

Db2 for z/OS: Lies, Damn lies and Statistics

Roy Boxwell

SEGUS & SOFTWARE ENGINEERING GmbH

Session code: A18

05.10.2017, 11:00

Platform: Db2 for z/OS



Agenda

- Quotes
- Db2 RUNSTATS basics
- IBM recommendations through the ages
- Db2 RUNSTATS advanced
 - SYSCOLDIST explained
- RUNSTATS real world Q&A
- RUNSTATS reversal

Quotes

"Lies, damn lies, and statistics" – Benjamin Disraeli,
Prime Minister of England (1868, 1874-1880)

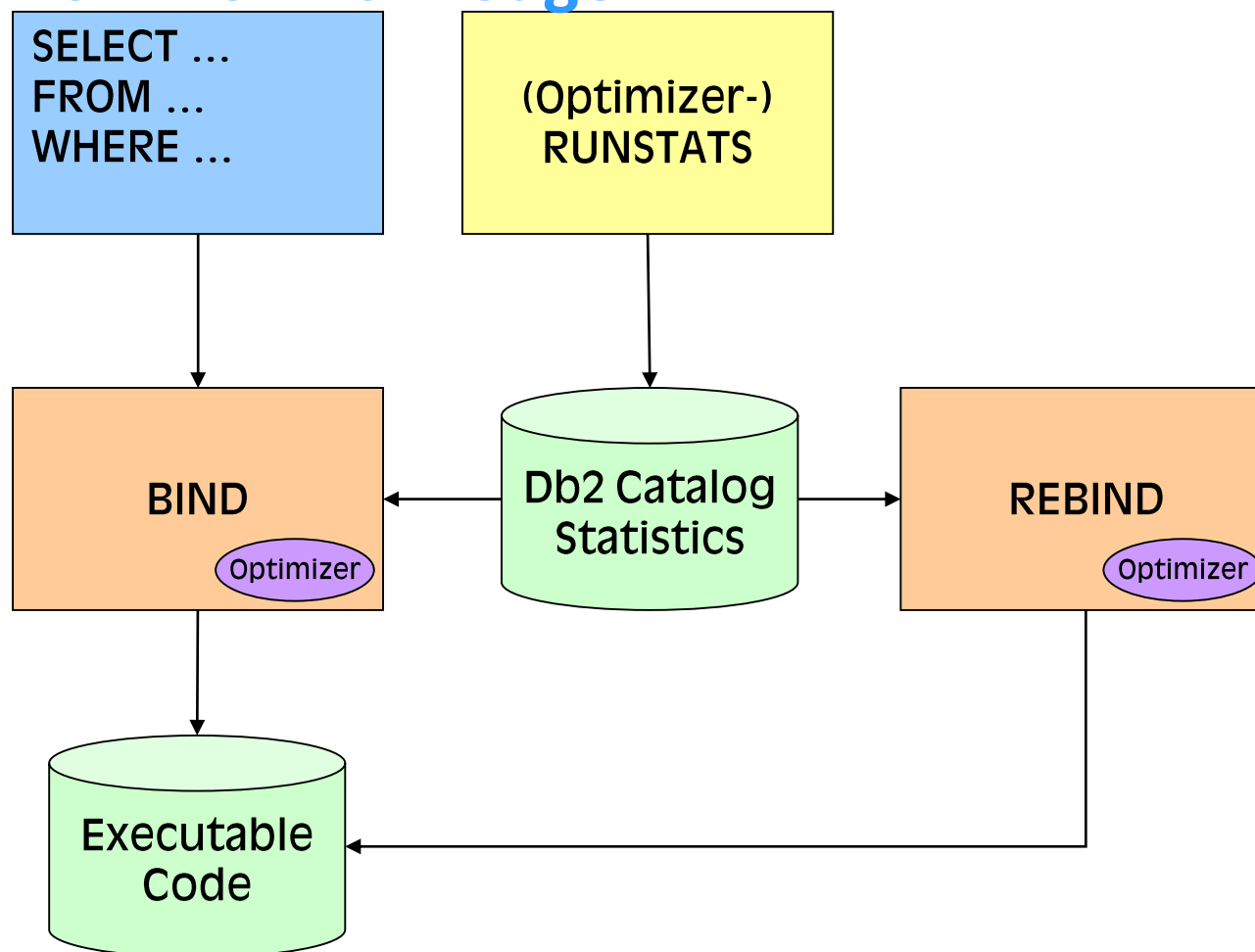
The above line may, or may not, have been spoken well over 100 years ago, but the need for statistics and, above all else, accurate statistics is more important than ever in the Db2 world of today.

Quotes

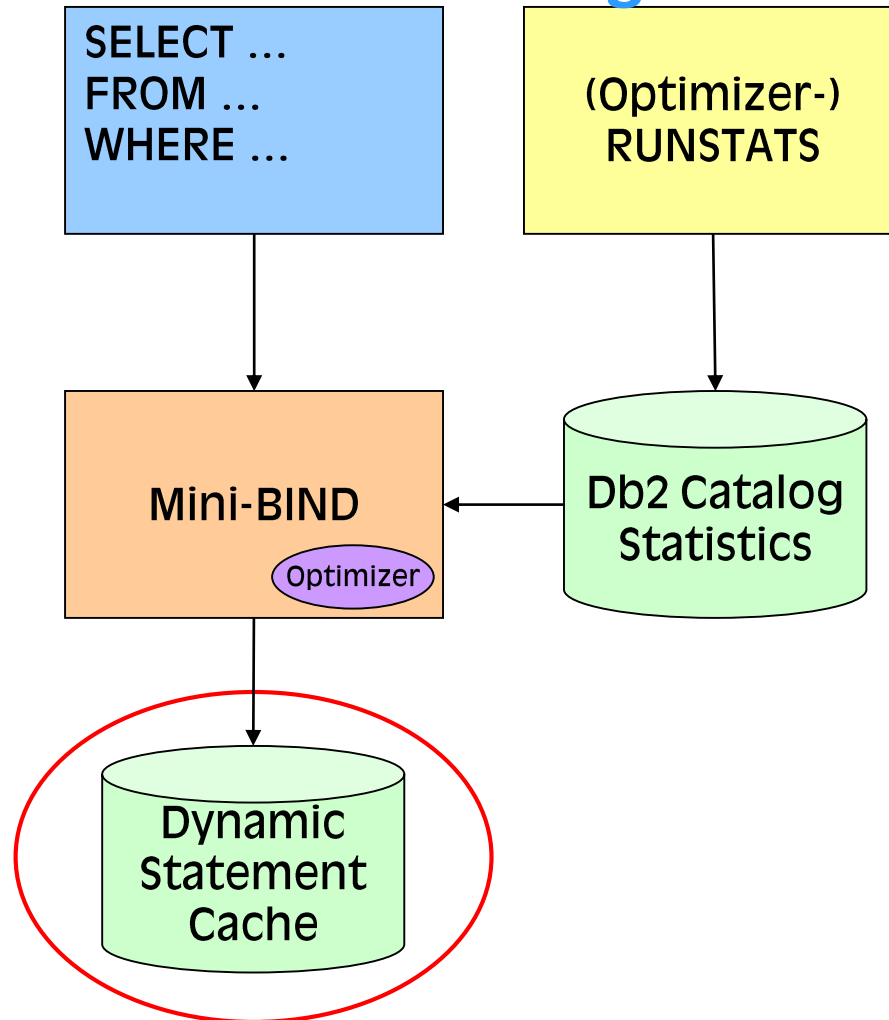
“All is mystery; but he is a slave who will not struggle to penetrate the dark veil.” – Benjamin Disraeli,
Prime Minister of England (1868, 1874-1880)

I guess he knew about the Db2 Optimizer as well!

Basic RUNSTATS knowledge



Basic RUNSTATS knowledge



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

LRU, Special RUNSTATS, ALTER, DROP, REVOKE, Db2 RESTART all 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
 - Optionally:
 - Invalidates the dynamic statement cache (It used to *always* do this up to Db2 12)
 - 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 an access path
 - Parallel 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

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

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

```
SHRLEVEL CHANGE
```

```
REPORT NO
```

```
UPDATE ALL
```

```
HISTORY NONE
```

```
INVALIDATECACHE YES
```

Catalog tables used for access path

- SYSIBM.SYSCOLDIST
- SYSIBM.SYSCOLSTATS Degree of parallelism only and, after APAR PK62804, also „sometimes“ used to bound filter factor estimates.
- SYSIBM.SYSCOLUMNNS
- SYSIBM.SYSINDEXES
- SYSIBM.SYSINDEXPART
- SYSIBM.SYSKEYTARGETS (same as SYSCOLUMNNS)
- SYSIBM.SYSKEYTGTDIST (same as SYSCOLDIST)
- SYSIBM.SYSROUTINES
- SYSIBM.SYSTABLES
- SYSIBM.SYSTABLESPACE
- SYSIBM.SYSTABSTATS

Columns used for access path decisions

SYSCOLDIST / SYSKEYTGTDIST

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

SYSCOLUMNS / SYSKEYTARGETS

COLCARDF (-1) 25/ CARDF (-1)
HIGH2KEY
LOW2KEY
n/a / STATS_FORMAT

SYSCOLSTATS

COLCARD
HIGHKEY
LOWKEY

SYSINDEXES

CLUSTERING*
CLUSTERRATIO (0)
CLUSTERRATIOF (0 use CR)
DATAPEATFACTORF (-1)
FIRSTKEYCARDF (-1) 25
FULLKEYCARDF (-1) 25
NLEAF (-1) Systables
CARD/300
NLEVELS (-1) 2

SYSINDEXPART

LIMITKEY*

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

SYSROUTINES

CARDINALITY* (-1) 10,000
INITIAL_INSTS* (-1) 40,000
INITIAL_IOS* (-1) 0
INSTS_PER_INVOC* (-1) 4,000
IOS_PER_INVOC* (-1) 0

SYSTABLES

CARDF (-1) 10,000
EDPROC*
NPAGES (-1)
CEILING (1 + CARD/20)
NPAGESF (-1 use NPAGES)
PCTROWCOMP

SYSTABLESPACE

NACTIVE (0)
CEILING(1 + CARD/20)
NACTIVEF (0 use NACTIVE)

SYSTABSTATS

CARDF (-1) 10,000
NPAGES (-1)
CEILING(1 + CARD/20)

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 simply trust that Db2
„knows what it is doing...”
- We can mess it all up.
- We can use clever software to see the current state of all these
statistics; good, bad, or awful.
- We can use even cleverer software to rescue us from bad stats

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 V3.1

Correlations in the catalog (DB2 Administration Guide)

- No change

IBM Recommendations DB2 V4.1

Correlations in the catalog (DB2 Administration Guide)

- No change

IBM Recommendations DB2 V5.1

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 V6.1

Correlations in the catalog (DB2 Administration Guide)

- No change

IBM Recommendations DB2 V7.1

Correlations in the catalog (DB2 Administration Guide)

- No change

IBM Recommendations DB2 V8.1 (1|2)

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

IBM Recommendations DB2 V8.1 (2|2)

Correlations in the catalog (DB2 Administration Guide)

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

- 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 V8.1

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 36 of the new book:
„DB2 Performance Monitoring and Tuning Guide“

- No change.

IBM Recommendations DB2 9

Filter factors and catalog statistics in Chapter 33 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 37 of:
„Managing Performance“

- No change.

IBM Recommendations DB2 10

Filter factors and catalog statistics in chapter 34 of:
„Managing Performance“

- KEYCARD use deleted. Now it is always active.

IBM Recommendations Db2 11

Correlations in the catalog in chapter 37 of:
„Managing Performance“

- No change.

IBM Recommendations Db2 11

Filter factors and catalog statistics in chapter 34 of:
„Managing Performance“

- SYSSTATFEEDBACK usage introduced for missing or conflicting statistics.

IBM Recommendations Db2 12

Correlations in the catalog in chapter 37 of:
„Managing Performance“

- No change.

IBM Recommendations Db2 12

Filter factors and catalog statistics in chapter 34 of:
„Managing Performance“

- SYSSTATFEEDBACK got one new reason - „STALE“.

SYSCOLDIST contents explained

- SYSCOLDIST is used for:
 - Frequencies
 - Cardinalities
 - Histograms
- Column TYPE can contain:
 - „C“ for cardinality values
 - „F“ for frequency values
 - „N“ for non-padded frequency values
 - „H“ for histogram 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.
- COLGROUPOCOLNO 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

For histogram data:

- 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
 - Cardinalities are good for: Everything!

Remember that a frequency without a cardinality is like a warm beer. Nice to look at, perhaps, but the optimizer will not touch it!

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 $1/5$ ($1/\text{COLCARD}$) 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 real world Q&A

We now know which data is used and where, so now comes a list of RUNSTATS questions:

- `FREQVAL NUMCOLS 3 COUNT 10` – What does this do?
- `FREQVAL NUMCOLS 3 COUNT 0` – What does this do?
- Use of `COLGROUP`
- Use of `HISTOGRAM`
- 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  
    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

Runstats Q & A

- Use of COLGROUP
 - Very powerful addition to basic RUNSTATS
 - Can lead to serious performance (PREPARE) problems if overdone!
 - Aim for less than a hundred COLGROUPs

Runstats Q & A

- Use of HISTOGRAM
 - Another very powerful addition to RUNSTATS
 - 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.

Runstats Q & A

- Use of TABLESAMPLE SYSTEM

TABLESAMPLE SYSTEM AUTO recommended for any table space with over 500,000 pages but not for LOBs.

The following is from the Utility Guide:

„When RUNSTATS TABLESPACE TABLE is run on a multi-table table space or a table space that is segmented and not partitioned, the TABLESAMPLE keyword is not used. Instead, all pages are scanned to collect statistics.“

Clear? Simply put – UTS only!

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... if the SQL of interest is first executed with atypical literal values then you get sub-optimal response times on all subsequent executions.
 - It is always a good idea to run a boot-strap SQL program after member restart to insert „good“ SQLs into the local DSC.

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 or HISTOGRAM?
 - For frequencies, the leading column information is replaced (and then the COLGROUPOCOLNO 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 cardinality data remains.
 - The histogram data remains.
- If you have ever run a RUNSTATS with FREQVAL or HISTOGRAM and since then without then you probably have „interesting“ data in the SYSCOLDIST.

Runstats Q & A

- What about PROFILE usage?

You must set up your profiles first and then you can happily update them with data thus making the RUNSTATS syntax easier.

However, if you have enabled the auto-update of profiles, which is **on** by default, then you must make sure that your profiles are not growing out of control!

All profile data lives in the SYSIBM.SYSTABLES_PROFILES table.

Runstats Q & A

- What about PROFILE usage?

The way you set up your system is to first populate your profile table by using the „dummy“ RUNSTATS syntax SET PROFILE and then either FROM EXISTING STATS *or* include normal COLGROUPs etc. in the syntax, both are not supported!

From then on, just adding USE PROFILE with/without the INCLUDE(NPI) syntax, and no other columns, colgroups, freqs etc., is all you need.

If no profile is found you will get an automatic

COLUMN ALL INDEX ALL

style runstats.

Runstats Q & A

- What about PROFILE usage?

Naturally you can also use the UPDATE PROFILE syntax to update the stored profiles. Again this is a „dummy“ RUNSTATS with the new desired columns, COLGROUPs, freqs etc.

Finally, if you are fed up with the Profile you can use the DELETE PROFILE syntax to clean up. Once again this is a „dummy“ RUNSTATS and no colgroup defs etc. are allowed in the syntax.

Runstats Q & A

- **Applying statistics recommendations to statistics profiles automatically**

This is new in Db2 12. To enable this feature you do nothing as it is, by default on. This is, in my opinion, a bad thing!

ZPARM STATFDBK_SCOPE set to ALL by default

ZPARM STATFDBK_PROFILE set to YES by default

SYSIBM.SYSTABLES column STATS_FEEDBACK set to Y by default

Out-of-the box it starts automatically creating (for TYPE='C' with NUMCOLS > 1 and TYPE='F' or 'H') and updating profiles...

Runstats Rescue (1 | 4)

Problem: An ill-timed RUNSTATS has killed at least one „key player“ SQL in production.
Telephones are ringing...

What can you do??

Runstats Rescue (2 | 4)

Our solution:

Schedule RunstatsRescue-batch job to maintain a history of optimizer relevant statistics (using a GDG).

If (when!) a (dynamic) SQL statement performs badly:

- Point RunstatsRescue to the STMT
 - RunstatsRescue shows the associated tablespaces/indexspaces for statistics recovery
- Specify since when the SQL access path degraded
 - RunstatsRescue checks if a RUNSTATS was executed since then and shows the details per object
 - RunstatsRescue verifies potential object (re-) creation in the timeframe

Runstats Rescue (3 | 4)

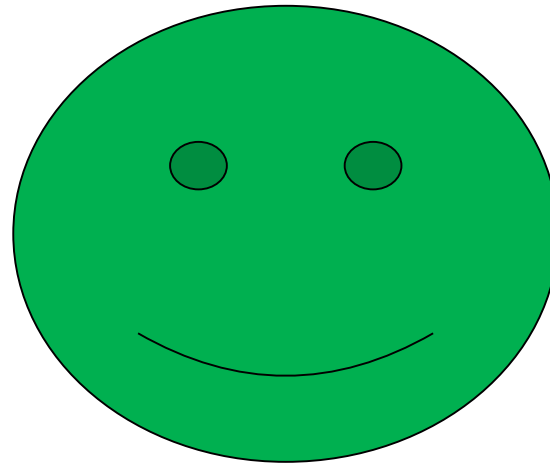
Our solution:

RunstatsRescue then generates a job to

- Restore the statistics from it's repository
- Flush the dynamic statement from the cache
 - If static SQL it creates a REBIND

Runstats Rescue (4|4)

Typically from first call to resolution is less than 10 minutes.



Conclusion

- Be careful with your Db2 Catalog statistics
→ Garbage In - Garbage Out!
- Guard your access paths as if they were gold!
- Control the number and complexity of COLGROUPs – It impacts PREPARE time remember!
- Plan a way to get your old statistics back as quickly as you can!

Roy Boxwell
SEGUS & SOFTWARE ENGINEERING GmbH
r.boxwell@seg.de

Session code: A18

*Please fill out your session
evaluation before leaving! or session
evaluation before leaving!*

