IBM Software Group

ZS08 - DB2 RUNSTATS Utility: What's Old, What's New, What's a FREQVAL?

Bryan F. Smith
bfsmith@us.ibm.com

IBM Information Management software

ON DEMAND BUSINESS

Abstract

- This presentation reviews the basics of the RUNSTATS utility (What it does; Why you need to run it; How DB2 uses the information), and explores new statistics collected on data and indexes, including: partition level information on Data Partitioned Secondary Indexes; non-uniform distributon statistics on non-indexed columns; and historical statistics. Upon completion of this session, the attendee, whose skill level may range from low to high, will be able to understand how to get the most out of DB2's statistics and operate at optimal efficiency.

Topics

- Why RUNSTATS?
- Commonly asked questions (about the stats)
- Rebinding considerations
- Reorg recommendations
- When is RUNSTATS needed?
- New/changed data statistics
- New/changed index statistics
- Handling part level statistics for DPSIs
- Distribution Statistics Enhanced
- HISTORY statistics changes
- Flushing the dynamic statement cache
- What statistics should I gather?

Why RUNSTATS?

- The RUNSTATS utility computes statistics on a specified table space or index and updates the DB2 catalog
- Two types of statistics
  - Access path statistics

    Those used by BIND/PREPARE in its process of optimization to determine access path (some can also be used to help determine when to reorg)
  - Space

    Those used by the DBA to monitor space usage; to assist in capacity planning; to help determine when to reorg; etc.

Statistics gathered by RUNSTATS

Access path statistic
Access path (not used)
Space statistic

Table in DSNDB06.SYSDBASE
Table in DSNDB06.SYSHIST
Table in DSNDB06.SYSSTATS

Collected from table space scan
both
Collected from index scan

- SYSIBM.SYSTABLES_HIST
  - CARD/F
  - NPAGES/F
  - PCTPAGES
  - PCTROWCOMP
  - AVGROWLEN
  - SPACEF
- SYSIBM.SYSTABLESPACE
  - NACTIVE/F
  - AVGROWLEN
  - SPACEF
- SYSIBM.SYSINDEXES_HIST
  - CLUSTERRATIO/F
  - CLUSTERED
  - FIRSTKEYCARD/F
  - FULLKEYCARD/F
  - NLEAF
  - NLEVELS
  - AVGKEYLEN
  - SPACEF

aggregates

- SYSIBM.SYSTABSTATS_HIST
  - CARD/F
  - NPAGES
  - PCTPAGES
  - NACTIVE
  - PCTROWCOMP
- SYSIBM.SYSTABLEPART_HIST
  - AVGROWLEN
  - CARD/F
  - DSNUM
  - EXTENTS
  - NEARINDREF
  - FARINDREF
  - PAGESAVE
  - PERCACTIVE
  - PERCDROP
  - SPACE/F
  - PQTY
  - SQTY
  - SECQTYI

aggregates

- SYSIBM.SYSINDEXSTATS_HIST
  - FIRSTKEYCARD/F
  - FULLKEYCARD/F
  - NLEAF
  - NLEVELS
  - IOFACTOR
  - PREFETCHFACTOR
  - KEYCOUNT/F
  - CLUSTERRATIO/F
  - FULLKEYCARDDATA
- SYSIBM.SYSINDEXPART_HIST
  - AVGKEYLEN
  - CARDF
  - DSNUM
  - EXTENTS
  - FAROFFPOSF
  - LEAFNEAR
  - LEAFFAR
  - NEAROFFPOS
  - LEAFDIST
  - PSUEDO_DEL_ENTRIES
  - SPACEF
  - PQTY
  - SECQTYI

- SYSIBM.SYSLOBSTATS_HIST
  - FREESPACE
  - ORGRATIO
  - AVGSIZE
- SYSIBM.SYSCOLUMNS_HIST
  - COLCARD/F
  - HIGH2KEY
  - LOW2KEY
  - STATS_FORMAT

aggregates

- SYSIBM.SYSCOLSTATS
  - COLCARD
  - HIGHKEY
  - HIGH2KEY
  - LOWKEY
  - LOW2KEY
  - COLCARDDATA
  - STATS_FORMAT
- SYSIBM.SYSCOLDIST_HIST
  - CARDF
  - COLGROUPCOLNO
  - COLVALUE
  - TYPE
  - FREQUENCY/F
  - NUMCOLUMNS

aggregates

- SYSIBM.SYSCOLDISTSTATS
  - CARDF
  - COLGROUPCOLNO
  - COLVALUE
  - TYPE
  - FREQUENCY/F
  - NUMCOLUMNS
  - KEYCARDDATA

Statistics gathered by RUNSTATS

Access path statistic
Access path (not used)
Space statistic

Table in DSNDB06.SYSDBASE
Table in DSNDB06.SYSHIST
Table in DSNDB06.SYSSTATS

Collected from table space scan
both
Collected from index scan

- SYSIBM.SYSTABLES_HIST
  - CARD/F
  - NPAGES/F
  - PCTPAGES
  - PCTROWCOMP
  - AVGROWLEN
  - SPACEF
- SYSIBM.SYSTABLESPACE
  - NACTIVE/F
  - AVGROWLEN
  - SPACEF
- SYSIBM.SYSINDEXES_HIST
  - CLUSTERRATIO/F
  - CLUSTERED
  - FIRSTKEYCARD/F
  - FULLKEYCARD/F
  - NLEAF
  - NLEVELS
  - AVGKEYLEN
  - SPACEF

aggregates

- SYSIBM.SYSTABSTATS_HIST
  - CARD/F
  - NPAGES
  - PCTPAGES
  - NACTIVE
  - PCTROWCOMP
- SYSIBM.SYSTABLEPART_HIST
  - AVGROWLEN
  - CARD/F
  - DSNUM
  - EXTENTS
  - NEARINDREF
  - FARINDREF
  - PAGESAVE
  - PERCACTIVE
  - PERCDROP
  - SPACE/F
  - PQTY
  - SQTY
  - SECQTYI

aggregates

- SYSIBM.SYSINDEXSTATS_HIST
  - FIRSTKEYCARD/F
  - FULLKEYCARD/F
  - NLEAF
  - NLEVELS
  - IOFACTOR
  - PREFETCHFACTOR
  - KEYCOUNT/F
  - CLUSTERRATIO/F
  - FULLKEYCARDDATA
- SYSIBM.SYSINDEXPART_HIST
  - AVGKEYLEN
  - CARDF
  - DSNUM
  - EXTENTS
  - FAROFFPOSF
  - LEAFNEAR
  - LEAFFAR
  - NEAROFFPOS
  - LEAFDIST
  - PSUEDO_DEL_ENTRIES
  - SPACEF
  - PQTY
  - SECQTYI

- SYSIBM.SYSLOBSTATS_HIST
  - FREESPACE
  - ORGRATIO
  - AVGSIZE

aggregates

- SYSIBM.SYSCOLUMNS_HIST
  - COLCARD/F
  - HIGH2KEY
  - LOW2KEY
  - STATS_FORMAT
- SYSIBM.SYSCOLSTATS
  - COLCARD
  - HIGHKEY
  - HIGH2KEY
  - LOWKEY
  - LOW2KEY
  - COLCARDDATA
  - STATS_FORMAT
- SYSIBM.SYSCOLDIST_HIST
  - CARDF
  - COLGROUPCOLNO
  - COLVALUE
  - TYPE
  - FREQUENCY/F
  - NUMCOLUMNS

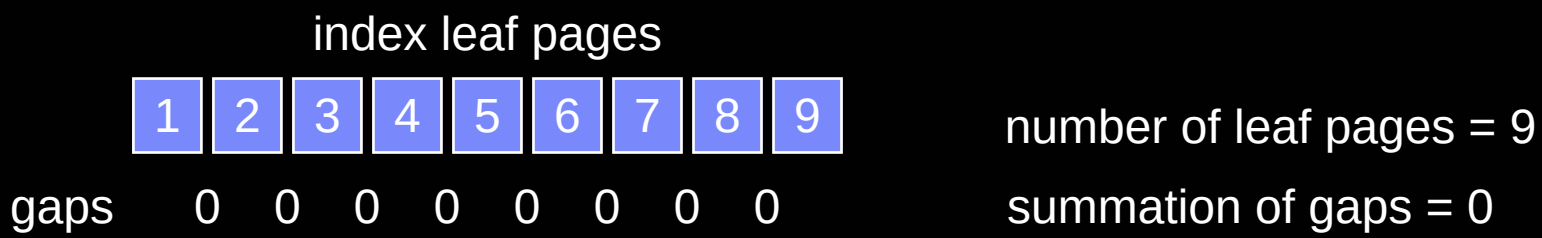
aggregates

- SYSIBM.SYSCOLDISTSTATS
  - CARDF
  - COLGROUPCOLNO
  - COLVALUE
  - TYPE
  - FREQUENCY/F
  - NUMCOLUMNS
  - KEYCARDDATA

Commonly asked questions about the stats

- What is SYSIBM.SYSINDEXPART.LEAFDIST?
  - LEAFDIST is 100 times the average number of pages between successive leaf pages of the index

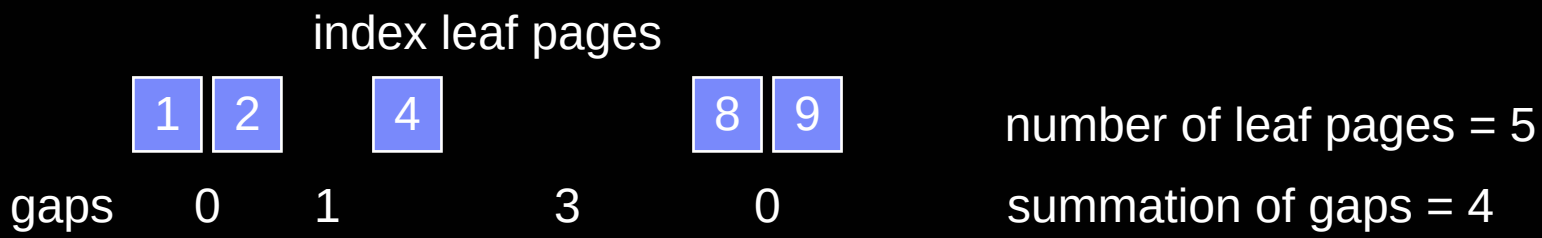
$$\text{LEAFDIST} = 100 \times \frac{\text{summation of distance between pages}}{\text{Number of leaf pages}}$$



LEAFDIST = 100 * (0/9) = 0 (%)

Commonly asked questions about the stats

- Another example of LEAFDIST



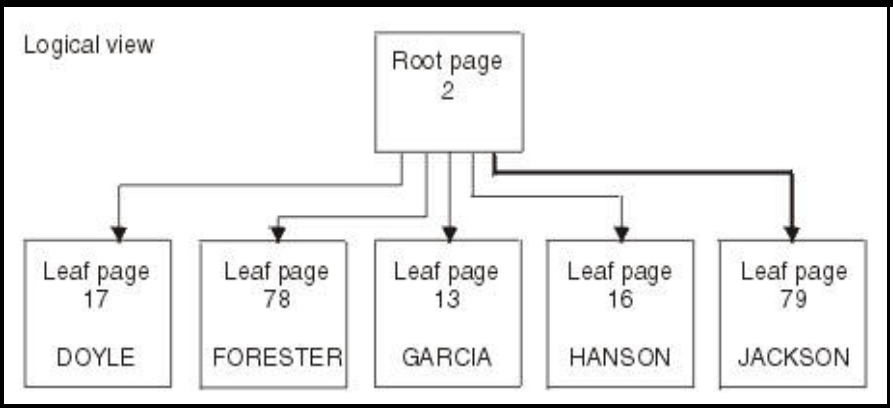
LEAFDIST = 100 * (4/5) = 80

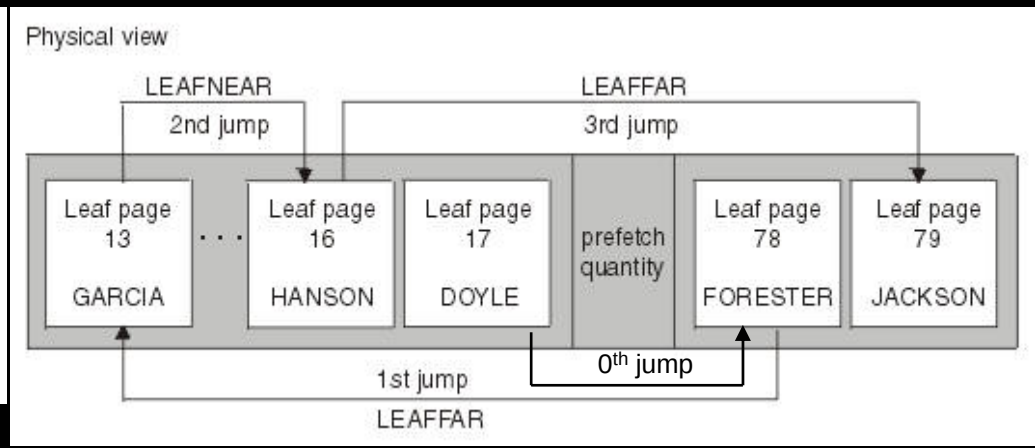
- If there were more gaps than active pages, LEAFDIST would be larger
- FREEPAGE on an index can certainly affect the calculation of LEAFDIST
- We used to use this value to determine when to reorg an index, but now we have better stats to determine this (LEAFFAR/NEAR)

Commonly asked questions about the stats

- What is SYSIBM.SYSINDEXPART.LEAFNEAR and LEAFFAR?
  - LEAFNEAR/FAR measure the disorganization of physical leaf pages
    - Number of pages that are not in an optimal position due to
      - index pages being deleted or
      - index leaf page splits caused by an insert that cannot fit onto a full page

Logical and physical views of an index in which LEAFNEAR=1 and LEAFFAR=3

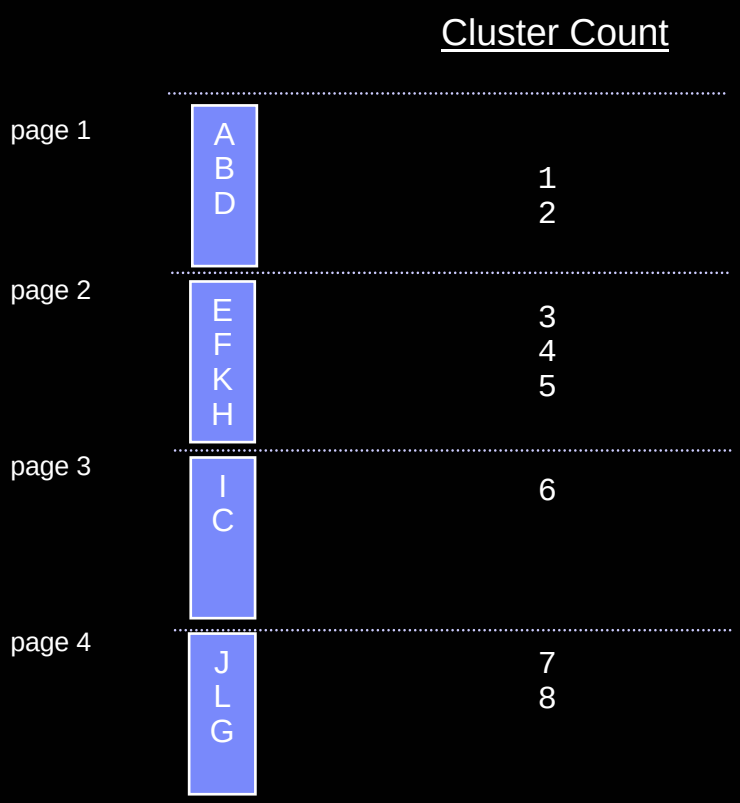


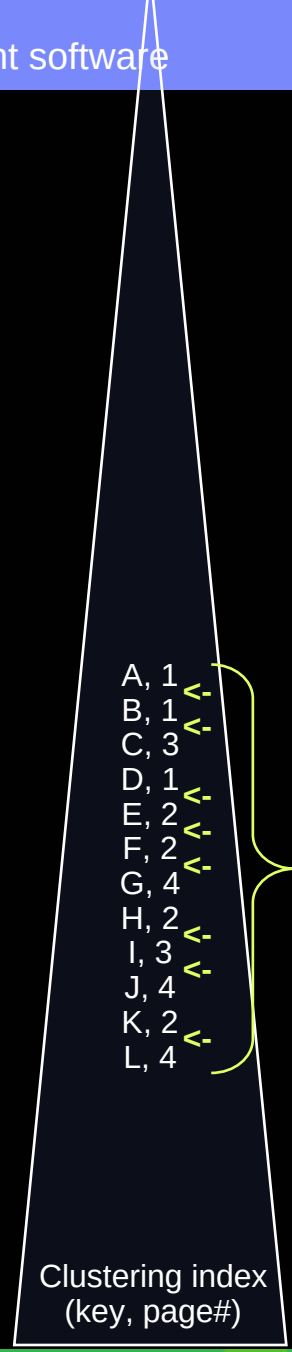


Commonly asked questions about the stats

- SYSIBM.SYSINDEXES.CLUSTERRATIO
  - An access path statistic that can also helps in determining when to reorg
  - % of the rows that are in cluster order
  - Rows are counted as being “clustered” if they are in a greater or equal page number of the previous row
  - This is a statistic that describes the data in the table(space), even though it is reported in SYSINDEXES – REORG INDEX will never affect this statistic

CLUSTERRATIO





CC incremented

8

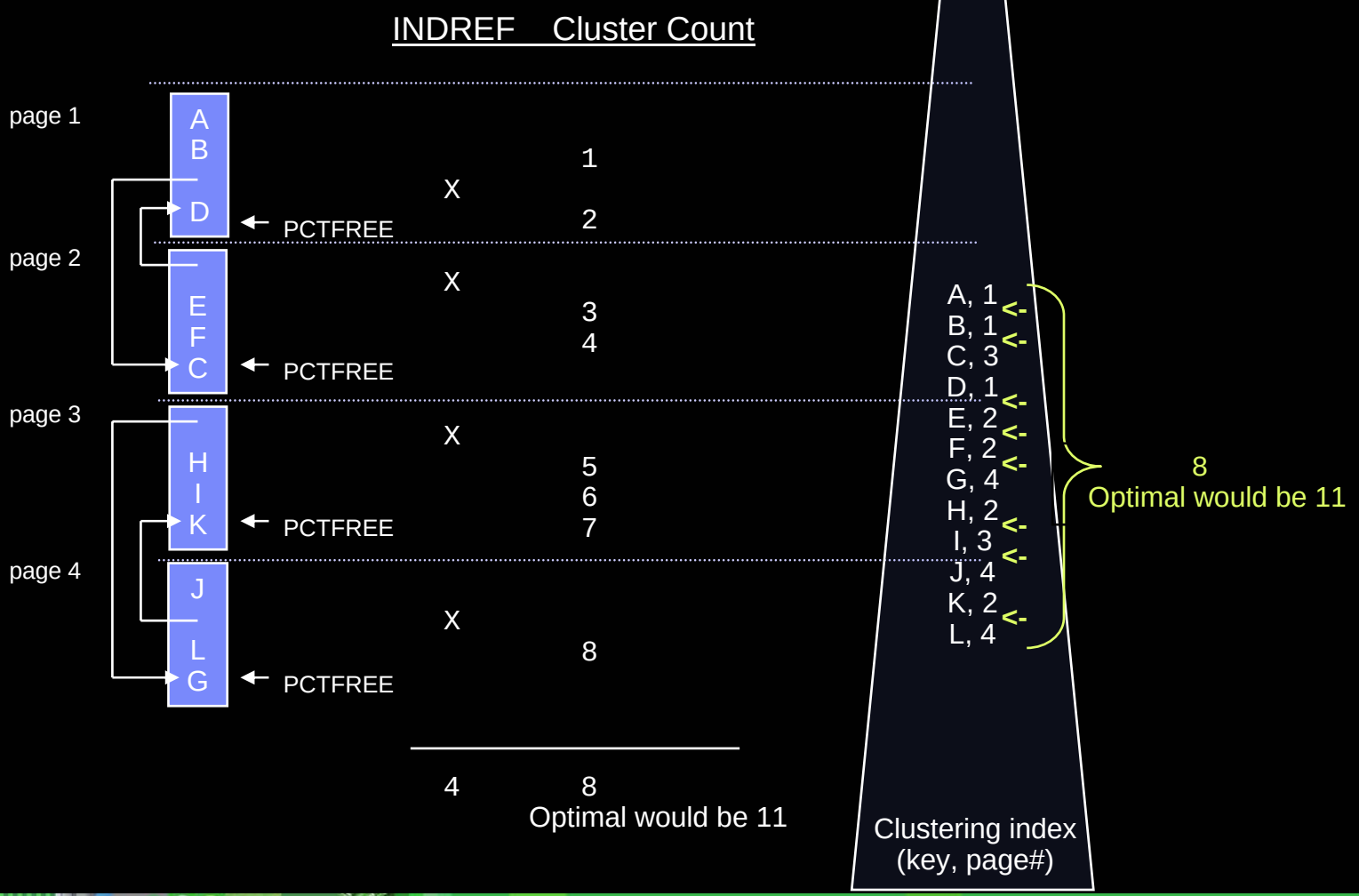
Optimal would be 11

Commonly asked questions about the stats

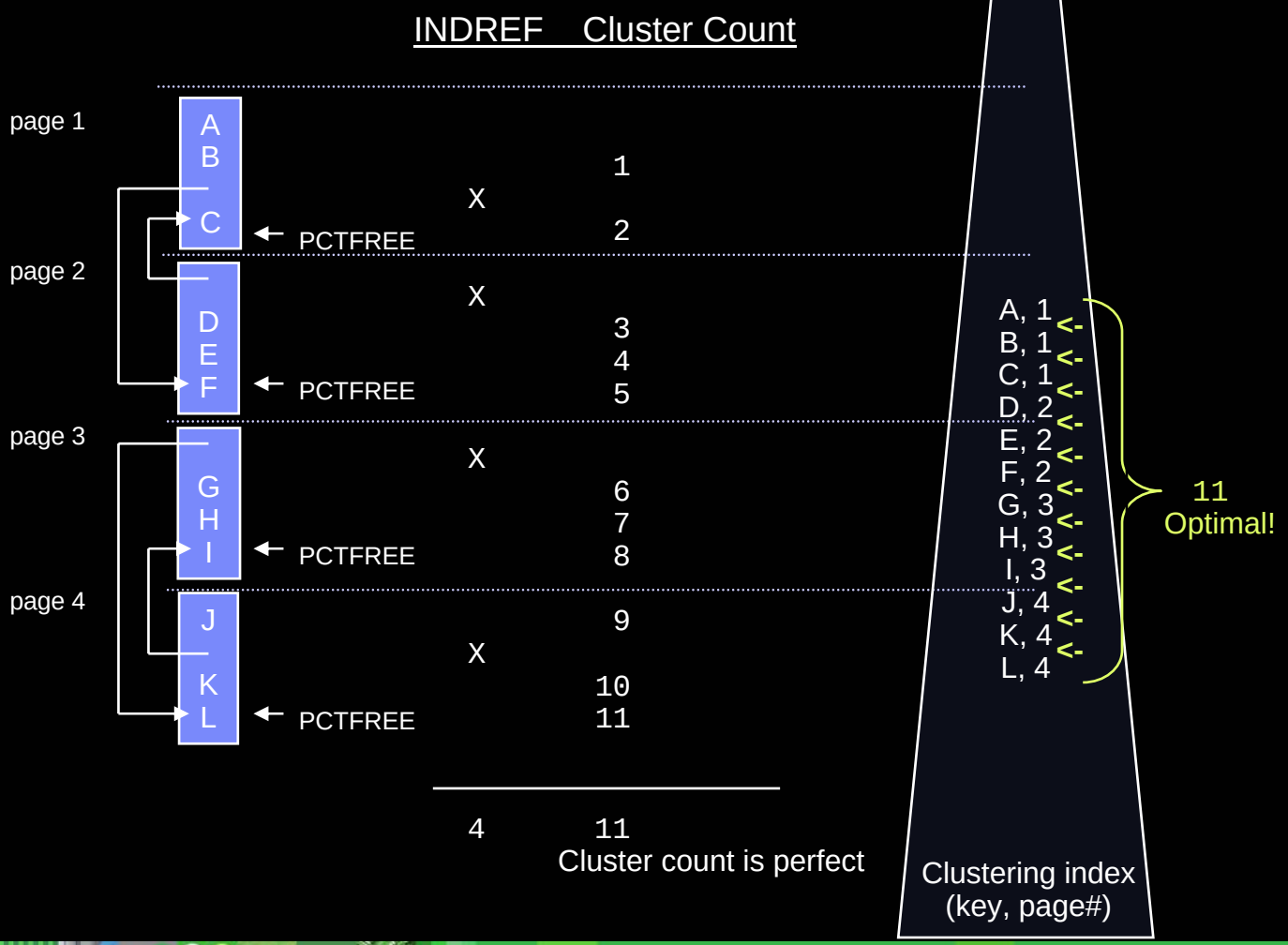
- How does NEAR|FAR INDREF and NEAR|FAR OFFPOS contribute to CLUSTERRATIO?
- *INDREF correlates closely with the cluster count if the keys are in cluster order and then rows are relocated to another page, but we can create cases where these stats are correlated and cases where they are not correlated
- *OFFPOS directly affects the cluster count. A single “jump” counts as two OFFPOS’, so the cluster count should be ½ of the sum of the *OFFPOS’.

- SYSIBM.SYSTABLEPART_HIST
  - NEARINDREF
  - FARINDREF
- SYSIBM.SYSINDEXES_HIST
  - CLUSTERRATIO/F
- SYSIBM.SYSINDEXPART_HIST
  - FAROFFPOSF
  - NEAROFFPOS

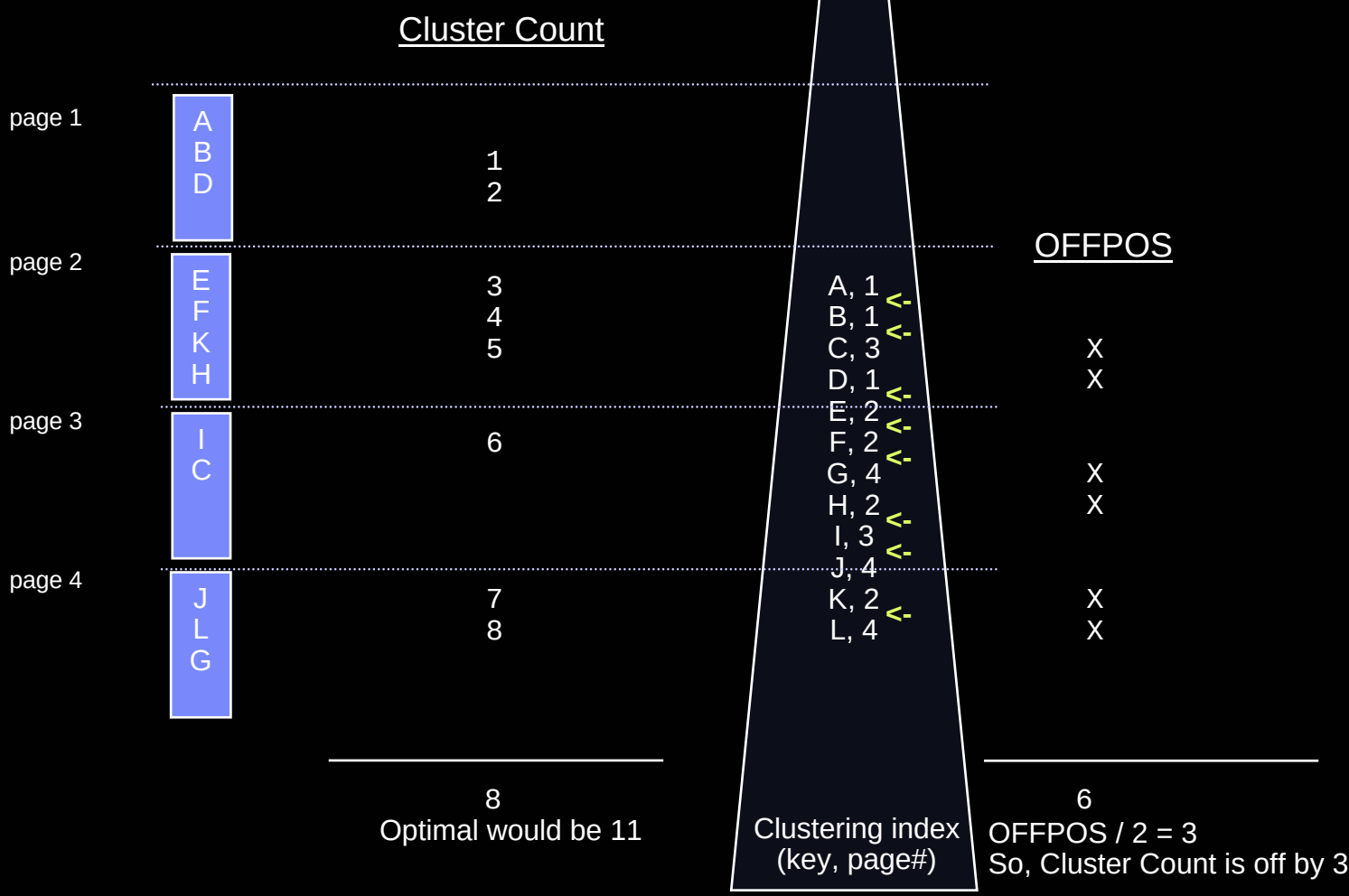
Example where INDREF is correlated with Cluster Count -> CLUSTERRATIO



Example where INDREF is not correlated with Cluster Count -> CLUSTERRATIO



Example where OFFPOS is correlated with Cluster Count -> CLUSTERRATIO



Exercise for the reader…

We just saw an example where *OFFPOS is correlated to the cluster count (which is used to compute CLUSTERATIO). Can an example be created showing non-correlation between these two metrics?

Commonly asked questions

- Can you collect stats and have them stored in the catalog without affecting any binds/prepares?
  - Yes (by specifying UPDATE NONE HISTORY ALL)
- Should you collect statistics on the DB2 Catalog?
  - Yes. Will it benefit DB2 processing like BIND or PREPARE?
    - No, …but SQL against the catalog can benefit
- Is there any difference between running

  RUNSTATS TABLESPACE DB1.TS1 INDEX (ALL)

  vs.

  RUNSTATS TABLESPACE DB1.TS1

  RUNSTATS INDEX(ALL) TABLESPACE DB1.TS1 -- ??
  - No, they are semantically equivalent, but you could run these two utility statements in parallel to reduce overall elapsed time
- Can/should you update the statistics in the DB2 Catalog?
  - It depends
- What is the semantic difference between RUNSTATS TABLESPACE and RUNSTATS TABLESPACE TABLE (ALL)?
  - The TABLE keyword triggers collection of column statistics

Rebinding considerations

- Consider the following guidelines regarding when to rebind
  - CLUSTERRATIOF changes to less or more than 80% (a value of 0.80)
  - NLEAF changes more than 20% from the previous value
  - NLEVELS changes
  - NPAGES changes more than 20% from the previous value
  - NACTIVEF changes more than 20% from the previous value
  - The range of HIGH2KEY to LOW2KEY range changes more than 20% from the range previously recorded
  - Cardinality changes more than 20% from previous range
  - Distribution statistics change the majority of the frequent column values

Reorg recommendations

- These are generic and do not apply in all cases – there is no absolutely reliable statistic as to when reorganization of table spaces or indexes should occur; however, understanding the rules of thumb will help in understanding data disorganization
- If reorg for performance, then track performance over time
- DSNACCOR usage

Reorg table space (incl. LOB) recommendations

- Consider running REORG TABLESPACE in the following situations:
  - Real-time statistics (TABLESPACESTATS)
    - (REORGNEARINDREF+REORGFARINDREF (number of overflow rows since the last Reorg))/TOTALROWS > 5% in data sharing, >10% in non-data sharing
    - REORGUNCLUSTINS (number of records inserted since the last Reorg that are not well-clustered)/TOTALROWS > 10%
      - Irrelevant if predominantly random access
      - REORGUNCLUSTINS is only an indication of the insert behavior and is correlated to the cluster ratio only if there are no updates or deletes. To prevent DSNACCOR from triggering on these, identify such objects and put them in exception list
    - REORGINSERTS (number of records inserted since the last Reorg)/TOTALROWS > 25%
    - REORGDELETES (number of records deleted since the last Reorg)/TOTALROWS > 25%
    - EXTENTS (number of extents) > 254
    - REORGDISORGLOB (number of LOBs inserted since the last Reorg that are not perfectly chunked)/TOTALROWS > 10%
    - REORGMASSDELETE > 0 (set for mass deletes on seg tsps and on DROP on multi-table tsps)
  - RUNSTATS
    - PERCDROP > 10%
    - SYSIBM.SYSLOBSTATS.ORGRATIO < 50% (changed to a value 0-100 in PQ96460 on V7/V8)
    - (NEARINDREF + FARINDREF) / CARDF > 10% non-data-sharing, > 5% if data sharing
    - FAROFFPOSF / CARDF > 10%
      - Or, if index is a clustering index, CLUSTERRATIOF < 90% (irrelevant if predominantly random access)
  - Other
    - The table space is in advisory or informational reorg pending status as the result of an ALTER TABLE statement
    - An index on the table space is in advisory-REBUILD-pending stats (ARBDP) as the result of an ALTER statement

Reorg table space (incl. LOB) recommendations

- Consider running REORG TABLESPACE in the following situations:
  - Real-time statistics (TABLESPACESTATS)
    - (REORGNEARINDREF+REORGFARINDREF (number of overflow rows since the last Reorg))/TOTALROWS > 5% in data sharing, >10% in non-data sharing
    - REORGUNCLUSTINS (number of records inserted since the last Reorg that are not well-clustered)/TOTALROWS > 10%
      - Irrelevant if predominantly random access
      - REORGUNCLUSTINS is only an indication of the insert behavior and is correlated to the cluster ratio only if there are no updates or deletes. To prevent DSNACCOR from triggering on these, identify such objects and put them in exception list
    - REORGINSERTS (number of records inserted since the last Reorg)/TOTALROWS > 25%
    - REORGDELETES (number of records deleted since the last Reorg)/TOTALROWS > 25%
    - EXTENTS (number of extents) > 254
    - REORGDISORGLOB (number of LOBs inserted since the last Reorg that are not perfectly chunked)/TOTALROWS > 10%
    - REORGMASSDELETE > 0 (set for mass deletes on seg tsps and on DROP on multi-table tsps)
  - RUNSTATS
    - PERCDROP > 10%
    - SYSIBM.SYSLOBSTATS.ORGRATIO < 50% (changed to a value 0-100 in PQ96460 on V7/V8)
    - (NEARINDREF + FARINDREF) / CARDF > 10% non-data-sharing, > 5% if data sharing
    - FAROFFPOSF / CARDF > 10%
      - Or, if index is a clustering index, CLUSTERRATIOF < 90% (irrelevant if predominantly random access)
  - Other
    - The table space is in advisory or informational reorg pending status as the result of an ALTER TABLE statement
    - An index on the table space is in advisory-REBUILD-pending stats (ARBDP) as the result of an ALTER statement

V9 RTS

Reorg table space (incl. LOB) recommendations

- Consider running REORG TABLESPACE in the following situations:
  - Real-time statistics (TABLESPACESTATS)
    - (REORGNEARINDREF+REORGFARINDREF (number of overflow rows since the last Reorg))/TOTALROWS > 5% in data sharing, >10% in non-data sharing
    - REORGUNCLUSTINS (number of records inserted since the last Reorg that are not well-clustered)/TOTALROWS > 10%
      - Irrelevant if predominantly random access
      - REORGUNCLUSTINS is only an indication of the insert behavior and is correlated to the cluster ratio only if there are no updates or deletes. To prevent DSNACCOR from triggering on these, identify such objects and put them in exception list
    - REORGINSERTS (number of records inserted since the last Reorg)/TOTALROWS > 25%
    - REORGDELETES (number of records deleted since the last Reorg)/TOTALROWS > 25%
    - EXTENTS (number of extents) > 254
    - REORGDISORGLOB (number of LOBs inserted since the last Reorg that are not perfectly chunked)/TOTALROWS > 10%
    - REORGMASSDELETE > 0 (set for mass deletes on seg tsps and on DROP on multi-table tsps)
  - RUNSTATS
    - PERCDROP > 10%
    - SYSIBM.SYSLOBSTATS.ORGRATIO < 50% (changed to a value 0-100 in PQ96460 on V7/V8)
    - (NEARINDREF + FARINDREF) / CARDF > 10% non-data-sharing, > 5% if data sharing
    - FAROFFPOSF / CARDF > 10%
      - Or, if index is a clustering index, CLUSTERRATIOF < 90% (irrelevant if predominantly random access)
  - Other
    - The table space is in advisory or informational reorg pending status as the result of an ALTER TABLE statement
    - An index on the table space is in advisory-REBUILD-pending stats (ARBDP) as the result of an ALTER statement

V9 RTS

Reorg table space (incl. LOB) recommendations

- Consider running REORG TABLESPACE in the following situations:
  - Real-time statistics (TABLESPACESTATS)
    - (REORGNEARINDREF+REORGFARINDREF (number of overflow rows since the last Reorg))/TOTALROWS > 5% in data sharing, >10% in non-data sharing
    - REORGUNCLUSTINS (number of records inserted since the last Reorg that are not well-clustered)/TOTALROWS > 10%
      - Irrelevant if predominantly random access
      - REORGUNCLUSTINS is only an indication of the insert behavior and is correlated to the cluster ratio only if there are no updates or deletes. To prevent DSNACCOR from triggering on these, identify such objects and put them in exception list
    - REORGINSERTS (# of records inserted since the last Reorg)/TOTALROWS > 25%
    - REORGDELETES (# of records deleted since the last Reorg)/TOTALROWS > 25%
    - EXTENTS (number of extents) > 254
    - REORGDISORGLOB (number of LOBs inserted since the last Reorg that are not perfectly chunked)/TOTALROWS > 10%
    - REORGMASSDELETE > 0 (set for mass deletes on seg tsps and on DROP on multi-table tsps)
  - Other
    - Tsp is in adv or info reorg pending status as result of an ALTER TABLE statement
    - Index on the tsp is in adv-REBUILD-pend state (ARBDP) as result an ALTER stmnt

Reorg index recommendations

- Consider running REORG INDEX in the following cases:
  - Real-time statistics (INDEXSPACESTATS)
    - REORGPSEUDODELETES (number of index entries pseudo-deleted since the last Reorg)/TOTALENTRIES > 10% in non-data sharing, 5% if data sharing as pseudo-deleted entry can cause S-lock/unlock in Insert for unique index
    - REORGLEAFFAR (number of index leaf page splits since the last Reorg and the new leaf page far from the original leaf page)/NACTIVE > 10%
    - REORGINSERTS (number of index entries inserted since the last Reorg)/TOTALENTRIES > 25%
    - REORGDELETES (number of index entries inserted since the last Reorg)/TOTALENTRIES > 25%
    - REORGAPPENDINSERT / TOTALENTRIES > 20%
    - EXTENTS (number of extents) > 254
  - RUNSTATS
    - LEAFFAR / NLEAF > 10% (NLEAF is a column in SYSIBM.SYSINDEXES and SYSIBM.SYSINDEXPART)
    - PSEUDO_DEL_ENTRIES / CARDF > 10% for non-data sharing and > 5% for data sharing
  - Other
    - The index is in advisory REORG-pending status (AREO*) or advisory-REBUILD-pending status (ARBDP) as the result of an ALTER statement

Reorg index recommendations

- Consider running REORG INDEX in the following cases:
  - Real-time statistics (INDEXSPACESTATS)
    - REORGPSEUDODELETES (number of index entries pseudo-deleted since the last Reorg)/TOTALENTRIES > 10% in non-data sharing, 5% if data sharing as pseudo-deleted entry can cause S-lock/unlock in Insert for unique index
    - REORGLEAFFAR (number of index leaf page splits since the last Reorg and the new leaf page far from the original leaf page)/NACTIVE > 10%
    - REORGINSERTS (number of index entries inserted since the last Reorg)/TOTALENTRIES > 25%
    - REORGDELETES (number of index entries inserted since the last Reorg)/TOTALENTRIES > 25%
    - REORGAPPENDINSERT / TOTALENTRIES > 20%
    - EXTENTS (number of extents) > 254
  - RUNSTATS
    - LEAFFAR / NLEAF > 10% (NLEAF is a column in SYSIBM.SYSINDEXES and SYSIBM.SYSINDEXPART)
    - PSEUDO_DEL_ENTRIES / CARDF > 10% for non-data sharing and > 5% for data sharing
  - Other
    - The index is in advisory REORG-pending status (AREO*) or advisory-REBUILD-pending status (ARBDP) as the result of an ALTER statement

Reorg index recommendations

- Consider running REORG INDEX in the following cases:
  - Real-time statistics (INDEXSPACESTATS)
    - REORGPSEUDODELETES (number of index entries pseudo-deleted since the last Reorg)/TOTALENTRIES > 10% in non-data sharing, 5% if data sharing as pseudo-deleted entry can cause S-lock/unlock in Insert for unique index
    - REORGLEAFFAR (number of index leaf page splits since the last Reorg and the new leaf page far from the original leaf page)/NACTIVE > 10%
    - REORGINSERTS (number of index entries inserted since the last Reorg)/TOTALENTRIES > 25%
    - REORGDELETES (number of index entries inserted since the last Reorg)/TOTALENTRIES > 25%
    - REORGAPPENDINSERT / TOTALENTRIES > 20%
    - EXTENTS (number of extents) > 254
  - ~~RUNSTATS~~
    - ~~LEAFFAR / NLEAF > 10% (NLEAF is a column in SYSIBM.SYSINDEXES and SYSIBM.SYSINDEXPART)~~
    - ~~PSEUDO_DEL_ENTRIES / CARDF > 10% for non-data sharing and > 5% for data sharing~~
  - Other
    - The index is in advisory REORG-pending status (AREO*) or advisory-REBUILD-pending status (ARBDP) as the result of an ALTER statement

When is RUNSTATS needed?

- When the data changes sufficiently to warrant new statistics
  - REORG of tbsp or index (use inline stats if data changes significantly)
  - LOAD REPLACE of tbsp (use inline stats if data changes significantly)
  - After "significant" application changes for the tablespace or index
    - Periodically (weekly, monthly) except for read only data?
    - Application tracks updates with activity tables?
    - After percentage of pages changed since last RUNSTATS (RTS)?
- Understand implications for access paths!
- SHRLEVEL
  - REFERENCE drains writers
  - CHANGE runs like application with ISOLATION (UR)

    (claim reader for allocation duration)

New/Changed Data Statistics (V8)

- SPACEF at the table space level
  - 4096 partitions can hold a lot of data!
- HIGHKEY/HIGH2KEY/LOWKEY/LOW2KEY expanded
  - From CHAR(8) to VARCHAR(2000)
    - 8 bytes not adequate for multi-byte character representations especially with Unicode
  - Optimizer has better information to estimate filter factors and determine access paths
- AVGROWLEN at the table space/partition level
  - V7 only collected at the table level
  - Useful for estimating current number of rows of table space from file size without having to run RUNSTATS
  - Conversely, can calculate table space size allocation more accurately
  - UNLOAD utility space allocation
  - REORG & LOAD space allocation
    - work datasets
    - sort space

- SYSIBM.SYSTABLESPACE
  - AVGROWLEN
  - SPACEF
- SYSIBM.SYSTABLEPART_HIST
  - AVGROWLEN

Calculating table space size with AVGROWLEN

- For example:
  - Number of records = 100000
  - Maximum record size = 130 bytes
  - Average record size = 80 bytes
  - Page size = 4 KB
  - PCTFREE = 5
  - FREEPAGE = 20
  - MAXROWS = 255

Calculating table space size with AVGROWLEN

- Using the maximum row size, you get the following results:
  - Usable page size = 4074 × 0.95 = 3870 bytes
  - Records per page = MIN(MAXROWS, FLOOR(3870 / 130)) = 29
  - Pages used = 2 + CEILING(100000 / 29) = 3451
  - Total pages = FLOOR(3451 × 21 / 20) = 3624
  - Estimated number of kilobytes = 3624 × 4 = 14496 KB
- Or using the AVGROWLEN
  - Usable page size = 4074 × 0.95 = 3870 bytes
  - Records per page = MIN(MAXROWS, FLOOR(3870 / 80)) = 48
  - Pages used = 2 + CEILING(100000 / 48) = 2085
  - Total pages = FLOOR(2085 × 21 / 20) = 2189
  - Estimated number of kilobytes = 2189 × 4 = 8756 KB

A difference of 5740MB

New/Changed Index Statistics

- AVGKEYLEN at the index/partition level
  - Needed for non-padded indexes
  - Useful for estimating current cardinality of index space without having to run RUNSTATS
  - Conversely, can calculate index space size allocation more accurately

- SYSIBM.SYSINDEXES_HIST
  - AVGKEYLEN

- SYSIBM.SYSINDEXPART_HIST
  - AVGKEYLEN

Calculating index space size with AVGKEYLEN

- For example:
  - Unique Index NOT PADDED
  - Number of records in table = 100000
  - Key is a single column defined as VARCHAR(100) NOT NULL
  - Max Key Len = 102
  - AVGKEYLEN = 62
  - PCTFREE = 5.
  - FREEPAGE = 4.

Calculating index space size using max

- Calculate total leaf pages using max key size
  - Space per key = 102 + 7 = 109
  - Usable space per page = FLOOR((100 - 5) × 4038/100) = 3844
  - Entries per page = FLOOR(3844 / 109) = 35
  - Total leaf pages = CEILING(100000 / 35) = 2858
- Calculate total nonleaf pages using max key size
  - Space per key = 102 + 7 = 109
  - Usable space per page = FLOOR(MAX(90, (100 - 5)) × (4046/100) = 3836
  - Entries per page = FLOOR(3836 / 109) = 35
  - Minimum child pages = MAX(2, (35 + 1)) = 36
  - Level 2 pages = CEILING(2858 / 36) = 80
  - Level 3 pages = CEILING(80 / 36) = 3
  - Level 4 pages = CEILING(3 / 36) = 1
  - Total nonleaf pages = (80 + 3 + 1) = 84
- Calculate total space required using max key size
  - Free pages = FLOOR(2585 / 4) = 646
  - Tree pages = MAX(2, (2585 + 84)) = 2669
  - Space map pages = CEILING((2669 + 646)/8131) = 1
  - Total index pages = MAX(4, (1 + 2669 + 646 + 1)) = 3317
- TOTAL SPACE REQUIRED, in KB = 4 × (3317 + 2) = 13276 KB

Calculating index space size using AVGKEYLEN

- Calculate total leaf pages using AVGKEYLEN
  - Space per key = 102 + 7 = 69
  - Usable space per page = FLOOR((100 - 5) × 4038/100) = 3844
  - Entries per page = FLOOR(3844 / 69) = 55
  - Total leaf pages = CEILING(100000 / 55) = 1819
- Calculate total nonleaf pages using AVGKEYLEN
  - Space per key = 62 + 7 = 69
  - Usable space per page = FLOOR(MAX(90, (100 - 5)) × (4046/100) = 3836
  - Entries per page = FLOOR(3836 / 69) = 55
  - Minimum child pages = MAX(2, (55 + 1)) = 56
  - Level 2 pages = CEILING(2858 / 56) = 56
  - Level 3 pages = CEILING(80 / 56) = 1
  - Total nonleaf pages = (56 + 1) = 57
- Calculate total space required using AVGKEYLEN
  - Free pages = FLOOR(1819 / 4) = 454
  - Tree pages = MAX(2, (1819 + 57)) = 1876
  - Space map pages = CEILING((1819 + 57)/8131) = 1
  - Total index pages = MAX(4, (1 + 1819 + 57 + 1)) = 2332
- TOTAL SPACE REQUIRED, in KB = 4 × (2332 + 2) = 9336 KB

3 instead of 4!

A difference of 3940KB

Part level statistics for DPSIs

- Statistics are not kept at the partition level for logical partitions of NPIs
- Data Partitioned Secondary Indexes need to have the same partition independence and capabilities (from a statistics gathering perspective) as classic partitioning indexes.
- Partition level statistics for DPSIs are stored in SYSCOLDISTSTATS with rollup to SYSCOLDIST
- Rollup requires SYSCOLDISTSTATS rows to be sorted requiring new parameters
  - SORTDEVT (defaults to SYSALLDA)
  - SORTNUM
- If not specified then SORT will use sort product defaults
- Can also use FORCEROLLUP to aggregate partition level statistics when not all partitions have statistics

Distribution Statistics Enhanced

- As queries become…
  - more complex
  - less predictable

  …Data skew becomes more important
- Problem with skewed data and regular statistics
  - Optimizer assumes inaccurate distribution of values
  - Less efficient join sequence could be chosen
  - Less efficient method of accessing individual tables
- DSTATS program could be downloaded to collect statistical data for non-indexed columns
  - Great improvement in access path selection, however
  - Run separate from RUNSTATS
  - Slow with big impact to DB2 work file database

Filter factors and catalog statistics

- SYSCOLDIST contains frequency (or distribution)
- If frequency statistics do not exist, DB2 assumes that the data is uniformly distributed
- For example:

AGE_CATEOGRY	FREQUENCY
INFANT	5%
CHILD	15%
ADOLESCENT	25%
ADULT	40%
SENIOR	15%

- Default filter factor is 1/5 (1/COLCARDF), or 20%, to estimate the number of rows that qualify for any predicate value.
- For AGE_CATEGORY='ADULT' underestimate by 50%.
- For AGE_CATEGORY='INFANT' overestimate by 400%.

Distribution Statistics Enhanced

- Non-uniform distribution statistics on non-indexed columns
  - Now part of RUNSTATS
  - Significant performance improvement - no impact on DB2 work file and data only has to be scanned once
  - Uses external sort requiring new parameters
    - SORTDEVT
    - SORTNUM
    - If not specified then SORT will use sort product defaults
- Extend non-uniform to collect on index or non-index
  - most frequent values
  - least frequent values
  - both
- As part of this, the previous limit of 10 names in the COLUMN parameter has been removed.

Example using new RUNSTATS Distribution Statistics

- Select all conference attendees that meet the following characteristics:
  - Gender = Female
  - Status = Single
  - Age = 38

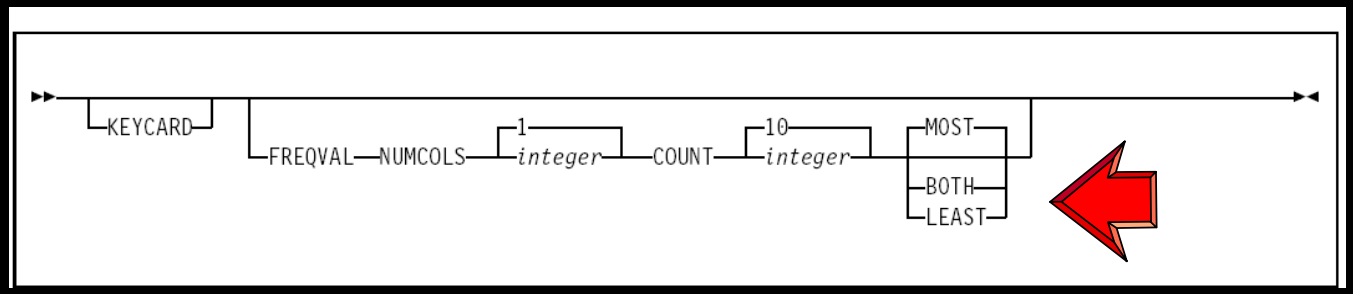
Example using new RUNSTATS Distribution Statistics…

	Pred	Dflt FF	Col card	Calc FF	NUD FF
Gender	=	1/25	2	1/2	2/5
Status	=	1/25	2	1/2	2/15
Age	=	1/25	40	1/40	1/4
Est * Card		<1		9	20

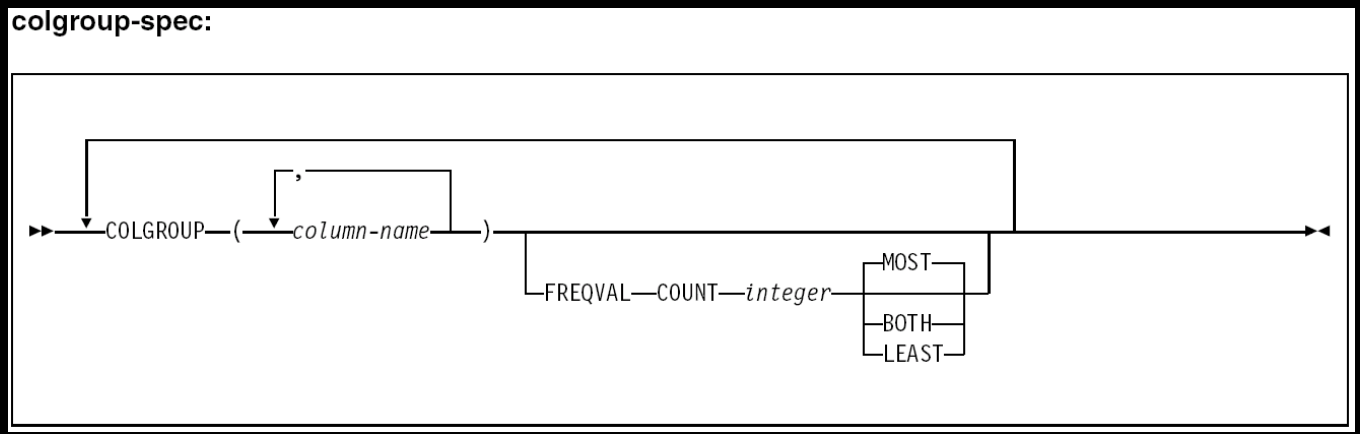
Table cardinality = 1500

Distribution Statistics Enhanced

- Changed/new syntax



RUNSTATS INDEX
REBUILD, REORG INDEX



RUNSTATS TABLESPACE

Distribution Statistics Enhanced

- Example: Collect distribution statistics for specific columns in a table space and retrieve the most and least frequently occurring values. Collect statistics for the columns EMPLEVEL, EMPGRADE, and EMPSALARY and use the FREQVAL and COUNT keywords to collect the 10 most frequently occurring values for each column and the 10 least frequently occurring values for each column.
- RUNSTATS TABLESPACE DSN8D81A.DSN8S81E

  TABLE(DSN8810.DEPT)

  COLGROUP(EMPLEVEL,EMPGRADE,EMPSALARY)

  FREQVAL COUNT 10 BOTH

Distribution Statistics Enhanced

- Example: Collect distribution statistics for specific columns in a table space and retrieve the most and least frequently occurring values. Collect statistics for the columns EMPLEVEL, EMPGRADE, and EMPSALARY and use the FREQVAL and COUNT keywords to collect the 10 most frequently occurring values for each column and the 10 least frequently occurring values for each column.
- RUNSTATS TABLESPACE DSN8D81A.DSN8S81E

  TABLE(DSN8810.DEPT)

  COLGROUP(EMPLEVEL,EMPGRADE,EMPSALARY)

  FREQVAL COUNT 10 BOTH

Not *currently* collected via in-line statistics from LOAD and REORG

Sorting

- SORT occurs in the following cases for RUNSTATS
  - Distribution (frequency) statistics for RUNSTATS TABLESPACE TABLE COLGROUP processing
    - An additional SORT occurs if FREQVAL is specified along with COLGROUP
  - In the case of INDEX processing for DPSI type index, SORT occurs during FREQVAL processing

HISTORY statistics without updating main statistics

- V7 required update of main catalog statistics if history statistics were wanted
- V8 relaxes this and history statistics can now be kept without updating current statistics.
  - Monitor statistics such as SYSTABLES.CARDF
  - No surprises for dynamic SQL access paths
  - CAUTION: If you use this you have to be remember that your static packages bound in that time frame may not have used the statistics in the history tables.
- For example,
  - in V7 UPDATE NONE HISTORY OPTIMIZER was prohibited.
  - in V8 UPDATE NONE HISTORY OPTIMIZER is allowed and you can monitor statistics changes over time without concern that access paths may change.

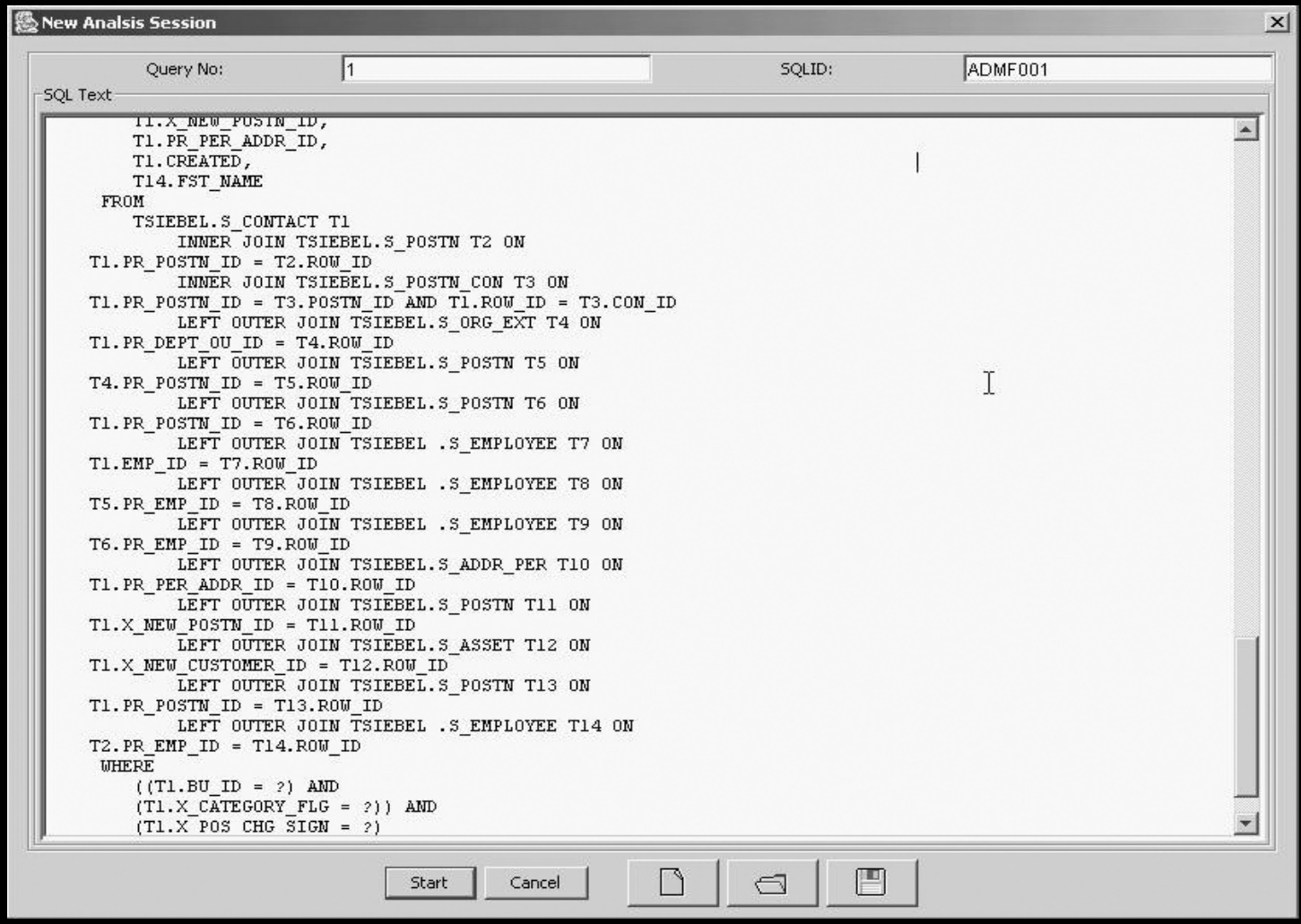
Flushing the dynamic statement cache

- RUNSTATS with UPDATE NONE REPORT NO
- Any statement in the Dynamic Statement Cache which is dependent on the affected table space or index space will be removed from the cache.
- Why? If users manually update the statistics in the catalog tables, the related dynamic SQL in the cache needs to be invalidated and the next prepare of the statements will cause the access paths to be reevaluated.
- Granularity is at the table space/index level (not the table level)

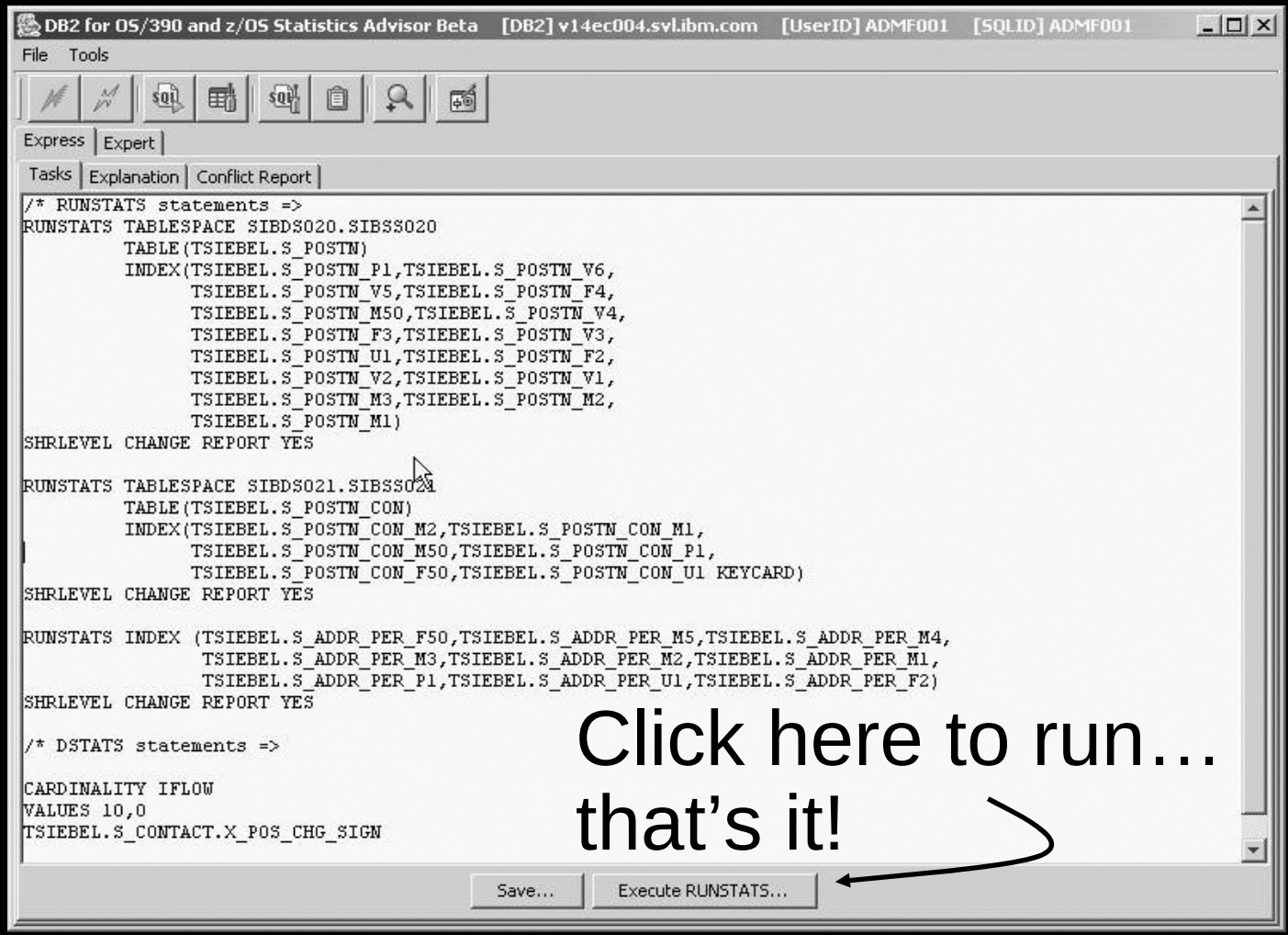
What statistics should I gather?

- No simple answer
  - Some collect no or insufficient statistics
    - Prime reason for poor performing access paths
  - Do you want to collect statistics on every column and permutations of combination of columns?
    - No way!
- Requires similar analysis of SQL as for index design
  - Have to include columns which you may not benefit from adding to an index
  - Analysis of queries labor intensive
  - Iterative process analyzing explain data (as always)

Input SQL, Click start

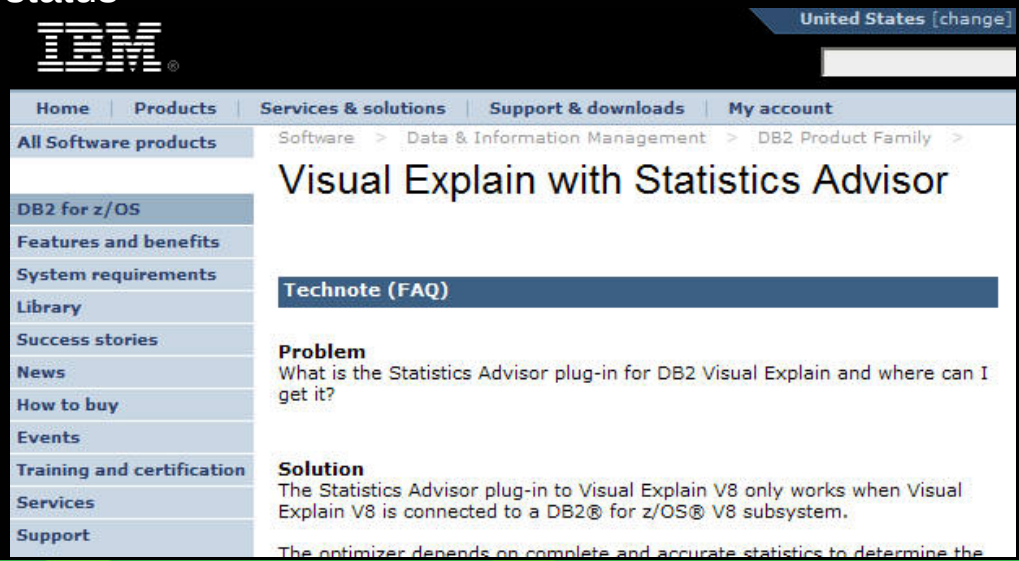


Suggestions for one Siebel query



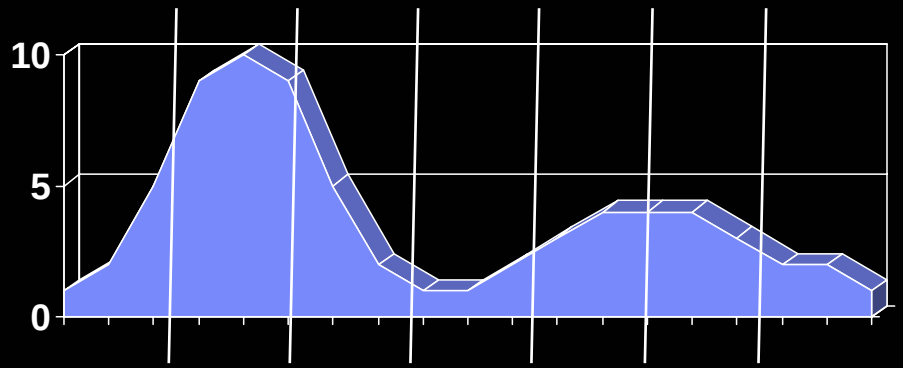
Statistics Advisor Current Status

- Statistics Advisor is integrated with VE now as a no-charge item
- Used as a serviceability tool
  - Service team use prototype on real problems
  - Demonstrates research of automation of query analysis
- Identifying, addressing areas of improvement
  - Move forward from prototype status



DB2 V9 for z/OS Changes to RUNSTATS

- New histogram statistics
  - Think of these as frequency distribution statistics on a range of data
  - Ideal for numeric, date, and time data types



- CPU reduction

Summary

- Why RUNSTATS?
- Commonly asked questions (about the stats)
- Rebinding considerations
- Reorg recommendations
- When is RUNSTATS needed?
- New/changed data statistics
- New/changed index statistics
- Handling part level statistics for DPSIs
- Distribution Statistics Enhanced
- HISTORY statistics without updating main statistics
- Flushing the dynamic statement cache
- What statistics should I gather?

IBM Information On Demand 2006
ibm.com/events/InformationOnDemand

IBM Information On Demand 2006
October 15-20, 2006
Anaheim, California

- The premier information management event for business and IT executives, managers, professionals, DBA's and developers.
- Select from over 800 sessions: a 2 1/2 day business leadership track with 180 sessions and a 5 day technical track with 650 sessions.
- Latest strategy and product announcements
- Large Expo Center
- Hands on labs
- One on ones with executives and specialists
- Birds of a Feather roundtables

Why attend:

- Participate in the PREMIER discussion on the future of Information Management
- Learn how the transformation to Information as a Service will help you unlock business value and drive competitive advantage
- Hear how your peers are realizing ROI
- Understand the roadmap to long term strategic advantage
- Learn best practices in your industry
- Receive the best in technical education and free certification
- Extensive opportunities for networking with both your peers and industry experts

TAKE BACK CONTROL

References

- DB2 UDB for z/OS home page
  http://www.software.ibm.com/data/db2/os390/
- utilities@work
  http://www.ibm.com/software/data/db2imstools/details/html/us_text.html
- The IDUG Solutions Journal March 1999 - Volume 6, Number 1
  Improving DB2 for OS/390 Query Performance with DSTATS By Steve Bower
  http://www.idug.org/neo_apps/cfmfiles/mainnavbar.cfm?body=/journal/index.html
- DB2 UDB for z/OS and OS/390 Version 7 Performance Topics, SG24-6129
- DB2 UDB for z/OS and OS/390 Version 7: Using the Utilities Suite, SG24-6289
- DB2 UDB for z/OS Version 8 What's New
  http://www-3.ibm.com/software/data/db2/os390/v8/dsnwnj1.pdf
- DB2 UDB for z/OS Version 8 Administration Guide
- DB2 UDB for z/OS Version 8 Utilities Guide and Reference

DB2 UDB for z/OS information resources

- Information center
  http://publib.boulder.ibm.com/infocenter/dzichelp/index.jsp
- Information roadmap
  http://ibm.com/software/db2zos/roadmap.html
- DB2 UDB for z/OS library page
  http://ibm.com/software/db2zos/library.html
- Examples trading post
  http://ibm.com/software/db2zos/exHome.html
- DB2 for z/OS support
  http://ibm.com/software/db2zos/support.html
- Official Introduction to DB2 for z/OS
  http://ibm.com/software/data/education/bookstore

Disclaimers & Trademarks*

Information in this presentation about IBM's future plans reflect current thinking and is subject to change at IBM's business discretion.

You should not rely on such information to make business plans. Any discussion of OEM products is based upon information which has been publicly available and is subject to change.

The following terms are trademarks or registered trademarks of the IBM Corporation in the United States and/or other countries: AIX, AS/400, DATABASE 2, DB2, OS/390, OS/400, ES/9000, MVS/ESA, Netfinity, RISC, RISC SYSTEM/6000, SYSTEM/390, SQL/DS, VM/ESA, IBM, Lotus, NOTES. The following terms are trademarks or registered trademarks of the MICROSOFT Corporation in the United States and/or other countries: MICROSOFT, WINDOWS, ODBC

Bryan F. Smith

IBM

bfsmith@us.ibm.com