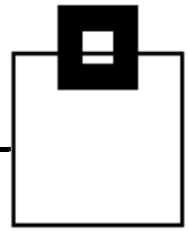
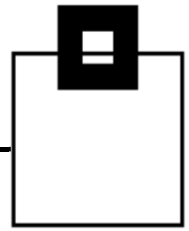


A RUNSTATS is a RUNSTATS is a RUNSTATS – or not?!

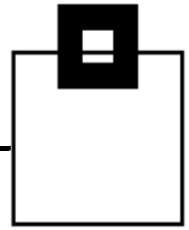
Ulf Heinrich
SEGUS, Inc
u.heinrich@segus.com

Objectives



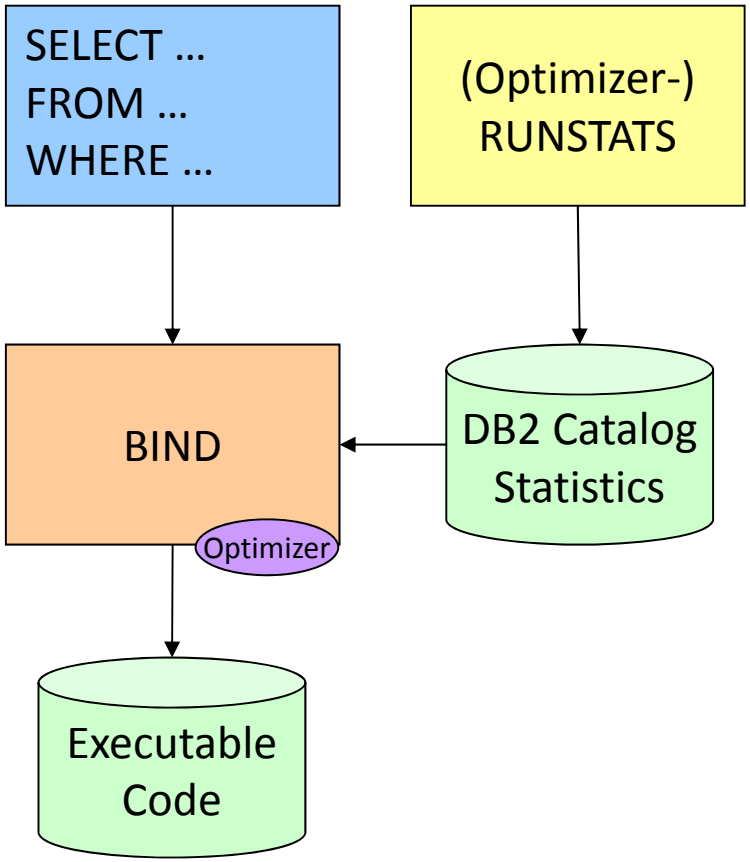
- RUNSTATS then and now
- Peculiarities with generated RUNSTATS
- Contents of SYSCOLDIST
- Cleaning up SYSCOLDIST
- Checking all of your statistics

Agenda



- DB2 Statistics
  - DB2 Catalog Statistics versus DB2 Realtime Statistics (RTS)
  - DB2 Optimizer and access path relevance
- DB2 RUNSTATS basics
- IBM recommendations through the ages
- DB2 RUNSTATS advanced
  - SYSCOLDIST explained
- RUNSTATS real world Question & Answers

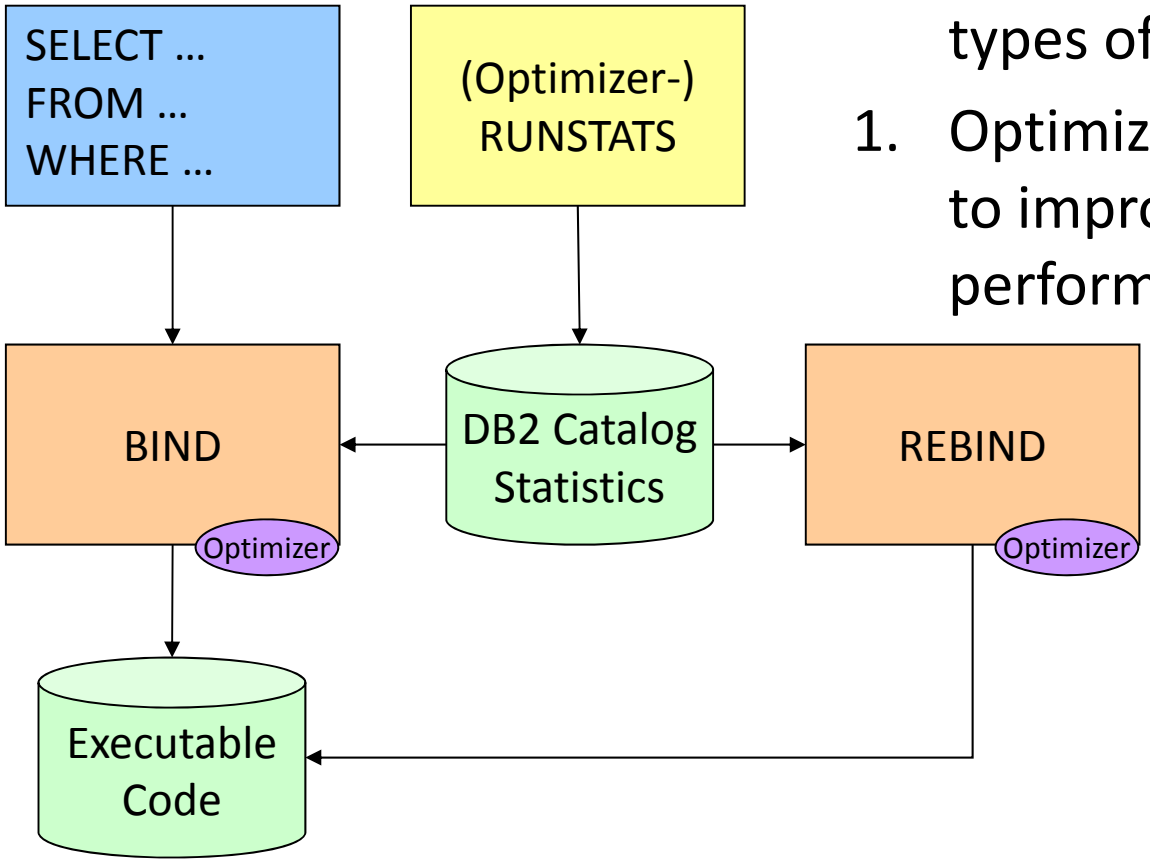
RTS vs. DB2 Catalog Statistics



Historically we have two types of RUNSTATS:

1. Optimizer RUNSTATS to improve performance

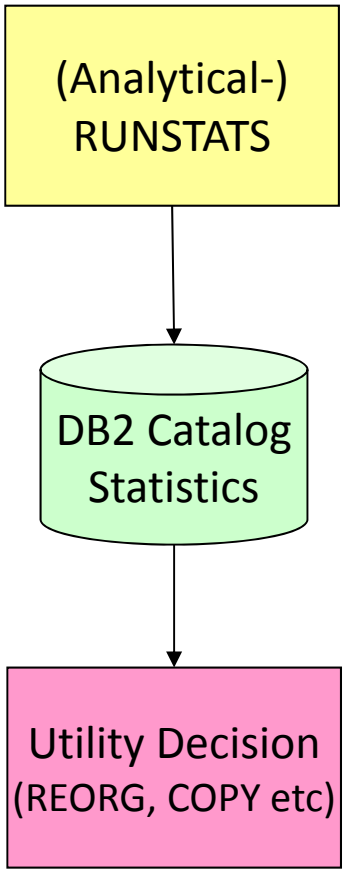
RTS vs. DB2 Catalog Statistics



Historically we have two types of RUNSTATS:

1. Optimizer RUNSTATS to improve performance

RTS vs. DB2 Catalog Statistics



Historically we have two types of RUNSTATS:

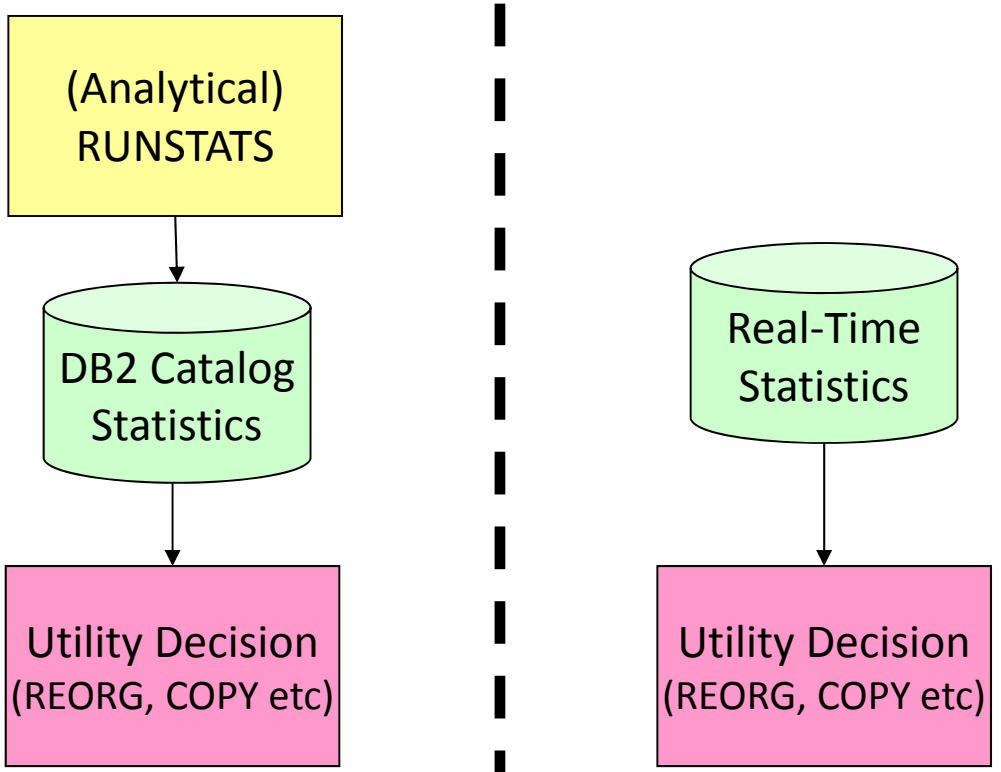
1. Optimizer RUNSTATS to improve performance
2. Analytical RUNSTATS for administrative tasks (e.g., threshold-based utilities)

RTS vs. DB2 Catalog Statistics

Since DB2 V7 we have two repositories

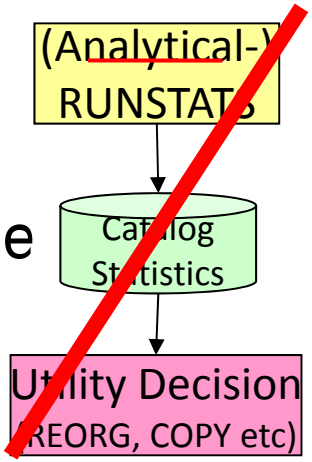
RUNSTATS → DB2 catalog

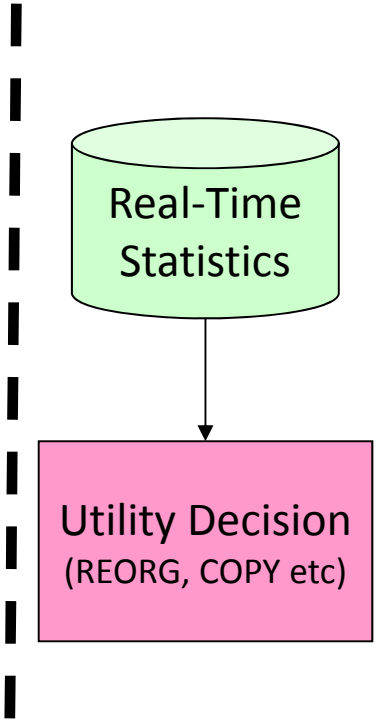
RTS → RTS objects



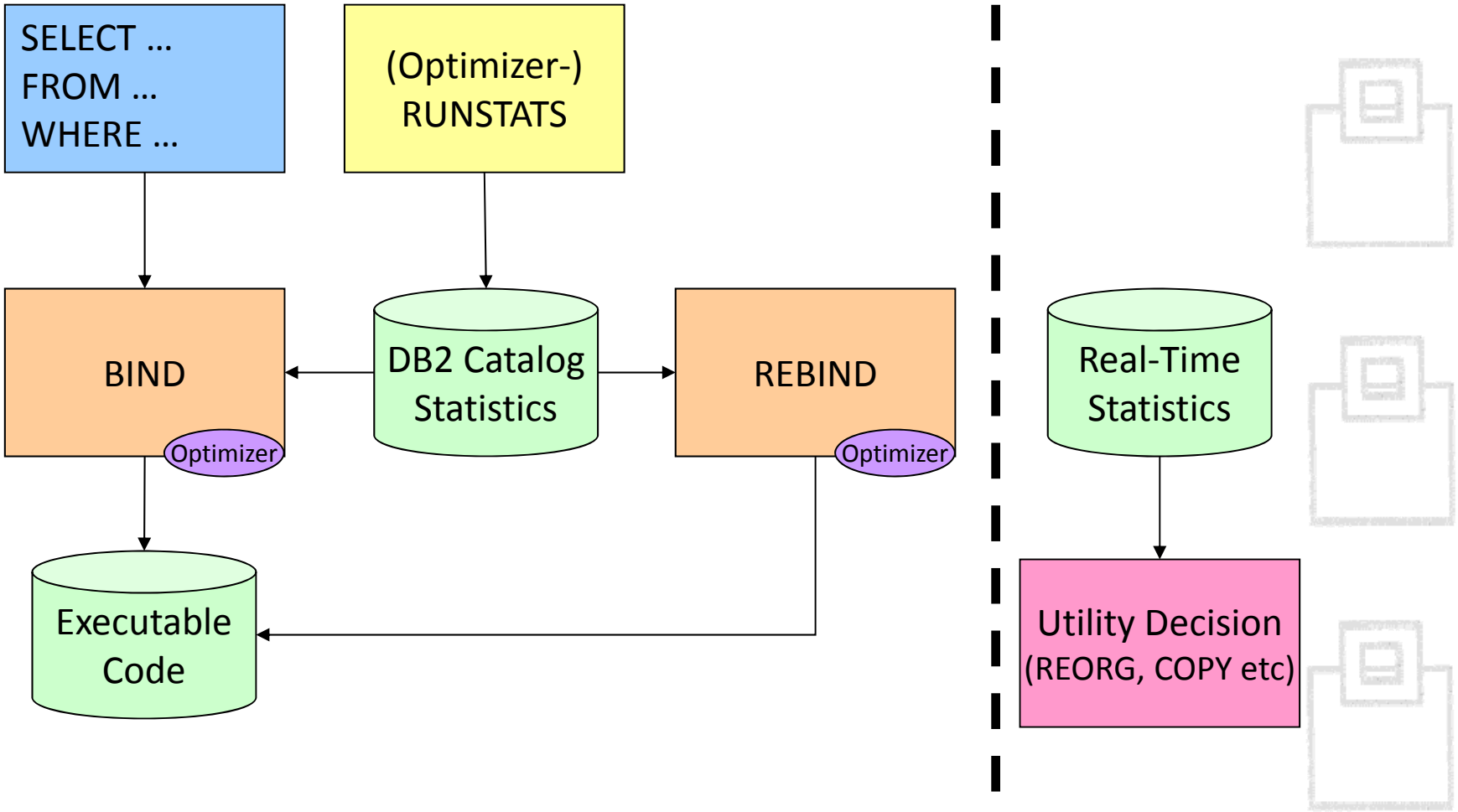
RTS vs. DB2 Catalog Statistics

- Stop using catalog statistics for analytical RUNSTATS
- Use Real-Time Statistics for accurate statistics
- Save CPU Eliminate analytical RUNSTATS
- Eliminate the „lag“ time
- Secure your optimizer base
  - Be careful with autonomic statistics

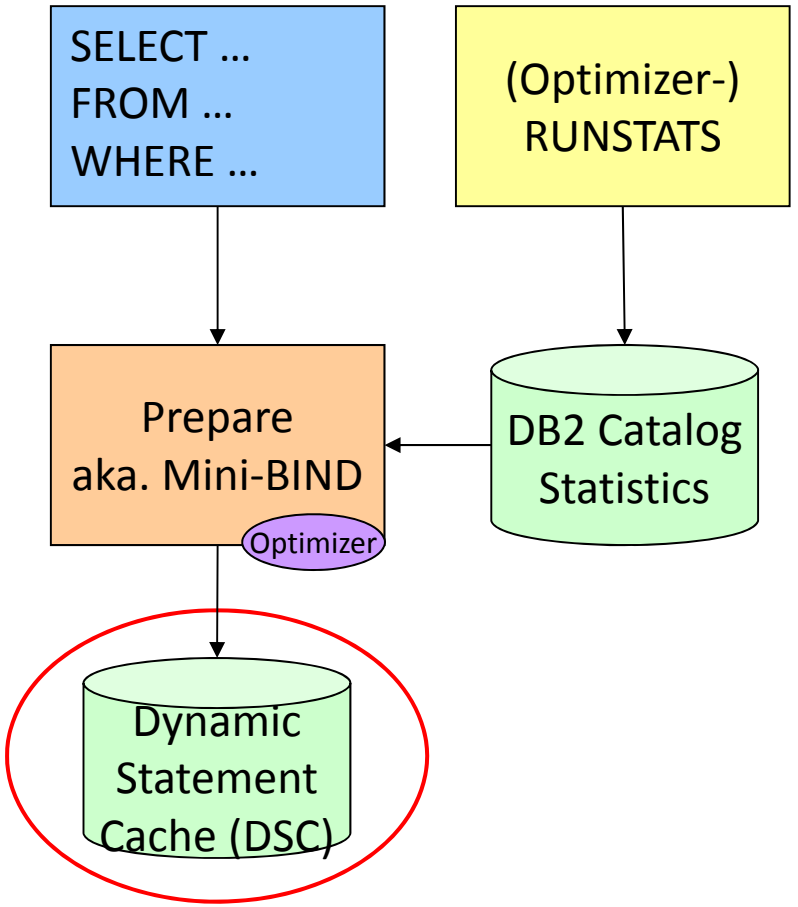




RTS vs. DB2 Catalog Statistics



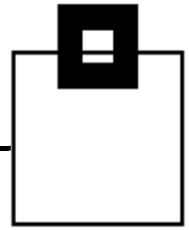
RTS vs. DB2 Catalog Statistics



Access Paths for dynamic SQL are determined on the fly and stored in the DSC.

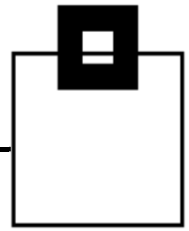
Last Recent Used queue (LRU), RUNSTATS, ALTER, DROP, REVOKE, DB2 RESTART invalidates and flushes the DSC for an object.

Basic RUNSTATS knowledge



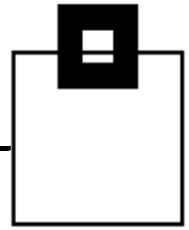
- The RUNSTATS utility
  - Gathers summary information about the characteristics of data in table spaces, indexes and partitions
  - Invalidates the dynamic statement cache
  - Optionally:
    - Reports the statistics
    - Updates the DB2 catalog
    - Updates the DB2 catalog history tables

Basic RUNSTATS knowledge



- Different types of RUNSTATS
  - RUNSTATS tablespace
  - RUNSTATS index
  - REORG / LOAD with Inline RUNSTATS
- Different types of statistics
  - Pure access path statistics
    - Those used by BIND in its process of optimization to determine access path
  - Parallelism access path statistics
    - Those used by BIND in its process of optimization to determine the degree of parallelism
  - Space statistics
    - Those used by the DBA to monitor space usage; to assist in capacity planning; to determine frequency of reorg; etc.

Basic RUNSTATS knowledge



- The RUNSTATS Utility

```
RUNSTATS TABLESPACE <DB>.<TS>
 TABLE(<CR>.<TB>)
   COLGROUP (<CO_A>, <CO_B>)
     FREQVAL COUNT 10 MOST
     HISTOGRAM NUMQUANTILES 100
   TABLESAMPLE SYSTEM/SAMPLE 25
   USE PROFILE
```

TABLE

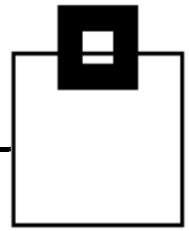

```
 INDEX(ALL)
   KEYCARD FREQVAL NUMCOLS 1 COUNT 10
           FREQVAL NUMCOLS 2 COUNT 10
           FREQVAL NUMCOLS 3 COUNT 10
              HISTOGRAM NUMCOLS 4 NUMQUANTILES 100
   USE PROFILE
```

INDEX


```
   REPORT NO
   UPDATE ALL
   HISTORY NONE
   SET/UPDATE PROFILE
```

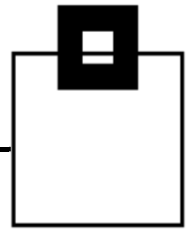
GENERAL

Basic RUNSTATS knowledge



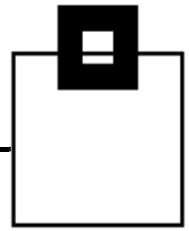
- When to run RUNSTATS
  - After a table is loaded
  - After an index is physically created
  - After running extensive updates, deletions, or insertions in a table space
  - After running any of the following utilities without collecting inline statistics: RECOVER TABLESPACE, REBUILD INDEX, or REORG TS/IX
  - Before running REORG with the OFFPOSLIMIT, INDREFLIMIT, or LEAFDISTLIMIT options
  - After running the ALTER TABLE ROTATE PARTITION statement run RUNSTATS with REORG
  - When the distribution of the data changes
  - When the values over which the data is distributed change

Basic RUNSTATS knowledge



- When to run RUNSTATS
  - Manually
    1. Determine need
    2. Generate/Execute job
  - Automated static
    1. Setup parameters and options
  - Automated dynamic
    1. Setup parameters and options
    2. Setup thresholds and exceptions
  - Autonomic (use DSNTIJRT to setup/configure Stored Procedures (SPs) & User Defined Functions (UDFs))
    1. Setup time window (SYSIBM.SYSAUTOTIMEWINDOWS)
    2. Setup object profiles (SYSTABLES_PROFILES)
    3. Define object in-/excludes (restrict-ts option for SP)

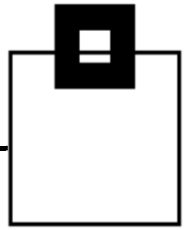
Catalog tables used for access path



- SYSIBM.SYSCOLDIST
- SYSIBM.SYSCOLSTATS *
- SYSIBM.SYSCOLUMNS
- SYSIBM.SYSINDEXES
- SYSIBM.SYSINDEXPART
- SYSIBM.SYSKEYTARGETS 9 and above (same as SYSCOLUMNS)
- SYSIBM.SYSKEYTGTDIST 9 and above (same as SYSCOLDIST)
- SYSIBM.SYSROUTINES
- SYSIBM.SYSTABLES
- SYSIBM.SYSTABLESPACE
- SYSIBM.SYSTABSTATS

* degree of parallelism only and, after APAR PK62804, also „sometimes“ used to bound filter factor estimates.

Columns used for access path decisions



SYSCOLDIST / SYSKEYTGTDIST
CARDF
COLGROUPCOLNO / KEYGROUPKEYNO
COLVALUE / KEYVALUE
FREQUENCYF
HIGHVALUE
LOWVALUE
NUMCOLUMNS / NUMKEYS
QUANTILENO
STATSTIME

SYSCOLUMNS / SYSKEYTARGETS
COLCARDF / CARDF
HIGH2KEY
LOW2KEY
n/a / STATS_FORMAT

SYSCOLSTATS
COLCARD
HIGHKEY
LOWKEY

SYSINDEXES
CLUSTERING*
CLUSTERRATIO
CLUSTERRATIOF
DATAREPEATFACTORF
FIRSTKEYCARDF
FULLKEYCARDF
NLEAF
NLEVELS

SYSINDEXPART
LIMITKEY*

* Columns are not updated by RUNSTATS
_ Columns are not updatable

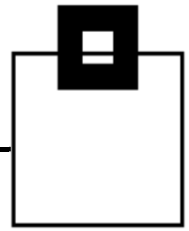
SYSROUTINES
CARDINALITY*
INITIAL_INSTS*
INITIAL_IOS*
INSTS_PER_INVOC*
IOS_PER_INVOC*

SYSTABLES
CARDF
EDPROC*
NPAGES
NPAGESF
PCTROWCOMP

SYSTABLESPACE
NACTIVE
NACTIVEF

SYSTABSTATS
CARD
CARDF
NPAGES

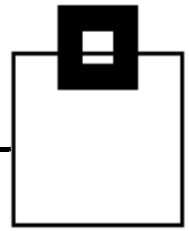
So what?



OK, we now know all the info that DB2 uses to choose access paths. What can we do with this info?

- We can change it to, hopefully, improve an SQL.
- We can delete some of it to, hopefully, improve an SQL.
- We can insert into it to, hopefully, improve an SQL.
- We can ignore it and trust that DB2 „knows what it is doing…“
- We can mess it all up.
- We can use clever add-ons to <u>really</u> mess it all up!
- We can use clever software to see the current state of all these statistics; good, bad, or awful.

IBM Recommendations DB2 V2.3

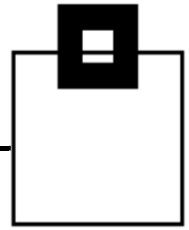


Correlations in the catalog (DB2 Administration Guide)

- Relationships exist among certain columns of certain tables:
  - Columns within SYSCOLUMNS
  - Columns in the tables SYSCOLUMNS and SYSINDEXES
  - Columns in the tables SYSCOLUMNS and SYSCOLDIST (well actually SYSFIELDS in those days!)

- If you plan to update some values, keep in mind the following correlations:
  - COLCARDF and FIRSTKEYCARDF/FULLKEYCARDF
  - COLCARDF, LOW2KEY and HIGH2KEY. For non-default COLCARDF

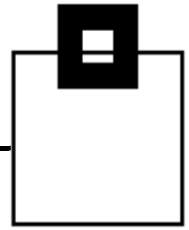
IBM Recommendations DB2 V3



Correlations in the catalog (DB2 Administration Guide)

- No change

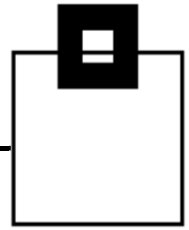
IBM Recommendations DB2 V4



Correlations in the catalog (DB2 Administration Guide)

- No change

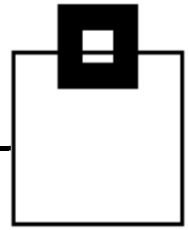
IBM Recommendations DB2 V5



Correlations in the catalog (DB2 Administration Guide)

- Relationships exist among certain columns of certain tables:
  - Columns in the tables SYSCOLUMNS, SYSCOLDIST, and SYSINDEXES
- If you plan to update values, keep in mind the following correlation:
  - CARDF in SYSCOLDIST. CARDF is related to COLCARDF and FIRSTKEYCARDF and FULLKEYCARDF. It must be at minimum:
    - A value between FIRSTKEYCARDF and FULLKEYCARDF if the index contains the same set of columns
    - A value between MAX(colcardf of each col) and the product of all the columns COLCARDFs in the group

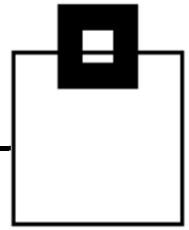
IBM Recommendations DB2 V6



Correlations in the catalog (DB2 Administration Guide)

- No change

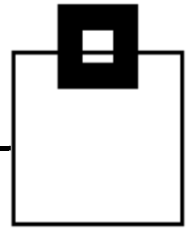
IBM Recommendations DB2 V7



Correlations in the catalog (DB2 Administration Guide)

- No change

IBM Recommendations DB2 V8

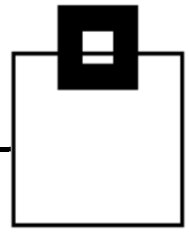


Correlations in the catalog (DB2 Administration Guide)

If you plan to update values, keep in mind the following correlations:

- The COLCARDF, LOW2KEY, and HIGH2KEY gained a note: „If the COLCARDF is 1 or 2 DB2 uses LOW2KEY and HIGH2KEY as domain statistics to generate frequencies“
- CARDF in SYSTABLES. CARDF must be equal or larger than any other cardinalities, such as COLCARDF, FIRSTKEYCARDF, FULLKEYCARDF, and CARDF in SYSCOLDIST
- FREQUENCYF and COLCARDF or CARDF. The number of frequencies collected must be less than or equal to COLCARDF for the column or CARDF for the column group
- FREQUENCYF. The sum of frequencies collected for a column or column group must be less than or equal to 1

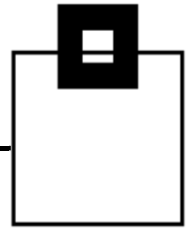
IBM Recommendations DB2 V8



In the „Filter factors and catalog statistics“ chapter (DB2 Administration Guide)

- Recommendation: If query performance is not satisfactory, consider the following actions:
  - Collect cardinality statistics on all columns that are used as predicates in a WHERE clause.
  - Collect frequencies for all columns with a low cardinality that are used as COL op literal predicates.
  - Collect frequencies for a column when the column can contain default data, the default data is skewed, and the column is used as a COL op literal predicate.
  - Collect KEYCARD on all candidate indexes.
  - Collect column group statistics on all join columns.

IBM Recommendations DB2 9

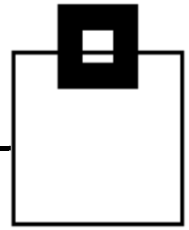


Correlations in the catalog in Chapter 39 of the new book:

„DB2 Performance Monitoring and Tuning Guide“

- No change.

IBM Recommendations DB2 9

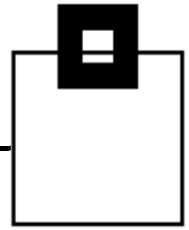


Filter factors and catalog statistics in Chapter 11 of the new :

„DB2 Performance Monitoring and Tuning Guide“

- New section all about HISTOGRAM statistics.

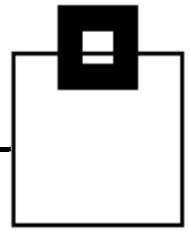
IBM Recommendations DB2 10



Correlations in the catalog in chapter 41 of :
„Managing Performance “

- No change.

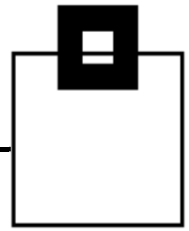
IBM Recommendations DB2 10



Filter factors and catalog statistics in chapter 11 of:
„Managing Performance“

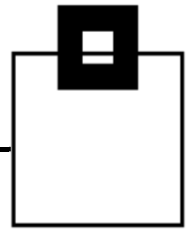
- KEYCARD use deleted. Now it is always active.

SYSCOLDIST contents explained



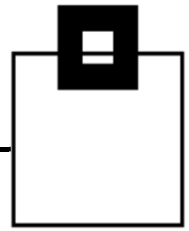
- SYSCOLDIST is used for:
  - Frequencies
  - Cardinalities
  - Histograms in 9 and above
- Column TYPE can contain C, F, H or N
  - If V8 and above „N“ for non-padded frequency values
- Regardless of the TYPE value, columns TBOWNER, TBNAME, NAME, COLGROUPCOLNO, NUMCOLUMNS, and STATSTIME are used for the same purpose.

SYSCOLDIST contents explained



- TBOWNER, TBNAME, and NAME (first column name only) columns are also the non-unique index.
- COLGROUPCOLNO for a single column object is an empty string, for a multi-column object contains a string of two byte SMALLINT, which contains the column numbers from the original table.
- NUMCOLUMNS is the number of columns in this group.
- STATSTIME is the time when RUNSTATS inserted this entry, which is also used when there is a complete duplicate so that DB2 uses the last inserted value.

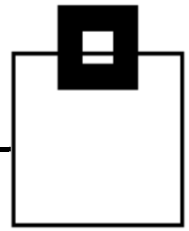
SYSCOLDIST contents explained



For frequency data:

- COLVALUE contains the actual data from the 1 – n columns in the group simply concatenated together.
  - This data might be readable or it might not.
  - Caution must be used when inserting or changing this data as numeric and date, time forms must be the internal DB2 format and not the external format
  - For example, the high bit must be flipped for numbers and watch out for NULLable columns!
- FREQUENCYF contains a floating point value between 0.0 and 1.0 , which is the frequency that this value occurs for this group in the table.

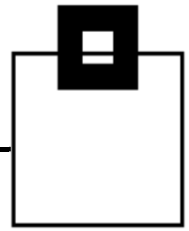
SYSCOLDIST contents explained



For cardinality data:

- CARDF contains a floating point value
- Which is how many different values for this group occur in the data.

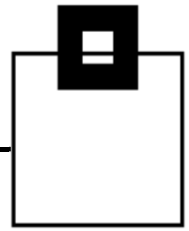
SYSCOLDIST contents explained



For histogram data:

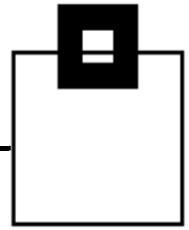
- COLVALUE contains the actual data from the 1 – n columns in the quantile simply concatenated together.
  - This data might be readable or it might not.
  - Caution must be used when inserting or changing this data as numeric and date, time forms must be the internal DB2 format and not the external format
  - For example,the high bit must be flipped for numbers and watch out for NULLable columns!

SYSCOLDIST contents explained



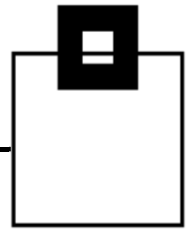
- FREQUENCYF contains a floating point value between 0.0 and 1.0 which is the frequency that this value occurs for this quantile.
- HIGHVALUE and LOWVALUE contain the upper and lower boundaries of this QUANTILENO.
- QUANTILENO is the ordinary number of this quantile (1 - 100)

SYSCOLDIST contents explained



- SYSCOLDIST in a nut shell:
  - Frequencies are good for - COL op literal
  - Histograms are possibly good for - COL op literal
    - But better for range predicates!
  - Cardinalities are good for - everything!

SYSCOLDIST contents explained

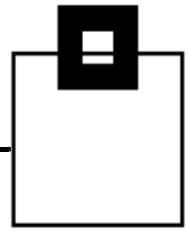


- Example: Importance of cardinality and frequency. If no frequency data existed in the SYSCOLDIST DB2 would assume that all values are equally distributed.
  - Default filter factor would be 1/5 (1/COLCARDF) or 20%

Category	Frequency
Infant	5%
Child	15%
Adolescent	25%
Adult	40%
Senior	15%

This would lead the optimizer to under estimate by 50% for ADULT and to over estimate by 400% for INFANT.

Runstats Q & A

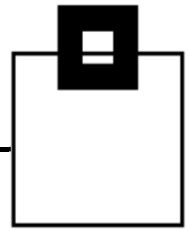


We now know which data is used and where, so now comes a

list of RUNSTATS questions:

- What does FREQVAL NUMCOLS 3 COUNT 10, or COUNT 0 do?
- Use of COLGROUP in DB2 V8
- Use of HISTOGRAM in DB2 9
- Use of SAMPLE & TABLESAMPLE
- Use of REOPT(ONCE)
- Does use of REORG INDEX with inline statistics cause problems?
- What happens to frequencies & cardinalities when not specified in the RUNSTATS utility run?

Runstats Q & A

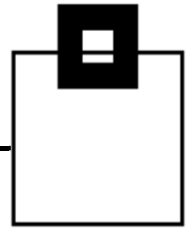


- FREQVAL NUMCOLS 3 COUNT 10
  - NUMCOLS is the number of leading index columns to sample;
    you actually need to provide:

```
RUNSTATS ROYTEST2.ROYTEST2
  TABLE(ALL) INDEX(ALL KEYCARD
  FREQVAL NUMCOLS 1 COUNT 10
  FREQVAL NUMCOLS 2 COUNT 10
  FREQVAL NUMCOLS 3 COUNT 10)
```

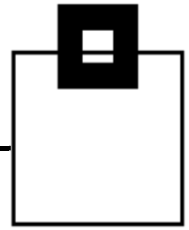
    To get the results you expect!

Runstats Q & A



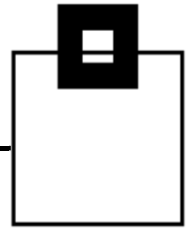
- FREQVAL NUMCOLS 3 COUNT 0
  - Like a „hidden feature“
  - Deletes all of the multi-column frequencies
  - Does NOT delete any entries created by COLGROUP processing in V8.

Runstats Q & A



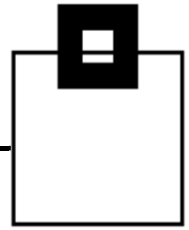
- Use of COLGROUP in DB2 V8
  - Very powerful addition since DB2 V8
  - Completely removes the need for the old DSTATS program
  - DSTATS had serious performance problems and column type limitations, which made its use complex, time consuming, and error prone.

Runstats Q & A



- Use of HISTOGRAM in DB2 9
  - Very powerful addition since DB2 9
  - Should not be used for EVERY table!
  - HISTOGRAM must be weighed up and evaluated on a case by case basis
- Note: If you start to see performance problems then just delete all TYPE H rows from SYSCOLDIST for the relevant TABLES(s)

Runstats Q & A



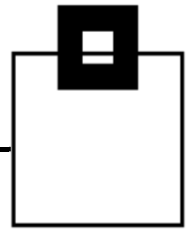
- Use of SAMPLE
  - Only for non-indexed columns
  - Test results show it has an impact on indexed columns as well.
  - Problem with SAMPLE:
    Even using SAMPLE 100, DB2 still gets it wrong…

Col	1	5	10	20	21	22	23	24	25	99	100
TBNAME	62	121	162	178	178	180	180	178	180	180	180
CLNAME	11647	11647	5442	4854	11647	5057	5340	3293	6707	7210	7168

The actual values were 181 and 6871.

Be very careful when using SAMPLE, or just use SYSTEM.

Runstats Q & A

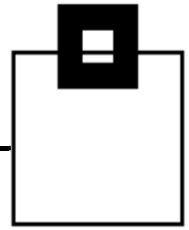


- Use of TABLESAMPLE SYSTEM in DB2 10 and above
  - TABLESAMPLE SYSTEM AUTO recommended for any table space with over 500,000 pages.
  - Note restrictions:
    - Not for LOBs
    - The following is from the Utility Guide:

      „When TABLESAMPLE is specified and the target table space is a multi-table table space or a table space that is segmented, but not partitioned, DB2 runs RUNSTATS with SAMPLE 25 instead of the TABLESAMPLE option.“

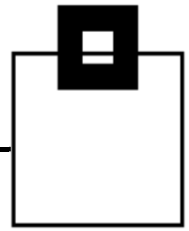
      Clear?

Runstats Q & A



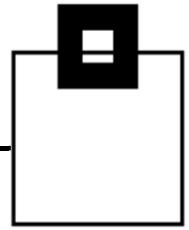
- Use of REOPT(ONCE) in DB2
  - This is a very interesting addition to DB2 as it enables DB2 to do its dynamic SQL mini-bind only once. This can be very good for performance… or not…
  - SAP only: A change that SAP did in ecc5 to use the REOPT(ONCE) could be a CPU killer. The first run SQL might not actually reflect the normal SQL that is executed over the day. The way out of this problem is to actually delete all frequency records from the SYSCOLDIST for the relevant tables.

Runstats Q & A



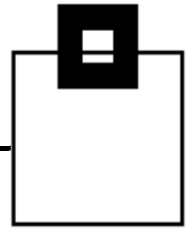
- Does use of REORG INDEX with inline statistics cause problems?
  - YES!
  - The inline stats will *only* update the index statistics and *not* any of the table ones.
  - This leads, very quickly, to the statistics „drifting apart“.
  - Solution: Either run a table space RUNSTATS after an index REORG, manually update the table statistics, or do not use inline statistics.

Runstats Q & A



- What happens to your frequencies, cardinalities and histogram data when you do a RUNSTATS run without using FREQVAL, KEYCARD and/or HISTOGRAM?
  - For frequencies, the leading column information is replaced (and then the COLGROUPCOLNO is correctly set to an empty string) and all the other data stays in the catalog. This can be a good and a bad thing.
    - The cardinalities remains.
    - The histogram data remain.

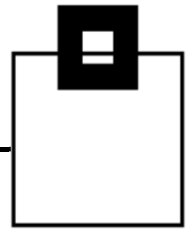
Runstats Q & A



- If you have ever run a RUNSTATS with either FREQVAL or KEYCARD and since then **without** these keywords you probably have „interesting“ data in the SYSCOLDIST.

  → Luckily the latest DB2 version force FREQVAL.

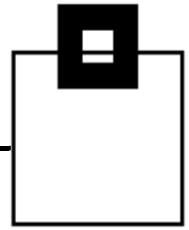
Example: Garbage in the SYSCOLDIST



- A colleague is on holiday and the telephone rings with end-users or management saying that the system performance is bad. You grab your RUNSTATS JCL, submit the job and all is well.

- Or is it?

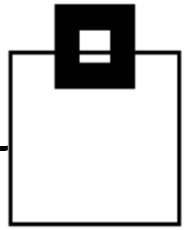
Example: Garbage in the SYSCOLDIST



- Here is the query that you have found:

```
SELECT A.ICTYPE , A.TIMESTAMP
FROM SYSIBM.SYSCOPY A
WHERE A.ICTYPE IN ('D','I','F','S','W','Y','R','X','Z')
  AND A.ICBACKUP NOT IN ('RP','RB')
  AND A.DBNAME  = ?
  AND A.TSNAME  = ?
  AND (A.DSNUM  = ? OR (? > 0 AND A.DSNUM = 0))
  AND A.TIMESTAMP = (SELECT MAX(B.TIMESTAMP)
                     FROM SYSIBM.SYSCOPY B
                     WHERE B.ICTYPE IN
    ('D','I','F','S','W','Y','R','X','Z')
                       AND B.ICBACKUP NOT IN
   ('LB','RP','RB')
                       AND B.DBNAME  = ?
                       AND B.TSNAME  = ?
                       AND (B.DSNUM  = ? OR (? > 0 AND
   B.DSNUM = 0))    )
WITH UR;
```

Example: Garbage in the SYSCOLDIST



- This is the actual RUNSTATS you should run…

```
RUNSTATS TABLESPACE DSNDB06.SYSCOPY
         TABLE(SYSIBM.SYSCOPY)
         COLGROUP(ICBACKUP) FREQVAL COUNT 10
         COLGROUP(ICTYPE) FREQVAL COUNT 1
         COLGROUP(DSNUM) FREQVAL COUNT 1
         SORTDEVT SYSDA
SHRLEVEL CHANGE REPORT YES
```

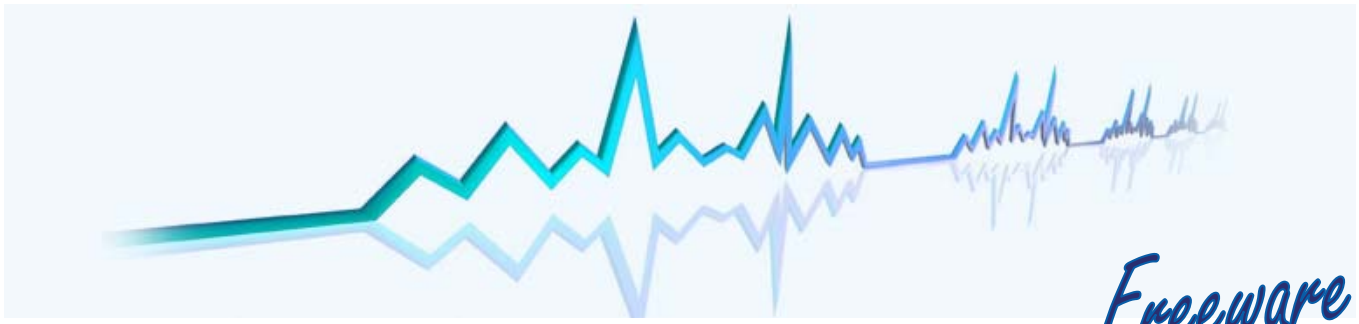
Statistics HealthCheck

- Now that you have learned everything that RUNSTATS does, the question is:
  - How do I know that the statistics I have are correct?
  - The answer is: run **Statistics HealthCheck**
  - How do I get the exact RUNSTATS to fix catalog inconsistencies detected?
  - The answer is: run **Statistics HealthCheck**

Freeware

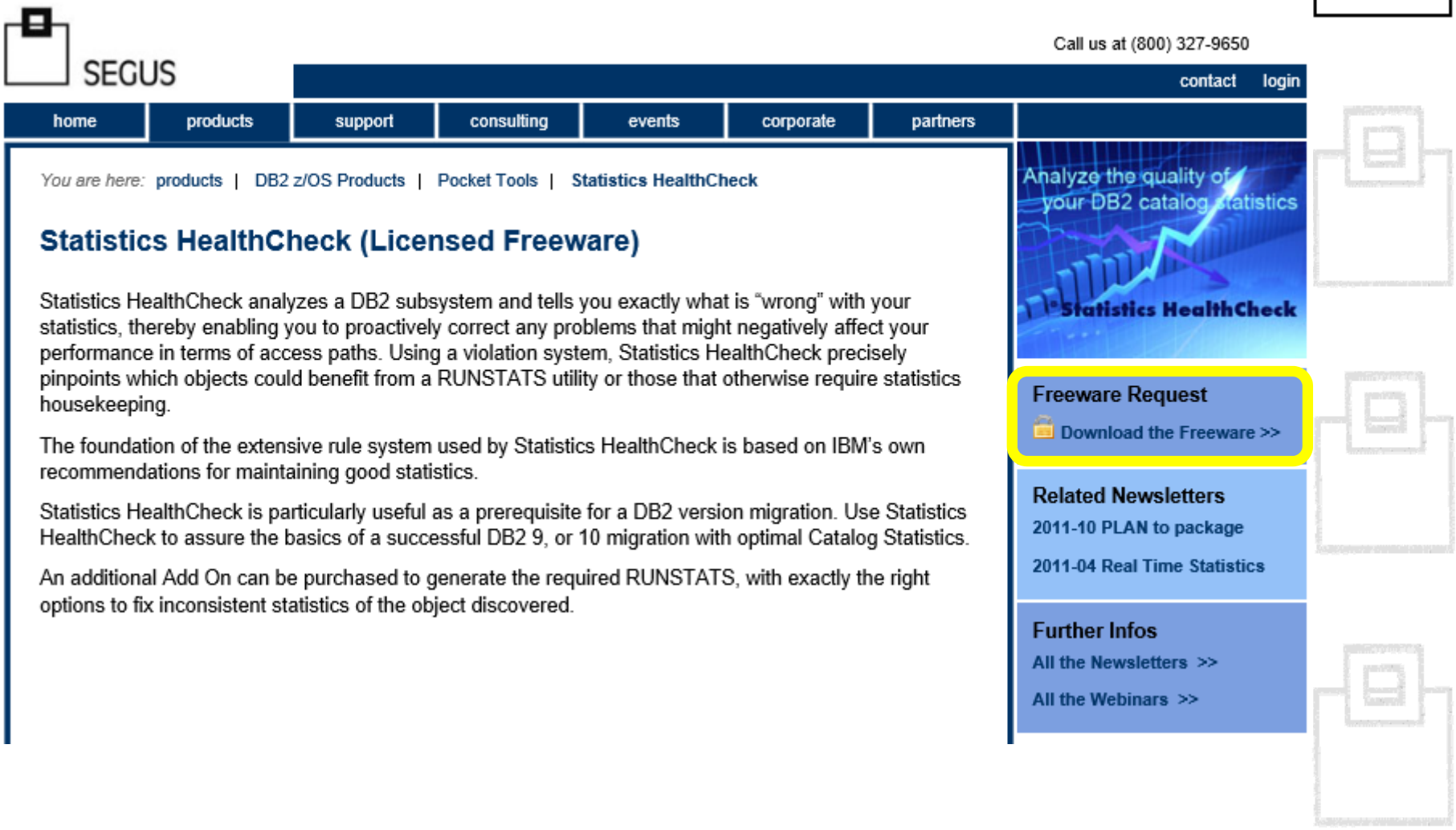
Pocket Tool

Statistics HealthCheck



Request your copy at http://www.segus.com
→ products → DB2 z/OS Products → Pocket Tools → Statistics HealthCheck

Request the freeware @ segus.com



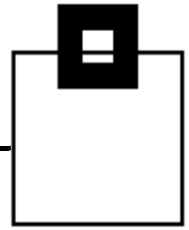
Statistics HealthCheck

- To support regular health checks, we offer a licensed freeware to analyze a DB2 subsystem and tell you exactly what is "wrong" with your statistics, thereby enabling you to proactively correct any problems that might negatively affect your performance in terms of

Statistics HealthCheck

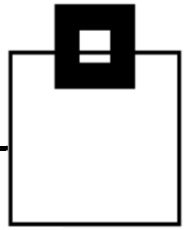
- Quickly, and effectively, analyzes and judges the quality of your entire DB2 catalog
- Knows the heightened sensitivity in DB2 with regard to bad statistics
- All current DB2 versions are supported
  - DB2 9
  - DB2 10
  - DB2 11
- Reporting based on an easy to use batch analysis

Statistics HealthCheck – sample output



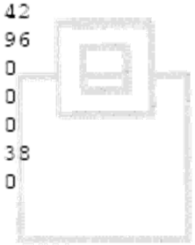
```
Tables read . . . . . : 1286
Tables ignored . . . . . : 0
Tables checked . . . . . : 1286
Perfect tables . . . . . : 1
Indexes checked . . . . . : 1731
Columns checked . . . . . : 22332
Number of critical rule violations . . . . . : 35
Number of serious rule violations . . . . . : 821
Number of warning rule violations . . . . . : 3815
Number of informational rule violations . . . . : 25701
Total number of rule violations . . . . . : 30372
Number of other findings . . . . . : 302

Critical violations
Frequency < 0 . . . . . : 0
Frequency > 100 . . . . . : 0
Frequency count > cardf . . . . . : 0
Frequency count > colcardf . . . . . : 2
Frequency sum < 0 . . . . . : 0
Frequency sum > 100 . . . . . : 0
SYSCOLDIST cardf outside allowable range . . . . : 0
SYSCOLUMNS - low2key high2key empty . . . . . : 4
SYSCOLUMNS aggregate colcardf < SYSCOLDIST cardf : 0
SYSCOLUMNS 1st cardf <> SYSINDEXES firstkeycardf : 0
SYSCOLUMNS 1st cardf <> SYSINDEXES fullkeycardf  : 1
SYSINDEXES 1 col ix firstkeycardf <> fullkeycardf: 4
SYSTABLES cardf < SYSCOLDIST cardf . . . . . : 0
SYSTABLES cardf < SYSCOLUMNS colcardf . . . . . : 18
SYSTABLES cardf < SYSINDEXES firstkeycardf . . . : 0
SYSTABLES cardf < SYSINDEXES fullkeycardf . . . : 1
SYSTABLES no RUNSTATS . . . . . : 5
```

Statistics HealthCheck – sample output

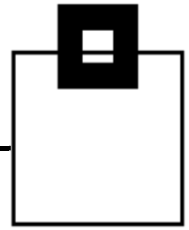


```
Serious violations
Missing leading column index frequencies . . . . : 0
Missing multi-column index cardinalities . . . . : 820
SYSCOLDIST cardf group <> SYSINDEXES fullkeycardf: 0
SYSCOLDIST cardf sub-set > super-set . . . . . : 0
SYSCOLUMNS - ALTERs . . . . . : 0
SYSINDEXES fullkeycard < SYSINDEXES firstkeycard : 0
SYSTABLES cardf <> unique index fullkeycardf . . : 1

Informational violations
Number of index parts <> number of parts . . . . : 0
SYSCOLDIST colgroup with a default colcardf . .  : 0
SYSCOLDIST colno > number of columns . . . . . : 0
SYSCOLDIST duplicate cardinalities . . . . . : 0
SYSCOLUMNS - low2key > high2key . . . . . : 1
SYSCOLUMNS - SYSINDEXES statstime not equal . .  : 4456
SYSINDEXPART - SYSINDEXES statstime not equal .  : 1735
SYSINDEXPART statistics missing . . . . . : 0
SYSTABLES - SYSCOLUMNS statstime not equal . . . : 19470
SYSTABLES - SYSINDEXES statstime not equal . . . : 19
SYSTABLES - SYSTABLESPACE statstime not equal .  : 0
SYSTABLES - SYSTABSTATS statstime not equal . .  : 0
SYSTABLES cardf < sum of SYSINDEXPART cardf . .  : 0
SYSTABLES cardf > sum of SYSINDEXPART cardf . .  : 0
SYSTABLES cardf < SYSINDEXPART cardf . . . . . : 20
SYSTABLESPACE - SYSTABSTATS no. parts not equal  : 0

Other findings
SYSCOLDIST not used in index cardinality records : 0
SYSCOLDIST not used in index frequency records . : 10
SYSINDEXES clustering = Y and clustered = N . .  : 292
```

```
Warning violations
Missing multi-column index frequencies . . . . . : 3635
SYSCOLDIST large differences found in STATSTIME  : 4
SYSCOLDIST statistics obsolete . . . . . : 42
SYSROUTINES with default statistics . . . . . : 96
SYSTABLES - SYSTABSTATS sum cardf not equal. . . : 0
SYSTABLES - SYSTABSTATS sum npages not equal . . : 0
SYSTABLES default cardf . . . . . : 0
SYSTABLES statistics obsolete . . . . . : 38
SYSTABLESPACE - SYSTABSTATS sum nactive not equal: 0
```

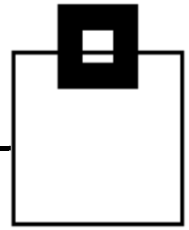


Statistics HealthCheck



- The Pocket Tool add-on additionally generates RUNSTATS with exactly the right options to fix detected inconsistencies:
  - Considering INDEX, TABLESPACE, TABLE, COLUMN
  - Considering COLGROUPS
  - Considering FREQVAL NUMCOLS
  - Considering HISTOGRAM STATISTICS

Conclusion



- RUNSTATS enhancements in DB2 V8, 9, and 10 open up significant access path improvements
- Be careful with your DB2 Catalog statistics

  → Garbage in - Garbage Out!
- Don't forget to REBIND in order to exploit the new DB2 version that you paid for
- Real-time statistics provide current statistics, instantly without RUNSTATS

  → Good for tasks like threshold-based utilities

  → Statistics for volatile tables without compromising access paths

  → Provides individual data for clone and base table