRUNS	The number of runs generated for a sort of a new table.
SNMERGES	The number of merges needed during a sort.
SNCCOLS	The number of columns as sort input for a composite table.
SCROWS	The number of rows as sort input for a composite table.
SCRECSZ	The record size for a composite table.
SCPAGES	The page size for a composite table.
SCRUNS	The number of runs generated during the sort of a composite table.
SCMERGES	The number of merges needed during a sort of a composite table.
COMPCARD	The total composite cardinality.
COMPCOST	The total cost.
EXPLAIN_TIME	The EXPLAIN timestamp.
GROUP_MEMBER	The member name of the DB2 subsystem that executed EXPLAIN. The column is blank if the DB2 subsystem was not in a data sharing environment when EXPLAIN was executed.
UNCERTAINTY	Describes the uncertainty factor of inner table index access. It is aggregated from uncertainty of inner table probing predicates. A larger value indicates a higher uncertainty. 0 indicates no uncertainty or uncertainty not considered.
UNCERTAINTY_1T	Describes the uncertainty factor of the ONECOMPROWS column of the table. It is aggregated from all local predicates on the table. A larger value indicates a higher uncertainty. 0 indicates no uncertainty or uncertainty not considered.
SECTNOI	The section number of the statement. The value is taken from the same column in SYSPACKSTMT or SYSSTMT tables and can be used to join tables to reconstruct the access path for the statement. This column is applicable only for static statements. The default value of -1 indicates that EXPLAIN information was captured in DB2 10 for z/OS or earlier versions of DB2.
VERSION	The version identifier for the package. Applies only to an embedded EXPLAIN statement executed from a package or to a statement that is explained when binding a package. A blank indicates that the column is not applicable. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
IXSCAN_SKIP_DUPS	Specifies whether duplicate index key values are skipped during index scan. @ 'Y' = Duplicate key values are skipped. @ 'N' = Duplicate key values are not skipped.
IXSCAN_SKIP_SCREEN	Specifies whether key ranges that are disqualified by index screening predicates are skipped during an index scan. @ 'Y' = Disqualified key ranges are skipped. @ 'N' = Key ranges are not skipped.

Column name	Description
EARLY_OUT	Specifies whether fetching from the table stops after the first qualified row. @ 'Y' = Internal fetching stops after the first qualified row. @ 'N' = Internal fetching continues after the first qualified row. @ Blank = The EXPLAIN information was captured in a previous release, or the EXPLAIN information was captured for a package that was bound in a previous release.
EXPANSION_REASON	Applies only to statements that reference archive tables or temporal tables. For other statements, this column is blank.

DSN_FILTER_TABLE

The filter table contains information about how predicates are used during query processing.

Column name	Description
QUERYNO	A number intended to identify the statement being explained. For a row produced by an EXPLAIN statement, specify the number in the QUERYNO clause. For a row produced by non-EXPLAIN statements, specify the number by using the QUERYNO clause, which is an optional part of the SELECT, INSERT, UPDATE, MERGE, and DELETE statement syntax. Otherwise, DB2 assigns a number based on the line number of the SQL statement in the source program. @ FETCH statements do not each have an individual QUERYNO assigned to them. Instead, DB2 uses the QUERYNO of the DECLARE CURSOR statement for all corresponding FETCH statements for that cursor. @ When the values of QUERYNO are based on the statement number in the source program, values greater than 32,767 are reported as 0. Hence, in a very long program, the value is not guaranteed to be unique. If QUERYNO is not unique, use the value of TIMESTAMP, which is always unique. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, and if the QUERYNO clause is specified, then DB2 uses its value. Otherwise, DB2 assigns a number based on the line number of the SQL statement in the nonlinear SQL function, or native SQL procedure.
QBLOCKNO	A number that identifies each query block within a query.
PLANNO	A number that identifies each mini-plan within a query block.
APPLNAME	The application plan name. @ When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
PROGNAME	The program name (binding an application) or the package name (binding a package). This applies only to embedded EXPLAIN statements and to statements explained as the result of binding a plan or package. A blank indicates that the column is not applicable. @ When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.

Column name	Description
COLLID	The collection ID: @ DSNDYNAMICSQLCACHE—The row originates from the dynamic statement cache. @ DSNEXPLAINMODEYES—The row originates from an application that specifies YES for the value of the CURRENT EXPLAIN MODE special register. @ DSNEXPLAINMODEEXPLAIN—The row originates from an application that specifies EXPLAIN for the value of the CURRENT EXPLAIN MODE special register. @ When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
ORDERNO	The sequence number of evaluation. It indicates the order in which the predicate is applied within each stage.
PREDNO	A number used to identify a predicate within a query.
STAGE	Indicates at which stage the predicate is evaluated. The possible values are as follows: @ Matching @ Screening @ Pagerange @ Stage 1 @ Stage 2
EXPLAIN_TIME	The EXPLAIN timestamp.
GROUP_MEMBER	The member name of the DB2 subsystem that executed EXPLAIN. The column is blank if the DB2 subsystem was not in a data sharing environment when EXPLAIN was executed.
SECTNOI	The section number of the statement. The value is taken from the same column in SYSPACKSTMT or SYSSTMT tables and can be used to join tables to reconstruct the access path for the statement. This column is applicable only for static statements. The default value of -1 indicates that EXPLAIN information was captured in DB2 10 for z/OS or earlier versions of DB2.
COLLID	The collection ID: @ DSNDYNAMICSQLCACHE—The row originates from the dynamic statement cache. @ DSNEXPLAINMODEYES—The row originates from an application that specifies YES for the value of the CURRENT EXPLAIN MODE special register. @ DSNEXPLAINMODEEXPLAIN—The row originates from an application that specifies EXPLAIN for the value of the CURRENT EXPLAIN MODE special register. @ When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
VERSION	The version identifier for the package. This applies only to an embedded EXPLAIN statement executed from a package or to a statement that is explained when binding a package. A blank indicates that the column is not applicable. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
PUSHDOWN	Specifies whether the predicate is pushed down into the index manager or data manager subcomponents for evaluation: @ I = The index manager subcomponent evaluates the predicate. @ D = The data manager subcomponent evaluates the predicate. @ Blank = The predicate is not pushed down for evaluation.
EXPANSION_REASON	Applies only to statements that reference archive tables or temporal tables. For other statements, this column is blank.

DSN_FUNCTION_TABLE

The function table contains information about the cost of user-defined functions used in an SQL statement.

Column name	Description
QUERYNO	A number intended to identify the statement being explained. For a row produced by an EXPLAIN statement, specify the number in the QUERYNO clause. For a row produced by non-EXPLAIN statements, specify the number by using the QUERYNO clause, which is an optional part of the SELECT, INSERT, UPDATE, MERGE, and DELETE statement syntax. Otherwise, DB2 assigns a number based on the line number of the SQL statement in the source program. When the values of QUERYNO are based on the statement number in the source program, values greater than 32,767 are reported as 0. Hence, in a very long program, the value is not guaranteed to be unique. If QUERYNO is not unique, use the value of TIMESTAMP, which is always unique. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, and if the QUERYNO clause is specified, then DB2 uses its value. Otherwise, DB2 assigns a number based on the line number of the SQL statement in the nonline SQL function, or native SQL procedure.
APPLNAME	The name of the application plan for the row. This applies only to embedded EXPLAIN statements executed from a plan or to statements explained when binding a plan. This column is blank if not applicable. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, then this column is not used.
PROGNAME	The name of the program or package containing the statement being explained. For statements explained dynamically, such as with QMF or SPUFI, the associated plan/package is listed. This column is blank if not applicable. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
COLLID	The collection ID: When the row originates from the dynamic statement cache. When the row originates from an application that specifies YES for the value of the CURRENT EXPLAIN MODE special register. When the row originates from an application that specifies EXPLAIN for the value of the CURRENT EXPLAIN MODE special register. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
GROUP_MEMBER	The member name of the DB2 that executed EXPLAIN, or is blank.

Column name	Description
EXPLAIN_TIME	The time at which the statement is processed. This time is the same as the BIND_TIME column in PLAN_TABLE.
SCHEMA_NAME	The schema name of the function invoked in the explained statement.
FUNCTION_NAME	The name of the function invoked in the explained statement.
SPEC_FUNC_ID	The specific name of the function invoked in the explained statement.
FUNCTION_TYPE	The type of function invoked in the explained statement: @ SU = Scalar function @ TU = Table function
VIEW_CREATOR	The creator of the view if the function specified in the FUNCTION_NAME column is referenced in a view definition. Otherwise, this column is blank.
VIEW_NAME	The name of the view if the function specified in the FUNCTION_NAME column is referenced in a view definition. Otherwise, this column is blank.
PATH	The value of the SQL path that was used to resolve the schema name of the function.
FUNCTION_TEXT	The text of the function reference (the function name and parameters). If the function reference is more than 1,500 bytes, this column contains the first 1,500 bytes. For functions specified in fixed notation, FUNCTION_TEXT contains only the function name. For example, for a function named /, which overloads the SQL divide operator, if the function reference is A/B, FUNCTION_TEXT contains only /, not A/B.
FUNC_VERSION	For a version of a non-inline SQL scalar function, this column contains the version identifier. For all other cases, this column contains a zero length string.
SECURE	Indicates whether the user-defined function is secure.
SECTNOI	The section number of the statement. The value is taken from the same column in SYSPACKSTMT or SYSSTMT tables and can be used to join tables to reconstruct the access path for the statement. This column is applicable only for static statements. The default value of -1 indicates that EXPLAIN information was captured in DB2 10 for z/OS or earlier versions of DB2.
VERSION	The version identifier for the package. This applies only to an embedded EXPLAIN statement executed from a package or to a statement that is explained when binding a package. A blank indicates that the column is not applicable. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
EXPANSION_REASON	Applies only to statements that reference archive tables or temporal tables. For other statements, this column is blank.

DSN_KEYTGTDIST_TABLE

The key-target distribution table contains non-uniform index expression statistics obtained dynamically by the DB2 optimizer.

Column name	Description
QUERYNO	A number intended to identify the statement being explained. For a row produced by an EXPLAIN statement, specify the number in the QUERYNO clause. For a row produced by non-EXPLAIN statements, specify the number by using the QUERYNO clause, which is an optional part of the SELECT, INSERT, UPDATE, MERGE, and DELETE statement syntax. Otherwise, DB2 assigns a number based on the line number of the SQL statement in the source program. When the values of QUERYNO are based on the statement number in the source program, values greater than 32,767 are reported as 0. Hence, in a very long program, the value is not guaranteed to be unique. If QUERYNO is not unique, use the value of TIMESTAMP, which is always unique. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, and if the QUERYNO clause is specified, then DB2 uses its value. Otherwise, DB2 assigns a number based on the line number of the SQL statement in the nonline SQL function, or native SQL procedure.
APPLNAME	The application plan name. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
PROGNAME	The program name (binding an application) or the package name (binding a package). This applies only to embedded EXPLAIN statements and to statements explained as the result of binding a plan or package. A blank indicates that the column is not applicable. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
COLLID	The collection ID: DSN DYNAMIC SQL CACHE—The row originates from the dynamic statement cache. DSN EXPLAIN MODE YES—The row originates from an application that specifies YES for the value of the CURRENT EXPLAIN MODE special register. DSN EXPLAIN MODE EXPLAIN—The row originates from an application that specifies EXPLAIN for the value of the CURRENT EXPLAIN MODE special register. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
GROUP_MEMBER	The member name of the DB2 subsystem that executed EXPLAIN. The column is blank if the DB2 subsystem was not in a data sharing environment when EXPLAIN was executed.

Column name	Description
EXPLAIN_TIME	The EXPLAIN timestamp.
IXSCHEMA	The qualifier of the index.
IXNAME	The name of the index.
KEYSEQ	The numeric position of the key-target in the index.
KEYVALUE	Contains the data of a frequently occurring value. Statistics are not collected for an index on a ROWID column. If the value has a noncharacter data type, the data might not be printable.
TYPE	The type of statistics: @ C = Cardinality @ F = Frequent value @ H = Histogram
CARDF	For TYPE='C', the number of distinct values for the column group. For TYPE='H', the number of distinct values for the column group in a quantile indicated by the value of the QUANTILENO column.
KEYGROUPKEYNO	Contains a value that identifies the set of keys that are associated with the statistics. If the statistics are associated with more than one key, the column contains an array of SMALLINT key numbers with a dimension that is equal to the value in NUMKEYS. If the statistics are only associated with a single key, it contains 0.
NUMKEYS	The number of keys that are associated with the statistics.
FREQUENCYF	The percentage of rows in the table with the value that is specified in the KEYVALUE column when the number is multiplied by 100. For example, a value of '1' indicates 100 percent. A value of '.153' indicates 15.3 percent.
QUANTILENO	The ordinary sequence number of a quantile in the whole consecutive value range, from low to high. This column is not updateable.
LOWVALUE	For TYPE='H', this is the lower bound for the quantile indicated by the value of the QUANTILENO column. It is not used if the value of the TYPE column is not 'H'. This column is not updateable.
HIGHVALUE	For TYPE='H', this is the higher bound for the quantile indicated by the value of the QUANTILENO column. This column is not used if the value of the TYPE column is not 'H'. This column is not updateable.
EXPANSION_REASON	Applies only to statements that reference archive tables or temporal tables. For other statements, this column is blank.

DSN_PGRANGE_TABLE

The page range table contains information about qualified partitions for all page range scans in a query.

Column name	Description
QUERYNO	A number intended to identify the statement being explained. For a row produced by an EXPLAIN statement, specify the number in the QUERYNO clause. For a row produced by non-EXPLAIN statements, specify the number by using the QUERYNO clause, which is an optional part of the SELECT, INSERT, UPDATE, MERGE, and DELETE statement syntax. Otherwise, DB2 assigns a number based on the line number of the SQL statement in the source program. When the values of QUERYNO are based on the statement number in the source program, values greater than 32,767 are reported as 0. Hence, in a very long program, the value is not guaranteed to be unique. If QUERYNO is not unique, use the value of TIMESTAMP, which is always unique. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, and if the QUERYNO clause is specified, then DB2 uses its value. Otherwise, DB2 assigns a number based on the line number of the SQL statement in the nonline SQL function, or native SQL procedure.
QBLOCKNO	A number that identifies each query block within a query.
TABNO	The table number.
RANGE	The sequence number of the current page range.
FIRSTPART	The starting partition in the current page range.
LASTPART	The ending partition in the current page range.
NUMPARTS	The number of partitions in the current page range.
EXPLAIN_TIME	The EXPLAIN timestamp.
GROUP_MEMBER	The member name of the DB2 subsystem that executed EXPLAIN. The column is blank if the DB2 subsystem was not in a data sharing environment when EXPLAIN was executed.
SECTNOI	The section number of the statement. The value is taken from the same column in SYSPACKSTMT or SYSSTMT tables and can be used to join tables to reconstruct the access path for the statement. This column is applicable only for static statements. The default value of -1 indicates that EXPLAIN information was captured in DB2 10 for z/OS or earlier.
APPLNAME	The application plan name. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
PROGNAME	The program name (binding an application) or the package name (binding a package). This applies only to embedded EXPLAIN statements and to statements explained as the result of binding a plan or package. A blank indicates that the column is not applicable. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.

Column name	Description
COLLID	The collection ID: @ DSNDYNAMICSQLCACHE—The row originates from the dynamic statement cache. @ DSNEXPLAINMODEYES—The row originates from an application that specifies YES for the value of the CURRENT EXPLAIN MODE special register. @ DSNEXPLAINMODEEXPLAIN—The row originates from an application that specifies EXPLAIN for the value of the CURRENT EXPLAIN MODE special register. @ When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
VERSION	The version identifier for the package. Applies only to an embedded EXPLAIN statement executed from a package or to a statement that is explained when binding a package. A blank indicates that the column is not applicable. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
EXPANSION_REASON	Applies only to statements that reference archive tables or temporal tables. For other statements, this column is blank.

DSN_PGROUP_TABLE

The parallel group table contains information about the parallel groups in a query.

Column name	Description
QUERYNO	A number intended to identify the statement being explained. For a row produced by an EXPLAIN statement, specify the number in the QUERYNO clause. For a row produced by non-EXPLAIN statements, specify the number by using the QUERYNO clause, which is an optional part of the SELECT, INSERT, UPDATE, MERGE, and DELETE statement syntax. Otherwise, DB2 assigns a number based on the line number of the SQL statement in the source program. @ FETCH statements do not each have an individual QUERYNO assigned to them. Instead, DB2 uses the QUERYNO of the DECLARE CURSOR statement for all corresponding FETCH statements for that cursor. @ When the values of QUERYNO are based on the statement number in the source program, values greater than 32,767 are reported as 0. Hence, in a very long program, the value is not guaranteed to be unique. If QUERYNO is not unique, use the value of TIMESTAMP, which is always unique. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, and if the QUERYNO clause is specified, then DB2 uses it value. Otherwise, DB2 assigns a number based on the line number of the SQL statement in the nonline SQL function, or native SQL procedure.
QBLOCKNO	A number that identifies each query block within a query.
PLANNAME	The application plan name.
COLLID	The collection ID: @ DSNDYNAMICSQLCACHE—The row originates from the dynamic statement cache. @ DSNEXPLAINMODEYES—The row originates from an application that specifies YES for the value of the CURRENT EXPLAIN MODE special register. @ DSNEXPLAINMODEEXPLAIN—The row originates from an application that specifies EXPLAIN for the value of the CURRENT EXPLAIN MODE special register. @ When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.

Column name	Description
PROGNAME	The program name (binding an application) or the package name (binding a package). This applies only to embedded EXPLAIN statements and to statements explained as the result of binding a plan or package. A blank indicates that the column is not applicable. @ When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
EXPLAIN_TIME	The EXPLAIN timestamp.
VERSION	The version identifier for the package. This applies only to an embedded EXPLAIN statement executed from a package or to a statement that is explained when binding a package. A blank indicates that the column is not applicable. @ When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
GROUPID	The parallel group identifier within the current query block.
FIRSTPLAN	The plan number of the first contributing mini-plan associated within this parallel group.
LASTPLAN	The plan number of the last mini-plan associated with this parallel group.
CPU COST	The estimated CPU cost of this parallel group in milliseconds.
I/O COST	The estimated total I/O cost of this parallel group in milliseconds.
BESTTIME	The estimated elapsed time for each parallel task for this parallel group.
DEGREE	The degree of parallelism for this parallel group determined at bind time. The maximum parallelism degree if the table space is large is 255; otherwise, it is 64.
MODE	The parallel mode: @ I = I/O parallelism @ C = CPU parallelism @ X = Multiple CPU Sysplex parallelism (highest level) @ N = No parallelism
REASON	The reason for downgrading parallelism mode.
LOCALCPU	The number of CPUs currently online when preparing the query.
TOTALCPU	The total number of CPUs in Sysplex. LOCALCPU and TOTALCPU are different only for the DB2 coordinator in a sysplex.
FIRSTBASE	The table number of the table that on which partitioning is performed.
LARGETS	Value is Y if the table space is large in this group.
PARTKIND	The partitioning type: @ L = Logical partitioning @ P = Physical partitioning
GROUPTYPE	Indicates what operations this parallel group contains: table access, join, or sort (A, AJ, or AJS).
ORDER	The ordering requirement of this parallel group: @ N = No order; results need no ordering @ T = Natural order; ordering required but results already ordered if accessed via index @ K = Key order; ordering achieved by sort—results ordered by sort key @ This value applies only to parallel sort.
STYLE	The input/output format style of this parallel group. The column is blank for I/O parallelism. For other modes: @ RIRO = Records IN, Records OUT @ WIRO = Work file IN, Records OUT @ WIWO = Work file IN, Work file OUT
RANGEKIND	The range type: @ K = Key range @ L = IN-list elements partitioning @ P = Page range @ R = Record range partitioning

Column name	Description
NKEYCOLS	The number of interesting key columns—that is, the number of columns that will participate in the key operation for this parallel group.
LOWBOUND	The low bound of the parallel group.
HIGHBOUND	The high bound of the parallel group.
LOWKEY	The low key of range if partitioned by key range.
HIGHKEY	The high key of range if partitioned by key range.
FIRSTPAGE	The first page in range if partitioned by page range.
LASTPAGE	The last page in range if partitioned by page range.
APPLNAME	The application plan name. @ When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
SECTNOI	The section number of the statement. The value is taken from the same column in SYSPACKSTMT or SYSSTMT tables and can be used to join tables to reconstruct the access path for the statement. This column is applicable only for static statements. The default value of -1 indicates that EXPLAIN information was captured in DB2 10 for z/OS or earlier versions of DB2.
EXPANSION_REASON	Applies to only statements that reference archive tables or temporal tables. For other statements, this column is blank.

DSN_PREDICAT_TABLE

The predicate table contains information about all the predicates in a query.

Column name	Description
QUERYNO	A number intended to identify the statement being explained. For a row produced by an EXPLAIN statement, specify the number in the QUERYNO clause. For a row produced by non-EXPLAIN statements, specify the number by using the QUERYNO clause, which is an optional part of the SELECT, INSERT, UPDATE, MERGE, and DELETE statement syntax. Otherwise, DB2 assigns a number based on the line number of the SQL statement in the source program. @ FETCH statements do not each have an individual QUERYNO assigned to them. Instead, DB2 uses the QUERYNO of the DECLARE CURSOR statement for all corresponding FETCH statements for that cursor. @ When the values of QUERYNO are based on the statement number in the source program, values greater than 32,767 are reported as 0. Hence, in a very long program, the value is not guaranteed to be unique. If QUERYNO is not unique, use the value of TIMESTAMP, which is always unique. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, and if the QUERYNO clause is specified, then DB2 uses its value. Otherwise, DB2 assigns a number based on the line number of the SQL statement in the nonline SQL function, or native SQL procedure.
QBLOCKNO	A number that identifies each query block within a query.

Column name	Description
APPLNAME	The application plan name. @ When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
PROGNAME	The program name (binding an application) or the package name (binding a package). This applies only to embedded EXPLAIN statements and to statements explained as the result of binding a plan or package. A blank indicates that the column is not applicable. @ When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
PREDNO	A number that identifies a predicate within a query.
TYPE	A string that indicates the type or the operation of the predicate. The possible values are as follows: @ AND @ OR @ EQUAL @ RANGE @ BETWEEN @ IN @ LIKE @ NOT LIKE @ EXISTS @ COMPOUND @ NOTEXIST @ SUBQUERY @ HAVING @ OTHERS
LEFT_HAND_SIDE	Indicates the column name if the left-hand side (LHS) of the predicate is a table column (LHS_TABNO > 0). Other possible values are as follows: @ VALUE @ COLEXP @ NONCOLEXP @ CORSUB @ NONCORSUB @ SUBQUERY @ EXPRESSION @ Blanks
LEFT_HAND_PNO	Indicates the column name if the LHS of the predicate is a table column (LHS_TABNO > 0). Other possible values are as follows: @ VALUE @ COLEXP @ NONCOLEXP @ CORSUB @ NONCORSUB @ SUBQUERY @ EXPRESSION @ Blanks
LHS_TABNO	Indicates a number that uniquely identifies the corresponding table reference within a query if the LHS of the predicate is a table column.
LHS_QBNO	Indicates a number that uniquely identifies the corresponding query block number within a query if the LHS of the predicate is a table column.
RIGHT_HAND_SIDE	Indicates the column name if the right-hand side (RHS) of the predicate is a table column (RHS_TABNO > 0). Other possible values are as follows: @ VALUE @ COLEXP @ NONCOLEXP @ CORSUB @ NONCORSUB @ SUBQUERY @ EXPRESSION @ Blanks
RIGHT_HAND_PNO	Indicates the second child predicate if the predicate is a compound predicate (AND/OR). However, this column is not reliable when predicate tree consolidation happens.
RHS_TABNO	Indicates a number that uniquely identifies the corresponding table reference within a query if the RHS of the predicate is a table column.
RHS_QBNO	Indicates a number that uniquely identifies the corresponding query block within a query if the RHS of the predicate is a subquery.
FIULTER_FACTOR	The estimated filter factor.
BOOLEAN_TERM	Specifies whether this predicate can be used to determine the truth value of the whole WHERE clause.
SEARCHARG	Specifies whether this predicate can be processed by data manager (DM) stage 1. If it cannot, use the relational data service (RDS) stage 2 to handle it, which is more costly.

Column name	Description
AFTER_JOIN	Indicates the predicate evaluation phase: @ A = After join @ D = During join @ Blank = Not applicable
ADDED_PRED	Indicates whether the predicate is generated by DB2, and why the predicate is added: @ Blank = DB2 did not add predicate @ 'B' = Bubble up @ 'C' = Correlation @ 'J' = Join @ 'K' = LIKE for expression-based index @ 'L' = Localization @ 'P' = Push down @ 'R' = Page range @ 'S' = Simplification @ 'T' = Transitive closure
REDUNDANT_PRED	Specifies whether the predicate is a redundant predicate, which means evaluation of other predicates in the query already determines the result that the predicate provides.
DIRECT_ACCESS	Specifies whether the predicate is direct access, which means you can navigate directly to the row through ROWID.
KEYFIELD	Indicates whether the predicate includes the index key column of the involved table.
EXPLAIN_TIME	The EXPLAIN timestamp.
MARKER	Specifies whether the predicate includes host variables, parameter markers, or special registers.
PARENT_PNO	The parent predicate number. If this predicate is a root predicate within a query block, this column is 0.
NEGATION	Indicates whether the predicate is negated via NOT.
LITERALS	The literal value or literal values separated by colon symbols.
CLAUSE	The clause where the predicate exists: @ HAVING = HAVING clause @ ON = ON clause @ WHERE = WHERE clause @ SELECT = The SELECT clause
GROUP_MEMBER	The member name of the DB2 subsystem that executed EXPLAIN. The column is blank if the DB2 subsystem was not in a data sharing environment when EXPLAIN was executed.
ORIGIN	Indicates the origin of the predicate: @ Blank = Generated by DB2 @ C = Column mask @ R = Row permission @ U = User-specified
UNCERTAINTY	Describes the uncertainty factor of a predicate's estimated filter factor. A bigger value indicates a higher degree of uncertainty. A zero value indicates no uncertainty or uncertainty not considered.
SECTNOI	The section number of the statement. The value is taken from the same column in SYSPACKSTMT or SYSSTMT tables and can be used to join tables to reconstruct the access path for the statement. This column is applicable only for static statements. The default value of -1 indicates that EXPLAIN information was captured in DB2 10 for z/OS or earlier versions of DB2.
COLLID	The collection ID: @ DSNDYNAMICSQLCACHE—The row originates from the dynamic statement cache. @ DSNEXPLAINMODEYES—The row originates from an application that specifies YES for the value of the CURRENT EXPLAIN MODE special register. @ DSNEXPLAINMODEEXPLAIN—The row originates from an application that specifies EXPLAIN for the value of the CURRENT EXPLAIN MODE special register. @ When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.

Column name	Description
VERSION	The version identifier for the package. This applies only to an embedded EXPLAIN statement executed from a package or to a statement that is explained when binding a package. A blank indicates that the column is not applicable. @ When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
EXPANSION_REASON	Applies only to statements that reference archive tables or temporal tables. For other statements, this column is blank.

DSN_PREDICATE_SELECTIVITY

The predicate selectivity table contains information about the selectivity of predicates that are used for access path selection. It is used as an input table for the BIND QUERY command when you specify selectivity overrides.

Column name	Description
QUERYNO	A number that identifies the statement that is being explained. The origin of the value depends on the context of the row. @ For rows produced by EXPLAIN statements, specify the number in the QUERYNO clause, which is an optional part of the SELECT, INSERT, UPDATE, MERGE, and DELETE statement syntax. @ For rows not produced by EXPLAIN statements, DB2 assigns a number that is based on the line number of the SQL statement in the source program. @ When the values of QUERYNO are based on the statement number in the source program, values that exceed 32,767 are reported as 0. However, in certain rare cases, the value is not guaranteed to be unique. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, and if the QUERYNO clause is specified, then DB2 uses its value. Otherwise, DB2 assigns a number based on the line number of the SQL statement in the non-inline SQL function, or native SQL procedure.
QBLOCKNO	A number that identifies each query block within a query. The values of the numbers are not in any particular order, nor are they necessarily consecutive.
APPLNAME	The name of the application plan for the row. This applies only to embedded EXPLAIN statements that are executed from a plan or to statements that are explained when binding a plan. A blank indicates that the column is not applicable. @ When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
PROGNAME	The name of the program or package containing the statement being explained. This applies only to embedded EXPLAIN statements and to statements explained as the result of binding a plan or package. A blank indicates that the column is not applicable. @ When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
SECTNOI	The section number of the statement. The value is taken from the same column in SYSPACKSTMT or SYSSTMT tables and can be used to join tables to reconstruct the access path for the statement. This column is applicable only for static statements. The default value of -1 indicates that EXPLAIN information was captured in Version 9 or earlier.

Column name	Description
COLLID	The collection ID: @ DSNDYNAMICSQLCACHE—The row originates from the dynamic statement cache. @ DSNEXPLAINMODEYES—The row originates from an application that specifies YES for the value of the CURRENT EXPLAIN MODE special register. @ DSNEXPLAINMODEEXPLAIN—The row originates from an application that specifies EXPLAIN for the value of the CURRENT EXPLAIN MODE special register. @ When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
VERSION	The version identifier for the package. This applies only to an embedded EXPLAIN statement executed from a package or to a statement that is explained when binding a package. A blank indicates that the column is not applicable. @ When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
PREDNO	The predicate number that identifies a specific predicate within a query.
INSTANCE	The selectivity instance. Used to group related selectivities.
SELECTIVITY	The selectivity estimate.
WEIGHT	The percentage of executions that have the specified selectivity. For example, a value of 0.25 means that 25 percent of the time when the query is executed, it has this selectivity.
ASSUMPTION	NULL indicates how the selectivity was estimated or is used when you specify one of the following values: @ 'NORMAL'—Selectivity is estimated by using the normal selectivity assumptions. @ 'OVERRIDE'—Selectivity is based on an override.
INSERT_TIME	The time when the row was inserted or updated.
EXPLAIN_TIME	The time when the EXPLAIN information was captured: @ All cached statements—When the statement entered the cache, in the form of a full-precision timestamp value @ Noncached static statements—When statement was bound, in the form of a full precision timestamp value @ Noncached dynamic statements—When EXPLAIN was executed, in the form of a value equivalent to a CHAR(16) representation of the time appended by four zeros
REMARKS	IBM internal use only.
EXPANSION_REASON	Applies only to statements that reference archive tables or temporal tables. For other statements, this column is blank.

DSN_QUERYINFO_TABLE

The query information table contains information about the eligibility of query blocks for automatic query rewrite, the MQTs that are considered for eligible query blocks, why ineligible query blocks are not eligible, and the acceleration of query blocks.

Column name	Description
QUERYNO	A number that identifies the statement that is being explained. The origin of the value depends on the context of the row. @ For rows produced by EXPLAIN statements, specify the number in the QUERYNO clause, which is an optional part of the SELECT, INSERT, UPDATE, MERGE, and DELETE statement syntax. @ For rows not produced by EXPLAIN statements, DB2 assigns a number that is based on the line number of the SQL statement in @ the source program. @ When the values of QUERYNO are based on the statement number in the source program, values that exceed 32,767 are reported as 0. However, in certain rare cases, the value is not guaranteed to be unique. @ When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, and if the QUERYNO clause is specified, then its DB2 uses its value. Otherwise, DB2 assigns a number based on the line number of the SQL statement in the non-inline SQL function, or native SQL procedure.
QBLOCKNO	A number that identifies each query block within a query. The values of the numbers are not in any particular order, nor are they necessarily consecutive.
QINAME1	TYPE='A' and REASON_CODE=0 indicates that this value is the name of the accelerator server to which the query is sent. @ TYPE='A' and REASON_CODE<>0 indicates that the query was not sent to an accelerator server. @ The REASON_CODE value indicates why the query was not sent to the accelerator server.
QINAME2	TYPE='A' and REASON_CODE=0 indicates that this value is the name of the location.
APPLNAME	The name of the application plan for the row. This applies only to embedded EXPLAIN statements that are executed from a plan or to statements that are explained when binding a plan. A blank indicates that the column is not applicable. @ When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
PROGNAME	The name of the program or package containing the statement being explained. This applies only to embedded EXPLAIN statements and to statements explained as the result of binding a plan or package. A blank indicates that the column is not applicable. @ When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
VERSION	The version identifier for the package. This applies only to an embedded EXPLAIN statement executed from a package or to a statement that is explained when binding a package. A blank indicates that the column is not applicable. @ When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
GROUP_MEMBER	The member name of the DB2 subsystem that executed EXPLAIN. The column is blank if the DB2 subsystem was not in a data sharing environment when EXPLAIN was executed.
COLLID	The collection ID: @ DSNDYNAMICSQLCACHE—The row originates from the dynamic statement cache. @ DSNEXPLAINMODEYES—The row originates from an application that specifies YES for the value of the CURRENT EXPLAIN MODE special register. @ DSNEXPLAINMODEEXPLAIN—The row originates from an application that specifies EXPLAIN for the value of the CURRENT EXPLAIN MODE special register. @ When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.

Column name	Description
SECTNOI	The section number of the statement. The value is taken from the same column in SYSPACKSTMT or SYSSTMT tables and can be used to join tables to reconstruct the access path for the statement. This column is applicable only for static statements. The default value of -1 indicates that EXPLAIN information was captured in Version 9 or earlier.
SEQNO	The sequence number for this row if QI_DATA exceeds the size of its column.
EXPLAIN_TIME	The time when the EXPLAIN information was captured: @ All cached statements—When the statement entered the cache, in the form of a full-precision timestamp value @ Noncached static statements—When the statement was bound, in the form of a full precision timestamp value @ Noncached dynamic statements—When EXPLAIN was executed, in the form of a value equivalent to a CHAR(16) representation of the time appended by four zeros
TYPE	The type of the output for this row: @ A = This row is for a query that DB2 attempts to run on an accelerator server. The value in column REASON_CODE indicates the outcome.
QI_DATA	When TYPE='A': @ For REASON_CODE values other than 0, this value is the description of the REASON_CODE value. @ For a REASON_CODE value of 0, this value is the query text, after it is converted for processing by the accelerator.
SERVICE_INFO	IBM internal use only.
QB_INFO_ROWID	IBM internal use only.
EXPANSION_REASON	Applies only to statements that reference archive tables or temporal tables. For other statements, this column is blank.

DSN_PTASK_TABLE

The parallel tasks table contains information about the parallel tasks in a query.

Column name	Description
QUERYNO	A number intended to identify the statement being explained. For a row produced by an EXPLAIN statement, specify the number in the QUERYNO clause. For a row produced by non-EXPLAIN statements, specify the number by using the QUERYNO clause, which is an optional part of the SELECT, INSERT, UPDATE, MERGE, and DELETE statement syntax. Otherwise, DB2 assigns a number based on the line number of the SQL statement in the source program. @ FETCH statements do not each have an individual QUERYNO assigned to them. Instead, DB2 uses the QUERYNO of the DECLARE CURSOR statement for all corresponding FETCH statements for that cursor. @ When the values of QUERYNO are based on the statement number in the source program, values greater than 32,767 are reported as 0. Hence, in a very long program, the value is not guaranteed to be unique. If QUERYNO is not unique, use the value of TIMESTAMP, which is always unique. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, and if the QUERYNO clause is specified, then DB2 uses it value. Otherwise, DB2 assigns a number based on the line number of the SQL statement in the nonlinear SQL function, or native SQL procedure.

Column name	Description
QBLOCKNO	A number that identifies each query block within a query.
APPLNAME	The application plan name. @ When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
PROGNAME	The program name (binding an application) or the package name (binding a package). This applies only to embedded EXPLAIN statements and to statements explained as the result of binding a plan or package. A blank indicates that the column is not applicable. @ When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
LPTNO	The parallel task number.
KEYCOLID	The key columns ID (KEY range only).
DPSI	Indicates whether a data partition secondary index (DPSI) is used.
LPTLOKEY	Low key value for this key column for this parallel task (KEY range only).
LPTHIKEY	High key value for this key column for this parallel task (KEY range only).
LPTLOPAG	Low page information if partitioned by page range.
LPTLHIPAG	High page information if partitioned by page range.
LPTLOPG#	Lower bound page number for this parallel task (page range or DPSI enabled only).
LPTHIPG#	The upper bound page number for this parallel task (page range or DPSI-enabled only).
LPTLOPT#	The lower bound partition number for this parallel task (page range or DPSI-enabled only).
KEYCOLDT	The data type for this key column (KEY range only).
KEYCOLPREC	The precision/length for this key column (KEY range only).
KEYCOLSCAL	The scale for this key column (KEY range with decimal data type only).
EXPLAIN_TIME	The EXPLAIN timestamp.
GROUP_MEMBER	The member name of the DB2 subsystem that executed EXPLAIN. The column is blank if the DB2 subsystem was not in a data sharing environment when EXPLAIN was executed.
SECTNOI	The section number of the statement. The value is taken from the same column in SYSPACKSTMT or SYSSTMT tables and can be used to join tables to reconstruct the access path for the statement. This column is applicable only for static statements. The default value of -1 indicates that EXPLAIN information was captured in DB2 10 for z/OS or earlier versions of DB2.

Column name	Description
COLLID	The collection ID: @ DSNDYNAMICSQLCACHE—The row originates from the dynamic statement cache. @ DSNEXPLAINMODEYES—The row originates from an application that specifies YES for the value of the CURRENT EXPLAIN MODE special register. @ DSNEXPLAINMODEEXPLAIN—The row originates from an application that specifies EXPLAIN for the value of the CURRENT EXPLAIN MODE special register. @ When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
VERSION	The version identifier for the package. This applies only to an embedded EXPLAIN statement executed from a package or to a statement that is explained when binding a package. A blank indicates that the column is not applicable. @ When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
EXPANSION_REASON	Applies only to statements that reference archive tables or temporal tables. For other statements, this column is blank.

DSN_QUERY_TABLE

The query table contains information about an SQL statement and displays the statement before and after query transformation in XML.

Column name	Description
QUERYNO	A number intended to identify the statement being explained. For a row produced by an EXPLAIN statement, specify the number in the QUERYNO clause. For a row produced by non-EXPLAIN statements, specify the number by using the QUERYNO clause, which is an optional part of the SELECT, INSERT, UPDATE, MERGE, and DELETE statement syntax. Otherwise, DB2 assigns a number based on the line number of the SQL statement in the source program. @ FETCH statements do not each have an individual QUERYNO assigned to them. Instead, DB2 uses the QUERYNO of the DECLARE CURSOR statement for all corresponding FETCH statements for that cursor. @ When the values of QUERYNO are based on the statement number in the source program, values greater than 32,767 are reported as 0. Hence, in a very long program, the value is not guaranteed to be unique. If QUERYNO is not unique, use the value of TIMESTAMP, which is always unique. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, and if the QUERYNO clause is specified, then DB2 uses its value. Otherwise, DB2 assigns a number based on the line number of the SQL statement in the nonlinear SQL function, or native SQL procedure.
TYPE	The type of the data in the NODE_DATA column.
QUERY_STAGE	The stage during query transformation when this row is populated.
SEQNO	Sequence number for this row if NODE_DATA exceeds the size of its column.
NODE_DATA	The XML data containing the SQL statement and its query block, table, and column information.
EXPLAIN_TIME	The EXPLAIN timestamp.

Column name	Description
QUERY_ROWID	The ROWID of the statement.
GROUP_MEMBER	The member name of the DB2 subsystem that executed EXPLAIN. The column is blank if the DB2 subsystem was not in a data sharing environment when EXPLAIN was executed.
HASHKEY	The hash value of the contents in NODE_DATA.
HASH_PRED	Indicates whether an SQL statement in the NODE_DATA column contains a parameter marker literal, a nonparameter marker literal, or no predicates.
SECTNOI	The section number of the statement. The value is taken from the same column in SYSPACKSTMT or SYSSTMT tables and can be used to join tables to reconstruct the access path for the statement. This column is applicable only for static statements. The default value of -1 indicates that EXPLAIN information was captured in DB2 10 for z/OS or earlier.
APPLNAME	The application plan name. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
PROGNAME	The program name (binding an application) or the package name (binding a package). This applies only to embedded EXPLAIN statements and to statements explained as the result of binding a plan or package. A blank indicates that the column is not applicable. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
COLLID	The collection ID: @ DSNDYNAMICSQLCACHE—The row originates from the dynamic statement cache. @ DSNEXPLAINMODEYES—The row originates from an application that specifies YES for the value of the CURRENT EXPLAIN MODE special register. @ DSNEXPLAINMODEEXPLAIN—The row originates from an application that specifies EXPLAIN for the value of the CURRENT EXPLAIN MODE special register. @ When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
VERSION	The version identifier for the package. This applies only to an embedded EXPLAIN statement executed from a package or to a statement that is explained when binding a package. A blank indicates that the column is not applicable. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
EXPANSION_REASON	Applies only to statements that reference archive tables or temporal tables. For other statements, this column is blank.

DSN_SORTKEY_TABLE

The sort key table contains information about sort keys for all the sorts that a query requires.

Column name	Description
QUERYNO	A number intended to identify the statement being explained. For a row produced by an EXPLAIN statement, specify the number in the QUERYNO clause. For a row produced by non-EXPLAIN statements, specify the number by using the QUERYNO clause, which is an optional part of the SELECT, INSERT, UPDATE, MERGE, and DELETE statement syntax. Otherwise, DB2 assigns a number based on the line number of the SQL statement in the source program. When the values of QUERYNO are based on the statement number in the source program, values greater than 32,767 are reported as 0. Hence, in a very long program, the value is not guaranteed to be unique. If QUERYNO is not unique, use the value of TIMESTAMP, which is always unique. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, and if the QUERYNO clause is specified, then DB2 uses its value. Otherwise, DB2 assigns a number based on the line number of the SQL statement in the nonline SQL function, or native SQL procedure.
QBLOCKNO	A number that identifies each query block within a query.
PLANNO	A number that identifies each mini-plan within a query block.
APPLNAME	The application plan name. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is blank.
PROGNAME	The program name (binding an application) or the package name (binding a package). This applies only to embedded EXPLAIN statements and to statements explained as the result of binding a plan or package. A blank indicates that the column is not applicable. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
COLLID	The collection ID: When the row originates from the dynamic statement cache. When the row originates from an application that specifies YES for the value of the CURRENT EXPLAIN MODE special register. When the row originates from an application that specifies EXPLAIN for the value of the CURRENT EXPLAIN MODE special register. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
SORTNO	The sequence number of the sort.
ORDERNO	The sequence of the sort key.
EXPTYPE	The type of the sort key. The possible values are as follows: COL EXP QRY
TEXT	The sort key text. This can be a column name, a scalar subquery, or a 'Record ID'.
TABNO	A number that uniquely identifies a corresponding table reference within a query.
COLNO	A number that uniquely identifies the corresponding column within a query. It is applicable only when the sort key is a column.
DATATYPE	The data type of the sort key. The possible values are as follows: HEXADECI MAL CHARACTER PACKED FIELD FIXED(31) FIXED(15) DATE TIME VARCHAR PACKED FLD FLOAT TIMESTAMP UNKNOWN DATA TYPE

Column name	Description
LENGTH	The length of the sort key.
CCSID	IBM internal use only.
ORDERCLASS	IBM internal use only.
EXPLAIN_TIME	The EXPLAIN timestamp.
GROUP_MEMBER	The member name of DB2 subsystem that executed EXPLAIN. This is blank if the DB2 subsystem was not in a data sharing environment when EXPLAIN was executed.
SECTNOI	The section number of the statement. The value is taken from the same column in SYSPACKSTMT or SYSSTMT tables and can be used to join tables to reconstruct the access path for the statement. This column is applicable only for static statements. The default value of -1 indicates that EXPLAIN information was captured in DB2 10 for z/OS or earlier.
VERSION	The version identifier for the package. This applies only to an embedded EXPLAIN statement executed from a package or to a statement that is explained when binding a package. A blank indicates that the column is not applicable. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
EXPANSION_REASON	Applies only to statements that reference archive tables or temporal tables. Otherwise, it is blank.

DSN_SORT_TABLE

The sort table contains information about sort operations that a query requires.

Column name	Description
QUERYNO	A number intended to identify the statement being explained. For a row produced by an EXPLAIN statement, specify the number in the QUERYNO clause. For a row produced by non-EXPLAIN statements, specify the number by using the QUERYNO clause, which is an optional part of the SELECT, INSERT, UPDATE, MERGE, and DELETE statement syntax. Otherwise, DB2 assigns a number based on the line number of the SQL statement in the source program. Fetch statements do not each have an individual QUERYNO assigned to them. Instead, DB2 uses the QUERYNO of the DECLARE CURSOR statement for all corresponding FETCH statements for that cursor. When the values of QUERYNO are based on the statement number in the source program, values greater than 32,767 are reported as 0. Hence, in a very long program, the value is not guaranteed to be unique. If QUERYNO is not unique, use the value of TIMESTAMP, which is always unique. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, and if the QUERYNO clause is specified, then DB2 uses its value. Otherwise, DB2 assigns a number based on the line number of the SQL statement in the non-line SQL function, or native SQL procedure.
QBLOCKNO	A number that identifies each query block within a query.
PLANNO	A number that identifies each mini-plan within a query block.

Column name	Description
APPLNAME	The application plan name.
PROGNAME	The program name (binding an application) or package name (binding a package).
COLLID	The collection ID: @ DSNDYNAMICSQLCACHE—The row originates from the dynamic statement cache. @ DSNEXPLAINMODEYES—The row originates from an application that specifies YES for value of CURRENT EXPLAIN MODE special register. @ DSNEXPLAINMODEEXPLAIN—The row originates from an application that specifies EXPLAIN for value of CURRENT EXPLAIN MODE special register. @ If the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
SORTC	Reasons for a sort of a composite table, using a bitmap of the following values: @ G = Group By @ O = Order By @ J = Join @ U = Uniqueness
SORTN	Reasons for a sort of a composite table, using a bitmap of following values: @ G = Group By @ O = Order By @ J = Join @ U = Uniqueness
SORTNO	The sequence of the sort.
KEYSIZE	The sum of the lengths of the sort keys.
ORDERCLASS	IBM internal use only.
EXPLAIN_TIME	The EXPLAIN timestamp.
GROUP_MEMBER	The member name of DB2 subsystem that executed EXPLAIN. This column is blank if the DB2 subsystem was not in a data sharing environment when EXPLAIN was executed.
SECTNOI	The section number of the statement. The value is taken from the same column in SYSPACKSTMT or SYSSTMT tables and can be used to join tables to reconstruct the access path for the statement. This column is applicable only for static statements. The default value of -1 indicates that EXPLAIN information was captured in DB2 10 for z/OS or earlier.
VERSION	The version identifier for a package. This applies only to an embedded EXPLAIN statement executed from a package or to a statement that is explained when binding a package. A blank indicates that the column is not applicable. @ When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
EXPANSION_REASON	Applies only to statements that reference archive tables or temporal tables. Otherwise, it is blank.

DSN_STATEMENT_CACHE_TABLE

The statement cache table contains information about the SQL statements in the statement cache.

Column name	Description
STMT_ID	An EDM unique token.
STMT_TOKEN	A user-provided identification string.

Column name	Description
COLLID	The collection ID: @ DSNDYNAMICSQLCACHE—The row originates from the dynamic statement cache. @ DSNEXPLAINMODEYES—The row originates from an application that specifies YES for value of CURRENT EXPLAIN MODE special register. @ DSNEXPLAINMODEEXPLAIN—The row originates from an application that specifies EXPLAIN for value of the CURRENT EXPLAIN MODE special register. @ If the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is blank.
PROGRAM_NAME	The name of the package or DBRM that performed the initial PREPARE.
INV_DROPALT	Invalidated by DROP/ALTER.
INV_REVOKE	Invalidated by REVOKE.
INV_LRU	Removed from the cache by LRU.
INV_RUNSTATS	Invalidated by RUNSTATS.
CACHED_TS	The timestamp when the statement was cached.
USERS	The number of current users of a statement. These are the users who have prepared or executed the statement during their current unit of work.
COPIES	The number of copies of the statement owned by all threads in the system.
LINES	The precompiler line number from the initial PREPARE.
PRIMAUTH	The primary authorization ID of the user who did the initial PREPARE.
CURSQLID	The CURRENT SQLID of the user who did the initial PREPARE.
BIND_QUALIFIER	The bind object qualifier for unqualified table names.
BIND_ISO	ISOLATION bind option: @ UR = Uncommitted read @ CS = Cursor stability @ RS = Read stability @ RR =Repeatable read
BIND_CDATA	DATA CURRENTDATA bind option: @ Y = CURRENTDATA(YES) @ N = CURRENTDATA(NO)
BIND_DYNRL	DYNAMICRULES bind option: @ B = DYNAMICRULES(BIND) @ R = DYNAMICRULES(RUN)
BIND_DEGRE	CURRENT DEGREE value: @ A = ANY @ 1 = 1
BIND_SQLRL	CURRENT RULES value: @ D = DB2 @ S = SQL
BIND_CHOLD	Cursor WITH HOLD bind option: @ Y = Initial PREPARE was done for a cursor WITH HOLD. @ N = Initial PREPARE was not done for a cursor WITH HOLD.
STAT_TS	The timestamp of statistics when IFCID 318 is started.
STAT_EXEC	This column is deprecated. Use STAT_EXECB instead.
STAT_GPAG	This column is deprecated. Use STAT_GPAGB instead.
STAT_SYNR	This column is deprecated. Use STAT_SYNRB instead.
STAT_WRIT	This column is deprecated. Use STAT_WRITB instead.
STAT_EROW	This column is deprecated. Use STAT_EROWB instead.
STAT_PROW	This column is deprecated. Use STAT_PROWB instead.

Column name	Description
STAT_SORT	This column is deprecated. Use STAT_SORTB instead.
STAT_INDX	This column is deprecated. Use STAT_INDXB instead.
STAT_RSCN	This column is deprecated. Use STAT_RSCNB instead.
STAT_PGRP	This column is deprecated. Use STAT_PGRPB instead.
STAT_ELAP	Accumulated elapsed time used for the statement.
STAT_CPU	Accumulated CPU time used for the statement.
STAT_SUS_SYNO	Accumulated wait time for synchronous I/O.
STAT_SUS_LOCK	Accumulated wait time for lock and latch requests.
STAT_SUS_SWIT	Accumulated wait time for synchronous execution unit switch.
STAT_SUS_GLCK	Accumulated wait time for global locks.
STAT_SUS_OTHR	Accumulated wait time for read activity done by another thread.
STAT_SUS_OTHW	Accumulated wait time for write activity done by another thread.
STAT_RIDLMT	This column is deprecated. Use STAT_RIDLMTB instead.
STAT_RIDSTOR	This column is deprecated. Use STAT_RIDSTORB instead.
EXPLAIN_TS	When the statement cache table is populated.
SCHEMA	The CURRENT SCHEMA value.
STMT_TEXT	The statement text.
STMT_ROWID	The statement ROWID.
BIND_RO_TYPE	The current specification of the REOPT option for a statement: @ N = REOPT(NONE) @ 1 = REOPT(ONCE) or its equivalent @ A = REOPT(AUTO) or its equivalent @ 0 = No need for REOPT(AUTO)
BIND_RA_TOT	The total number of REBIND commands issued for the dynamic statement because of the REOPT(AUTO) option.
GROUP_MEMBER	The name of the DB2 data sharing member that inserted the row. This column is null if it is not in a data sharing environment.
STAT_GPAGB	The number of getpage operations that are performed.
STAT_SYNRB	The number of synchronous buffer reads that are performed.
STAT_WRITB	The number of buffer write operations that are performed.
STAT_EROWB	The number of rows that are examined.
STAT_PROWB	The number of rows that are processed.
STAT_SORTB	The number of sorts that are performed.
STAT_EXECB	The number of times this statement has been run. For a statement with a cursor, this is the number of OPENS.
STAT_INDXB	The number of index scans that are performed.
STAT_RSCNB	The number of table space scans that are performed.

Column name	Description
STAT_PGRPB	The number of parallel groups that are created.
STAT_RIDLIMTB	The number of times an RID list was not used because the number of RIDs would have exceeded DB2 limits.
STAT_RIDSTORB	The number of time an RID list was not used because sufficient storage is not available to hold the list of RIDs.
LITERAL_REPL	Identifies cached statements where literal values are replaced by '&' @ R = Statement is prepared with CONCENTRATE STATEMENTS WITH LITERALS behavior and literal constants in the statement have been replaced with '&' @ D = Statement is a duplicate statement instance with different literal reusability criteria. @ Blank = Literal values are not replaced.
STAT_SUS_LATCH	Accumulated wait time for latch requests.
STAT_SUS_PLATCH	Accumulated wait time for page latch requests.
STAT_SUS_DRAIN	Accumulated wait time for a drain lock requests.
STAT_SUS_CLAIM	Accumulated wait time for claim count requests.
STAT_SUS_LOG	Accumulated wait time for the log writer requests.
EXPANSION_REASON	Applies only to statements that reference archive tables or temporal tables. For other statements, this column is blank.

DSN_STATEMNT_TABLE

The statement table contains information about the estimated cost of specified SQL statements.

Column name	Description
QUERYNO	A number intended to identify the statement being explained. For a row produced by an EXPLAIN statement, specify the number in the QUERYNO clause. For a row produced by non-EXPLAIN statements, specify the number by using the QUERYNO clause, which is an optional part of the SELECT, INSERT, UPDATE, MERGE, and DELETE statement syntax. Otherwise, DB2 assigns a number based on the line number of the SQL statement in the source program. @ FETCH statements do not each have an individual QUERYNO assigned to them. Instead, DB2 uses the QUERYNO of the DECLARE CURSOR statement for all corresponding FETCH statements for that cursor. @ When the values of QUERYNO are based on the statement number in the source program, values greater than 32,767 are reported as 0. Hence, in a very long program, the value is not guaranteed to be unique. If QUERYNO is not unique, use the value of TIMESTAMP, which is always unique. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, and if the QUERYNO clause is specified, then DB2 uses its value. Otherwise, DB2 assigns a number based on the line number of the SQL statement in the nonlinear SQL function, or native SQL procedure.

Column name	Description
APPLNAME	The name of the application plan for the row. This applies only to embedded EXPLAIN statements executed from a plan or to statements explained when binding a plan. This column is blank if not applicable. If the SQL statement is embedded in a non-inline SQL function or native SQL procedure, the column is blank.
PROGNAME	The name of the program or package containing the statement being explained. For statements explained dynamically, such as with QMF or SPUFI, the associated plan/package is listed. This column is blank if not applicable. @ If the SQL statement is embedded in a non-inline SQL function or native SQL procedure, the column is blank.
COLLID	The collection ID: @ DSNDYNAMICSQLCACHE—The row originates from the dynamic statement cache. @ DSNEXPLAINMODEYES—The row originates from an application that specifies YES for the value of the CURRENT EXPLAIN MODE special register. @ DSNEXPLAINMODEEXPLAIN—The row originates from an application that specifies EXPLAIN for the value of the CURRENT EXPLAIN MODE special register. @ When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
GROUP_MEMBER	The member name of the DB2 that executed EXPLAIN, or blank.
EXPLAIN_TIME	The time the statement is processed, which is the same as BIND_TIME in PLAN_TABLE.
STMT_TYPE	The type of statement being explained: @ SELECT = SELECT @ INSERT = INSERT @ UPDATE = UPDATE @ DELETE = DELETE @ MERGE = MERGE @ TRUNCA = TRUNCATE @ SELUPD = SELECT with FOR UPDATE OF @ DELCUR = DELETE WHERE CURRENT OF CURSOR @ UPDCUR = UPDATE WHERE CURRENT OF CURSOR
COST_CATEGORY	Specifies whether DB2 was forced to use default values when making its estimates: @ A = DB2 had enough information to make a cost estimate without using default values. @ B = Some condition exists for which DB2 was forced to use default values. See the values in REASON to determine why DB2 was unable to put this estimate in cost category A.
PROCMS	The estimated processor cost in milliseconds for the SQL statement, rounded up to the next integer value. The maximum value for this cost is 2,147,483,647 milliseconds, which is equivalent to approximately 24.8 days. If the estimated value exceeds this maximum, the column reports the maximum value.
PROCSU	The estimated processor cost in service units for the SQL statement, rounded up to the next integer value. The maximum value for this cost is 2,147,483,647 service units. If the estimated value exceeds this maximum, the column reports the maximum value.

Column name	Description
REASON	Reasons for putting an estimate into cost category B: @ HAVING CLAUSE—A subselect in SQL statement contains a HAVING clause. @ HOST VARIABLES—The statement uses host variables, parameter markers, or special registers. @ REFERENTIAL CONSTRAINTS—Referential constraints of the type CASCADE or SET NULL exist on the target table of a DELETE statement. @ TABLE CARDINALITY—Cardinality statistics are missing for one or more of the tables used in the statement. @ UDF—The statement uses user-defined functions. @ TRIGGERS—Triggers are defined on the target table of an INSERT, UPDATE, or DELETE statement. @ MATERIALIZATION—Statistics are missing because the statement uses materialized views or nested table expressions.
STMT_ENCODE	The encoding scheme of the statement. If the statement represents a single CCSID set, the possible values are as follows: @ A = ASCII @ E = EBCDIC @ U = Unicode @ If the statement has multiple CCSID sets, the column value is M.
TOTAL_COST	The overall estimated cost of the statement. Use this only for reference.
SECTNOI	The section number of the statement. This value is taken from the same column in SYSPACKSTMT or SYSSTMT tables and can be used to join tables to reconstruct the access path for the statement. This column is applicable only for static statements. The default of -1 indicates that EXPLAIN information was captured in DB2 10 for z/OS or earlier versions of DB2.
VERSION	The version identifier for the package. This applies only to an embedded EXPLAIN statement executed from a package or to a statement that is explained when binding a package. A blank indicates that the column is not applicable. @ When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
EXPANSION_REASON	Applies only to statements that reference archive tables or temporal tables. For other statements, this column is blank.

DSN_STAT_FEEDBACK

This table contains recommendations for capturing missing or conflicting statistics that are defined during EXPLAIN. Collecting these statistics by using the RUNSTATS utility might improve query performance.

Column name	Description
QUERYNO	A number intended to identify the statement being explained. For a row produced by an EXPLAIN statement, specify the number in the QUERYNO clause. For a row produced by non-EXPLAIN statements, specify the number by using the QUERYNO clause, which is an optional part of the SELECT, INSERT, UPDATE, MERGE, and DELETE statement syntax. Otherwise, DB2 assigns a number based on the line number of the SQL statement in the source program. When the values of QUERYNO are based on the statement number in the source program, values greater than 32,767 are reported as 0. Hence, in a very long program, the value is not guaranteed to be unique. If QUERYNO is not unique, use the value of TIMESTAMP, which is always unique. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, and if the QUERYNO clause is specified, then DB2 uses its value. Otherwise, DB2 assigns a number based on the line number of the SQL statement in the nonline SQL function, or native SQL procedure.
APPLNAME	The name of application plan for the row. This applies only to embedded EXPLAIN statements that are executed from a plan or to statements that are explained when binding a plan. A blank indicates that the column is not applicable. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, the column is blank.
PROGNAME	The program name (binding an application) or the package name (binding a package). This applies only to embedded EXPLAIN statements and to statements explained as the result of binding a plan or package. A blank indicates that the column is not applicable. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, the column is blank.
COLLID	The collection ID: DSNDDYNAMICSQLCACHE—The row originates from the dynamic statement cache. DSNEXPLAINMODEYES—The row originates from an application that specifies YES for the value of the CURRENT EXPLAIN MODE special register. DSNEXPLAINMODEEXPLAIN—The row originates from an application that specifies EXPLAIN for the value of the CURRENT EXPLAIN MODE special register. If the SQL statement is embedded in a non-inline SQL function or native SQL procedure, the column is blank.
EXPLAIN_TIME	The EXPLAIN timestamp.
GROUP_MEMBER	The member name of the DB2 subsystem that executed EXPLAIN. The column is blank if the DB2 subsystem was not in a data sharing environment when EXPLAIN was executed.
SECTNOI	The section number of statement. The value is taken from the same column in SYSPACKSTMT or SYSSTMT tables and can be used to join tables to reconstruct the access path for the statement. This column is applicable only for static statements. The default value of -1 indicates that EXPLAIN information was captured in DB2 10 for z/OS or earlier.
VERSION	The version identifier for package. This applies only to an embedded EXPLAIN statement executed from a package or to a statement that is explained when binding a package. A blank indicates that the column is not applicable. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, the column is blank.

Column name	Description
TBCREATOR	The creator of the table.
TBNAME	The name of the table.
IXCREATOR	The creator of the index.
IXNAME	The name of the index.
COLNAME	The name of the column.
NUMCOLUMNS	The number of columns in the column group.
COLGROUPCOLNO	A hex representation that identifies the set of columns associated with the statistics. If the statistics are associated with only one 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.
TYPE	The type of statistic to collect: @' C' = Cardinality @' F' = Frequency @' H' = Histogram @' I' = Index @' T' = Table
DBNAME	The name of the database.
TSNAME	The name of the table space.
REASON	The reason that the statistic was recommend: @' BASIC'—A basic statistic value for a column table or index is missing. No statistics were collected for the identified object. @' KEYCARD'—The cardinalities of index key columns are missing. @' LOWCARD'—The cardinality of the column is a low value, which indicates that data might be skewed. @' NULLABLE'—Distribution statistics are not available for a nullable column, which indicates that data might be skewed. @' DEFAULT'—A predicate references a value that is probably a default value, which indicates that data might be skewed. @' RANGEPRD'—Histogram statistics are not available for a range predicate. @' PARALLEL'—Parallelism could be improved by uniform partitioning of key ranges. @' CONFLICT'—Another statistic contains a value that conflicts with the value of this statistic. Such conflicts usually occur because statistics were collected for related objects at different times. @' COMPPFIX'—Multiple-column cardinality statistics are needed for an index compound filter factor.
REMARKS	Free-form text for extensibility.

DSN_STRUCT_TABLE

The structure table contains information about the query blocks in a query.

Column name	Description
QUERYNO	A number intended to identify the statement being explained. For a row produced by an EXPLAIN statement, specify the number in the QUERYNO clause. For a row produced by non-EXPLAIN statements, specify the number by using the QUERYNO clause, which is an optional part of the SELECT, INSERT, UPDATE, MERGE, and DELETE statement syntax. Otherwise, DB2 assigns a number based on the line number of the SQL statement in the source program. When the values of QUERYNO are based on the statement number in the source program, values greater than 32,767 are reported as 0. Hence, in a very long program, the value is not guaranteed to be unique. If QUERYNO is not unique, use the value of TIMESTAMP, which is always unique. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, and if the QUERYNO clause is specified, then DB2 uses its value. Otherwise, DB2 assigns a number based on the line number of the SQL statement in the nonline SQL function, or native SQL procedure.
QBLOCKNO	A number that identifies each query block within a query.
APPLNAME	The application plan name. If the SQL statement is embedded in a non-inline SQL function or native SQL procedure, the column is blank.
PROGNAME	The program name (binding an application) or the package name (binding a package). This applies only to embedded EXPLAIN statements and to statements explained as the result of binding a plan or package. A blank indicates that the column is not applicable. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, the column is blank.
PARENT	Parent query block number of current query block in structure of SQL text. This is the same as the PARENT_QBLOCKNO in the PLAN_TABLE.
ROWCOUNT	The estimated number of rows returned by RDS (query cardinality).
ATOPEN	Specifies whether the query block is moved up for do-at-open processing. The value is Y if processing is done at open, or N otherwise.
CONTEXT	Indicates the context of the current query block. Possible values are as follows: TOP LEVEL UNION UNION ALL PREDICATE TABLE EXP UNKNOWN
ORDERNO	Not used.
DOATOPEN_PARENT	The parent query block number of the current query block. This is the do-at-open parent if the query block processing is done at open; this value might differ from the PARENT_QBLOCKNO in the PLAN_TABLE.
QBLOCK_TYPE	The type of the current query block: SELECT INSERT UPDATE DELETE SELUPD DELCUR UPDCUR CURSUS NCOSUS TABLE X TRIGGR UNION UNIONA CTE This column is equivalent to the QBLOCK_TYPE column in PLAN_TABLE, except for CTE.
EXPLAIN_TIME	The EXPLAIN timestamp.
GROUP_MEMBER	The member name of DB2 subsystem that executed EXPLAIN. This column is blank if the DB2 subsystem was not in a data sharing environment when EXPLAIN was executed.

Column name	Description
ORIGIN	Indicates the origin of the query block: @ Blank = Generated by DB2 @ C = Column mask @ R = Row permission @ U = User-specified
SECTNOI	The section number of the statement. The value is taken from the same column in SYSPACKSTMT or SYSSTMT tables and can be used to join tables to reconstruct the access path for the statement. This column is applicable only for static statements. The default value of -1 indicates that EXPLAIN information was captured in DB2 10 for z/OS or earlier.
COLLID	The collection ID: @ DSNDYNAMICSQLCACHE—The row originated from the dynamic statement cache. @ DSNEXPLAINMODEYES—The row originated from an application that specifies YES for the value of the CURRENT EXPLAIN MODE special register. @ DSNEXPLAINMODEEXPLAIN—The row originated from an application that specifies EXPLAIN for value of CURRENT EXPLAIN MODE. @ If the SQL statement is embedded in a non-inline SQL function or native SQL procedure, the column is blank.
VERSION	The version identifier for the package. This applies only to an embedded EXPLAIN statement executed from a package or to a statement that is explained when binding a package. A blank indicates that the column is not applicable. If the SQL statement is embedded in a non-inline SQL function or native SQL procedure, the column is blank.
EXPANSION_REASON	Applies only to statements that reference archive tables or temporal tables. For other statements, this column is blank.

DSN_VIEWREF_TABLE

The view reference table contains information about all views and MQTs used to process a query.

Column name	Description
QUERYNO	A number intended to identify the statement being explained. For a row produced by an EXPLAIN statement, specify the number in the QUERYNO clause. For a row produced by non-EXPLAIN statements, specify the number by using the QUERYNO clause, which is an optional part of the SELECT, INSERT, UPDATE, MERGE, and DELETE statement syntax. Otherwise, DB2 assigns a number based on the line number of the SQL statement in the source program. @ FETCH statements do not each have an individual QUERYNO assigned to them. Instead, DB2 uses the QUERYNO of the DECLARE CURSOR statement for all corresponding FETCH statements for that cursor. @ When the values of QUERYNO are based on the statement number in the source program, values greater than 32,767 are reported as 0. Hence, in a very long program, the value is not guaranteed to be unique. If QUERYNO is not unique, use the value of TIMESTAMP, which is always unique. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, and if the QUERYNO clause is specified, then DB2 uses its value. Otherwise, DB2 assigns a number based on the line number of the SQL statement in the nonlinear SQL function, or native SQL procedure.

Column name	Description
APPLNAME	The application plan name. When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, this column is not used and is blank.
PROGNAME	The program name (binding an application) or the package name (binding a package). This applies only to embedded EXPLAIN statements and to statements explained as the result of binding a plan or package. A blank indicates that the column is not applicable. @ When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, the column is blank.
VERSION	The version identifier for the package. This applies only to an embedded EXPLAIN statement executed from a package or to a statement that is explained when binding a package. A blank indicates that the column is not applicable. @ When the SQL statement is embedded in a non-inline SQL function or native SQL procedure, the column is blank.
CREATOR	The authorization ID of the owner of the object.
NAME	The name of the object.
TYPE	The type of the object: @ V = View @ R = MQT that has been used to replace the base table for rewrite @ M = MQT
MQTUSE	IBM internal use only.
EXPLAIN_TIME	The EXPLAIN timestamp.
GROUP_MEMBER	The member name of the DB2 subsystem that executed EXPLAIN. This column is blank if the DB2 subsystem was not in a data sharing environment when EXPLAIN was executed.
SECTNOI	The section number of the statement. The value is taken from the same column in SYSPACKSTMT or SYSSTMT tables and can be used to join tables to reconstruct the access path for the statement. This column is applicable only for static statements. The default value of -1 indicates that EXPLAIN information was captured in DB2 10 for z/OS or earlier.
EXPANSION_REASON	Applies only to statements that reference archive tables or temporal tables. For other statements, this column is blank.

APPENDIX F

Predicates

The following table shows the predicates and whether they are indexable and/or stage 1. For more information, refer to *IBM DB2 11 for z/OS Managing Performance*—SC19-4060-07.

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
value BETWEEN COL1 AND COL2	N	N
COL BETWEEN COL1 AND COL2	N	N
COL BETWEEN expression1 AND expression2	Y	Y
COL LIKE 'pattern'	Y	Y
COL IN (list)	Y	Y
COL <> value	N	Y
COL <> noncol expr	N	Y

Predicate Type	Indexable	Stage 1
COL IS NOT NULL	Y	Y
COL NOT BETWEEN value1 AND value2	N	Y
COL NOT BETWEEN noncol expr1 AND noncol expr2	N	Y
value NOT BETWEEN COL1 AND COL2	N	N
COL NOT IN (list)	N	Y
COL NOT LIKE ' char'	N	Y
COL LIKE '%char'	N	Y
COL LIKE '_char'	N	Y
COL LIKE host variable	Y	Y
T1.COL = T2 col expr	Y	Y
T1.COL op T2 col expr	Y	Y
T1.COL <> T2 col expr	N	Y
T1.COL1 = T1.COL2	Y	Y
T1.COL1 op T1.COL2	Y	Y
T1.COL1 <> T1.COL2	N	N
COL=(noncor subq)	Y	Y
COL = ANY (noncor subq)	Y	Y
COL = ALL (noncor subq)	N	N
COL op (noncor subq)	Y	Y
COL op ANY (noncor subq)	Y	Y
COL op ALL (noncor subq)	Y	Y
COL <> (noncor subq)	N	Y
COL <> ANY (noncor subq)	N	N
COL <> ALL (noncor subq)	N	N
COL IN (noncor subq)	Y	Y
(COL1,...COLn) IN (noncor subq)	Y	Y
COL NOT IN (noncor subq)	N	N
(COL1,...COLn) NOT IN (noncor subq)	N	N
COL = (cor subq)	N	N
COL = ANY (cor subq)	Y	Y
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

Predicate Type	Indexable	Stage 1
COL <> ALL (cor subq)	N	N
COL IN (cor subq)	Y	Y
(COL1,...COLn) IN (cor subq)	N	N
COL NOT IN (cor subq)	N	N
(COL1,...COLn) NOT IN (cor subq)	N	N
COL IS DISTINCT FROM value	N	Y
COL IS NOT DISTINCT FROM value	Y	Y
COL IS DISTINCT FROM noncol expr	N	Y
COL IS NOT DISTINCT FROM noncol expr	Y	Y
T1.COL1 IS DISTINCT FROM T2.COL2	N	N
T1.COL1 IS NOT DISTINCT FROM T2.COL2	N	N
T1.COL1 IS DISTINCT FROM T2 col expr	N	Y
T1.COL1 IS NOT DISTINCT FROM T2 col expr	Y	Y
COL IS DISTINCT FROM (noncor subq)	N	Y
COL IS NOT DISTINCT FROM (noncor subq)	Y	Y
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
XMLEXISTS	Y	N
NOT XMLEXISTS	N	N

Advisory and Restrictive States

The following table shows the advisory and restrictive states that you can set on a DB2 object, as well as the actions that you can take to remove the status. For more information, refer to *IBM DB2 11 for z/OS Utilities*—SC19-4067-01.

Status Code	Status Name	Affected Objects	Corrective Action(s)
ACHKP	Auxiliary CHECK pending	Base table space, LOB table spaces	Update or delete invalid LOBs and XML objects by using SQL. @ Run CHECK DATA with appropriate SCOPE option to verify the validity of LOBs and XML objects.
AUXW	Auxiliary warning	Base table space	Update or delete invalid LOBs and XML by using SQL. @ If an orphan LOB or a version mismatch exists between the base table and the auxiliary index, use REPAIR to delete the LOB from the LOB table space. @ Run CHECK DATA to verify the validity of LOBs and XML objects.
		LOB table space	Update or delete invalid LOBs and XML by using SQL. @ If an orphan LOB or a version mismatch exists between the base table and the auxiliary index, use REPAIR to delete the LOB from the LOB table space. @ Run CHECK LOB to verify the validity of the LOBs and XML objects.

Status Code	Status Name	Affected Objects	Corrective Action(s)
CHECKP	CHECK pending	Table space, base table space	Check and correct RI constraints by using CHECK DATA. @ If a table space is in both REORG-pending and CHECK-pending (or auxiliary CHECK-pending) status, run REORG first and then use CHECK DATA.
		Partitioning index, nonpartitioning index, index on auxiliary table	Run CHECK INDEX on the index. @ If errors, run REBUILD INDEX.
		LOB table space	Run CHECK LOB. If errors, correct defects in LOB table space with REPAIR, and then run CHECK LOB again.
COPY	COPY pending	Table space, table space partition	Make an image copy (best action), use –START DATABASE(<i>db</i>) SPACENAM(<i>ts</i>) ACCESS FORCE, or run REPAIR and reset the COPY flag.
DBETE	Database exception table (DBET) error	Table space, partition, index, index partition, logical index partition	Contact IBM support.
GRECP	Group buffer pool (GBP) recover pending	Table space, index space	RECOVER the object, or use the START DATABASE command.
ICOPY	Informational COPY pending	Partitioned index, nonpartitioned index, index on auxiliary table	Copy the affected index.
		NOT LOGGED table space	Copy the affected table space.
LPL	Logical page list	Table spaces, index space	START DATABASE ACCESS R/W or R/O. @ Run RECOVER or REBUILD INDEX utility. @ Run LOAD REPLACE. @ DROP the object.
ARDBP	Advisory REBUILD pending	Index	Run REBUILD on affected index.
RBDP	REBUILD pending	Physical or logical index partition	Run REBUILD or RECOVER on the affected index partition.
RBDP*		Logical partitions of nonpartitioned secondary indexes	Run REBUILD INDEX PART or RECOVER on the affected logical partitions.
PSRBD		Nonpartitioned secondary index, index on auxiliary table	Run REBUILD INDEX ALL, RECOVER, or REBUILD INDEX.

Status Code	Status Name	Affected Objects	Corrective Action(s)
			<i>Note:</i> The following actions also reset the REBUILD status: @ LOAD REPLACE with table space or partition @ REPAIR SET INDEX with NORBDPEND on index part (however, this action does not correct inconsistencies) @ Start database ACCESS FORCE (however, this action does not correct inconsistencies) @ REORG INDEX SORTDATA on the index
RECP	RECOVER pending	Table space	Run the RECOVER utility on the affected object.
		Table space partition	Recover the logical partition.
		Index on auxiliary table	Run REBUILD INDEX, RECOVER INDEX, or REORG SORTDATA.
		Index space	Run one of the following utilities on the affected index space: @ REBUILD INDEX @ RECOVER INDEX @ REORG INDEX SORTDATA
		Any	The following actions also reset the RECOVER status: @ LOAD REPLACE with table space or partition @ REPAIR SET TABLESPACE or INDEX with NORCVRPEND on index part (however, this action doesn't correct inconsistencies) @ Start database ACCESS FORCE (however, this action doesn't correct inconsistencies)
REFP	Refresh pending	Table space, index space	Run a LOAD REPLACE. The object will also be in RECP or RBDP status and will need appropriate action taken.
REORP	REORG pending	Table space	Perform one of the following: @ LOAD REPLACE on entire table space @ REORG TABLESPACE SHRLEVEL NONE @ REORG TABLESPACE PART <i>n:m</i> SHRLEVEL NONE @ REORG TABLESPACE REFERENCE or CHANGE
		Partitioned table space	<i>For rows <= 32 KB:</i> @ Run REORG TABLESPACE SHRLEVEL NONE SORTDATA. @ <i>For rows > 32 KB:</i> @ Run REORG TABLESPACE UNLOAD ONLY. @ Run LOAD TABLESPACE FORMAT UNLOAD.
AREO*	Advisory REORG	Table space	Run one of the following utilities: @ REORG TABLESPACE @ LOAD REPLACE @ REPAIR TABLESPACE
		Index space	Run one of the following utilities: @ REORG TABLESPACE @ LOAD REPLACE @ REORG INDEX @ REPAIR INDEX

Status Code	Status Name	Affected Objects	Corrective Action(s)
AREOR	Advisory REORG	Table space	Run one of the following utilities: @ REORG TABLESPACE @ REPAIR TABLESPACE
		Index space	Run one of the following utilities: @ REORG TABLESPACE @ LOAD REPLACE @ REBUILD INDEX @ REPAIR INDEX
PRO	Persistent Read Only	Table space, partition	Run utilities that update data at the partition level.
RESTP	Restart pending	Table space, partition, index space, physical index partition	Objects are unavailable until backout work is complete, or until restart is canceled and a conditional restart or cold start is performed.
STOPE	Stop error	Table space, index space	RECOVER the table space or index space.
WEPR	Write error page range	Page range in error	Run a RECOVER utility on affected data.

References

The following references were used in the creation of this book:

IBM Manuals

- *IBM DB2 11 for z/OS Administration*—SC19-4050-04
- *IBM DB2 11 for z/OS Application Programming and SQL Guide*—SC19-4051-01
- *IBM DB2 11 for z/OS Command Reference*—SC19-4054-00
- *IBM DB1 11 for z/OS Data Sharing: Planning and Administration*—SC19-4055-00
- *IBM DB2 11 for z/OS Introduction*—SC19-4058-07
- *IBM DB2 11 for z/OS Managing Performance*—SC19-4060-07
- *IBM DB2 11 for z/OS Managing Security*—SC19-4061-04
- *IBM DB2 11 for z/OS pureXML Guide*—SC19-4064-83
- *IBM DB2 11 for z/OS SQL Reference*—SC19-4066-00
- *IBM DB2 11 for z/OS Utilities*—SC19-4067-01

IBM Redbooks

- *IBM DB2 11 for z/OS Performance Topics*—SG24-8222-00
- *IBM DB2 11 for z/OS Technical Overview*—SG24-8180-00
- *Subsystem and Transaction Monitoring and Tuning in DB2 11 for z/OS*—SG24-8182-00

IBM Knowledge Center

- www-01.ibm.com/support/knowledgecenter

Other

- *YLA DB2 11 for z/OS Reference Guide*

DB2 11 for z/OS Database Administration: Certification Study Guide (Exam 312)

Susan Lawson

First Edition

First Printing—June 2016

© Copyright 2016 Susan Lawson. All rights reserved.

Printed in Canada. *All rights reserved.* This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, contact mcbooks@mcpressonline.com.

Every attempt has been made to provide correct information. However, the publisher and the author do not guarantee the accuracy of the book and do not assume responsibility for information included in or omitted from it.

The following are trademarks of International Business Machines Corporation in the United States, other countries, or both: IBM, DB2, DB2 Connect, IMS, MVS, Optim, Parallel Sysplex, pureQuery, pureXML, RACF, System z, VTAM, z/Architecture, and z/OS. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>.

Java, Oracle, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates. Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both. Microsoft and Windows are trademarks of the Microsoft Corporation in the United States, other countries, or both. UNIX is a registered trademark of The Open Group in the United States, other countries, or both.

All other product names are trademarked or copyrighted by their respective manufacturers.

Portions of this book are sourced from the following IBM publications:

- *IBM DB2 11 for z/OS Administration - SC19-4050-04*
- *IBM DB2 11 for z/OS Application Programming and SQL Guide - SC19-4051-01*
- *IBM DB2 11 for z/OS Command Reference - SC19-4054-00*
- *IBM DB2 11 for z/OS Data Sharing: Planning and Administration - SC19-4055-00*
- *IBM DB2 11 for z/OS Introduction - SC19-4058-07*
- *IBM DB2 11 for z/OS Managing Performance - SC19-4060-07*
- *IBM DB2 11 for z/OS Managing Security - SC19-4061-04*
- *IBM DB2 11 for z/OS pureXML Guide - SC19-4064-83*
- *IBM DB2 11 for z/OS SQL Reference - SC19-4066-00*
- *IBM DB2 11 for z/OS Utilities - SC19-4067-01*

Reprint Courtesy of International Business Machines Corporation, © International Business Machines Corporation.

MC Press offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include custom covers and content particular to your business, training goals, marketing focus, and branding interest.

MC Press Online, LLC

Corporate Offices: 3695 W. Quail Heights Court, Boise, ID 83703-3861 USA

Sales and Customer Service: (208) 629-7275 ext. 500;

service@mcpressonline.com

Permissions and Bulk/Special Orders: mcbooks@mcpressonline.com

www.mcpressonline.com • www.mc-store.com

ISBN: 978-1-58347-397-9