

Statistics@KBC

Dirk Beauson

Special thx to Peter Cornelissen for the recent RUNSTATS study and for preparing the slides



Statistics @ KBC

- Why do we run RUNSTATS ?
- RUNSTATS Process
- RUNSTATS and Utilities
- Release Process Issues
- Miscellaneous

Statistics @ KBC

- Why do we run RUNSTATS ?
- RUNSTATS Process
- RUNSTATS and Utilities
- Release Process Issues
- Miscellaneous

Statistics @ KBC

- Why do we run RUNSTATS ?
- RUNSTATS Process
 - Daily Process
 - Exclude Table
- RUNSTATS and Utilities
- Release Process Issues
- Miscellaneous

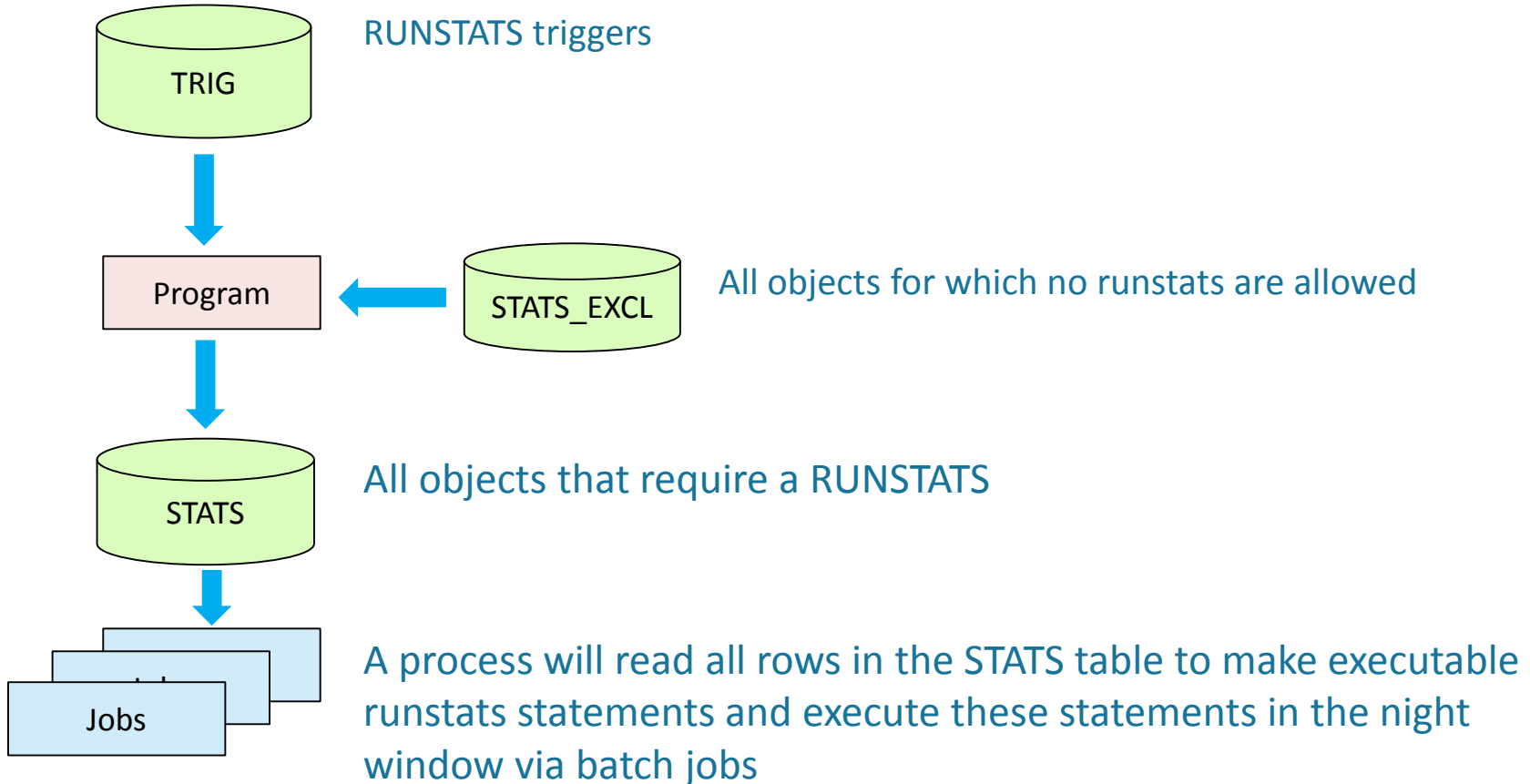
Runstats Process @ KBC (1)

- We daily run an application to select objects who need RUNSTATS

And run the RUNSTATS for those objects

- Default : TABLE ALL SAMPI  INDEX ALL
- We use our own triggers  DSNACCOX :
 - More than 20% of data CHANGED
 - Last STATSTIME older than 75 days
- We can also EXCLUDE objects

Runstats Process @ KBC (2)



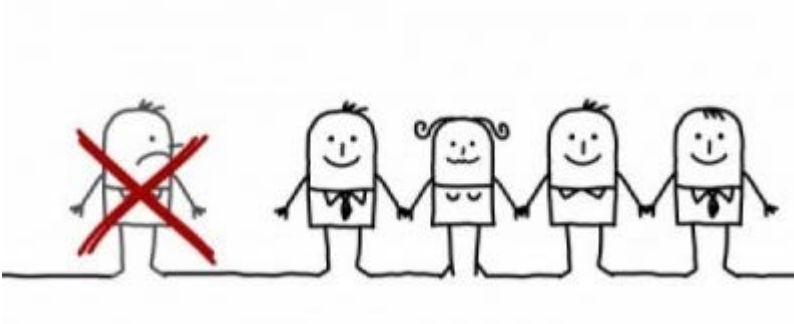
DB2V11 and re-evaluation KBC RUNSTATS Triggers

- One of the recommendations by the roll out of DB2 11 was a mass rebind of all the packages V9 or before to V10 before migration from 10 to 11 (APPLCOMPAT zparm : New SQL functionality available in V11 NFM cannot be used until package is bound with APPLCOMPAT value of V11 R1)
- Before running the REBINDs,
we checked the last STATSTIME of all the objects
and ran runstats on all the tablespaces
where the last STATStime was more than 75 days ago.
- After that we did the mass rebinds

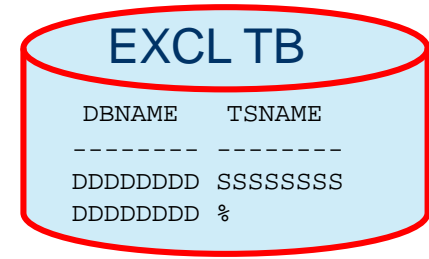
On that moment we saw that our process of taking runstats was not optimal, because a lot of STATSTIMEs of objects were not up to date.

So we decided to add the 75 days trigger.

EXCLUDE Table Usage



EXCLUDE Table Usage (1)



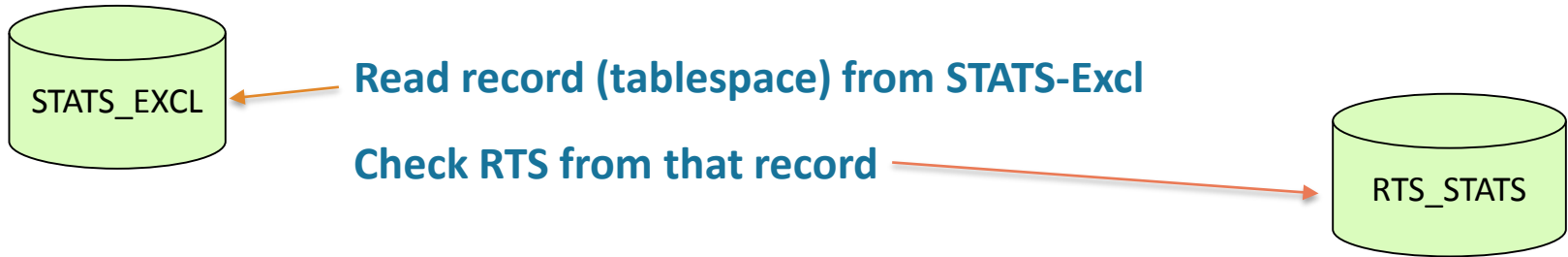
The diagram shows a cylinder representing a table named 'EXCL TB'. Inside the cylinder, there is a header section with 'DBNAME' and 'TSNAME' separated by a dashed line. Below the header, there are three rows of data: 'DDDDDDDD SSSSSSSS', 'DDDDDDDD %', and 'DDDDDDDD %'.

DBNAME	TSNAME
DDDDDDDD	SSSSSSSS
DDDDDDDD	%
DDDDDDDD	%

- STATS_EXCL Table contains :
 - all the new tablespaces that were created **empty** in a release
 - All critical tables that may not be runstats (flip-flop, volatile tables, etc)
- These tables will be excluded in the daily runstats jobs
- These tables will also be handled differently if we prepare a BMC Change Manager worklist
- The purpose is
 - to keep this table as small as possible
 - to remove records on this table as quickly as possible

EXCLUDE Table Usage (2)

We have an automatic Cleanup process that :



If tablespace contains a predefined number of records, then the tablespace will be deleted on STATS_EXCL.

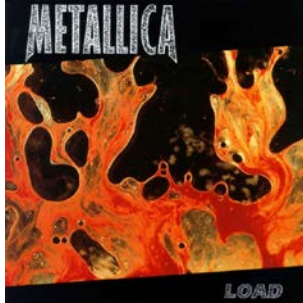
After that, the tablespace will then be runstats in the weekly process.

Statistics @ KBC

- Why do we run RUNSTATS ?
- RUNSTATS Process
- **RUNSTATS and Utilities**
- Release Process Issues
- Miscellaneous

RUNSTATS and UTILITIES

LOAD



REORG



To RUNSTATS or not to RUNSTATS

FLIP/FLOP



RUNSTATS and Utilities

- **LOAD**
- FLIP/FLOP Designs
- REORG

RUNSTATS & LOAD (1)

Standard LOAD process :

Operational tables : full load and new statistics

- only loads during the release moments
- if the impact on operational processes is small or nihil



```
LOAD DATA
RESUME NO
SHRLEVEL NONE
REPLACE
COPYDDN TEMPLIC
STATISTICS TABLE(ALL)
SAMPLE 25
INDEX(ALL) KEYCARD
REPORT NO
UPDATE ALL
```

RUNSTATS & LOAD (2)

Standard LOAD process :

Warehouse (Month/Day) tables : full load and new statistics

- Daily or Monthly full load

```
LOAD DATA
RESUME NO
SHRLEVEL NONE
REPLACE
COPYDDN TEMPLIC
STATISTICS TABLE(ALL)
SAMPLE 25
INDEX(ALL) KEYCARD
REPORT NO
UPDATE ALL
```


RUNSTATS and Utilities

- LOAD
- FLIP/FLOP Designs
- REORG

RUNSTATS & FLIP/FLOP Designs (1)



FLIP/FLOP Designs :

- 2 or more Tables or a Partitioned Table
 - 2 versions : Active/Passive table
 - 8 versions : Weekly wrap around
 - ...
- Starts empty
- Always available
- Every day we use one set of tables or one set of partitions and we delete the data on the set of tables or set of partitions that will be used tomorrow

RUNSTATS & FLIP/FLOP Designs (2)

FLIP/FLOP Designs and RUNSTATS :

- At Create Time all objects get good statistics
- Insert Objects in EXCLUDE table
- Mass deletes via DUMMY LOAD **while maintaining statistics**

LOAD DATA

RESUME NO
SHRLEVEL NONE
REPLACE
COPYDDN TMPLOC
KEEPDICTIONARY
LOG NO
DISCARDS 0
ENFORCE CONSTRAINTS
INTO TABLE XXXX.YYYYYY

**NO RUNSTATS !!!
KEEPDICTIONARY !!!**

RUNSTATS and Utilities

- LOAD
- FLIP/FLOP Designs
- REORG
 - STD REORG Process
 - EXCLUDE Table

RUNSTATS & REORG (1)

Standard REORG process (used only in BMC Change Manager Worklist) :

```
REORG TABLESPACE DDDDDDDD.SSSSSSSS  
LOG NO  
UNLDDN SYSREC  
UNLOAD CONTINUE  
WORKDDN (SYSUT1,SORTOUT)  
STATISTICS TABLE (ALL) SAMPLE 25 INDEX (ALL) KEYCARD
```

Standard ONLINE REORG process :

```
REORG TABLESPACE LIST COPYLST  
COPYDDN COPYDS  
WORKDDN WORKDS  
UNLDDN RECDS  
SHRLEVEL(CHANGE)  
DEADLINE CURRENT TIMESTAMP +4 HOURS  
DRAIN_WAIT 12 RETRY 2 RETRY_DELAY 120  
MAXRO 12 LONGLOG TERM DELAY 1200 TIMEOUT TERM  
STATISTICS TABLE (ALL) SAMPLE 25 INDEX (ALL) KEYCARD
```



RUNSTATS & REORG (2)

Way of working during release in Worklist BMC Change Manager

If a Table is registered in the EXCLUDE-table, the goal is to maintain the current stats.

Therefore, reorgs used during a release moment in a Worklist will be manipulated to do NO RUNSTATS.

```
-REOR 000500
      REORG TABLESPACE DDDDDDDD.SSSSSSSS
      UNLDDN SYR1001
          COPYDDN (SYCL0001)
      STATISTICS
          HISTORY NONE
          INDEX (ALL)
          TABLE ALL
          UPDATE NONE
          REPORT YES
      SORTKEYS
      SORTDATA LOG NO
      SHRLEVEL REFERENCE
-SYNC 000550 REORG OF TABLESPACE DDDDDDDD.SSSSSSSS COMPLETE
```

RUNSTATS & REORG (3)

Way of working outside Worklist BMC Change Manager

Same condition as above

```
OPTIONS TEMPLATEDD ODATEMPL
LISTDEF COPYLST
  INCLUDE TABLESPACE DDDDDDDD.SSSSSSSS PARTLEVEL
--
  QUIESCE LIST COPYLST
--
  REORG TABLESPACE LIST COPYLST
    COPYDDN COPYDS  WORKDDN WORKDS  UNLDDN RECDS
    SHRLEVEL(CHANGE)
    DEADLINE CURRENT TIMESTAMP +4 HOURS
    DRAIN_WAIT 12 RETRY 2 RETRY_DELAY 120
    MAXRO 12 LONGLOG TERM DELAY 1200 TIMEOUT TERM
    STATISTICS TABLE (ALL) SAMPLE 25 INDEX (ALL) KEYCARD
    UPDATE NONE REPORT YES
```

Statistics @ KBC

- Why do we run RUNSTATS ?
- RUNSTATS Process
- RUNSTATS and Utilities
- Release Process Issues
- Miscellaneous

Release Process Issues (1)

As we said in the intro..... one of our major concerns is to have good access paths for the queries in our programs. That's why it is important to have good statistics.

For the existing tables, that is no problem



But.....

- What with new tables ?
- What with a new column that stays empty, but will be used in an index?

Because, we see a lot of projects that start with empty tables/empty columns ...

Release Process Issues

Ways of working :

- Copying Data from Test to Accept and Production
- Migrate Statistics from Test to Accept and Production
- Manipulating the DB2 Catalog

Working with test data (1)

TEST



Unload data on test ENVIRONMENT

ACC/PRD



Create new objects



Load testdata on new objects (RUNSTATS)



Dummy Load



Insert new tables in STATS_EXCL



Bind packages

Working with test data(2)

- PRO :
 - It's a simple method to have good statistics, before you do the binds of the packages
- CONS :
 - You're working with data, so you have to make good arrangements with the develop team to ensure to have a good set of data
 - Your scheme of jobs that will be executed during a release becomes more complicated
 - If you have more than one table with RI between them, you must pay attention on the good order of the loads

Release Process Issues

Ways of working :

- Copying Data from Test to Accept and Production
- **Migrate Statistics from Test to Accept and Production**
- Manipulating the DB2 Catalog

Using BMC migrate statistics (1)

- BMC Workbench enables you to migrate the access path statistics for :
 - A set of objects from one subsystem to another
 - From one schema to another within the same subsystem.
- For the selected objects, BMC Workbench migrates the table, tablespace and index statistics that affect the access path.
- You can collect the data for one or more objects
- You can manipulate the collected info :
 - Delete info from the objects you don't need
 - Manipulate info of the selected objects

Using BMC migrate statistics (2)

```
# SQL EXPLORER 11.02.00#
# STATISTICS OBTAINED FROM GDTC 2017-03-15-15.26.51.705899
#
# RECORD FORMAT FOR EACH OBJECT TYPE
#
SYSTABLESPACE
# :TS,DBNAME,NAME,NACTIVE,STATSTIME

SYSTABLES
# :TB,CREATOR,NAME,CARD,NPAGES,PCTROWCOMP,STATSTIME

SYSTABSTATS
# :TA,PARTITION,CARD,NPAGES,PCTPAGES,NACTIVE,PCTROWCOMP,IBMREQD,
STATSTIME

SYSCOLUMNS
# :CO,NAME,COLCARD,HIGH2KEY,LOW2KEY,STATSTIME

# :C1,LOWVALUE
# :C2,HIGHVALUE

# :CS,10,CNAME,PARTITION,COLCARD,STATSTIME,IBMREQD,STATS_FORMAT,COLCARDATA
# :CS,20,CNAME,PARTITION,HIGHKEY
# :CS,21,CNAME,PARTITION,HIGH2KEY
# :CS,22,CNAME,PARTITION,LOWKEY
# :CS,23,CNAME,PARTITION,LOW2KEY
```

```
SYSINDEXES
# :IX,CREATOR,NAME,NLEAF,NLEVELS,CLUSTERRATIO,FIRSTKEYCARD,FULLKEYCARD,STATSTIME

# :KT,10,SCHEMA,NAME,KEYSEQ,STATSTIME,STATS_FORMAT
# :KT,20,SCHEMA,NAME,KEYSEQ,HIGH2KEY
# :KT,21,SCHEMA,NAME,KEYSEQ,LOW2KEY

# :KD,10,SCHEMA,NAME,KEYSEQ,STATSTIME,CARDF,NUMKEYS,IBMREQD,FREQUENCYF,QUANTILEN
# :KD,20,SCHEMA,NAME,KEYSEQ,KEYVALUE
# :KD,21,SCHEMA,NAME,KEYSEQ,KEYGROUPKEYNO
# :KD,22,SCHEMA,NAME,KEYSEQ,HIGHVALUE
# :KD,23,SCHEMA,NAME,KEYSEQ,LOWVALUE

# :KS,10,SCHEMA,NAME,KEYSEQ,PARTITION,STATSTIME,CARDF,IBMREQD,STATS_FORMAT
# :KS,20,SCHEMA,NAME,KEYSEQ,PARTITION,HIGHKEY
# :KS,21,SCHEMA,NAME,KEYSEQ,PARTITION,HIGH2KEY
# :KS,22,SCHEMA,NAME,KEYSEQ,PARTITION,LOWKEY
# :KS,23,SCHEMA,NAME,KEYSEQ,PARTITION,LOW2KEY

# :KA,10,SCHEMA,NAME,KEYSEQ,PARTITION,STATSTIME,CARDF,IBMREQD,TYPE,NUMKEYS,FREQU
# :KA,20,SCHEMA,NAME,KEYSEQ,PARTITION,KEYVALUE
# :KA,21,SCHEMA,NAME,KEYSEQ,PARTITION,KEYGROUPKEYNO
# :KA,22,SCHEMA,NAME,KEYSEQ,PARTITION,HIGHVALUE
# :KA,23,SCHEMA,NAME,KEYSEQ,PARTITION,LOWVALUE

# :RT,SCHEMA,NAME,ROUTINETYPE,SPECIFICNAME,PARAM_COUNT,CARDINALITY,INITIAL_INSTS,
#
```

Using BMC migrate statistics(3)

TEST



Collect STATS info of the objects in DB2 Catalog Tables via BMC tool

ACC/PRD

Create New Objects

Run the BMC tool to insert STATS info on the Catalog Tables

Insert new table(s) in Stats_EXCL

Bind packages

Using BMC migrate statistics(4)

- PRO :

- You can easily create new empty tables and update the DB2 Catalog with info from test, to have good Statistics
- You don't have to worry about RI between new tables, because you don't work with data on the tables

- CONS :

- You're working with statistics based on the DB2 Catalog of test, so you have to make good arrangements with the develop team to ensure to have a good set Statistics in Test
- Your scheme of jobs that will be executed during a release is less complicated, but still extra steps need to be executed

Release Process Issues

Ways of working :

- Copying Data from Test to Accept and Production
- Migrate Statistics from Test to Accept and Production
- **Manipulating the DB2 Catalog**

Manipulate DB2 Catalog (1)

- In the 2 first methods, we depend on the quality of the data in TEST
- So we decided to make a more simple method to manipulate columns of the DB2 Catalog :
 - SYSTABLESPACE
 - SYSTABLES
 - SYSINDEXES
 - SYSCOLUMNS

Manipulate DB2 Catalog (2)

- After the creation of the BMC Change Manager worklist of a database, we do a full scan of the worklist to collect :
 - Info Create/Alter tables
 - Info Create indexes
 - Overview fields of index + COLCARDF of these field (informative)
 - Load of tables in WL
 - Check tablespaces in EITSTATS_EXCL

Manipulate DB2 Catalog (3)

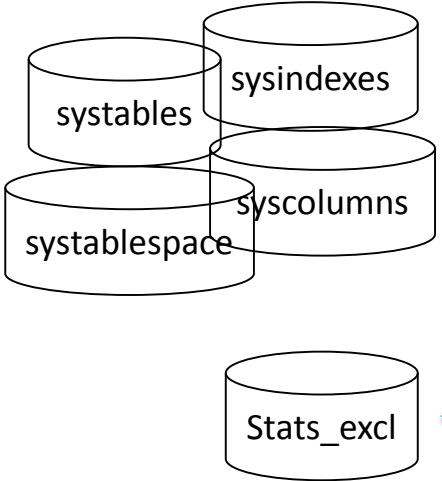
ACC/PRD

Worklist BMC

```
-SYNC 000250 START OF EXECUTION SYMPOINT
.....
-SQL 000350 CREATE TABLESPACE CBSCMO25 IN CBDCM01
      MAXPARTITIONS 16
      USING STOGROUP CBGRP001
      PRIORITY 10800
....
-SYNC 000450 END OF CREATE TABLESPACE SECTION
.....
-SQL 000500 CREATE TABLE xxxxxx.CBTCMO25
      (
        ELB_USER_NO
      ....
```

Evaluate worklist BMC

```
CBTCMO25 XXXXXX          1T
N CBX2CMO25              1I
 * CEV_NTF_NO             1C
 * TYPE_CD                1C
Y CBX1CMO25              1I
 * CEV_NTF_NO             1C
 * CEV_NTF_PMTR_NO       1C
```



Create update statements on DB2 Catalog

(If necessary) Insert record

Manipulate DB2 Catalog (5)

```
SET CURRENT SQLID = 'XXXXX';
```

```
UPDATE SYSIBM.SYSTABLES  
SET CARD      = 100000  
  ,CARDF     = 100000  
  ,NPAGES    = 2000  
  ,NPAGESF   = 2000  
WHERE NAME    = 'CBTXXX01'  
  AND TYPE    = 'T'  
  AND CREATOR = 'YYYY';  
COMMIT;
```

```
UPDATE SYSIBM.SYSINDEXES  
SET FIRSTKEYCARDF = 100000  
  ,FULLKEYCARDF  = 100000  
  ,NLEAF         = 1000  
  ,NLEVELS      = 2  
  ,CLUSTERRATIO = 99  
  ,CLUSTERED    = 'N'  
WHERE NAME      = 'CBX1XXX01'  
  AND CREATOR   = 'YYYY';  
COMMIT;
```

```
UPDATE SYSIBM.SYSINDEXES  
SET FIRSTKEYCARDF = 80000  
  ,FULLKEYCARDF  = 100000  
  ,NLEAF         = 1000  
  ,NLEVELS      = 2  
  ,CLUSTERRATIO = 95  
  ,CLUSTERED    = 'N'  
WHERE NAME      = 'CBX2XXX01'  
  AND CREATOR   = 'YYYY';  
COMMIT;
```

```
UPDATE SYSIBM.SYSCOLUMNS  
SET COLCARDF = 100000  
WHERE NAME    = 'COL_NM_1'  
  AND TBNAME  = 'CBTXXX01'  
  AND TBcreator = 'YYYY';  
COMMIT;
```

```
UPDATE SYSIBM.SYSCOLUMNS  
SET COLCARDF = 100000  
WHERE NAME    = 'FQC_NO'  
  AND TBNAME  = 'CBTXXX01'  
  AND TBcreator = 'YYYY';  
COMMIT;
```

We didn't do this before !!!

Since a few months we saw that DB2 wasn't using a correct access path if we didn't gave the colcardf in syscolumns of index fields a good value

Manipulate DB2 Catalog (6)

- PRO :

- You can easily create new empty tables and update the DB2 Catalog with a pre-defined Script
- You're **NOT** working with data or statistics from test, but with a self defined Script to update the DB2 Catalog
- You don't have to worry about RI between new tables, because you don't work with data on the tables

- CONS :

- Your scheme of jobs that will be executed during release is less complicated, but you still have to run an extra job.

Statistics @ KBC

- Why do we run RUNSTATS ?
- RUNSTATS Process
- RUNSTATS and Utilities
- Release Process Issues
- **Miscellaneous**

Miscellaneous

- Adding Index and RUNSTATS, not enough ???
- RUNSTATS and IDAA
- Use of SYSSTATFEEDBACK

Adding New Index and RUNSTATS

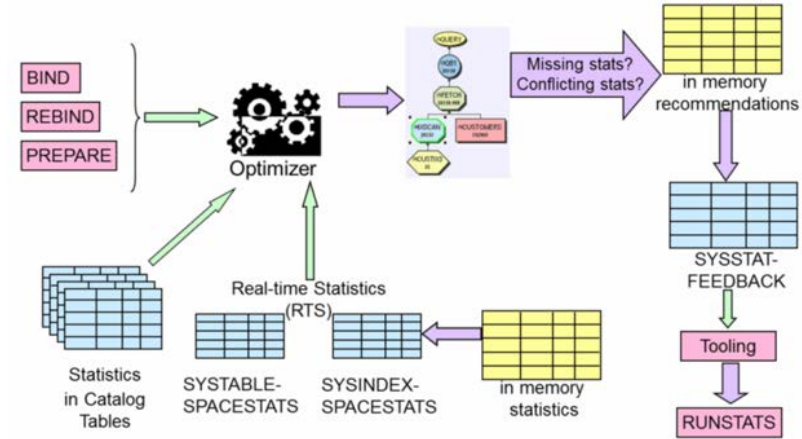
- We recently had a problem concerning RUNSTATS :
 - We created a new well designed INDEX to solve a performance problem
 - Ran RUNSTATS on that new INDEX
 - REBIND the PACKAGE, **BUT** DB2 **didn't use** the new INDEX
 - Ran RUNSTATS on TS level, INDEX ALL
 - REBIND the PACKAGE, **AND** DB2 **did use** the new INDEX
 - I think I once heard that on an IDUG or other seminar
 - We already had 2 of these cases !!!
 - We are going to imbed this in our RUNSTATS strategy !!!

RUNSTATS and IDAA

- IDAA needs statistics on the DB2 OBJECT
- If NOT
- The query probably won't be accelerated

Use of SYSSTATFEEDBACK

- Investigated the new process in DB2 11
- Conclusion :
 - Many recommendations on test DB2
 - Lots of work to interpret recommendations
 - Some recommendations are useful, but not always
 - NO guarantee that the recommendations will change the access paths
- For the moment our own RUNSTATS process does what we expect
- So we didn't changed it





Questions