A Look Under the Hood of CBO
The 10053 Event

Wolfgang Breitling
breitliw@centrexcc.com
Agenda

Overview
Trace contents
Table, Index and Column Statistics
Cost Calculations
  Single Table Access Costs
  Join Costs
Event 10053

Event 10053 details the choices made by the CBO in evaluating the execution path for a query

Event 10053 externalizes the information that the optimizer uses in generating a plan for a query
Setting Event 10053

for your own session

on:

alter session set events
'10053 trace name context forever[, level {1|2}]'

off:

alter session set events
'10053 trace name context off'
Setting Event 10053

for another session

on:

sys.dbms_system.set_ev
(<sid>, <serial#>, 10053, {1|2}, '')

off:

sys.dbms_system.set_ev
(<sid>, <serial#>, 10053, 0, '')
Trace Generation

When the statement is parsed by the CBO

1. the statement is parsed
   and
2. the statement is parsed by the CBO
Trace Contents

- Query
- Parameters used by the optimizer
- Base Statistical Information
- Base Table Access Cost
- Join Order and Method Computations
- Recosting for special features
Because the relationship between employee and department is one-to-one:

```sql
QUERY

select dname, ename from emp, dept
where emp.deptno = dept.deptno
and ename = :b1
```
Parameters used by the Optimizer

OPTIMIZER_FEATURES_ENABLE
OPTIMIZER_INDEX_CACHING
OPTIMIZER_INDEX_COST_ADJ
OPTIMIZER_PERCENT_PARALLEL
OPTIMIZER_DYNAMIC_SAMPLING = 1
HASH_MULTIBLOCK_IO_COUNT
DB_FILE_MULTIBLOCK_READ_COUNT
HASH_AREA_SIZE
SORT_AREA_SIZE
BASE STATISTICAL INFORMATION

**************************
Table stats  Table: EMP  Alias: EMP

TOTAL :: CDN: 72130  NBLKS: 900  AVG_ROW_LEN: 42

Column: DEPTNO  Col#: 8  Table: EMP  Alias: EMP

NDV: 12  NULLS: 0  DENS: 3.1935e-05

FREQUENCY HISTOGRAM: #BKT: 339  #VAL: 12

-- Index stats

INDEX NAME: EMP_1  COL#: 1

TOTAL :: LVLS: 1  #LB: 283  #DK: 73227  LB/K: 1  DB/K: 1  CLUF: 5392

INDEX NAME: EMP_2  COL#: 2


INDEX NAME: EMP_3  COL#: 8

TOTAL :: LVLS: 2  #LB: 483  #DK: 12  LB/K: 40  DB/K: 389  CLUF: 4673
Table Statistics

trace:

Table stats    Table: EMP    Alias: EMP
   TOTAL ::  CDN: 72130  NBLKS: 903  AVG_ROW_LEN: 39

dba_tables:

NUM_ROWS             : 72130
BLOCKS               : 903
EMPTY_BLOCKS         : 0
AVG_SPACE            : 0
AVG_ROW_LEN          : 39
Index Statistics

trace:

-- Index stats
  INDEX NAME: EMP_1  COL#: 1
  TOTAL ::  LVLS: 1   #LB: 308  #DK: 72130  LB/K: 1  DB/K: 1  CLUF: 4922
  INDEX NAME: EMP_2  COL#: 2
  TOTAL ::  LVLS: 2   #LB: 352  #DK: 42  LB/K: 8  DB/K: 378  CLUF: 15883
  INDEX NAME: EMP_3  COL#: 8

dba_indexes:

BLEVEL            : 2
LEAF_BLOCKS       : 352
DISTINCT_KEYS     : 42
AVG_LEAF_BLOCKS_PER_KEY : 8
AVG_DATA_BLOCKS_PER_KEY : 378
CLUSTERING_FACTOR : 15883
Column Statistics

trace:

Column:      ENAME  Col#: 2      Table: EMP    Alias: EMP
            NDV: 42        NULLS: 0         DENS: 2.3810e-02
            NO HISTOGRAM: #BKT: 1 #VAL: 2

dba_tab_columns:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUM_DISTINCT</td>
<td>42</td>
</tr>
<tr>
<td>LOW_VALUE</td>
<td>4144414D53</td>
</tr>
<tr>
<td>HIGH_VALUE</td>
<td>77617264</td>
</tr>
<tr>
<td>DENSITY</td>
<td>.0238095238095238</td>
</tr>
<tr>
<td>NUM_NULLS</td>
<td>0</td>
</tr>
<tr>
<td>NUM_BUCKETS</td>
<td>1</td>
</tr>
</tbody>
</table>
Single Table Access Path

SINGLE TABLE ACCESS PATH
TABLE: EMP     ORIG CDN: 72130  ROUNDED CDN: 1717  CMPTD CDN: 1717
Access path: tsc  Resc: 88  Resp: 88
Access path: index (equal)
   Index: EMP_2
TABLE: EMP
   RSC_CPU: 0   RSC_IO: 397
IX_SEL: 0.0000e+00  TB_SEL: 2.3810e-02
Access path: index (equal)
   Index: EMP_2
TABLE: EMP
   RSC_CPU: 0   RSC_IO: 16
IX_SEL: 2.3810e-02  TB_SEL: 2.3810e-02
Access path: index (no sta/stp keys)
   Index: EMP_3
TABLE: EMP
   RSC_CPU: 0   RSC_IO: 485
IX_SEL: 1.0000e+00  TB_SEL: 1.0000e+00
Cardinality Estimate

TABLE: EMP
ORIG CDN: 72130
ROUNDED CDN: 1717
CMPTD CDN: 1717

TB_SEL: 2.3810e-02
Column: ENAME ... NDV: 42 ... DENS: 2.3810e-02

ORIG CDN * TB_SEL = CMPTD CDN
72130 * 2.3810e-02 = 1717.415
Base Access Plans

2 Table Scan
3 Index Unique
4 Index Range
5 Index And-Equal
23 index fast full scan
Cardinality Estimate

<table>
<thead>
<tr>
<th>Column: ENAME</th>
<th>Col#: 2</th>
<th>Table: EMP</th>
<th>Alias: EMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>NDV: 14</td>
<td>NULLS: 0</td>
<td>DENS: 1.6667e-001</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE: EMP</th>
<th>ORIG CDN: 855</th>
<th>CMPTD CDN: 143</th>
</tr>
</thead>
<tbody>
<tr>
<td>BEST_CST: 1.00</td>
<td>PATH: 2</td>
<td>Degree: 1</td>
</tr>
</tbody>
</table>

\[
142.5 = 855 \times 1.6667e^{-001}
\]

CMPTD CDN = ORIG CDN * FF
Table Scan Cost

Table stats  Table: D  Alias: D
TOTAL :: CDN: 115630  NBLKS: 4339
SCAN_CST: 265

4339 / 265 = 16.373

Table stats  Table: A  Alias: A
TOTAL :: CDN: 454503  NBLKS: 8975
SCAN_CST: 548

8975 / 548 = 16.377

SCAN_CST = NBLKS / k
Table Scan Cost and multi_block_read_count

April 24, 2003 © Centrex Consulting Corporation, Wolfgang Breitling
<table>
<thead>
<tr>
<th>Predicate</th>
<th>Filter factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>c1 = value</td>
<td>1/c1.num_distinct</td>
</tr>
<tr>
<td>c1 like value</td>
<td>1/c1.num_distinct</td>
</tr>
<tr>
<td>c1 &gt; value</td>
<td>(Hi - value) / (Hi - Lo)</td>
</tr>
<tr>
<td>c1 &lt; value</td>
<td>(value - Lo) / (Hi - Lo)</td>
</tr>
</tbody>
</table>
## Predicates and Filter Factors without Bind Variables

<table>
<thead>
<tr>
<th>Predicate</th>
<th>filter factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_1 \geq value$</td>
<td>((\text{Hi} - \text{value}) / (\text{Hi} - \text{Lo})) + $c_1\text{.num_distinct}$</td>
</tr>
<tr>
<td>$c_1 \leq value$</td>
<td>((\text{value} - \text{Lo}) / (\text{Hi} - \text{Lo})) + $c_1\text{.num_distinct}$</td>
</tr>
<tr>
<td>$c_1$ between $\text{upper}$ and $\text{lower}$</td>
<td>((\text{upper} - \text{lower}) / (\text{Hi} - \text{Lo})) + $2 \times c_1\text{.num_distinct}$ or the selectivity of $c_1 \leq \text{upper}$ if that is smaller</td>
</tr>
</tbody>
</table>
## Predicates and Filter Factors with Bind Variables

<table>
<thead>
<tr>
<th>Predicate</th>
<th>Filter Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>c1 = :b1</code></td>
<td><code>c1.density</code></td>
</tr>
<tr>
<td><code>c1 like :b1</code></td>
<td>`{ 5.0e^{-02}</td>
</tr>
<tr>
<td>`c1 { &gt;</td>
<td>&gt;=</td>
</tr>
<tr>
<td><code>c1 between :b1 and :b2</code></td>
<td><code>2.5e^{-03} \times (5.0e^{-02} \times 5.0e^{-02})</code></td>
</tr>
</tbody>
</table>
### Predicates and Filter Factors

#### Combining Predicates

<table>
<thead>
<tr>
<th>Predicate</th>
<th>filter factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1 AND P2</td>
<td>FF1 * FF2</td>
</tr>
<tr>
<td>P1 OR P2</td>
<td>FF1 + FF2 - FF1 * FF2</td>
</tr>
<tr>
<td>NOT P1</td>
<td>1 - FF1</td>
</tr>
</tbody>
</table>
Column Statistics and Histograms

- Value Based Histogram
  # buckets = NDV

- Height Based Histogram
  # buckets < NDV
Value Based Histogram

- Predicate matches one of the values in the histogram:

  \[
  \begin{align*}
  &< \\ &<= \\ &=
  
  \text{selectivity} = \frac{\text{EP of prior row}}{\text{num\_rows}} \quad \frac{\text{EP of matching row}}{\text{num\_rows}} \quad \frac{\text{difference}}{\text{num\_rows}}
  \end{align*}
  \]

- Example:

<table>
<thead>
<tr>
<th>table</th>
<th>column</th>
<th>EP</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSTREESELECT06</td>
<td>SELECTOR_NUM</td>
<td>10001</td>
<td>651</td>
</tr>
<tr>
<td>PSTREESELECT06</td>
<td>SELECTOR_NUM</td>
<td>10242</td>
<td>664</td>
</tr>
</tbody>
</table>

  \[
  \begin{align*}
  \text{selector\_num} < 664 & \quad \text{selectivity} = 10001 / \text{num\_rows} \\
  \text{selector\_num} \leq 664 & \quad \text{selectivity} = 10242 / \text{num\_rows} \\
  \text{selector\_num} = 664 & \quad \text{selectivity} = (10242-10001) / \text{num\_rows}
  \end{align*}
  \]
Value Based Histogram

- Predicate does not match one of the values in the histogram. Since this is a value base histogram that means there are no rows in the table with that value for the column and therefore the selectivity should be 0. However, the optimizer uses the density from the column statistics as selectivity.

- Bind Variable predicate: The selectivity is taken as $1/\text{num\_distinct}$, effectively ignoring the histogram.
Height Based Histogram

- the
Index Access Costs

Unique scan  blevel + 1

Fast full scan  leaf_blocks / k

Index-only  blevel + FF*leaf_blocks

Range scan  blevel + FF*leaf_blocks + FF*clustering_factor
## Index Access Costs

<table>
<thead>
<tr>
<th>INDEX#</th>
<th>Col#</th>
<th>LVLS</th>
<th>#LB</th>
<th>#DK</th>
<th>CLUF</th>
</tr>
</thead>
<tbody>
<tr>
<td>8417</td>
<td>27, 1</td>
<td>1</td>
<td>13100</td>
<td>66500</td>
<td>1469200</td>
</tr>
<tr>
<td>8418</td>
<td>1, 12, 7</td>
<td>2</td>
<td>19000</td>
<td>74700</td>
<td>1176500</td>
</tr>
<tr>
<td>8419</td>
<td>3, 1, 4, 2</td>
<td>2</td>
<td>31000</td>
<td>49700</td>
<td>118000</td>
</tr>
<tr>
<td>15755</td>
<td>1, 12, 8</td>
<td>1</td>
<td>12600</td>
<td>18800</td>
<td>1890275</td>
</tr>
</tbody>
</table>

Col#: 1, NDV: 10, DENS: 1.0000e-001
Col#: 12, NDV: 8, DENS: 1.2500e-001
Col#: 8, NDV: 33, DENS: 3.0303e-001

\[ \begin{align*}
2 & \quad 2 \\
+ 19000 \times 1.0000e^{-1} \times 1.2500e^{-1} & = 237.5 \\
+ 1176500 \times 1.0000e^{-1} \times 1.2500e^{-1} & = 14706.25 \\
& = 14945.75
\end{align*} \]

Access path: index (scan)  INDEX#: 8418  CST: 14947
### Index Access Costs

<table>
<thead>
<tr>
<th>INDEX#</th>
<th>Col#</th>
<th>LVLS</th>
<th>#LB</th>
<th>#DK</th>
<th>CLUF</th>
</tr>
</thead>
<tbody>
<tr>
<td>8417</td>
<td>27, 1</td>
<td>1</td>
<td>13100</td>
<td>66500</td>
<td>1469200</td>
</tr>
<tr>
<td>8418</td>
<td>1, 12, 7</td>
<td>2</td>
<td>19000</td>
<td>74700</td>
<td>1176500</td>
</tr>
<tr>
<td>8419</td>
<td>3, 1, 4, 2</td>
<td>2</td>
<td>31000</td>
<td>49700</td>
<td>118000</td>
</tr>
<tr>
<td>15755</td>
<td>1, 12, 8</td>
<td>1</td>
<td>12600</td>
<td>18800</td>
<td>1890275</td>
</tr>
</tbody>
</table>

Col#: 1  NDV: 10  DENS: 1.0000e-001
Col#: 12 NDV: 8   DENS: 1.2500e-001
Col#: 8   NDV: 33 DENS: 3.0303e-001

\[
1 \quad 1 \\
+ 12600 \times 1.0000e^{-1} \times 1.2500e^{-1} \times 3.0303e^{-1} \quad 47.73 \\
+ 1890275 \times 1.0000e^{-1} \times 1.2500e^{-1} \times 3.0303e^{-1} \quad 7160.13 \\
\frac{7208.86}{7208.86}
\]

Access path: index (equal)  INDEX#: 15755  CST: 7209
Default Index Statistics

INDEX#: 23574  COL#: 1
  TOTAL ::  LVLS: 1  #LB: 25  #DK: 100  LB/K: 1  DB/K: 1
  CLUF: 800

INDEX#: 23575  COL#: 2
  TOTAL ::  LVLS: 1  #LB: 25  #DK: 100  LB/K: 1  DB/K: 1
  CLUF: 800

INDEX#: 23576  COL#: 8
  TOTAL ::  LVLS: 1  #LB: 25  #DK: 100  LB/K: 1  DB/K: 1
  CLUF: 800
Default Table Statistics

Table stats  Table: EMP  Alias: EMP

TOTAL :: (NOT ANALYZED)  CDN: 2240  NBLKS: 55
SCAN_CST: 4  AVG_ROW_LEN: 100

Table stats  Table: EMP  Alias: EMP

TOTAL ::  CDN: 4457  NBLKS: 55
SCAN_CST: 4  AVG_ROW_LEN: 36

CDN = NBLKS * (db_block_size – 24) / 100
Default Column Statistics

Column: ENAME  Col#: 2  Table: EMP  Alias: E
NO STATISTICS (using defaults)
NDV: 70    NULLS: 0    DENS: 1.4286e-002

Column: HIREDATE  Col#: 5  Table: EMP  Alias: E
NO STATISTICS (using defaults)
NDV: 70    NULLS: 0    DENS: 1.4286e-002

DENS = NBLKS * m
Join Costs

1. NL Join
   \[ \text{join cost} = \text{cost of accessing outer table} + \left( \text{cardinality of outer table} \times \text{cost of accessing inner table} \right) \]

2. SM Join
   \[ \text{join cost} = \left( \text{cost of accessing outer table} + \text{outer sort cost} \right) + \left( \text{cost of accessing inner table} + \text{inner sort cost} \right) \]

3. HA Join
   \[ \text{join cost} = \left( \text{cost of accessing outer table} \right) + \left( \text{cost of building hash table} \right) + \left( \text{cost of accessing inner table} \right) \]
NL Join

join cost = cost of outer table access
+ (cardinality of outer table * cost of inner table access)

Outer table: cost: 1, cdn: 4, rcz: 11, resp: 1
Inner table: EMP
  Access path: tsc, Resc: 4
  Join resc: 17, Resp: 17  \[ 17 = 1 + 4 \times 4 \]
Join Cardinality

Join cardinality:  36 = outer (4) * inner (107) *
    sel (8.3333e-002)  [flag=0]

join selectivity = 1/max[ NDV(t1.c1), NDV(t2.c2) ]
    * [ (card t1 - # t1.c1 NULLs) / card t1 ]
    * [ (card t2 - # t2.c2 NULLs) / card t2 ]
SM Join

\[ \text{join cost} = (\text{cost of accessing outer table} + \text{outer sort cost}) \]
\[ + (\text{cost of accessing inner table} + \text{inner sort cost}) \]

Outer table:
- resc: 1  cdn: 4  rcz: 11  deg: 1  resp: 1

Inner table: EMP
- resc: 4  cdn: 107  rcz: 13  deg: 1  resp: 4
SM Join

SORT resource  Sort statistics
Sort width: 3  Area size: 43008  Degree: 1
Blocks to Sort: 1  Row size: 23  Rows: 4
Initial runs: 1  Merge passes: 1  Cost / pass: 2
Total sort cost: 2

SORT resource  Sort statistics
Sort width: 3  Area size: 43008  Degree: 1
Blocks to Sort: 1  Row size: 25  Rows: 107
Initial runs: 1  Merge passes: 1  Cost / pass: 2
Total sort cost: 2

Merge join  Cost: 8  Resp: 8  \[
((1 + 2) + (4 + 2))\]
HA Join

join cost = (cost of outer table access) + (cost of building hash table) + (cost of inner table access)

Outer table: resc: 1  cdn: 4  rcz: 11  deg: 1  resp: 1

Inner table: EMP
  resc: 4  cdn: 107  rcz: 13  deg: 1  resp: 4

Hash join one ptn: 1  Deg: 1
  hash_area: 32  buildfrag: 33  probefrag: 1  ppasses: 2

Hash join  Resc: 6  Resp: 6  \[1 + 4 + 1\]
Multi-table Joins

![Diagram of multi-table joins]
Multi-table Joins

SINGLE TABLE ACCESS PATH

TABLE: A
    ORIG CDN: 683620  CMPTD CDN: 29
    BEST_CST: 467.00  PATH: 4  Degree: 1

TABLE: L1
    ORIG CDN: 125263  CMPTD CDN: 1
    BEST_CST: 33.00  PATH: 2  Degree: 1

TABLE: L
    ORIG CDN: 238504  CMPTD CDN: 129
    BEST_CST: 3.00  PATH: 4  Degree: 1
Multi-table Joins
Multi-table Joins

| TABLE: A | CMPTD CDN: | 29 |
| TABLE: L1 | CMPTD CDN: | 1 |
| TABLE: L | CMPTD CDN: | 129 |

<table>
<thead>
<tr>
<th>cost</th>
<th>card</th>
<th>operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>105</td>
<td>1</td>
<td>SELECT STATEMENT</td>
</tr>
<tr>
<td>105</td>
<td>1</td>
<td>SORT GROUP BY</td>
</tr>
<tr>
<td>102</td>
<td>1</td>
<td>NESTED LOOPS</td>
</tr>
<tr>
<td>99</td>
<td>1</td>
<td>NESTED LOOPS</td>
</tr>
<tr>
<td>33</td>
<td>1</td>
<td>TABLE ACCESS FULL L1</td>
</tr>
<tr>
<td>66</td>
<td>29</td>
<td>TABLE ACCESS BY LOCAL INDEX ROWID A:6-6</td>
</tr>
<tr>
<td>2</td>
<td>29</td>
<td>INDEX RANGE SCAN A_ACC:6-6</td>
</tr>
<tr>
<td>3</td>
<td>129</td>
<td>INDEX RANGE SCAN L</td>
</tr>
</tbody>
</table>
Metalink Notes

40656.1  Supposedly a note about event 10053. Not externally available (yet).
75713.1  Important Customer Information about numeric EVENTS
35934.1  Cost Based Optimizer - Common Misconceptions and Issues
66030.1  Relationship between optimizer_max_permutations and optimizer_search_limit
32895.1  SQL Parsing Flow Diagram
68992.1  Predicate Selectivity
104817.1 Discussion on Oracle Joins - Costs - Algorithms & Hints
67522.1  Why is my index not used?
More Metalink Notes

62364.1  Hints and Subqueries
46234.1  Interpreting Explain plan
33089.1  Troubleshooting Guide: SQL Tuning
1031826.6  Histograms: An Overview
72539.1  Interpreting Histogram Information
77228.1  How to Tell if a Table has been analyzed
70075.1  Use of bind variables in queries
31412.1  Select to show Optimizer Statistics for CBO
43214.1  Autotrace Option in 7.3
Resources

Oracle University - Course ID: 65340
Oracle8i: Everything You Always Wanted to Know about the Optimizer

asktom.oracle.com (Thomas Kyte)
www.ixora.com.au (Steve Adams)
www.hotsos.com (Cary Millsap)
www.orapub.com (Craig Shallahamer)
www.jlcomp.demon.co.uk (Jonathan Lewis)
www.oraperf.com (Anjo Kolk)
www.evdbt.com (Tim Gorman)
Which Plan is better?

### a) Plan:

<table>
<thead>
<tr>
<th>cost</th>
<th>card</th>
<th>operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>2,979</td>
<td>446</td>
<td>SELECT STATEMENT</td>
</tr>
<tr>
<td>2,979</td>
<td>446</td>
<td>SORT ORDER BY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FILTER</td>
</tr>
<tr>
<td>2,955</td>
<td>446</td>
<td>HASH JOIN</td>
</tr>
<tr>
<td>10</td>
<td>13,679</td>
<td>TABLE ACCESS FULL E</td>
</tr>
<tr>
<td>2,901</td>
<td>49,755</td>
<td>HASH JOIN</td>
</tr>
<tr>
<td>737</td>
<td>8,629</td>
<td>HASH JOIN</td>
</tr>
<tr>
<td>5</td>
<td>45</td>
<td>HASH JOIN</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>TABLE ACCESS FULL A</td>
</tr>
<tr>
<td>1</td>
<td>15</td>
<td>TABLE ACCESS FULL D</td>
</tr>
<tr>
<td>731</td>
<td>316,380</td>
<td>TABLE ACCESS FULL B</td>
</tr>
<tr>
<td>1,953</td>
<td>239,142</td>
<td>TABLE ACCESS FULL C</td>
</tr>
</tbody>
</table>

### b) Plan:

<table>
<thead>
<tr>
<th>cost</th>
<th>card</th>
<th>operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>792</td>
<td>1</td>
<td>SELECT STATEMENT</td>
</tr>
<tr>
<td>792</td>
<td>1</td>
<td>SORT ORDER BY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FILTER</td>
</tr>
<tr>
<td>790</td>
<td>1</td>
<td>HASH JOIN</td>
</tr>
<tr>
<td>760</td>
<td>83</td>
<td>HASH JOIN</td>
</tr>
<tr>
<td>758</td>
<td>11</td>
<td>NESTED LOOPS</td>
</tr>
<tr>
<td>749</td>
<td>1</td>
<td>HASH JOIN</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>TABLE ACCESS FULL A</td>
</tr>
<tr>
<td>731</td>
<td>28,762</td>
<td>TABLE ACCESS FULL B</td>
</tr>
<tr>
<td>9</td>
<td>239,142</td>
<td>TABLE ACCESS BY INDEX ROWID C</td>
</tr>
<tr>
<td>4</td>
<td>239,142</td>
<td>INDEX RANGE SCAN C_IX0</td>
</tr>
<tr>
<td>1</td>
<td>15</td>
<td>TABLE ACCESS FULL D</td>
</tr>
<tr>
<td>10</td>
<td>13,679</td>
<td>TABLE ACCESS FULL E</td>
</tr>
</tbody>
</table>
Analysis of the Explain Plan

<table>
<thead>
<tr>
<th>cost</th>
<th>card</th>
<th>operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>792</td>
<td>1</td>
<td>SELECT STATEMENT</td>
</tr>
<tr>
<td>792</td>
<td>1</td>
<td>SORT ORDER BY FILTER</td>
</tr>
<tr>
<td>790</td>
<td>1</td>
<td>HASH JOIN</td>
</tr>
<tr>
<td>760</td>
<td>83</td>
<td>HASH JOIN</td>
</tr>
<tr>
<td>758</td>
<td>11</td>
<td>NESTED LOOPS</td>
</tr>
<tr>
<td>749</td>
<td>1</td>
<td>HASH JOIN</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>TABLE ACCESS FULL A</td>
</tr>
<tr>
<td>731</td>
<td>28,762</td>
<td>TABLE ACCESS FULL B</td>
</tr>
<tr>
<td>9</td>
<td>239,142</td>
<td>TABLE ACCESS BY INDEX ROWID C</td>
</tr>
<tr>
<td>4</td>
<td>239,142</td>
<td>INDEX RANGE SCAN C_IX0</td>
</tr>
<tr>
<td>1</td>
<td>15</td>
<td>TABLE ACCESS FULL D</td>
</tr>
<tr>
<td>10</td>
<td>13,679</td>
<td>TABLE ACCESS FULL E</td>
</tr>
</tbody>
</table>
Quiz

Q When is a statement parsed by the rule based optimizer rather than the cost based optimizer?

Q When is a statement parsed by the cost based optimizer rather than the rule based optimizer?

Q How do you guarantee that a SQL statement gets parsed in order to generate a 10053 trace but avoid that it actually gets executed?
## Value Based Histogram

<table>
<thead>
<tr>
<th>table</th>
<th>column</th>
<th>EP</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSTREESELECT06</td>
<td>SELECTOR_NUM</td>
<td>9278</td>
<td>218</td>
</tr>
<tr>
<td>PSTREESELECT06</td>
<td>SELECTOR_NUM</td>
<td>9519</td>
<td>588</td>
</tr>
<tr>
<td>PSTREESELECT06</td>
<td>SELECTOR_NUM</td>
<td>9760</td>
<td>638</td>
</tr>
<tr>
<td>PSTREESELECT06</td>
<td>SELECTOR_NUM</td>
<td>10001</td>
<td>651</td>
</tr>
<tr>
<td>PSTREESELECT06</td>
<td>SELECTOR_NUM</td>
<td>10242</td>
<td>664</td>
</tr>
<tr>
<td>PSTREESELECT06</td>
<td>SELECTOR_NUM</td>
<td>18164</td>
<td>715</td>
</tr>
<tr>
<td>PSTREESELECT06</td>
<td>SELECTOR_NUM</td>
<td>26086</td>
<td>720</td>
</tr>
<tr>
<td>PSTREESELECT06</td>
<td>SELECTOR_NUM</td>
<td>26338</td>
<td>1508</td>
</tr>
<tr>
<td>PSTREESELECT06</td>
<td>SELECTOR_NUM</td>
<td>26590</td>
<td>1696</td>
</tr>
<tr>
<td>PSTREESELECT06</td>
<td>SELECTOR_NUM</td>
<td>26854</td>
<td>2564</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PSTREESELECT06</td>
<td>SELECTOR_NUM</td>
<td>87490</td>
<td>10088</td>
</tr>
<tr>
<td>PSTREESELECT06</td>
<td>SELECTOR_NUM</td>
<td>97545</td>
<td>10112</td>
</tr>
<tr>
<td>PSTREESELECT06</td>
<td>SELECTOR_NUM</td>
<td>107600</td>
<td>10114</td>
</tr>
<tr>
<td>PSTREESELECT06</td>
<td>SELECTOR_NUM</td>
<td>117655</td>
<td>10115</td>
</tr>
<tr>
<td>PSTREESELECT06</td>
<td>SELECTOR_NUM</td>
<td>127710</td>
<td>10116</td>
</tr>
<tr>
<td>PSTREESELECT06</td>
<td>SELECTOR_NUM</td>
<td>143163</td>
<td>10127</td>
</tr>
</tbody>
</table>
### Height Based Histogram

<table>
<thead>
<tr>
<th>table</th>
<th>column</th>
<th>EP</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSTREESELECT06</td>
<td>SELECTOR_NUM</td>
<td>2</td>
<td>218</td>
</tr>
<tr>
<td>PSTREESELECT06</td>
<td>SELECTOR_NUM</td>
<td>4</td>
<td>715</td>
</tr>
<tr>
<td>PSTREESELECT06</td>
<td>SELECTOR_NUM</td>
<td>5</td>
<td>720</td>
</tr>
<tr>
<td>PSTREESELECT06</td>
<td>SELECTOR_NUM</td>
<td>6</td>
<td>2564</td>
</tr>
<tr>
<td>PSTREESELECT06</td>
<td>SELECTOR_NUM</td>
<td>8</td>
<td>3449</td>
</tr>
<tr>
<td>PSTREESELECT06</td>
<td>SELECTOR_NUM</td>
<td>10</td>
<td>4235</td>
</tr>
<tr>
<td>PSTREESELECT06</td>
<td>SELECTOR_NUM</td>
<td>11</td>
<td>6904</td>
</tr>
<tr>
<td>PSTREESELECT06</td>
<td>SELECTOR_NUM</td>
<td>13</td>
<td>7294</td>
</tr>
<tr>
<td>PSTREESELECT06</td>
<td>SELECTOR_NUM</td>
<td>15</td>
<td>8206</td>
</tr>
<tr>
<td>PSTREESELECT06</td>
<td>SELECTOR_NUM</td>
<td>17</td>
<td>8274</td>
</tr>
<tr>
<td>PSTREESELECT06</td>
<td>SELECTOR_NUM</td>
<td>18</td>
<td>9343</td>
</tr>
<tr>
<td>PSTREESELECT06</td>
<td>SELECTOR_NUM</td>
<td>19</td>
<td>9879</td>
</tr>
<tr>
<td>PSTREESELECT06</td>
<td>SELECTOR_NUM</td>
<td>21</td>
<td>10112</td>
</tr>
<tr>
<td>PSTREESELECT06</td>
<td>SELECTOR_NUM</td>
<td>24</td>
<td>10114</td>
</tr>
<tr>
<td>PSTREESELECT06</td>
<td>SELECTOR_NUM</td>
<td>26</td>
<td>10115</td>
</tr>
<tr>
<td>PSTREESELECT06</td>
<td>SELECTOR_NUM</td>
<td>28</td>
<td>10116</td>
</tr>
<tr>
<td>PSTREESELECT06</td>
<td>SELECTOR_NUM</td>
<td>32</td>
<td>10127</td>
</tr>
</tbody>
</table>
Wolfgang Breitling
Centrex Consulting Corporation
www.centrexcc.com