RUNSTATS challenges for optimal query performance

Terry Purcell

IBM Silicon Valley Lab

Session code: B01

Tues June 5th 2018

Db2 for z/OS

Agenda

- Importance of statistics to optimizer
- Common mistakes with FREQVAL
- RUNSTATS conflicts
- Integrating supplemental statistics
- Optimizer recommended statistics
- RUNSTATS Simplification/Performance

Why did the optimizer choose that access path?

- This is a cost based optimizer
  - It chose what it believes to be the lowest cost access path
- If the access path is not optimal, then the real question is
  - Why did the optimizer think that was the lowest cost access path?

OR

- What doesn't the optimizer know???

What's important for Optimizer to Know?

- We all recognize the need for RUNSTATS for optimizer:
  - Size of the objects
    - NPAGESF, NLEAF, NLEVELS etc.
  - Selectivity or number of records/keys
    - COLCARDF, FREQVAL etc.
  - Other important statistics
    - CLUSTERRATIOF, PCTROWCOMP etc.
  - This stuff gets collected by default.
    - Are there other important RUNSTATS inputs?
    - Other inputs not collected by RUNSTATS?

Let's Oversimplify Optimizer Costing...

- Optimizer assigns Filter Factors (FFs) for each WHERE/ON predicate
- FFs are combined to determine the total filtering per object
  - Multiply “AND” predicate FFs
    - Available multi-column cardinality statistics determine “degree” of multiplication
      - KEYCARD
      - COLGROUP(C1,C2)
  - Add “OR” predicate FFs
  - FF accuracy and how to combine these is important for costing
    - Index matching
    - Total index filtering
    - Total table level filtering

Agenda

- Importance of statistics to optimizer
- Common mistakes with FREQVAL (or HISTOGRAMs)
- RUNSTATS conflicts
- Integrating supplemental statistics
- Optimizer recommended statistics
- RUNSTATS Simplification/Performance

Common misuse of FREQVAL NUMCOLS

- In an effort to simplify standard RUNSTATS jobs, we see

```
RUNSTATS TABLE(ALL)
         INDEX(ALL) FREQVAL NUMCOLS 10 COUNT 10
```

  - Thinking this means:
    - Frequencies on the first column
    - Frequencies on the first two columns
    - Frequencies on the first three columns
    - …
    - Frequencies on the first ten columns

Does that mean I should correct it with…….?

- NO…..although this is the correct way to collect multi-column frequencies

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

  - Multi-column frequencies (FREQVALs) are rarely used
  - The above is ANOTHER common MISTAKE

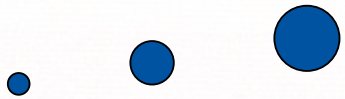
Why are Multi-Column FREQVALs a Mistake?

- For the optimizer to use this (FREQVAL on 10 columns)…..

```
RUNSTATS TABLE (ALL)
         INDEX (ALL)
             .....
             FREQVAL NUMCOLS 10 COUNT 10
```

  - You need an SQL that looks like this……

```
SELECT * FROM TABLE
WHERE C1 = 'A'
AND C2 = 10
AND C3 = 100
AND C4 = 'SMITH'
AND C5 = 1000
.....
AND C10 = 'Y'
```



To use a 10 column FREQVAL – you need 10 equals predicates with literals!

What about other FREQVAL or HISTOGRAM options? Mistakes???

- Recent customer default RUNSTATS options…..

```
RUNSTATS TABLESPACE LIST TSLIST TABLE ALL INDEX(ALL) KEYCARD
FREQVAL NUMCOLS 1 COUNT 15 BOTH
FREQVAL NUMCOLS 2 COUNT 15 BOTH
.....
FREQVAL NUMCOLS 10 COUNT 15 BOTH
HISTOGRAM NUMCOLS 1
HISTOGRAM NUMCOLS 2
.....
HISTOGRAM NUMCOLS 10
```

BOTH (or LEAST) not as valuable as MOST

HISTOGRAMS utilized less than FREQVAL

- What is the harm in “too many” statistics?
  - Every BIND/REBIND/Prepare/Explain
    - Optimizer must read in ALL available statistics ($$$)

Basic Statistics Recommendations

- Basic statistics foundation

```
RUNSTATS TABLE (ALL)
  INDEX (ALL) KEYCARD
```

KEYCARD is the default as of Db2 10

- Supplement with more detailed statistics as needed
  - Distribution statistics
    - Frequencies
    - Histograms
  - Multi-column cardinality statistics
- Multi-column frequency or histograms are NOT recommended by default

- Easily identify important supplemental statistics with Statistics Advisor
  - (Free) Single query Statistics Advisor (Data Studio/DSM)
  - Workload Statistics Advisor (Optim Workload Query Tuner/DSM) – or your preferred vendor's product
  - Db2 Optimizer Recommended Statistics (Db2 11 & 12)

What if I have "excess" statistics? How to clean up (without risk)?

- Multi-column FREQVAL (and histograms) do have some value
  - While NOT recommended by default
  - Question is HOW to remove without risk of regression?

  - If you have 5 col, 8 col, 10 col by default…….
    - You are UNLIKELY to be getting value after the 2nd or 3rd

  - You can remove FREQVAL by specifying COUNT 0 (does not apply to HISTOGRAM)

```
FREQVAL NUMCOLS 4 COUNT 0 .....
```

    - DELETE FROM SYSIBM.SYSCOLDIST WHERE TYPE = 'H' AND NUMCOLUMNS > 3
    - Do NOT simply STOP collecting – as these will stay forever

To recap – FREQVAL and HISTOGRAMS

- HISTOGRAMs (Single or Multi-column) and/or Multi-column FREQVAL
  - Are NOT recommended by default
  - Collect "AS NEEDED" based upon identified SQL
    - By tooling or optimizer statistics recommendations (discussed later)

- Using BOTH or LEAST
  - Are NOT recommended
  - USE MOST only

- Recommended default is ONLY the following (without superfluous FREQVAL or HISTOGRAMs)

```
RUNSTATS TABLE (ALL)
 INDEX (ALL) KEYCARD
```

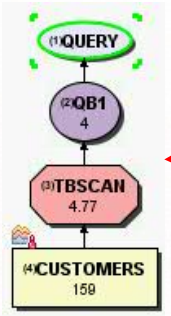
Agenda

- Importance of statistics to optimizer
- Common mistakes with FREQVAL
- RUNSTATS conflicts
- Integrating supplemental statistics
- Optimizer recommended statistics
- RUNSTATS Simplification/Performance

Mistimed RUNSTATS Collection

- Is RUNSTATS run on tables and indexes at different times?
  - WARNING – Optimizer does not account for all statistics discrepancies
  - In general:
    - Table statistics are used for data access costing
    - Index statistics are used for index costing
  - Example:

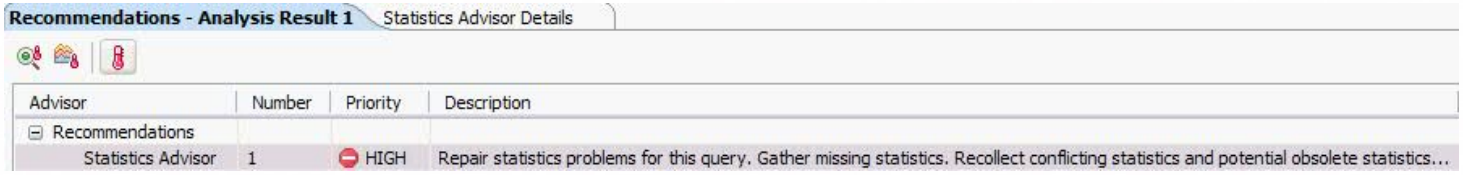
```
SELECT *
FROM CUSTOMERS
WHERE CUSTNO BETWEEN ? and ?
```



Tablespace scan, instead of unique index

Mistimed RUNSTATS Example Results

- Statistics Advisor
  - Identified conflicting statistics




```
Statistics Advisor Summary Report
Analysis start time: 2011-05-02 13:31:34.403818
Analysis end time:   2011-05-02 13:31:36.558377

==============================

TABLE SYSADM.CUSTOMERS
  Table type:       Table
  Cardinality:      159.0
  Collection time:  2010-01-01 00:00:00.000001
  Statistics status: conflicting


INDEXES:

SYSADM.CUSTIX1C (CUSTNO)
  First key cardinality:  192960.0
  Full key cardinality:   192960.0
  Data repetition factor: 2757.0
  Collection time:        2011-05-02 08:28:44.544249
  Statistics status:      conflicting
```

Table has 159 rows.
Stats collected in 2010.

Index has 192960 rows.
Stats collected in 2011.

- NOTE: Stats Advisor conflicts indicate a problem in your RUNSTATS processes

Identifying Statistics Conflicts

- Identify statistics conflicts by querying the catalog directly
  - Tables with table cardinality less than index full key cardinality
  - Columns with more single column frequencies than values
  - Table cardinality less than column cardinality
  - Column cardinality not equal to index first key cardinality for an index leading with that column
  - And many more...
- Statistics Advisor identifies many statistics conflicts – on a query or workload basis
- Db2 optimizer recommended statistics identifies conflicts based on queries bound/prepared

Identifying Statistics Conflicts

- Tables with table cardinality less than index full key cardinality

```
SELECT SUBSTR(T.CREATOR,1,18) TBCREATOR
      ,SUBSTR(T.NAME,1,10)    TBNAME
      ,SUBSTR(I.CREATOR,1,18) IXCREATOR
      ,SUBSTR(I.NAME,1,10)    IXNAME
      ,T.CARDF
      ,I.FULLKEYCARDF
FROM SYSIBM.SYSTABLES T
    ,SYSIBM.SYSINDEXES I
WHERE T.CREATOR = I.TBCREATOR
  AND T.NAME = I.TBNAME
  AND T.CARDF < I.FULLKEYCARDF
  AND T.CARDF >= 0
```

Statistics have been collected

Identifying Statistics Conflicts

- Columns with more single column frequencies than values

```
SELECT SUBSTR(C.TBCREATOR,1,10) TBCREATOR
      ,SUBSTR(C.TBNAME,1,18)    TBNAME
      ,SUBSTR(C.NAME,1,18)      COLNAME
      ,C.COLCARDF
      ,COUNT(*)                 NUM_FREQS
FROM SYSIBM.SYSCOLUMNS C
    ,SYSIBM.SYSCOLDIST D
WHERE C.TBCREATOR = D.TBOWNER
  AND C.TBNAME = D.TBNAME
  AND C.NAME = D.NAME
  AND D.TYPE = 'F'
  AND D.NUMCOLUMNS = 1
  AND C.COLCARDF >= 0
GROUP BY C.TBCREATOR, C.TBNAME, C.NAME, C.COLCARDF
HAVING COUNT(*) > C.COLCARDF
```

Frequency statistics

Single column distribution statistics

More frequencies than column values

Identifying Statistics Conflicts

- Column cardinality matches index first key cardinality

```
SELECT SUBSTR(I.TBCREATOR,1,18) TBCREATOR
      ,SUBSTR(I.TBNAME,1,18)    TBNAME
      ,SUBSTR(I.CREATOR,1,18)   IXCREATOR
      ,SUBSTR(I.NAME,1,18)      IXNAME
      ,SUBSTR(C.NAME,1,18)      COLNAME
      ,I.FIRSTKEYCARDF
      ,C.COLCARDF
FROM SYSIBM.SYSINDEXES I
INNER JOIN SYSIBM.SYSKEYS K
   ON I.CREATOR = K.IXCREATOR
  AND I.NAME = K.IXNAME
INNER JOIN SYSIBM.SYSCOLUMNS C
   ON I.TBCREATOR = C.TBCREATOR
  AND I.TBNAME = C.TBNAME
  AND K.COLNO = C.COLNO
WHERE K.COLSEQ = 1
  AND I.IX_EXTENSION_TYPE = ' '
  AND I.FIRSTKEYCARDF >= 0
  AND C.COLCARDF >= 0
  AND (I.FIRSTKEYCARDF < C.COLCARDF * 0.9
    OR I.FIRSTKEYCARDF > C.COLCARDF * 1.1)
```

First index column

Simple Index (not IOE or XML)

Ideally they match, but check within 10%

To recap – Collecting RUNSTATS at different dates/times

- The optimizer (V11/12) will identify STALE or CONFLICTing statistics
  - These typically highlight a discrepancy in collection processes
  - BUT – the optimizer may NOT resolve the conflict
  - For example
    - Regular REORG with inline stats on indexes but NOT tablespaces
- Recommended NOT to introduce conflicts
  - Collect RUNSTATS on related objects – table(space) and its related indexes – at same time

Agenda

- Importance of statistics to optimizer
- Common mistakes with FREQVAL
- RUNSTATS conflicts
- Integrating supplemental statistics
- Optimizer recommended statistics
- RUNSTATS Simplification/Performance

How do I integrate supplemental statistics?

```
RUNSTATS LIST mylist
         TABLE (ALL)
         INDEX (ALL) KEYCARD

RUNSTATS mydb.myts
         TABLE(CUSTOMER)
         COLGROUP(CITY,ZIPCODE)
```

- Prior to Db2 10, must be two separate RUNSTATS steps

Mixing Regular and "targeted" RUNSTATS

- If I run the following

```
1. RUNSTATS TABLE(ALL) INDEX(ALL) KEYCARD

2. RUNSTATS TABLE(customer) COLGROUP (STATUS) FREQVAL COUNT 20

3. RUNSTATS TABLE(ALL) INDEX(ALL) KEYCARD
```

- Wont "Regular" RUNSTATS overwrite the "targeted"?
  - NO: RUNSTATS will only overwrite similar statistics
    - COLGROUP(STATUS) FREQVAL COUNT 20 is only overwritten if default statistics are collecting FREQVAL on this column
    - Is there an index leading with STATUS? Default is to collect top 10 (not top 20).

Db2 10 Simplifies Integration of Supplemental Statistics

- Statistics profiles
  - Integrate specialized statistics into generic RUNSTATS job
    - RUNSTATS TABLE (mytb) COLGROUP(STATUS)… **SET PROFILE**
      - Or if wanting to create profile from currently collected statistics
        - use **SET PROFILE FROM EXISTING STATS**
    - RUNSTATS … TABLE (mytb) …. **UPDATE PROFILE**
      - To update the profile
  - Next usage
    - RUNSTATS LIST mylist TABLE(ALL) **USE PROFILE**
      - Profile will be used for any table that has a profile

Db2 10 Simplifies Integration of Supplemental Statistics

- Statistics profile restrictions
  - Cannot USE PROFILE for a table without a defined profile. RUNSTATS will fail
    - Solved in Db2 11 – Default of COLUMN(ALL) INDEX(ALL) will be used
  - Cannot USE PROFILE via inline stats
    - Solved in Db2 12
  - DDL doesn't maintain statistics profiles (e.g. dropped index remains in profile)
    - Solved in Db2 12

Reset Access Path Statistics – Db2 11

- Reset statistics for a given object

```
RUNSTATS mydb.myts
         TABLE(mytb)
         RESET ACCESSPATH
```

  - Does not reset space statistics
  - Does not reset real time statistics (RTS)
- Critical to RESET to clear out old statistics if unknown what exists
  - Recommended to recollect valid statistics after RESET
  - Be aware – time between RESET and recollection may allow poor optimizer choices

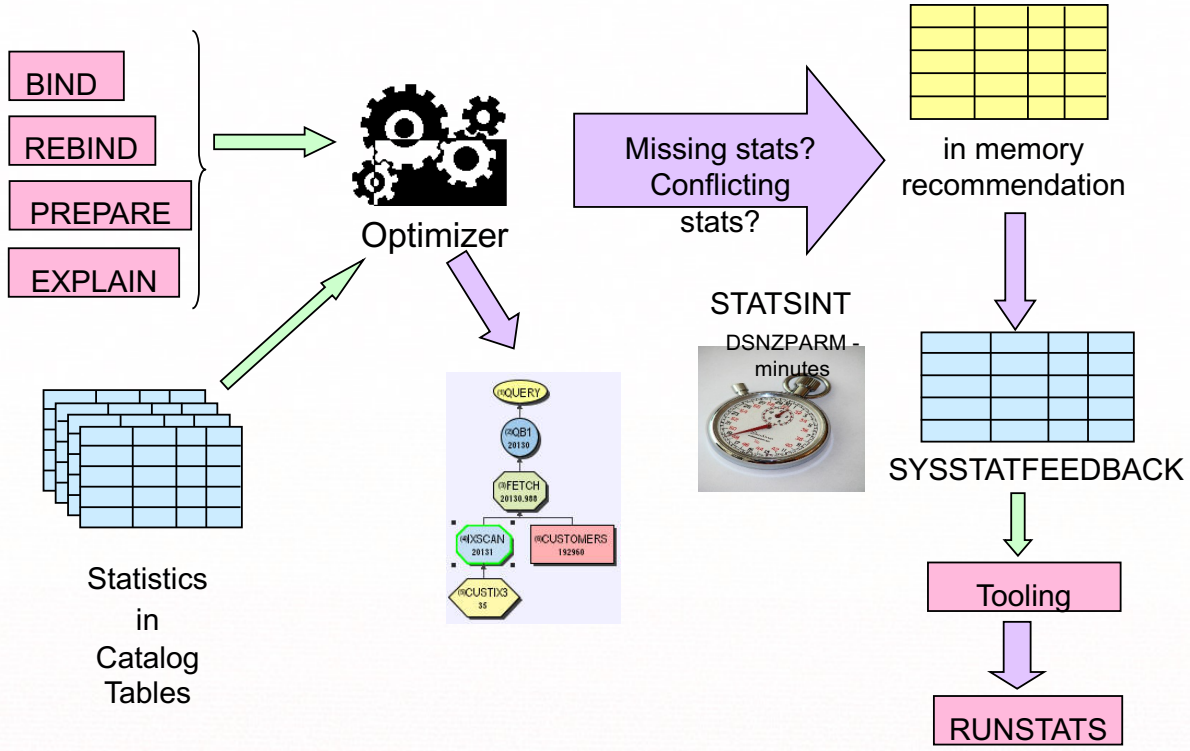
Agenda

- Importance of statistics to optimizer
- Common mistakes with FREQVAL
- RUNSTATS conflicts
- Integrating supplemental statistics
- Optimizer recommended statistics
- RUNSTATS Simplification/Performance

Db2 Optimizer and Statistics - Challenge

- Db2 cost-based optimizer relies on statistics about tables and indexes
- Customers often gather only standard or default statistics
  - E.g. RUNSTATS TABLE(ALL) INDEX(ALL) KEYCARD
- Queries would often perform better if Db2 optimizer could exploit more complete statistics
- Customers have difficulty knowing which statistics should be gathered
  - Many SQL statements
  - Many tables, indexes, column, keys
  - Statistics advisor tools provide beneficial input on statistics
    - Only apply to the queries or workloads submitted for analysis

Db2 11 – Optimizer externalization of missing statistics



Db2 11 Solution: Optimizer Externalization

- During access path calculation, optimizer will identify missing or conflicting statistics
  - On every BIND, REBIND, PREPARE and EXPLAIN
  - Recommendations are written asynchronously to SYSIBM.SYSSTATFEEDBACK
  - -ACCESS DB(...) SP(...) MODE(STATS)
    - Externalize in-memory RTS statistics and optimizer recommended statistics
      - In data sharing, externalization is done on all members
  - Db2 also provides statistics recommendations on EXPLAIN
    - Populates DSN_STAT_FEEDBACK synchronously

Optimizer Recommended Statistics: Controls

- ZPARM STATFDBK_SCOPE
  - NONE – Disable collection of recommended RUNSTATS
  - STATIC – Collect for static queries only
  - DYNAMIC – Collect for dynamic queries only
  - ALL – Collect for all SQL (default)
- SYSTABLES.STAT_FEEDBACK updateable column (table control)
  - Y|N – Indicates whether to externalize recommendations for this table
    - Y is the default. N means no externalization for this table

Optimizer Recommended Statistics: Column details

- Contents of SYSSTATFEEDBACK or DSN_STAT_FEEDBACK can be used to generate input to RUNSTATS

Column	Description
TBCREATOR	Creator of the table
TBNAME	Name of the table
IXCREATOR	Creator of the index
IXNAME	Name of the index
COLNAME	Name of the column
NUMCOLUMNS	Number of columns in the column group
COLGROUPCOLNO	Hex representation that identifies the set of columns associated with the statistics. If the statistics are only associated with a single column, the field contains a zero length. Otherwise, the field is an array of SMALLINT column numbers with a dimension equal to the value in NUMCOLUMNS.
TYPE	'T' – Table 'I' – Index 'C' – Cardinality 'F' – Frequency 'H' - Histogram

Optimizer Recommended Statistics: Column details

Column	Value	Description
REASON	BASIC	Basic statistic is missing
	KEYCARD	KEYCARD statistic is missing
	LOWCARD	Column has low cardinality, which indicates data skew is likely
	NULLABLE	Distribution statistics not available for a nullable column
	IXPROBE	Index probing was used to improve a very low filter factor
	DEFAULT	A predicate references a value that is probably a default value
	RANGEPRD	Histogram statistics not available for a range predicate
	PARALLEL	Parallelism could be improved by uniform partitioning of key ranges
	CONFLICT	Another statistic conflicts with this statistic
	COMPFFIX	Multi-column cardinality statistics are needed for an index compound filter factor

Optimizer Recommended Statistics: Example

TBCREATOR	TBNAME	IXCREATOR	IXNAME	DBNAME	TSNAME	COLNAME	TYPE
SYSADM	T1	SYSADM	T1_IX2	MYDB	MYTS1		I
SYSADM	T1			MYDB	MYTS1	C1	F
SYSADM	T2			MYDB	MYTS2	C2	H

RUNSTATS TABLESPACE MYDB.MYTS2
TABLE(SYSADM.T2)
COLGROUP(C2) HISTOGRAM NUMQUANTILE
100

RUNSTATS TABLESPACE MYDB.MYTS1
TABLE(SYSADM.T1)
INDEX(SYSADM.T1_IX2)
COLGROUP(C1) FREQVAL COUNT 10

Consuming recommendations in Db2 11

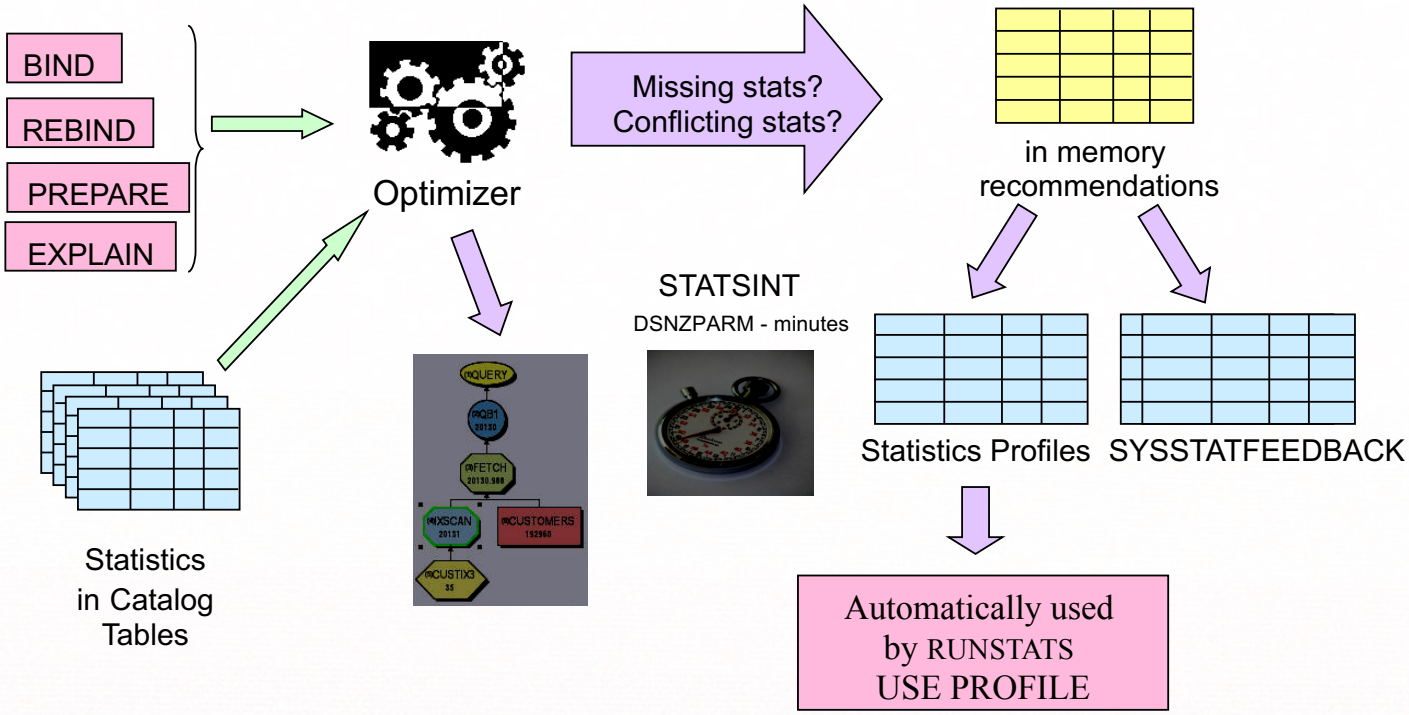
- Some reasons are more likely to provide value than others

- Focus on five reasons
  - BASIC
    - Basic statistics are needed (TABLE(ALL) INDEX(ALL))
  - LOWCARD
  - DEFAULT
  - NULLABLE
  - CONFLICT
    - Likely implies statistics were run on different objects at different times

Further notes about interpreting recommendations

- Db2 is only recommending that a statistic could have been used
  - This is not a guarantee that the statistic is needed.
  - There is still a benefit to try to 1st determine whether collecting the statistic may add value
    - For a TYPE='F' recommendation – is the data really skewed?
    - What value to use for "COUNT integer"?
      - 10 is a good default (better solution in Db2 12)
      - If COLCARDF<=10, then use COLCARDF-1
  - REASON should also be considered
    - For example - TYPE='F',REASON='NULLABLE'
      - If NULL is most frequently occurring, then you only need COUNT 1 (not 10)
    - NOTE: V11/12 APAR PI74408/PI76730
      - Improve performance and zIIP offload for RUNSTATS with non-indexed FREQVAL

Db2 12 – Optimizer externalization of missing statistics



Db2 12: Optimizer externalization of missing statistics

- Let Db2 manage which statistics are collected based on analysis of queries running in the system
  - Completely automated solution for what to collect
  - Statistics profiles built to include a complete set of statistics and not just those identified as missing
  - Frequency of collection still governed by existing tools/features (e.g. DSNACCOX)

- To enable:
  - Zparm STATFDBK_SCOPE must be set to allow recommendations
  - New zparm STATFDBK_PROFILE must be set to YES
  - Statistics jobs must specify USE PROFILE

Migration Considerations (V11 or 12) - Old statistics

- Old (stale) statistics
  - Customers often run “specialized” stats as a one-off to try to solve an issue or as a prior default.
    - These old statistics can become stale and cause access path issues
    - Simplest way to find these is to look for tables with rows having different STATSTIMEs in SYSCOLDIST
- Db2 11 delivers
  - RUNSTATS reset option
    - Sets relevant catalog values to -1, and clears tables such as SYSCOLDIST

```
RUNSTATS TABLESPACE db-name.ts-name
TABLE table-name RESET ACCESSPATH
```

    - Recommend running “regular” RUNSTATS after RESET
  - Blog post
    - http://www.worldofdb2.com/profiles/blogs/ensure-you-have-a-simple-base-of-statistics-before-exploiting-the

Do you have old statistics?

- Culprits are usually multi-column statistics or histograms
  - Quick check using this SQL
    - Remove ASAP – since future enhancements can incorrectly assume these are critical to collect
  - Cleanup becomes critical in Db2 12
    - Or if using SET PROFILE FROM EXISTING STATS (V10-12)

```
SELECT TYPE, NUMCOLUMNS, TBOWNER, TBNAME, NAME
, MIN(STATSTIME), COUNT(*)
FROM SYSIBM.SYSCOLDIST CD
WHERE STATSTIME < CURRENT TIMESTAMP - 1 MONTH
AND (TYPE IN ('C', 'H') OR NUMCOLUMNS > 1
  OR STATSTIME < CURRENT TIMESTAMP - 1 YEAR)
AND NOT EXISTS
(SELECT 1
FROM SYSIBM.SYSINDEXES I
WHERE I.TBCREATOR = CD.TBOWNER
AND   I.TBNAME = CD.TBNAME
AND   CD.STATSTIME BETWEEN I.STATSTIME - 8 DAYS
                       AND I.STATSTIME + 8 DAYS)
AND NOT EXISTS
(SELECT 1
FROM SYSIBM.SYSTABLES T
WHERE T.CREATOR = CD.TBOWNER
AND   T.NAME = CD.TBNAME
AND   CD.STATSTIME BETWEEN T.STATSTIME - 8 DAYS
                       AND T.STATSTIME + 8 DAYS)
GROUP BY TYPE, NUMCOLUMNS, TBOWNER, TBNAME, NAME
ORDER BY TYPE, NUMCOLUMNS, TBOWNER, TBNAME, NAME
WITH UR;
```

Db2 12 Statistics Feedback: Migration Considerations

- Prior to mass rebind after Db2 12 migration ensure Db2 is collecting relevant statistics

- Step 1 - Enable statistics feedback and statistics profile updates
- Step 2 - Rebind static packages with EXPLAIN(ONLY) to drive statistics profile updates
- Step 3 - Modify RUNSTATS jobs to USE PROFILE
- Step 4 – Execute -ACCESS DB(*) SP(*) MODE(STATS) to force externalization of any pending statistics recommendations
- Step 5 - Run RUNSTATS with USE PROFILE to collect newly recommended statistics

- NOTE: Customers can choose any process they wish
  - An EXPLAIN, BIND/REBIND or dynamic prepare will trigger (potential) RUNSTATS recommendations (and create profile in V12)

Agenda

- Importance of statistics to optimizer
- Common mistakes with FREQVAL
- RUNSTATS conflicts
- Integrating supplemental statistics
- Optimizer recommended statistics
- RUNSTATS Simplification/Performance

RUNSTATS Automated Count

- When collecting FREQVAL (frequencies)
  - What is the best “count” to use?
    - COLGROUP(C1) COUNT 10? COUNT 20?
  - The answer is:
    - Keep collecting until the data is “no longer” skewed
- Db2 12 adds this capability
  - Exclude the COUNT n keywords – and Db2 will automate the COUNT (with a max of 100)
  - NOTE: (APAR PI94111) Optimizer stats externalization utilizes this feature

RUNSTATS Simplification

- Only collect the following by default

```
RUNSTATS TABLE (ALL)
 INDEX (ALL) KEYCARD
```

- Additional options are based upon workload requirements
  - Do NOT collect multi-column FREQVAL or (single/multi-column) HISTOGRAMs by default

- Db2 11/12 allows you to embed complexity (additional options) inside a STATS PROFILE
  - In Db2 12 PROFILEs can be utilized by all methods of RUNSTATS executions

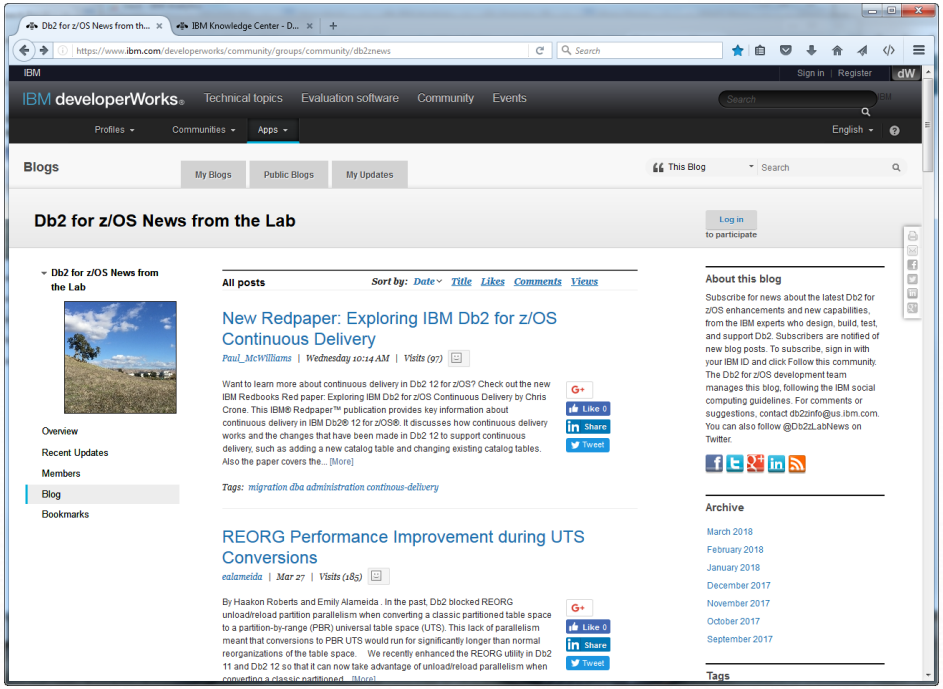
RUNSTATS Performance

- Ensure recent performance APARs are applied

  V11 APAR PI74408, V12 APAR PI76730
  Improve performance and zIIP offload for RUNSTATS with non-indexed FREQVAL

- Use TABLESAMPLE SYSTEM AUTO to improve RUNSTATS performance
  - Original SAMPLE option has less performance benefit
  - TABLESAMPLE n is NOT preferred
    - Due to impact for smaller tables

Db2 for z/OS News from the Lab blog
http://ibm.biz/db2znews



- Get the latest news from the IBMers who design and build Db2!
  - New capabilities in Db2 12 for z/OS continuous delivery
  - Enhancements in Db2 11 for z/OS
  - Helpful tips and best practices from Db2 for z/OS development
- Join the conversation
  - Subscribe to follow the blog
  - Become a member to comment
  - Follow us on Twitter: @Db2zLabNews

IDUG Db2 Tech Conference NA
Philadelphia, PA | April 29 - May 3, 2018

#IDUGDb2

Terry Purcell
IBM
tpurcel@us.ibm.com

Session code: B01

Please fill out your session evaluation before leaving!