IBM

Data Management

Tuning Queries Like a DB2 Developer

Jase Alpers
IBM Silicon Valley Lab

Disclaimer

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS AND/OR SOFTWARE.

HOWTO Analyze a query performance problem

- Demonstrate an approach to analyzing query performance problems
  - How to prepare a query for analysis
  - What information to collect
  - How to begin analysis of a query
    - Table, predicate analysis
    - Access path choices analysis
- Use walk-through of real world examples
- Highlight tools which automate some of the preparation
- Ultimate goal:
  - Provide a good starting point for how to approach query tuning

Agenda

- Preparing a query for analysis
- Primary cause of access path problems
- Query walkthrough
- Verify selectivity estimates

Preparing a query for analysis

- Format the query
- Annotate the query
- Automation available

Why format SQL?

- The optimizer doesn't care about formatting, humans do.
  - Make statement easy to read
  - Embed analysis information
  - Easier to spot transformations which will occur
  - Spot human errors
    - Ever forget a join predicate?
    - Forget parenthesis?
  - Easy to comment out portions of SQL to facilitate analysis

- How confident will you be in your analysis if
  - Referenced tables aren't easily identified
  - Predicates aren't easily identified

Format the statement

- Major clauses should start a new line
  - SELECT, FROM, ON, WHERE, GROUP BY, HAVING, ORDER BY
  - Each table on own line
    - → Lead with comma
    - → Use correlation names
  - Each on-clause predicate on it's own line
    - → Lead with AND/OR
  - Each WHERE clause predicate on it's own line
    - → Lead with AND/OR
  - Use parenthesis for OR to make meaning clear
  - Use correlation names on tables and predicates to make associations obvious

– Nested table expressions / subqueries
  - Indent, follow same guidelines as above

Sample unformatted query

```
EXPLAIN PLAN SET QUERYNO = 1  FOR
    SELECT   DISTINCT ITEM.ITEM_NBR AS ITEM_NBR, ITEM.PRDT_ID,
    STOREITEM.WK_STRT_DT AS WK_STRT_DT ,STOREITEM.DC_ID AS
    DC_ID FROM PROD.TIPA004_STITM_PROJ AS STOREITEM ,
    PROD.TITM001_ITEM AS ITEM WHERE ITEM.BUS_UNIT_ID = ‘GS‘ AND
    ITEM.BUS_UNIT_ID = STOREITEM.BUS_UNIT_ID AND
    ITEM.MJR_CATG_ID = '00754‘ AND  ITEM.INTMD_CATG_ID = '00043‘
    AND ITEM.ITEM_NBR = STOREITEM.ITEM_NBR AND
    ITEM.MJR_CATG_ID = STOREITEM.MJR_CATG_ID AND
    ITEM.INTMD_CATG_ID = STOREITEM.INTMD_CATG_ID AND
    STOREITEM.RTL_DEPT_NBR = 1 AND   AD_ITEM_FLG = 'Y‘ AND
    WK_STRT_DT = '2011-02-08';
```

Unformatted SQL, where to start?

Separate sections

```
EXPLAIN PLAN SET QUERYNO = 1  FOR
SELECTDISTINCT ITEM.ITEM_NBR AS ITEM_NBR, ITEM.PRDT_ID,
   STOREITEM.WK_STRT_DT AS WK_STRT_DT ,STOREITEM.DC_ID AS
   DC_ID
FROM PROD.TIPA004_STITM_PROJ AS STOREITEM ,
   PROD.TITM001_ITEM AS ITEM
WHERE ITEM.BUS_UNIT_ID = ‘GS‘ AND ITEM.BUS_UNIT_ID =
   STOREITEM.BUS_UNIT_ID AND      ITEM.MJR_CATG_ID = '00754‘ AND
     ITEM.INTMD_CATG_ID = '00043‘ AND ITEM.ITEM_NBR =
   STOREITEM.ITEM_NBR AND    ITEM.MJR_CATG_ID =
   STOREITEM.MJR_CATG_ID AND     ITEM.INTMD_CATG_ID =
   STOREITEM.INTMD_CATG_ID AND  STOREITEM.RTL_DEPT_NBR = 1
   AND  AD_ITEM_FLG = 'Y‘ AND WK_STRT_DT = '2011-02-08';
```

Separate sections

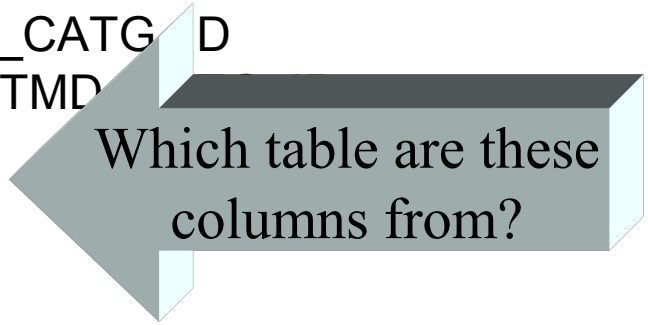
```
EXPLAIN PLAN SET QUERYNO = 1  FOR
SELECT DISTINCT ITEM.ITEM_NBR AS ITEM_NBR, ITEM.PRDT_ID,
    STOREITEM.WK_STRT_DT AS WK_STRT_DT ,STOREITEM.DC_ID AS
    DC_ID
FROM   PROD.TIPA004_STITM_PROJ AS STOREITEM
       ,PROD.TITM001_ITEM AS ITEM
WHERE ITEM.BUS_UNIT_ID = 'GS'
AND    ITEM.BUS_UNIT_ID = STOREITEM.BUS_UNIT_ID
AND    ITEM.MJR_CATG_ID = '00754'
AND    ITEM.INTMD_CATG_ID = '00043'
AND    ITEM.ITEM_NBR = STOREITEM.ITEM_NBR
AND    STOREITEM.AD_ITEM_FLG = 'Y'
AND    ITEM.MJR_CATG_ID = STOREITEM.MJR_CAT
AND    ITEM.INTMD_CATG_ID = STOREITEM.INTMD_CA    _ID
AND    RTL_DEPT_NBR = 1
AND    WK_STRT_DT = '2011-02-08';
```

1 line per table,
Lead with comma

1 line per predicate,
Lead with AND

Use correlation names

```
EXPLAIN PLAN SET QUERYNO = 1  FOR
SELECT DISTINCT ITEM.ITEM_NBR AS ITEM_NBR, ITEM.PRDT_ID,
    STOREITEM.WK_STRT_DT AS WK_STRT_DT ,STOREITEM.DC_ID AS
    DC_ID
FROM   PROD.TIPA004_STITM_PROJ AS STOREITEM
       ,PROD.TITM001_ITEM AS ITEM
WHERE ITEM.BUS_UNIT_ID = 'GS'
AND    ITEM.BUS_UNIT_ID = STOREITEM.BUS_UNIT_ID
AND    ITEM.MJR_CATG_ID = '00754'
AND    ITEM.INTMD_CATG_ID = '00043'
AND    ITEM.ITEM_NBR = STOREITEM.ITEM_NBR
AND    STOREITEM.AD_ITEM_FLG = 'Y'
AND    ITEM.MJR_CATG_ID = STOREITEM.MJR_CATG_ID
AND    ITEM.INTMD_CATG_ID = STOREITEM.INTMD
AND    RTL_DEPT_NBR = 1
AND    WK_STRT_DT = '2011-02-08';
```



Organize WHERE clause

```
EXPLAIN PLAN SET QUERYNO = 1  FOR
SELECT DISTINCT ITEM.ITEM_NBR AS ITEM_NBR, ITEM.PRDT_ID,
    STOREITEM.WK_STRT_DT AS WK_STRT_DT ,STOREITEM.DC_ID AS
    DC_ID
FROM   PROD.TIPA004_STITM_PROJ AS STOREITEM
          ,PROD.TITM001_ITEM AS ITEM
WHERE ITEM.BUS_UNIT_ID = STOREITEM.BUS_UNIT_ID
AND     ITEM.MJR_CATG_ID = STOREITEM.MJR_CAT
AND     ITEM.INTMD_CATG_ID = STOREITEM.INTMD_
AND     ITEM.ITEM_NBR = STOREITEM.ITEM_NBR
AND     ITEM.BUS_UNIT_ID = 'GS'
AND     ITEM.MJR_CATG_ID = '00754'
AND     ITEM.INTMD_CATG_ID = '00043'
AND     STOREITEM.AD_ITEM_FLG = 'Y'
AND     STOREITEM.RTL_DEPT_NBR = 1
AND     STOREITEM.WK_STRT_DT = '2011-02-08';
```

Formatted SQL ready for analysis

- Easily see the different components of the SQL
  - Can see each major section
  - Quickly see number of tables
  - Observe join predicates
  - Which tables have local predicates
  - Easy to see transformations
- Does this seem unimportant?
  - Speed of analysis reduces down time in a crisis
  - Well formatted SQL speeds analysis

Annotate the Query

- Embed information within the statement
  - Table information
    - NPAGES
    - CARDF
  - Column information for predicates (local and join)
    - COLCARDF
    - LOW2KEY and HIGH2KEY
    - Frequency/histogram statistics
  - Observe where the filtering is
    - Selectivity of a predicate is relative to table cardinality

Annotate the Query

```
EXPLAIN PLAN SET QUERYNO = 1  FOR
SELECT DISTINCT ITEM.ITEM_NBR AS ITEM_NBR, ITEM.PRDT_ID, STOREITEM.WK_STRT_DT AS
     WK_STRT_DT ,STOREITEM.DC_ID AS DC_ID
FROM         PROD.TIPA004_STITM_PROJ AS STOREITEM
                                                          CARDF = 4,142,397
                                                          NPAGES = 287,080
             ,PROD.TITM001_ITEM AS ITEM                   CARDF = 1,258,022
                                                          NPAGES = 212,121
WHERE ITEM.BUS_UNIT_ID = STOREITEM.BUS_UNIT_ID            COLCARDF = 12 / 12
AND   ITEM.MJR_CATG_ID = STOREITEM.MJR_CATG_ID            COLCARDF = 1024 / 998
AND   ITEM.INTMD_CATG_ID = STOREITEM.INTMD_CATG_ID        COLCARDF = 962 / 962
AND   ITEM.ITEM_NBR = STOREITEM.ITEM_NBR                  COLCARDF = 858,022 / 758,726
AND   ITEM.BUS_UNIT_ID = ‘GS‘                             COLCARDF = 12
AND   ITEM.MJR_CATG_ID = '00754‘                          COLCARDF = 1024
AND   ITEM.INTMD_CATG_ID = '00043‘                        COLCARDF = 962
AND   STOREITEM.AD_ITEM_FLG = 'Y‘                         COLCARDF = 2
AND  STOREITEM.RTL_DEPT_NBR = 1                           COLCARDF = 1057
AND   STOREITEM.WK_STRT_DT = '2011-02-08';                COLCARDF = 104
```

Identify Suspicious Predicates

- Which predicates might DB2 incorrectly estimate?
  - Range predicates
    - Especially with host variables / parameter markers
  - Predicate on low cardinality columns
    - Data skew is likely
  - Predicates with expressions
    - SUBSTR(C1,1,1)='A' ← Default FF of 1/25
  - Predicates with default values
    - EFF_DT < '9999-12-31'
    - STATUS_CD = ' '

Identify Suspicious Predicates

```
EXPLAIN PLAN SET QUERYNO = 1  FOR
SELECT DISTINCT ITEM.ITEM_NBR AS ITEM_NBR, ITEM.PRDT_ID, STOREITEM.WK_STRT_DT AS
     WK_STRT_DT ,STOREITEM.DC_ID AS DC_ID
FROM         PROD.TIPA004_STITM_PROJ AS STOREITEM
                                                                    CARDF = 4,142,397
                                                                    NPAGES = 287,080
             ,PROD.TITM001_ITEM AS ITEM                             CARDF = 1,258,022
                                                                    NPAGES = 212,121
WHERE ITEM.BUS_UNIT_ID = STOREITEM.BUS_UNIT_ID                      COLCARDF = 12 / 12
AND   ITEM.MJR_CATG_ID = STOREITEM.MJR_CATG_ID                      COLCARDF = 1024 / 998
AND   ITEM.INTMD_CATG_ID = STOREITEM.INTMD_CATG_ID                  COLCARDF = 962 / 962
AND   ITEM.ITEM_NBR = STOREITEM.ITEM_NBR                            COLCARDF = 858,022 / 758,726
AND   ITEM.BUS_UNIT_ID = ‘GS‘                                       COLCARDF = 12
AND   ITEM.MJR_CATG_ID = '00754‘                                    COLCARDF = 1024
AND   ITEM.INTMD_CATG_ID = '00043‘                                  COLCARDF = 962
AND   STOREITEM.AD_ITEM_FLG = 'Y‘                                   COLCARDF = 2
AND  STOREITEM.RTL_DEPT_NBR = 1                                     COLCARDF = 1057
AND   STOREITEM.WK_STRT_DT = '2011-02-08';                          COLCARDF = 104
```

See anything suspicious?

Identify Suspicious Predicates

```
EXPLAIN PLAN SET QUERYNO = 1  FOR
SELECT DISTINCT ITEM.ITEM_NBR AS ITEM_NBR, ITEM.PRDT_ID, STOREITEM.WK_STRT_DT AS
     WK_STRT_DT ,STOREITEM.DC_ID AS DC_ID
FROM         PROD.TIPA004_STITM_PROJ AS STOREITEM
                                                          CARDF = 4,142,397
                                                          NPAGES = 287,080
             ,PROD.TITM001_ITEM AS ITEM                   CARDF = 1,258,022
                                                          NPAGES = 212,121
WHERE ITEM.BUS_UNIT_ID = STOREITEM.BUS_UNIT_ID            COLCARDF = 12 / 12
AND   ITEM.MJR_CATG_ID = STOREITEM.MJR_CATG_ID            COLCARDF = 1024 / 998
AND   ITEM.INTMD_CATG_ID = STOREITEM.INTMD_CATG_ID        COLCARDF = 962 / 962
AND   ITEM.ITEM_NBR = STOREITEM.ITEM_NBR                  COLCARDF = 858,022 / 758,726
AND   ITEM.BUS_UNIT_ID = 'GS'                             COLCARDF = 12
AND   ITEM.MJR_CATG_ID = '00754'                          COLCARDF = 1024
AND   ITEM.INTMD_CATG_ID = '00043'                        COLCARDF = 962
AND   STOREITEM.AD_ITEM_FLG = 'Y'                         COLCARDF = 2
AND  STOREITEM.RTL_DEPT_NBR = 1                           COLCARDF = 1057
AND   STOREITEM.WK_STRT_DT = '2011-02-08';                COLCARDF = 104
```

Low column cardinality – likely to be skewed

Is this a default value?

Automation

- Data Studio
  - Query formatting
- Optim Query Tuner
  - Query formatting
  - Query annotation

Agenda

- Preparing a query for analysis
- Primary cause of access path problems
- Query walkthrough
- Verify selectivity estimates

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???

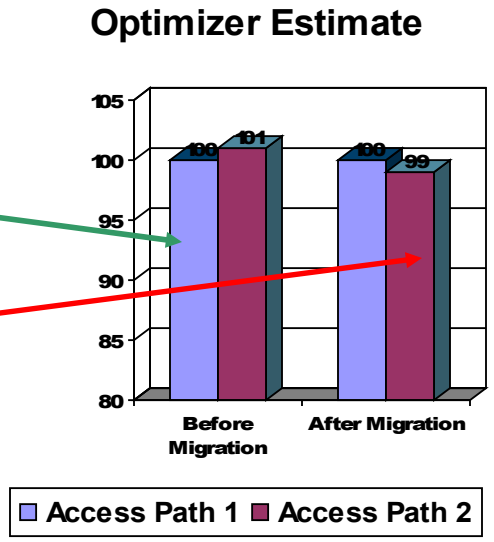
Why can performance fluctuate?

- When access path costs do not reflect reality
  - Eg. Actual performance of 2 access paths
    - Access path 1 = 100 msec
    - Access path 2 = 1 min

Best performing

600 X slower

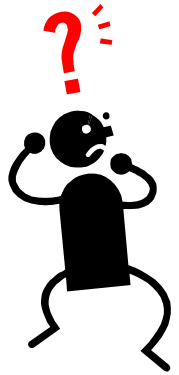
  - Before
    - Access path 1 wins
      - → Estimate 100msec vs 101msec
  - After migration/maint etc.
    - Access path 2 wins
      - → Estimate 100msec vs 99msec



The problem?

- After Rebind
  - Access path 2 cost improved from 101 to 99 msec
    - But reality is 1 min (approx 600X greater)

- Should the focus be on:
  - What caused the cost to change from 101 to 99?
  - OR, why the cost isn't closer to 1 min?



Why can optimizer estimate incorrectly?

- Incorrect selectivity estimates
  - Lack of statistics
    - Frequency statistics
    - Correlation statistics
    - Histogram statistics
  - Inability to use important statistics
    - Host variables or parameter markers preclude use of frequency of histogram statistics without REOPT (with the exception of the null value)
      - → Especially problematic coupled with range predicate (where actual FF could vary from 0% to 100%)
  - Difficult to estimate predicates
    - DATE(TIMESTAMP_COL) = '2011-02-01'
    - C1+C2 < 5

Agenda

- Preparing a query for analysis
- Primary cause of access path problems
- Query walkthrough
- Verify selectivity estimates

Query breakdown

```
SELECT  …
FROM      SETL_TRANS S
          ,BRANCH CUST
          ,BRANCH_ADDR A
WHERE S.ADV_ABA_R = ?
AND       S.PROCESS_DT < '9999-12-31‘
AND       S.TYPE_CD  IN ('A', ‘C’, ‘X’)
AND       S.CLR_CYCLE_CD IN ('EOD', 'IMD‘, ‘OPN’)
AND       S.STLMT_DT = ?
AND       S.ACCT_NUM   = CUST.ACCT_NUM
AND       CUST.CUST_EFCT_DT <= ?
AND       CUST.CUST_INACTV_DT >  ?
AND       A.ACCT_NUM = CUST.ACCT_NUM
AND       A.CUST_EFCT_DT    <= ?
AND       A.CUST_INACTV_DT    >  ?
AND       A.ADDR_TYP_CD   = ' '
```

See any suspicious predicates?

Identify suspicious predicates

```
SELECT  …
FROM      SETL_TRANS S
          ,BRANCH CUST
          ,BRANCH_ADDR A
WHERE S.ADV_ABA_R = ?
AND       S.PROCESS_DT < ‘9999-12-31’                    ← MAX DATE
AND       S.TYPE_CD  IN ('A', 'C', ‘X‘, ‘Z’)
AND       S.CLR_CYCLE_CD IN ('EOD', 'IMD‘, ‘OPN’)
AND       S.STLMT_DT = ?
AND       S.ACCT_NUM   = CUST.ACCT_NUM
AND       CUST.CUST_EFCT_DT <= ?                         ← Range with marker
AND       CUST.CUST_INACTV_DT >  ?                       ← Range with marker
AND       A.ACCT_NUM = CUST.ACCT_NUM
AND       A.CUST_EFCT_DT    <= ?                         ← Range with marker
AND       A.CUST_INACTV_DT    >  ?                       ← Range with marker
AND       A.ADDR_TYP_CD   = ' ‘                          ← COL = blank
```

Why are they suspicious?

Predicates with typical default often skewed.

```
AND     S.PROCESS_DT < '9999-12-31'      ← MAX DATE
AND     A.ADDR_TYP_CD   = ' '            ← COL = blank
```

Range predicates with parameter markers

- Impossible to estimate without literal

```
AND     CUST.CUST_EFCT_DT <= ?           ← Range with marker
AND     CUST.CUST_INACTV_DT >  ?         ← Range with marker

AND     A.CUST_EFCT_DT    <= ?           ← Range with marker
AND     A.CUST_INACTV_DT    >  ?         ← Range with marker
```

Embed statistics

```
SELECT  …
FROM          SETL_TRANS S                          CARDF 1,600,254          NPAGES 21,627
              ,BRANCH CUST                          CARDF 31,696             NPAGES 1132
              ,BRANCH_ADDR A                        CARDF 58,627             NPAGES 2791
WHERE         S.ADV_ABA_R = ?                                                COLCARDF 19,712
AND S.PROCESS_DT < ‘9999-12-31’                     COLCARDF 11
              LOW2KEY 2004-03-24      HIGH2KEY 2004-04-05
AND S.TYPE_CD  IN ('A', 'C', ‘X‘, ‘Z’)              COLCARDF 4
AND S.CLR_CYCLE_CD IN ('EOD', 'IMD', ‘OPN')         COLCARDF 3
AND S.STLMT_DT = ?                                  COLCARDF 13
AND S.ACCT_NUM   = CUST.ACCT_NUM                    COLCARDF 15360 / 26,527
AND CUST.CUST_EFCT_DT <= ?                          COLCARDF 2,496
              LOW2KEY 1994-09-02      HIGH2KEY 2004-04-06
AND CUST.CUST_INACTV_DT >  ?                        COLCARDF 279
              LOW2KEY 2004-03-04      HIGH2KEY 2004-04-07
AND A.ACCT_NUM = CUST.ACCT_NUM                      COLCARDF 26,527 / 26,527
AND A.CUST_EFCT_DT    <= ?                          COLCARDF 2,496
              LOW2KEY 1994-09-02      HIGH2KEY 2004-04-06
AND A.CUST_INACTV_DT    >  ?                        COLCARDF 274
              LOW2KEY ‘2004-03-04’    HIGH2KEY ‘2004-04-07’
AND A.ADDR_TYP_CD   = ‘ ‘                           COLCARDF 5
```

Literal predicate analysis (1)

```
SELECT  …
FROM          SETL_TRANS S                          CARDF 1,600,254          NPAGES 21,627
              ,BRANCH CUST                          CARDF 31,696             NPAGES 1132
              ,BRANCH_ADDR A                        CARDF 58,627             NPAGES 2791
WHERE         S.ADV_ABA_R = ?                       COLCARDF 19,712
AND S.PROCESS_DT < '9999-12-31'                     COLCARDF 11
              LOW2KEY 2004-03-24      HIGH2KEY 2004-04-05
AND S.TYPE_CD  IN ('A', 'C', 'X', 'Z')              COLCARDF 4
AND S.CLR_CYCLE_CD IN ('EOD', 'IMD', 'OPN')         COLCARDF 3
AND S.STLMT_DT = ?                                  COLCARDF 13
AND S.ACCT_NUM   = CUST.ACCT_NUM                    COLCARDF 15360 / 26,527
AND CUST.CUST_EFCT_DT <= ?                          COLCARDF 2,496
              LOW2KEY 1994-09-02      HIGH2KEY 2004-04-06
AND CUST.CUST_INACTV_DT >  ?                        COLCARDF 279
              LOW2KEY 2004-03-04      HIGH2KEY 2004-04-07
AND A.ACCT_NUM = CUST.ACCT_NUM                      COLCARDF 26,527 / 26,527
AND A.CUST_EFCT_DT    <= ?                          COLCARDF 2,496
              LOW2KEY 1994-09-02      HIGH2KEY 2004-04-06
AND A.CUST_INACTV_DT    >  ?                        COLCARDF 274
              LOW2KEY '2004-03-04'    HIGH2KEY '2004-04-07'
AND A.ADDR_TYP_CD   = ' '                           COLCARDF 5
```

Literal predicate analysis (2)

```
SELECT  …
FROM          SETL_TRANS S                          CARDF 1,600,254        NPAGES 21,627
              ,BRANCH CUST                          CARDF 31,696           NPAGES 1132
              ,BRANCH_ADDR A                        CARDF 58,627           NPAGES 2791
WHERE         S.ADV_ABA_R = ?                       COLCARDF 19,712
AND S.PROCESS_DT < ‘9999-12-31’                     COLCARDF 11
              LOW2KEY 2004-03-24        HIGH2KEY 2004-04-05

AND S.TYPE_CD  IN ('A', 'C', ‘X‘, ‘Z’)              COLCARDF 4
AND S.CLR_CYCLE_CD IN ('EOD', 'IMD', ‘OPN')         COLCARDF 3

AND S.STLMT_DT = ?                                  COLCARDF 13
AND S.ACCT_NUM   = CUST.ACCT_NUM                    COLCARDF 15360 / 26,527
AND CUST.CUST_EFCT_DT <= ?                          COLCARDF 2,496
              LOW2KEY 1994-09-02        HIGH2KEY 2004-04-06
AND CUST.CUST_INACTV_DT >  ?                        COLCARDF 279
              LOW2KEY 2004-03-04        HIGH2KEY 2004-04-07
AND A.ACCT_NUM = CUST.ACCT_NUM                      COLCARDF 26,527 / 26,527
AND A.CUST_EFCT_DT    <= ?                          COLCARDF 2,496
              LOW2KEY 1994-09-02        HIGH2KEY 2004-04-06
AND A.CUST_INACTV_DT    >  ?                        COLCARDF 274
              LOW2KEY ‘2004-03-04’      HIGH2KEY ‘2004-04-07’
AND A.ADDR_TYP_CD   = ‘ ‘                           COLCARDF 5
```

Literal predicate analysis (3)

```
SELECT  …
FROM          SETL_TRANS S                          CARDF 1,600,254          NPAGES 21,627
              ,BRANCH CUST                          CARDF 31,696             NPAGES 1132
              ,BRANCH_ADDR A                        CARDF 58,627             NPAGES 2791
WHERE         S.ADV_ABA_R = ?                                                COLCARDF 19,712
AND S.PROCESS_DT < '9999-12-31'                     COLCARDF 11
              LOW2KEY 2004-03-24      HIGH2KEY 2004-04-05
AND S.TYPE_CD  IN ('A', 'C', 'X', 'Z')              COLCARDF 4
AND S.CLR_CYCLE_CD IN ('EOD', 'IMD', 'OPN')         COLCARDF 3
AND S.STLMT_DT = ?                                  COLCARDF 13
AND S.ACCT_NUM   = CUST.ACCT_NUM                    COLCARDF 15360 / 26,527
AND CUST.CUST_EFCT_DT <= ?                          COLCARDF 2,496
              LOW2KEY 1994-09-02      HIGH2KEY 2004-04-06
AND CUST.CUST_INACTV_DT >  ?                        COLCARDF 279
              LOW2KEY 2004-03-04      HIGH2KEY 2004-04-07
AND A.ACCT_NUM = CUST.ACCT_NUM                      COLCARDF 26,527 / 26,527
AND A.CUST_EFCT_DT    <= ?                          COLCARDF 2,496
              LOW2KEY 1994-09-02      HIGH2KEY 2004-04-06
AND A.CUST_INACTV_DT    >  ?                        COLCARDF 274
              LOW2KEY '2004-03-04'    HIGH2KEY '2004-04-07'
AND A.ADDR_TYP_CD   = ' '                           COLCARDF 5
```

Range with marker predicate analysis (1)

```
SELECT  …
FROM          SETL_TRANS S                          CARDF 1,600,254          NPAGES 21,627
              ,BRANCH CUST                          CARDF 31,696             NPAGES 1132
              ,BRANCH_ADDR A                        CARDF 58,627             NPAGES 2791
WHERE         S.ADV_ABA_R = ?                       COLCARDF 19,712
AND S.PROCESS_DT < '9999-12-31'                     COLCARDF 11
              LOW2KEY 2004-03-24        HIGH2KEY 2004-04-05
AND S.TYPE_CD  IN ('A', 'C', 'X', 'Z')              COLCARDF 4
AND S.CLR_CYCLE_CD IN ('EOD', 'IMD', 'OPN')         COLCARDF 3
AND S.STLMT_DT = ?                                  COLCARDF 13
AND S.ACCT_NUM   = CUST.ACCT_NUM                    COLCARDF 15360 / 26,527

AND CUST.CUST_EFCT_DT <= ?                          COLCARDF 2,496
              LOW2KEY 1994-09-02        HIGH2KEY 2004-04-06
AND CUST.CUST_INACTV_DT >  ?                        COLCARDF 279
              LOW2KEY 2004-03-04        HIGH2KEY 2004-04-07

AND A.ACCT_NUM = CUST.ACCT_NUM                      COLCARDF 26,527 / 26,527

AND A.CUST_EFCT_DT    <= ?                          COLCARDF 2,496
              LOW2KEY 1994-09-02        HIGH2KEY 2004-04-06
AND A.CUST_INACTV_DT    >  ?                        COLCARDF 274
              LOW2KEY '2004-03-04'      HIGH2KEY '2004-04-07'

AND A.ADDR_TYP_CD   = ' '                           COLCARDF 5
```

Range predicate interpolation

Table 104. Default filter factors for interpolation

COLCARDF	Filter factor for OP	Filter Factor for LIKE / BETWEEN
>= 100,000,000	1 / 10,000	3 / 100,000
>= 10,000,000	1 / 3,000	1 / 10,000
>= 1,000,000	1 / 1,000	3 / 10,000
>= 100,000	1 / 300	1 / 1,000
>= 10,000	1 / 100	3 / 1,000
>= 1,000	1 / 30	1 / 100
>= 100	1 / 10	3 / 100
>= 2	1 / 3	1 / 10
= 1	1	1
>= 0	1 / 3	1 / 10

Note: Op is one of these operators: <, <=, >, >=.

COMMENT: This is DB2's documented guess for an impossible to estimate Filter factor.

Range with marker predicate analysis

Error is how different the optimizer's DEFAULT estimate is from ACTUAL filtering.

3) AND CUST.CUST_EFCT_DT <= ? COLCARDF 2,496
LOW2KEY 1994-09-02 HIGH2KEY 2004-04-06
ESTIMATED FF WITH LITERAL: = 100%
ESTIMATE WITH MARKER: 1/30 = 3% (97% error)

4) AND CUST.CUST_INACTV_DT > ? COLCARDF 279
LOW2KEY 2004-03-04 HIGH2KEY 2004-04-07
ESTIMATED FF WITH LITERAL: = 99%
ESTIMATE WITH MARKER: 1/10 = 10% (89% error)

5) AND A.CUST_EFCT_DT <= ? COLCARDF 2,496
LOW2KEY 1994-09-02 HIGH2KEY 2004-04-06
ESTIMATED FF WITH LITERAL: = 100%
ESTIMATE WITH MARKER: 1/30 = 3% (97% error)

6) AND A.CUST_INACTV_DT > ? COLCARDF 274
LOW2KEY '2004-03-04' HIGH2KEY '2004-04-07'
ESTIMATED FF WITH LITERAL: = 99%
ESTIMATE WITH MARKER: 1/10 = 10% (89% error)

Suspicious predicate analysis

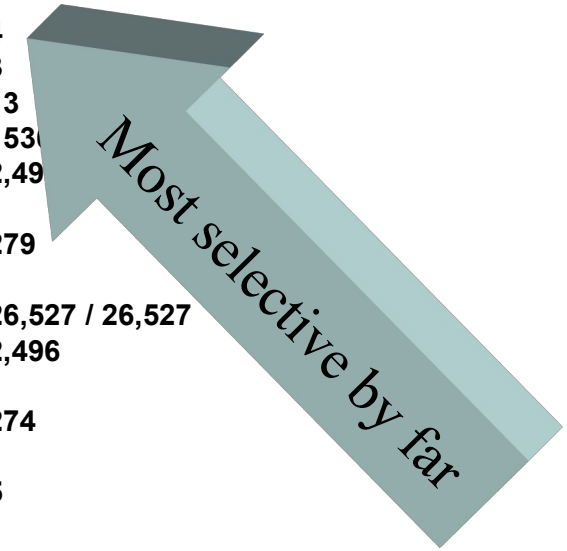
- Conclusion
  - The range predicates with parameter markers introduce significant filter factor error.
  - We should recognize that this filter factor error can cause significant cost estimation problems for the optimizer – possibly resulting in poor access path choice.

Where's the filtering?

WHERE S.ADV_ABA_R = ? COLCARDF 19,712
(Very selective predicate)
AND S.PROCESS_DT < '9999-12-31' COLCARDF 11
(This predicate doesn't filter anything, known from suspicious predicate analysis)
AND S.TYPE_CD IN ('A', 'C', 'X', 'Z') COLCARDF 4
(In-list looking for 4 values, COLCARDF 4 – not filtering)
AND S.CLR_CYCLE_CD IN ('EOD', 'IMD', 'OPN') COLCARDF 3
(In-list looking for 3 values, COLCARDF 3 – not filtering)
AND S.STLMT_DT = ? COLCARDF 13
(COL = LIT, COLCARDF 13 – somewhat filtering, but not great selectivity)
AND S.ACCT_NUM = CUST.ACCT_NUM COLCARDF 15360 / 26,527
(For the range predicates, we know that optimizer PERCIEVES them to be selective but In reality, they are not. This was determined during suspicious predicate analysis)
AND CUST.CUST_EFCT_DT <= ? COLCARDF 2,496
AND CUST.CUST_INACTV_DT > ? COLCARDF 279
AND A.ACCT_NUM = CUST.ACCT_NUM COLCARDF 26,527 / 26,527
AND A.CUST_EFCT_DT <= ? COLCARDF 2,496
AND A.CUST_INACTV_DT > ? COLCARDF 274
AND A.ADDR_TYP_CD = ' ' COLCARDF 5
(COL = blank. Probably this column is skewed on blank. COLCARDF 5, not typically Very filtering)

Where's the filtering?

```
SELECT  …
FROM        SETL_TRANS S                                  CARDF 1,600,254        NPAGES 21,627
           ,BRANCH CUST                                   CARDF 31,696           NPAGES 1132
           ,BRANCH_ADDR A                    CARDF 58,627                NPAGES 2791
WHERE       S.ADV_ABA_R = ?                                                COLCARDF 19,712
AND S.PROCESS_DT < '9999-12-31'               COLCARDF 11
           LOW2KEY 2004-03-24      HIGH2KEY 2004-04-05
AND S.TYPE_CD  IN ('A', 'C', 'X', 'Z')        COLCARDF 4
AND S.CLR_CYCLE_CD IN ('EOD', 'IMD', 'OPN')   COLCARDF 3
AND S.STLMT_DT = ?                            COLCARDF 13
AND S.ACCT_NUM   = CUST.ACCT_NUM              COLCARDF 153
AND CUST.CUST_EFCT_DT <= ?                    COLCARDF 2,49
      LOW2KEY 1994-09-02      HIGH2KEY 2004-04-06
AND CUST.CUST_INACTV_DT >  ?                  COLCARDF 279
      LOW2KEY 2004-03-04 HIGH2KEY 2004-04-07
AND A.ACCT_NUM = CUST.ACCT_NUM                COLCARDF 26,527 / 26,527
AND A.CUST_EFCT_DT    <= ?                    COLCARDF 2,496
      LOW2KEY 1994-09-02      HIGH2KEY 2004-04-06
AND A.CUST_INACTV_DT    >  ?                  COLCARDF 274
      LOW2KEY '2004-03-04'    HIGH2KEY '2004-04-07'
AND A.ADDR_TYP_CD   = ' '                     COLCARDF 5
```



Index analysis

- One significant input to the optimizer is…
  - Available indexes
  - What join sequence they encourage
- Some index performance considerations
  - Provide efficient access for local predicates
    - Encourages table to be outer table
  - Provide efficient access for join predicates
    - Encourage access to table as INNER table of join
  - Provide ordering to avoid sort
- Analysis:
  - Are there appropriate indexes to support this query?

Identify indexes

Table: SETL_TRANS
INDEX IXSTRN01
(PROCESS_DT, CLR_CYCLE_CD, ADV_ABA_R, TYPE_CD, ACCT_NUM, STLMT_DT)

TABLE: BRANCH

INDEX: IXBRNC01
(CUST_INACTV_DT, CUST_EFCT_DT)
INDEX: IXBRNC02
(ACCT_NUM, CUST_EFCT_DT)

TABLE: BRANCH_ADDR

INDEX: IXBRAD01
(CUST_INACTV_DT, CUST_EFCT_DT)
INDEX: IXBRAD02
(ACCT_NUM, ADDR_TYP_CD, CUST_EFCT_DT)

SETL_TRANS
local predicate index analysis

```
SELECT  …
FROM            SETL_TRANS S
WHERE           S.ADV_ABA_R = ?
AND  S.PROCESS_DT < ‘9999-12-31’
AND  S.TYPE_CD  IN ('A', 'C', ‘X‘, ‘Z’)
AND  S.CLR_CYCLE_CD IN ('EOD', 'IMD', ‘OPN')
AND  S.STLMT_DT = ?
```

CARDF 1,600,254 NPAGES 21,627
COLCARDF 19,712
COLCARDF 11
COLCARDF 4
COLCARDF 3
COLCARDF 13

Indexes
Table: AJT_SETL_TRANS
INDEX IXSTRN01

```
        (PROCESS_DT,
        ,CLR_CYCLE_CD
        ,ADV_ABA_R
        ,TYPE_CD
        , ACCT_NUM
        , STLMT_DT
        )
```

Not a good local filtering index.

← non-filtering range predicate (matching stopper)
← non-filtering in-list
← Selective COL = LIT predicate
← non-filtering in-list

← moderately filtering COL = LIT.

SETL_TRANS
join index analysis

```
SELECT  …
FROM            SETL_TRANS S
WHERE           S.ADV_ABA_R = ?
AND  S.PROCESS_DT < ‘9999-12-31’
AND  S.TYPE_CD  IN ('A', 'C', ‘X‘, ‘Z’)
AND  S.CLR_CYCLE_CD IN ('EOD', 'IMD', ‘OPN')
AND  S.STLMT_DT = ?
AND  S.ACCT_NUM   = CUST.ACCT_NUM
AND  S.ACCT_NUM = A.ACCT_NUM
```

CARDF 1,600,254 NPAGES 21,627
COLCARDF 19,712
COLCARDF 11
COLCARDF 4
COLCARDF 3
COLCARDF 13
COLCARDF 15,360 / 26,527
COLCARDF 15,360 / 26,527

Indexes
Table: AJT_SETL_TRANS
INDEX IXSTRN01

```
        (PROCESS_DT,
        ,CLR_CYCLE_CD
        ,ADV_ABA_R
        ,TYPE_CD
        , ACCT_NUM
        , STLMT_DT
        )
```

Does not support the join.

← non-filtering range predicate (matching stopper)
← non-filtering in-list
← Selective COL = LIT predicate
← non-filtering in-list
← Join predicate
← moderately filtering COL = LIT.

BRANCH
local predicate analysis

```
SELECT  …
FROM
                ,BRANCH CUST
WHERE           …
AND  CUST.CUST_EFCT_DT <= ?
AND  CUST.CUST_INACTV_DT >  ?
AND  S.ACCT_NUM   = CUST.ACCT_NUM
AND  A.ACCT_NUM = CUST.ACCT_NUM
```

CARDF 31,696 NPAGES 1132

COLCARDF 2,496
COLCARDF 279
COLCARDF 15360 / 26,527
COLCARDF 26,527 / 26,527

Indexes:

```
INDEX: IXBRNC01
        (CUST_INACTV_DT
        , CUST_EFCT_DT
        )
```

Looks more attractive than it is.
← High estimated filtering, poor actual filtering range predicate
← High estimated filtering, poor actual filtering range predicate

```
INDEX IXBRNC02
        (ACCT_NUM
        ,CUST_EFCT_DT
        )
```

No local matching indexable, not a good local filtering candidate.

← High estimated filtering, poor actual filtering range predicate

BRANCH
join predicate analysis

```
SELECT  …
FROM
                ,BRANCH CUST
WHERE           …
AND  CUST.CUST_EFCT_DT <= ?
AND  CUST.CUST_INACTV_DT >  ?
AND  S.ACCT_NUM   = CUST.ACCT_NUM
AND  A.ACCT_NUM = CUST.ACCT_NUM
```

CARDF 31,696 NPAGES 1132

COLCARDF 2,496
COLCARDF 279
COLCARDF 15360 / 26,527
COLCARDF 26,527 / 26,527

Indexes:

INDEX: IXBRNC01
(CUST_INACTV_DT
, CUST_EFCT_DT
)

No support for join.
← High estimated filtering, poor actual filtering range predicate
← High estimated filtering, poor actual filtering range predicate

INDEX IXBRNC02
(ACCT_NUM
,CUST_EFCT_DT
)

Supports the join!
← Matching indexable from join perspective
← High estimated filtering, poor actual filtering range predicate

BRANCH_ADDR
local predicate analysis

```
SELECT  …
FROM            …
                ,BRANCH_ADDR A
WHERE           …
AND  A.CUST_EFCT_DT   <= ?
AND  A.CUST_INACTV_DT   > ?
AND  A.ADDR_TYP_CD   = ' '
```

CARDF 58,627 NPAGES 2791

COLCARDF 2,496
COLCARDF 274
COLCARDF 5

Indexes:

INDEX: IXBRAD01
(CUST_INACTV_DT
, CUST_EFCT_DT
)

Looks more attractive than it really is.
← High estimated filtering, poor actual filtering range predicate
← High estimated filtering, poor actual filtering range predicate

INDEX: IXBRAD02
(ACCT_NUM
, ADDR_TYP_CD
, CUST_EFCT_DT
)

Cannot match on local filtering, not a good local candidate.

← COL=LIT, potentially skewed, not that selective. Can't be matching.
← High estimated filtering, poor actual filtering range predicate

BRANCH_ADDR
join predicate analysis

```
SELECT  …
FROM …
                ,BRANCH_ADDR A
WHERE           …
AND   A.CUST_EFCT_DT   <= ?
AND   A.CUST_INACTV_DT   > ?
AND   A.ADDR_TYP_CD   = ‘ ‘
AND   S.ACCT_NUM   = CUST.ACCT_NUM
AND   S.ACCT_NUM = A.ACCT_NUM
```

CARDF 58,627 NPAGES 2791

COLCARDF 2,496
COLCARDF 274
COLCARDF 5
COLCARDF 15,360 / 26,527
COLCARDF 15,360 / 26,527

Indexes:

INDEX: IXBRAD01
(CUST_INACTV_DT
, CUST_EFCT_DT
)

No join support
← High estimated filtering, poor actual filtering range predicate
← High estimated filtering, poor actual filtering range predicate

INDEX: IXBRAD02
(ACCT_NUM
, ADDR_TYP_CD
, CUST_EFCT_DT
)

Supports the join
← Join predicate
← COL=LIT, potentially skewed, not that selective. Can’t be matching.
← High estimated filtering, poor actual filtering range predicate

Overlay table size

Table: SETL_TRANS CARDF 1,600,254 NPAGES 21,627
INDEX IXSTRN01
(PROCESS_DT, CLR_CYCLE_CD, ADV_ABA_R, TYPE_CD, ACCT_NUM, STLMT_DT)

TABLE: BRANCH CARDF 31,696 NPAGES 1132
INDEX: IXBRNC02
(CUST_INACTV_DT, CUST_EFCT_DT)
INDEX: IXBRNC01
(ACCT_NUM, CUST_EFCT_DT)

TABLE: BRANCH_ADDR CARDF 58,627 NPAGES 2791

INDEX: IXBRAD01
(CUST_INACTV_DT, CUST_EFCT_DT)
INDEX: IXBRAD02
(ACCT_NUM, ADDR_TYP_CD, CUST_EFCT_DT)

Key:
RED = Range predicate, stops matching
BLUE: Join predicate
PURPLE: Local equals predicate / in-list

SETL_TRANS
join index analysis

```
SELECT  …
FROM            SETL_TRANS S
WHERE           S.ADV_ABA_R = ?
AND  S.PROCESS_DT < ‘9999-12-31’
AND  S.TYPE_CD  IN ('A', 'C', ‘X‘, ‘Z’)
AND  S.CLR_CYCLE_CD IN ('EOD', 'IMD', ‘OPN')
AND  S.STLMT_DT = ?
AND  S.ACCT_NUM   = CUST.ACCT_NUM
AND  S.ACCT_NUM = A.ACCT_NUM
```

CARDF 1,600,254 NPAGES 21,627
COLCARDF 19,712
COLCARDF 11
COLCARDF 4
COLCARDF 3
COLCARDF 13
COLCARDF 15,360 / 26,527
COLCARDF 15,360 / 26,527

Candidate indexes
Table: AJT_SETL_TRANS

INDEX opt_1 **(Support good local filtering + join ordering)**
(ADV_ABA_R **← COL = lit on most selective predicate to allow matching**
, STLMT_DT **← COL = lit on moderately selective predicate**
, ACCT_NUM)

INDEX opt_2 (Support for join)
(ACCT_NUM)

Note: Could also reverse the column sequence of STLMT_DT, ADV_ABA_R in the first option.

Summary of this SQL

- Indexes on BRANCH, BRANCH_ADDR look better than they are
  - Range predicate with parameter marker estimates 3% of rows qualify
  - In reality, 99% qualify
- Inefficient index available on SETL_TRANS table
  - No efficient outer table index available
  - No efficient inner table index available
  - This is the biggest table, with the best filter!!!
- Optimizer bad join method due to combination of above factors
  - Performed full scan of transaction index 26,000 times
- Resolution:
  - Providing new index on SETL_TRANS should provide more stable, faster access than ever before
  - REOPT, or providing literal values avoids the disaster without new index

Agenda

- Preparing a query for analysis
- Primary cause of access path problems
- Query walkthrough
- Verify selectivity estimates

Verify selectivity estimates

- Determine if optimizer's filter factor estimates are correct
  - Estimated filter factor
    DSN_PREDICAT_TABLE.FILTER_FACTOR
  - Actual filter factor
    Number of rows that qualify / table cardinality

- When the actual filtering differs significantly from the estimate
  - DB2 is less likely to choose an efficient access path
  - The chosen access path is more likely to be prone to access path change / regressions when
    - Running statistics
    - Applying maintenance
    - Upgrading DB2

Which index choice is better?

- Perfectly clustered index on RATE_PKG_CODE
- Good clustering index on LEVEL_ID

SELECT ...	
FROM CUSTOMER	CARDF = 137,526,126
WHERE LEVEL_ID BETWEEN ? AND ?	COLCARDF = 162332
	FF = 0.1%
AND LEVEL_ID2 BETWEEN ? AND ?	COLCARDF = 203254
AND TYPE_CODE BETWEEN ? AND ?	COLCARDF = 6
AND AGE_CODE BETWEEN ? AND ?	COLCARDF = 92
AND SEX_CODE BETWEEN ? AND ?	COLCARDF = 2
AND GROUP_CODE BETWEEN ? AND ?	COLCARDF = 104832
	FF = 0.1%
AND STATE_CODE BETWEEN ? AND ?	COLCARDF = 2
AND COUNTY_CODE BETWEEN ? AND ?	COLCARDF = 2
AND EMPLOYMENT_CODE BETWEEN ? AND ?	COLCARDF = 5

See any suspicious predicates?

Actual filtering

- What values are used for the parameter markers when the query performs poorly?
  - LEVEL_ID BETWEEN 5278 AND 5278
  - RATE_PKG_CODE BETWEEN '' and '9999'

```
SELECT COUNT(*)    = 102,642
FROM CUSTOMER
WHERE LEVEL_ID BETWEEN 5278 AND 5278
FOR FETCH ONLY WITH UR;

SELECT COUNT(*)    = 137,526,126
FROM CUSTOMER
WHERE GROUP_CODE BETWEEN '' AND '9999'
FOR FETCH ONLY WITH UR;
```

	Estimated FF	Actual FF
LEVEL_ID	0.1%	102642/137526126 = **0.07%**
GROUP_CODE	0.1%	137526126/137526126 = **100%**

Solutions

- Don't provide optimizer a choice
  - If the predicate doesn't filter any data, don't include it in the query
- Provide the actual value for optimization
  - REOPT
  - Code the literal value in the query
- Force the desired access path
  - Optimization hint
  - Query trick

Which join order is better?

- Generally better filtering early yields better performance
- Which table has better filtering?
  - T51
  - T07

```
SELECT ...
FROM T02907 T07

INNER JOIN F02951 T51

WHERE T07.TMA = ?
  AND T51.INIT_STATUS_CD BETWEEN '030' AND '089'

  AND T51.STATUS_CD IN ('5','7')
  AND T51.ORG LIKE ?
ORDER BY T07.NBR;
```

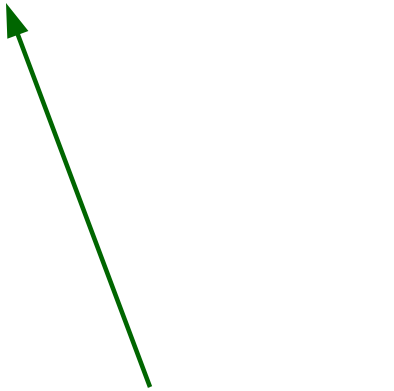
```
CARDF =  941360
NPAGES =  7583
CARDF =  521876
NPAGES =  43033
COLCARDF = 521
COLCARDF = 10
L2K / H2K = 31 / 89
COLCARDF = 25
COLCARDF = 23226
```

See any suspicious predicates?

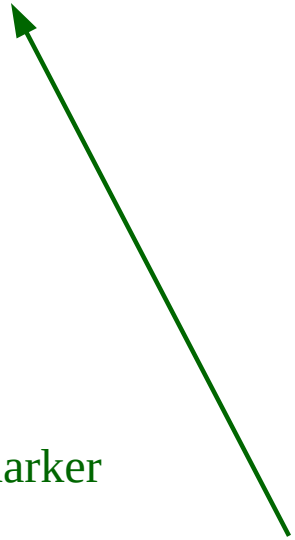
Suspicious predicates

```
SELECT ...
FROM T02907 T07

INNER JOIN F02951 T51

WHERE T07.TMA = ?
  AND T51.INIT_STATUS_CD BETWEEN '030' AND '089'

  AND T51.STATUS_CD IN ('5','7')
  AND T51.ORG LIKE ?
ORDER BY T07.NBR;
```

```
CARDF =  941360
NPAGES =  7583
CARDF =  521876
NPAGES =  43033
COLCARDF = 521
COLCARDF = 10
L2K / H2K = 31 / 89
COLCARDF = 25
COLCARDF = 23226
```



Range predicate with parameter marker



Status columns with low cardinality – Likely skewed

Estimated filter factors

```
SELECT ...
FROM T02907 T07

INNER JOIN F02951 T51

WHERE T07.TMA = ?

  AND T51.INIT_STATUS_CD BETWEEN '030' AND '089'


  AND T51.STATUS_CD IN ('5','7')

  AND T51.ORG LIKE ?

ORDER BY T07.NBR;
```

CARDF = 941360
NPAGES = 7583
CARDF = 521876
NPAGES = 43033
COLCARDF = 521
FF = 1/521 = 0.2%
COLCARDF = 10
L2K / H2K = 31 / 89
FF ~= 100%
COLCARDF =52
FF = 2/52 = 3%
COLCARDF = 23226
FF = 0.3%

- Equal predicate filtering on T07
- Good filtering via the LIKE and IN predicates
  - T51 looks more attractive as the outer

Actual table filtering

```
SELECT COUNT(*)     = 347821
FROM F02951 T51
WHERE T51.INIT_STATUS_CD BETWEEN '030' AND '089'
  AND T51.STATUS_CD IN ('5','7')
  AND T51.ORG LIKE 'Y11%'
FOR FETCH ONLY WITH UR;
```

```
SELECT COUNT(*)  = 1885
FROM T02907 T07
WHERE T07.TMA = 781
FOR FETCH ONLY WITH UR;
```

	Estimated FF	Actual FF
T07	0.2%	1885/941360 = **0.2%**
T51	0.3% * 3% = 0.009%	347821/521876 = **66%**

Where did the estimate go wrong?

Actual predicate filtering

```
SELECT COUNT(*)     = 521876
FROM F02951 T51
WHERE T51.INIT_STATUS_CD BETWEEN '030' AND '089'
FOR FETCH ONLY WITH UR;
```

```
SELECT COUNT(*)     = 421001
FROM F02951 T51
WHERE T51.ORG LIKE 'Y11%'
FOR FETCH ONLY WITH UR;
```

```
SELECT STATUS_CD, COUNT(*)
FROM F02951 T51
GROUP BY T51.STATUS_CD
ORDER BY 2 DESC
FOR FETCH ONLY WITH UR;
```

```
                +--------------------------+
                |    STATUS_CD  |          |
                +--------------------------+
  1_| 7             |   398456 | 76.3%
  2_| F             |    76504 | 14.7%
  3_| A             |    32451 |  6.2%
  4_| J             |     6872 |  1.3%
  5_| L             |     2144 |  0.4%
 .....
 42_| 5             |       23 |  0.004%
 .....only top 5 of 52 shown (plus value of 5)
```

	Estimated FF	Actual FF
INIT_STATUS_CD	~100%	521876/521876 = **100%**
ORG	0.3%	421001/521876 = **80.6%**
STATUS_CD	3%	(398456+23)/521876 = **76.3%**

Which join order is better?

```
SELECT ...
FROM T02907 T07

INNER JOIN F02951 T51

WHERE T07.TMA = ?


 AND T51.INIT_STATUS_CD BETWEEN '030' AND '089'



 AND T51.STATUS_CD IN ('5','7')


 AND T51.ORG LIKE ?


ORDER BY T07.NBR;
```

CARDF = 941360
NPAGES = 7583
CARDF = 521876
NPAGES = 43033
COLCARDF = 521
Estimated FF = 1/521 = 0.2%
Actual FF = 0.2%
COLCARDF = 10
L2K / H2K = 31 / 89
Estimated FF ~= 100%
Actual FF = 100%
COLCARDF =52
Estimated FF = 2/52 = 3%
Actual FF = 76.3%
COLCARDF = 23226
FF = 0.3%
Actual FF = 80.6%

- Problems
  - Data skew on STATUS_CD not represented to optimizer via frequency statistics
  - Default range predicate filtering factor on ORG
  - Data skew on ORG

Solution

- Provide missing information to optimizer
  - Capture frequency statistics on STATUS_CD
    COLGROUP(STATUS_CD) FREQVAL COUNT 10
  - Capture histogram statistics on ORG
    COLGROUP(ORG) HISTOGRAM NUMQUANTILES 100
  - Allow optimizer to use the histogram statistics on ORG
    - Code the literal value in the query
    - REOPT

- Does solving the problem on column STATUS_CD provide a stable access path without the overhead of REOPT?

Conclusion – What did we discuss?

- Format, annotate with key statistics
  - Tables – CARDF, NPAGES
  - Predicates – COLCARDF, LOW2KEY, HIGH2KEY, filter factor estimate
- Observe suspicious predicates
  - Predicates that are likely to be incorrectly estimated
- Primary cause of access path problems is incorrect selectivity estimates
  - Are table level estimates reasonable based on your knowledge?
    - → If you don't know – perform counts to find out if estimates are accurate
    - → If you don't know how selective things are, how will you know what the best path should be?
  - Are predicate level filtering estimates reasonable?
- Examine EXPLAIN output
  - Does optimizer choose the path you expect?
  - If not, you should have better understanding of what makes other access paths competitive, tuning can be more targeted
    - Eg. Certain predicate appears filtering, but is not.
    - Can use REOPT, or trick – targeted to solve a specific problem.
- Skilled targeted tuning is less susceptible to regression than blind tuning (where problem is not understood)

Tuning queries like a DB2 developer

Jase Alpers

IBM

alpersj@us.ibm.com