# How Customers use DB2 for z/OS
# - based on V7 Catalog Survey

# White Paper

June 2004

Chris Crone
Manfred Olschanowsky
Akira Shibamiya

International Business Machines Corporation
Software Solutions Division
Silicon Valley Laboratory
San Jose, California

# Abstract

This paper reports on the results of the statistical survey of DB2 catalogs from the DB2 for z/OS Version 7 customers.

# Acknowledgment

The authors would like to thank the customers of DB2 for z/OS Version 7 who willingly participated in this catalog survey and made interesting and valuable statistical information available.

# Introduction

The statistical survey of the DB2 catalogs from DB2 for z/OS Version 7 customers was conducted in 2002 and 2003. Some of the more interesting information derived from the survey are included in this report.

Understanding how DB2 customers use DB2 is important in setting priorities on the future product enhancement, both functional and performance. For example, a higher priority can be placed on the performance of data types which are more frequently used such as character and decimal. It also helps in creating the DB2 workload for internal performance testing which most closely simulates the representative customer workload. Information on  areas where customers could tune to get better performance, for example, bind option on resource acquire and release, isolation level, DYNAMICRULES, etc., are also provided.

The survey consists of about 170 SQL calls issued against DB2 catalog tables. Over 400 statistics are generated for each DB2 subsystem and the overall aggregate is calculated.

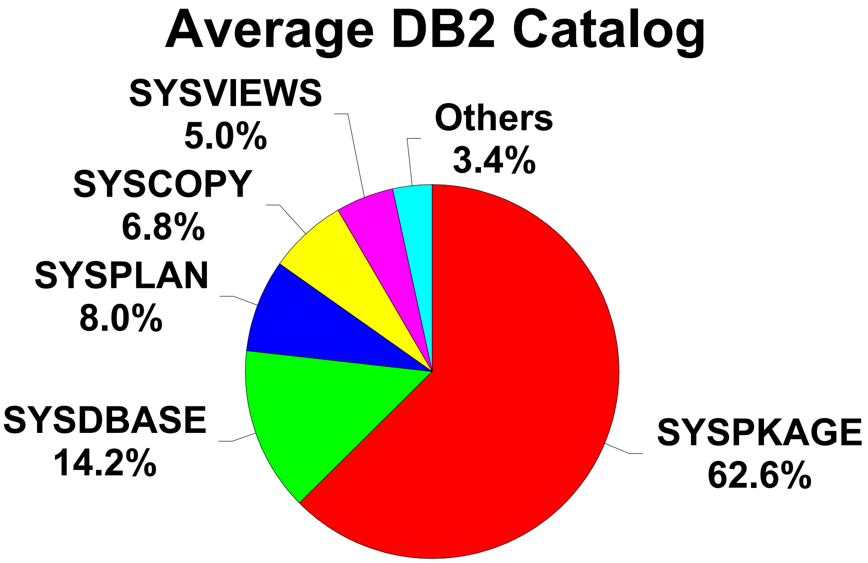
Many findings from this survey indicate, as discussed subsequently, that the most customer catalogs sampled were oriented toward transaction environment as opposed to the data warehouse query environment.

# Survey Results and Analysis

## 1. Size of Catalog

a. Out of 42 DB2 V7 catalogs surveyed, an average number of 4KB pages in the DB2 catalog was 300,000 or 1200MB, with the smallest one at 7464 pages and the largest one at 2 million pages or 8000MB. This is the sum of all 4K pages for 20 catalog tablespaces.

b. SYSPKAGE tablespace was  by far the biggest one in the DB2 catalog, followed by SYSDBASE, SYSPLAN, SYSCOPY, and SYSVIEWS tablespaces.



## 2. Tablespace (SYSTABLESPACE and SYSTABLEPART tables)

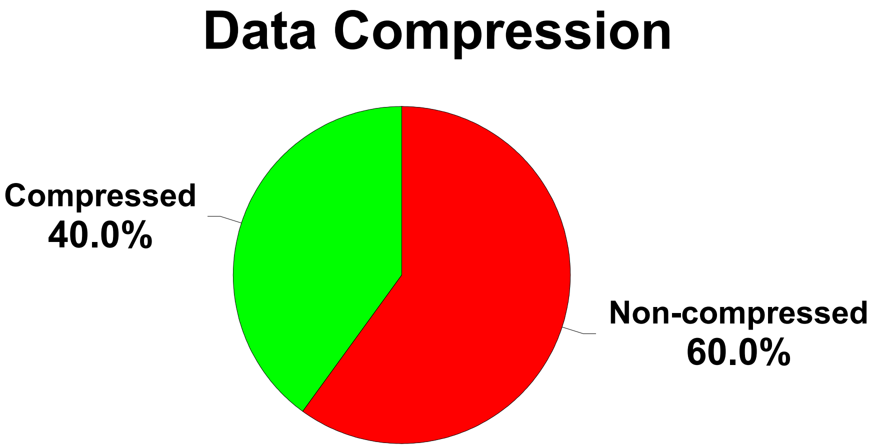
a. Out of 336,000 non partitioned tablespaces and partitions of partitioned tablespaces, 99% are DB2-managed and the remainder user-managed data sets. (Information derived from SYSTABLEPART catalog table)

   i. An average PCTFREE, or a percentage of each page left as free space, is 8 with a minimum of 0 and a maximum of 99. FREEPAGE, or a number of pages loaded before a page is left as free space, varies from 0 to 100 with an average of 4.5. The use of PCTFREE is generally recommended over FREEPAGE as it can result in fewer Buffer Manager Getpage, Read I/O, and Lock/Unlock requests in Update of variable-length record or compressed record. **In general, a recommendation is to start with defaults of 5%  PCTFREE and 0 FREEPAGE, and then refine tuning later for a specific situation only if needed.**

   ii. 40% of non partitioned tablespaces and partitions of partitioned tablespaces use DB2 data compression. If there are significant activities against compressed data, the use of zSeries (2064) processor is strongly recommended as the CPU cost of compressing or decompressing can be reduced significantly compared to 9672 processors, as zSeries is up to 5 times faster in compression and decompression compared to the previously fastest cmos (G6 turbo). For those  customers who are currently constrained from DBM1 virtual storage limit and thus unable to exploit DB2 data compression for many tablespaces and/or partitions will be able to fully exploit in DB2 for z/OS V8 as compression dictionaries will be moved above 2GB line such that there won't be any practical limit on the number of compression dictionaries to be kept in DBM1 virtual storage for open and compressed tablespaces and/or partitions.

   The compression ratio varies from a negative number to 97% maximum. 97% means what used to take 100 units of storage to save the data takes only 3 units after the DB2 data compression. The negative compression ratio means the DB2 data compression actually results in taking up more storage because of additional storage required for compression dictionary. For example, one page data after undergoing compression will take up 17 pages, one for data and 16 for compression dictionary, resulting in -1600 as a compression ratio (100%x(1-17)/1). This clearly shows that it does not pay to try to compress a small object. In order to more than offset the added size of the compression dictionary and some added CPU cost, the compression of less than 100 pages of  data is not recommended.

Over the years, we have observed a gradual but steady increase in the percentage of data that is compressed. If 40% of non partitioned tablespaces and partitions of partitioned tablespaces are being compressed, the majority of data stored in DB2 for z/OS subsystems, in terms of actual size, must be compressed, since the objects being compressed are typically much larger than the objects that are not compressed.



b. Out of 93,000 tablespaces, 3.6% are partitioned tablespaces, with an average number of partitions of 24, a minimum of 1, and a maximum of 254. (Information derived from SYSTABLESPACE catalog table)

   i. 68% of all tablespaces are segmented tablespaces with an average segment size of 36, a minimum of 4, and a maximum of 64. Segmented tablespaces compared to non segmented tablespaces including partitioned tablespaces have advantages in potentially more efficient delete, insert, update, and Drop Table processing. Scanning segmented tablespace containing multiple tables can be more efficient since only segments belonging to a given table need to be scanned.

   ii. 66% of all tablespaces have locksize ANY, as generally recommended, followed by 20% ROW, 7% TABLESPACE, and 5% PAGE level locking. Locksize TABLE and LOB (large object) are less than 1% each.

   iii. 24% of all tablespaces have LOCKMAX of 0, which means no lock escalation will occur. 9% have non zero LOCKMAX, with an average of 313,000, a minimum of 1, and a maximum of 3 million. This value indicates the maximum number of locks (row, page, table, tablespace, or LOB locks) an application process can acquire before lock escalation occurs. The remaining 67% have -1 which represents LOCKMAX SYSTEM, in which the value of field Locks per Tablespace on installation panel DSNTIPJ determines lock escalation.

   iv. LOCKPART is used by 1% of tablespaces. This option only applies to partitioned tablespaces which take up 3.6% of all tablespaces. This means 1%/3.6% or 28% of partitioned tablespaces have partition-level locking. This option locks at a finer partition level rather than at the grosser tablespace level and can be beneficial especially in data sharing environment by possibly promoting partition affinity by member and reducing inter DB2 read/write interest. This option will not be available in V8 of DB2 for z/OS as LOCKPART YES will be enforced in all cases.

   v. In terms of PAGESIZE, 4K is dominant at 98%, followed by 1% for 32K. 8K and 16K are quite rare as a majority of catalogs do not have them at all. Note that I/O striping is currently supported for 4K page, but not 8K, 16K, or 32K page. In DB2 for z/OS V8, DB2 catalogs will have

8K as well as 16K pages also.

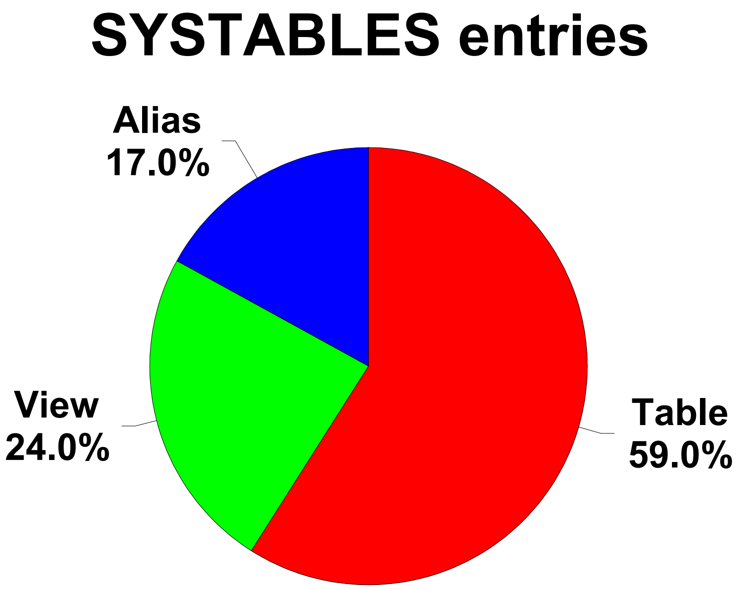
vi. 98% of tablespaces are created without DSSIZE, LARGE, LOB, and MEMBER CLUSTER options.

vii. 91% of all tablespaces have only one table. For the remaining 9%, an average number of tables in a tablespace is 16, with a minimum of 2 and a maximum of 2431. As stated already, a segmented tablespace can support many tables without performance problem. However, for a non-segmented tablespace, one table per tablespace is recommended. When there are many small tables, the use of segmented tablespace is particularly desirable to avoid having to have many data sets to be managed.

viii. 99% of all tablespaces have MAXROWS (the maximum number of rows that DB2 will store on a data page) greater than 0, with an average value of 253, a minimum of 1, and a maximum of 255 which is the default.

3. Table (SYSTABLES table) and View (SYSVIEWS table)

a. Out of 941,000 entries in the SYSTABLES table, 59% are for TABLEs, 24% are VIEWs, and 17% are ALIASes. There were also 2255 auxiliary tables for Large Objects and 860 created temp tables.



b. An average number of columns in a table was 16, with a minimum of 1 and a maximum of 703. (The maximum number of columns supported in a DB2 table is 750.) However, there was a wide variation among the customers, ranging from 2 columns average per table for one customer to 75 columns average per table for another customer. Processing hundreds of columns can add to a significant CPU time increase. For tables with many columns, the selective column processing, for example SELECT c1, c3, ... instead of SELECT *, or UPDATE of all columns in a row, is strongly recommended.

c. An average number of columns in a view is 19, with a minimum of 1 and a maximum of 487.

d. An average number of columns in a created temp table is 7 with a minimum of 1 and a maximum of 146.

e. Validation procedures (VALIDPROCS) occur less than 0.01% but edit procedures (EDITPROCS) occur 0.5%. Field procedures is discussed later in Table Column section.

f. The maximum length of any record in the table varies from 9 to 32714 bytes with an average of 370 bytes.

g. An average length of rows for the tables in a tablespace is 145 bytes, with a minimum of 1 and a maximum of 30165 bytes. If the tablespace is compressed, the value represents the compressed row length.

h. 99.6% of all tables have no AUDIT option. Audit Changes is used by 0.2% and Audit All is used by 0.2%.

i. 1.5% of all tables have Datacapture option.

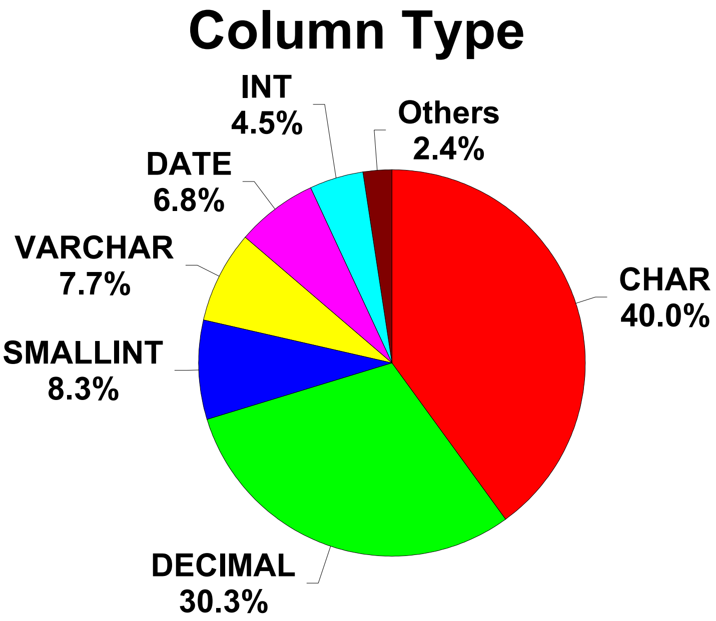
j. There are on average 1570 synonyms of tables or views in each catalog. (SYSSYNONYMS table)

k. Out of 221,000 views, 8% are created with the check option. (SYSVIEWS table)

## 4. Table Column (SYSCOLUMNS table)

a. Out of 12 million columns looked at, the CHAR datatype column was the most frequent, closely followed by the DECIMAL datatype column. Any column type with a zero occurrence is excluded.

| | Frequency | Minimum | Average | Maximum |
|---|---|---|---|---|
| | | Length | Length | Length |
| CHAR | 40.0% | 1 | 8.9 | 255 |
| DECIMAL | 30.3% | 1 | 9.5 | 31 |
| SMALLINT | 8.3% | 2 | 2 | 2 |
| VARCHAR | 7.7% | 0 | 390 | 32,700 |
| DATE | 6.8% | 4 | 4 | 4 |
| INTEGER | 4.5% | 4 | 4 | 4 |
| TIMESTMP | 1.8% | 10 | 10 | 10 |
| TIME | 0.5% | 3 | 3 | 3 |
| LONGVAR | 0.03% | 214 | 5,300 | 32,698 |
| BLOB | 0.02% | 2KB | 177MB | 2150MB |
| GRAPHIC | 0.01% | 1 | 109 | 127 |
| CLOB | 0.003% | 5KB | 63MB | 2150MB |
| ROWID | 0.003% | 17 | 17 | 17 |
| FLOAT | 0.001% | 4 | 6.4 | 8 |
| VARGRAPHIC | 0.0004% | 3 | 117 | 256 |



The CPU cost of column processing becomes important and sometimes dominant as the number of columns processed increases. This is particularly important in JDBC/SQLJ environment where Integer/Smallint has the lowest cost, but cost increases for Float, Char, Decimal, Date/Time, and Timestamp, in that order, with Timestamp being the most expensive type of column to process in JDBC/SQLJ.

b. Average lengths of CHAR and VARCHAR columns are 8.9 bytes and 390 bytes, respectively. CHAR type is more efficient for columns with relatively constant size. However, when column size varies widely, VARCHAR can be more efficient. The same observation holds for index keys also in DB2 for z/OS V8 with non-padded index key.

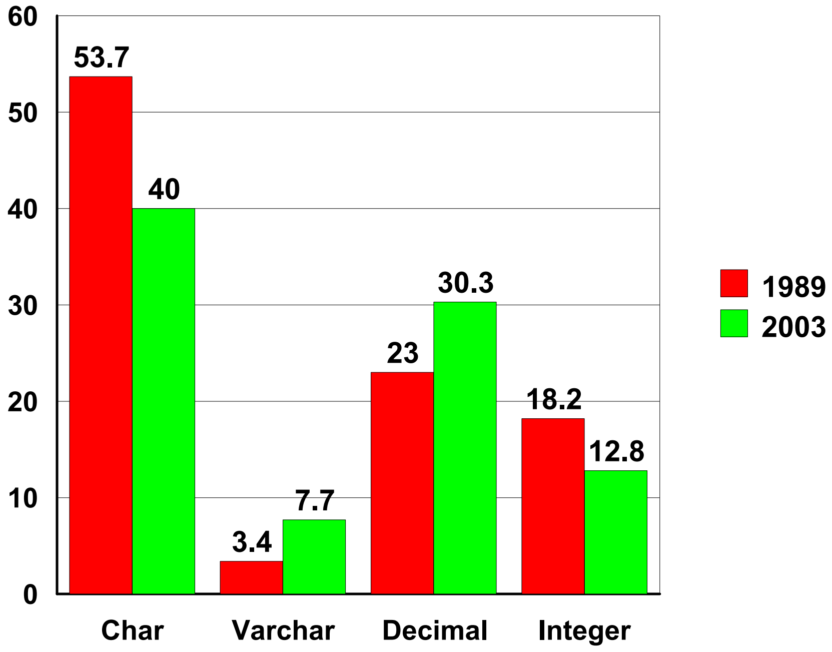
c. DECIMAL columns have an average precision of 9.5 bytes with a minimum of 1 byte and a maximum of 31 bytes which is the absolute maximum allowed. The position of the decimal point is determined by the precision and the scale of the number. The scale, which is the number of digits in the fractional part of the number, cannot be negative or greater than the precision, and has an average of 1.1, a minimum of 0, and a maximum of 19 in the survey.

d. 13% of columns are nullable.

e. Out of 12 million columns, 143 are identity columns (SYSSEQUENCES table) and 5300 have field procedures (also in SYSFIELDS table). Since identity columns were introduced in V6 of DB2 for OS/390 and therefore relatively new, the frequency of identity columns is expected to increase in the future.

f. Since GRAPHIC and VARGRAPHIC are used mostly in Chinese, Japanese, and Korean and unicode tables, the results based on 42 catalogs surveyed may be skewed.

g. It is interesting to note that over the years there is a slow but steady shift from CHAR to VARCHAR and INT/SMALLINT to DECIMAL as indicated in changing percentage values.



## 5. Index (SYSINDEXES and SYSINDEXPART tables)

a. Out of 426,000 indexes, all are type 2 indexes. That is, no type 1 index was found.

b. There are on average 1.2 indexes per table, indicating that the most customer catalogs surveyed here are oriented toward transaction environment in contrast to the data warehouse query environment.

c. A typical index level varies from 1 to 6 with an average of 2. There was one with 500 index levels and another with 10, but all others were 6 or less.

d. The COPY YES index introduced in V6 is not being used much (less than 0.01% of indexes) except by one customer.

e. 25% of indexes have non-unique index keys.

f. 63% of indexes are defined to be clustering and 58% are actually clustered, i.e. having 80% or higher cluster ratio. A high percentage is expected here since there is only 1.2 indexes per table on average.

g. 77% of indexes are multi-column index, which means 23% are single column index, with 3.4 columns average, with a minimum of 2 and a maximum of 44 columns.

h. Most index data sets are DB2-managed with only 1% user-managed. 1% here includes 118 indexes for DB2 system data sets which are always user-managed.

i. PCTFREE, which is a percentage of each leaf or nonleaf page that is left as free space, varies from 0 to 99 with an average of 10%. FREEPAGE, which is a number of pages that are loaded before a page is left as free space, ranges from 0 to 100 with an average of 4. The use of PCTFREE is recommended over FREEPAGE as it can reduce the number of index splits. A general recommendation is to start with defaults of 10% PCTFREE and 0 FREEPAGE and then make a further tuning later for a specific situation only if needed.

## 6. Application Plan (SYSPLAN table)

a. Out of 200,000 plans, 84% have validity checking performed at bind time, which can reduce the frequency of incremental bind.

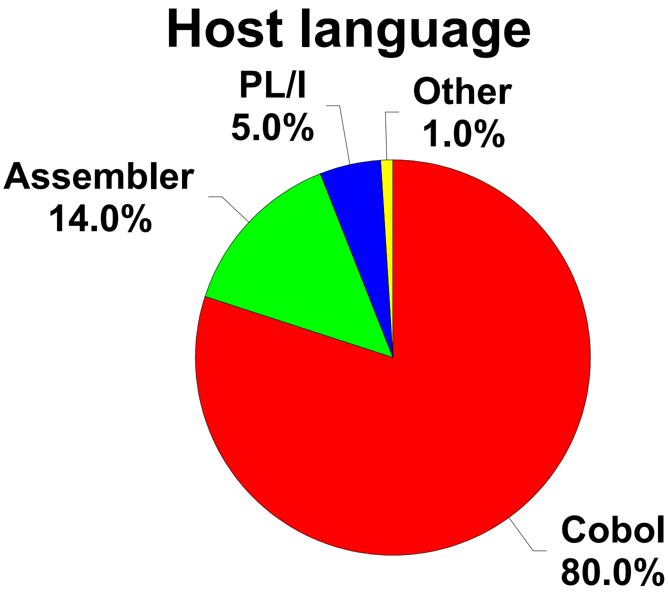
b. 96% of plans have isolation level Cursor Stability. This is recommended in general for performance. An isolation level for the SQL statement can be specified by a WITH clause, but more than 99.9% of 10 million statements in DBRMs do not specify an isolation level in a WITH clause. (SYSSTMT table)

c. 76% have a bind option of acquire resource at first use and release at commit, which is the default. This is recommended in general as “acquire at first use” acquires only the resources actually needed and uses less EDM pool storage and “release at commit” frees acquired resources sooner. 23% have acquire resource at first use and release at deallocate, which can avoid repeated acquiring and freeing of resources by keeping them for a longer period, thus reducing CPU time used. Only the most frequently used plans should be considered for release at deallocate option as it could tie up resources needed by other concurrently running applications. One of the frequently-occurring performance problems in a data sharing environment is a high false contention for parent (tablespace or partition) locks when release at commit option is used. However, in V8 of DB2 for z/OS, this problem has been removed so that the customers no longer need to use release at deallocate option to avoid a high false contention with a potential concurrency penalty.

   Acquire resource at allocation, which acquires resources referenced in all SQL statements in a plan whether needed or not, is not recommended unless most of SQL statements in a plan are executed most of the time and it occurs less than 1% in the survey.

d. 82% of plans have a plan authorization cache with an average size of 865 bytes, a minimum of 256 bytes, and a maximum of 4096 bytes. If there is a frequent plan authorization checking through the catalog, increasing the plan authorization cache may reduce this overhead.

e. Out of 242,000 DBRMs, the most predominant host language from a list of Assembler, Cobol, C, C++, Fortran, and PL/I associated with DBRM is Cobol at 80%, followed by Assembler at 14%, PL/I at 5%, and C at 0.9%. (SYSDBRM table)



f. DEFER(PREPARE) to defer PREPARE processing until OPEN is executed occurs 4% of the time.

g. The use of DEGREE(ANY) for query parallelism occurs 9% of the time.

h. The use of REOPT(VARS) for rebinding at execution time using input variable values occurs very infrequently at less than 0.01%. For an online transaction environment, REOPT(VARS) NO is overwhelmingly expected.

i. KEEPDYNAMIC bind option to avoid Prepare is used 0.1% of the time.

j. The use of DYNAMICRULES(BIND), which uses an authorization id of plan or package, occurs 14% of the time. This option can more effectively exploit the dynamic statement cache by increasing its hit ratio, compared to DYNAMICRULES(RUN) which uses an authorization id of application process or SQL authorization id.

k. The use of OPTHINT bind option to have Explain Plan Table used as input to the optimizer is practically non existent, as expected for a typical online transaction environment.

l. Many findings here indicate that the most customer catalogs surveyed here are oriented toward transaction environment in contrast to the data warehouse query environment.

m. PLSIZE or the size of the base section of a plan has an average of 8900 bytes, with a minimum of 1500 and a maximum of 350,000 bytes.

   i. AVGSIZE or the size of the section that contains SQL statements processed at bind time has an average of 3900 bytes, with a minimum of 228 bytes and a maximum of 115,000 bytes.

   ii. The total amount of EDM pool storage required for a given plan can be calculated by PLSIZE + N*AVGSIZE, where N is the number of RDS sections representing SQL statements.

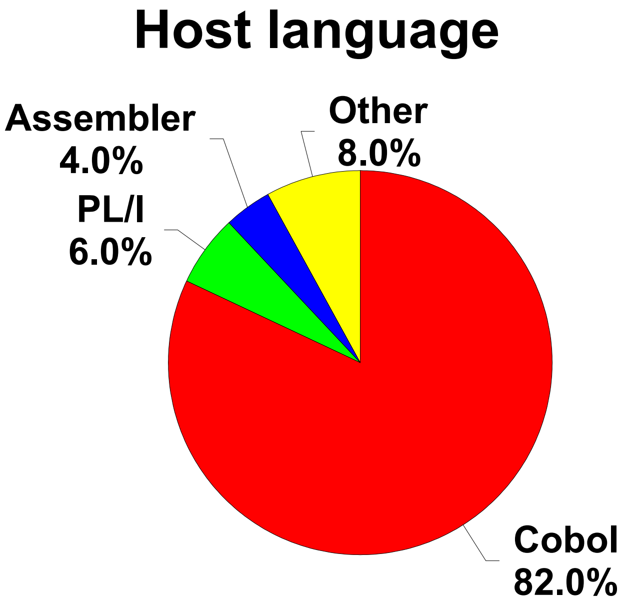
## 7. Package (SYSPACKAGE table)

a. Out of 2 million packages, 96% have validity checking performed at bind time, which can reduce the frequency of incremental bind.

b. 98% of packages have an isolation level Cursor Stability. This is recommended in general for performance. An isolation level for the SQL statement can be specified by WITH clause, but 99% of statements in a package do not. 1% of statements have WITH UR (Uncommitted Read) specified. (SYSPACKSTMT table)

c. 86% have bind option resource release at commit. This is recommended in general. Only the most frequently used packages should be considered for resource release at deallocate option, as explained in Application Plan section.

d. As it was for DBRMs, Cobol is the most predominant host language used at 82%. It is followed by PL/I at 6% and Assembler at 4%.



e. DEFER(PREPARE) to defer PREPARE processing until OPEN is executed occurs 1.6% of the time while 20% specified NO. The remainder did not specify this bind option and therefore it is inherited from the plan.

f. The use of DEGREE(ANY) for query parallelism is quite rare at 0.3%. If the package was migrated, DEGREE field will be blank.

g. The use of REOPT(VARS) for rebinding at execution time using input variable values occurs very infrequently at 0.02%. For an online transaction environment, REOPT(VARS) NO is overwhelmingly expected.

h. KEEPDYNAMIC bind option to avoid Prepare was used for 0.2% of the time.

i. The use of the OPTHINT bind option to have Explain Plan Table used as input to the optimizer is practically non existent, as expected for a typical online transaction environment.

j. Many findings here indicate that the most customer catalogs surveyed here were oriented toward transaction environment in contrast to the data warehouse query environment.

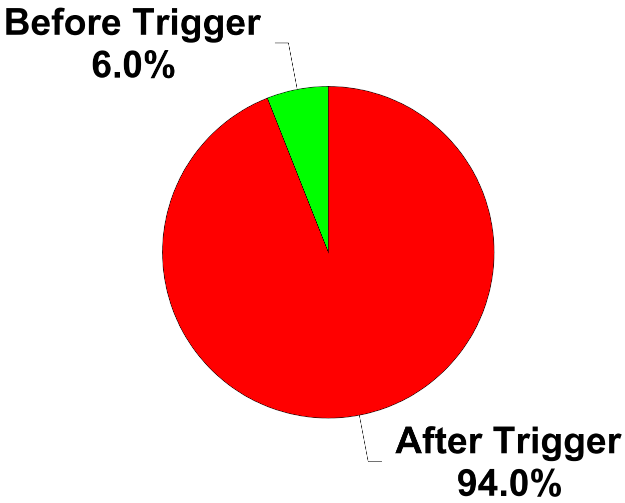
k. PKSIZE or the number of bytes for the base section of a package has an average of 2800 bytes, with a minimum of 376 bytes and a maximum of 200,000 bytes.

   i. The AVGSIZE or the number of bytes for the section that contains SQL statements processed at bind time has an average of 4400 bytes, with a minimum of 228 bytes and a maximum of 590,000 bytes.

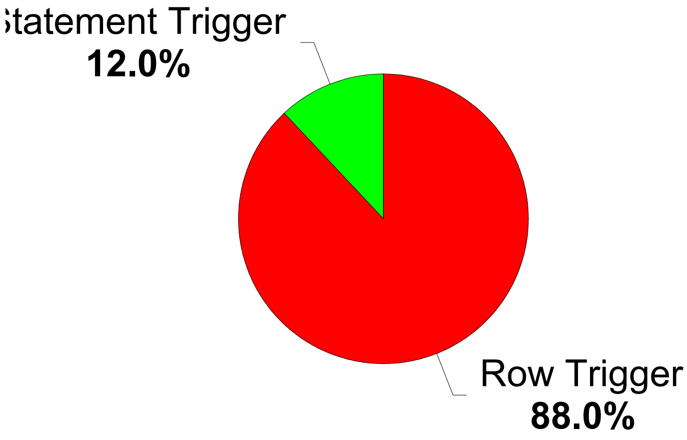
   ii. The total amount of EDM pool storage required for a given package can be calculated by PKSIZE + N*AVGSIZE, where N is the number of RDS sections representing SQL statements

## 8. Trigger (SYSTRIGGERS table)

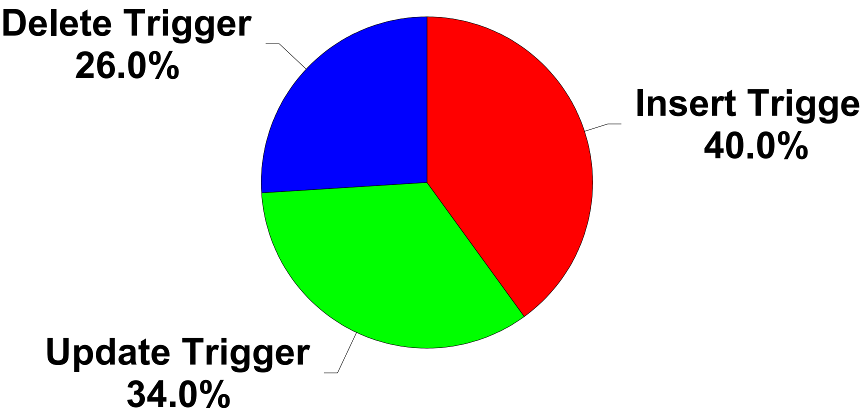
a. Out of 778 triggers, 94% are After Triggers while 6% are Before Triggers.



b. 88% are Row Triggers, which are invoked once per affected row, while 12% are Statement Triggers, which are invoked once per statement execution.



c. 40% are Insert Triggers, 34% Update Triggers, and 26% Delete Triggers.



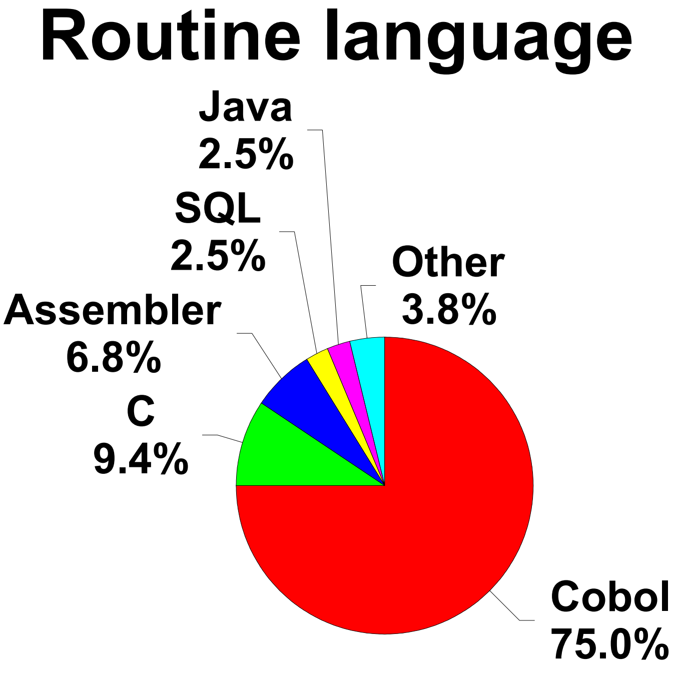
## 9. Routine (SYSROUTINES table)

a. Out of 3556 routines, 96% are stored procedures and the remainder are User Defined Functions (UDFs). Practically all are external UDFs or stored procedures rather than sourced UDFs or system-generated functions.

b. Of the 4% of routines that were UDFs, 3% are scalar functions and 1% table functions.

c. An average number of parameters for a routine is 10 with a minimum of 0 and a maximum of 90.

d. 75% of implementation language of the routine is Cobol, followed by 9.4% C, 6.8% Assembler, 2.5% each for Java and SQL, and 3.8% remainder including PL/I and REXX.



e. 82 external functions used SCRATCHPAD option, with an average length of 5514, a minimum of 4, and a maximum of 32763 bytes.

f. Parallel external function execution is used by 2%.

g. The STAYRESIDENT option is used by 20%. This is recommended to improve performance by avoiding routine reloading.

h. 90% of all routines run as Main routine and 10% as Subroutine. A PROGRAM TYPE of SUB is recommended, when it can be used, over MAIN to reduce the routine invocation overhead.

i. 88% of stored procedures have COMMIT_ON_RETURN NO, which is the default. COMMIT_ON_RETURN YES with WLM-managed routine, from remote callers, can reduce the number of messages by avoiding a separate message send/receive for commit.

## 10. Miscellaneous

a. Constraint Checking

   i. On average, 61 tables with check constraints are defined per catalog, with an average of 3.6 constraints, a minimum of 1 and a maximum of 83. Out of 61 such tables, 24 are provided by DB2 subsystem with a range of 1 to 10 constraints and an average of 4 per table. (SYSTABLES table)

   ii. There are on average 450 references to a column in a table check constraint (SYSCHECKDEP table).

   iii. There are on average 1127 referential constraints. (SYSRELS table)

   iv. There are on average 1300 unique constraints (primary key or unique key). (SYSTABCONST table)

b. Authorization

   i. There are on average 30,000 UPDATE or REFERENCES privileges that are held by users on individual columns of a table or view per catalog. (SYSCOLAUTH table)

   ii. There are on average 3000 privileges held by users over databases (SYSDBAUTH table), 66,000 over packages (SYSPACKAUTH table), 9400 over plans (SYSPLANAUTH table), 307,000 over tables and views (SYSTABAUTH table), 400 over routines (SYSROUTINEAUTH table) and 250 system privileges that are held by users (SYSUSERAUTH table) per catalog.

c. Image Copy data set can be stored on tape or DASD. The SYSCOPY catalog table indicates that out of more than 12 million Image Copy data sets, 37% are on DASD and 63% are on tape.

d. The SYSDATABASE catalog table shows that out of approximately 51,000 databases, 0.07% are for Temp databases, with a minimum of 0 and a maximum of 4 per catalog. Temp databases are used by declared temporary tables and scrollable cursors. 0.21% are for work file database, with a minimum of 1 and a maximum of 10 per catalog. There can be one work file database and Temp database for each non data sharing DB2 subsystem or for each member of data sharing system. All other databases are for normal user database.

e. Large Object

   i. There were 2255 auxiliary tables for LOB, with one catalog having 1832. (SYSAUXRELS table)

   ii. There were 640 LOB tablespaces, with one catalog having 581. (SYSLOBSTATS table) A base tablespace that is partitioned must have one auxiliary table for each partition of each LOB column.

   iii. Average size of LOB is 1.1MB with a minimum of 15 bytes and a maximum of 1070MB.

# Reference

i. V7 SQL Reference SC26-9944, Appendix D - DB2 catalog tables