

# Effects of Partitioning and Physical Ordering on Performance with Multi-org Oracle Financials

Mark W. Farnham

**RIGHTSIZING<sup>INC</sup>**

**36 West Street  
Lebanon, NH 03766  
Office: 603.448.1803  
[www.rightsizinginc.com](http://www.rightsizinginc.com)**

# Agenda

- Introduction
- Oracle Stores Rows in Blocks
- Multi-org by default tends to mix rows you want with rows you don't want
- Usually there is a useful account order
- Sorting one table before opening a new period can drive a useful order in the new period

# Agenda

- Reports should complete faster with less work
- Journal Imports coordinated in order complete faster with less work
- When we get to RDBMS 11g and beyond “REF” partitioning may allow
  - Guaranteed cohesion of rows to blocks by organization
  - Improved parallel non-interfering operations in a BORING layout

# Oracle stores rows in blocks

- **Row Selectivity vs. Block Selectivity**
  - High row selectivity returns a small fraction of the rows in a table
  - High block selectivity returns a small fraction of the blocks in a table
  - Predicates choose rows, but Oracle reads blocks

# Oracle stores rows in blocks

- **Row Selectivity vs. Block Selectivity**
  - How do they come to be different?
  - Order of insertion of rows
  - Row migration
  - Rows physically exist in exactly one order.

# Oracle stores rows in blocks

- **Row Selectivity vs. Block Selectivity**
  - If the rows are not in the order of the index used, you might still need to read a high fraction of blocks to get a small fraction of rows.
  - Consider ten rows of 100 in ten blocks.

# GL\_CODE\_COMBINATIONS

- **SEGMENT01 ...SEGMENT30**
- **FND\_ID\_FLEX\_SEGMENTS**
- break on id\_flex\_num skip 1;

# GL\_CODE\_COMBINATIONS

```
SQL> select id_flex_num,  
           application_column_name,segment_name,  
           segment_num,display_flag,display_size  
from fnd_id_flex_segments  
where application_id = 101  
order by id_flex_num,segment_num;
```





# GL\_CODE\_COMBINATIONS

<b>101</b>	<b>SEGMENT1</b>	<b>BUSINESS SEGMENT</b>	<b>1</b>	<b>Y</b>	<b>2</b>
	<b>SEGMENT2</b>	<b>COST CENTER</b>	<b>2</b>	<b>Y</b>	<b>5</b>
	<b>SEGMENT3</b>	<b>DIVISION</b>	<b>3</b>	<b>Y</b>	<b>2</b>
	<b>SEGMENT4</b>	<b>DEPARTMENT</b>	<b>4</b>	<b>Y</b>	<b>3</b>
	<b>SEGMENT5</b>	<b>ACCOUNT NUMBER</b>	<b>5</b>	<b>Y</b>	<b>4</b>
	<b>SEGMENT6</b>	<b>INTERCOMPANY</b>	<b>6</b>	<b>Y</b>	<b>5</b>

# GL\_CODE\_COMBINATIONS

```

50105 SEGMENT1  BENEFIT  TYPE      1  Y  2
      SEGMENT10 ACCOUNT  NUMBER  2  Y  5
      SEGMENT11 FUTURE  USE      3  Y  3
    
```

- Notice that the “101” (previous slide) and “50105” in the `id_flex_num` column corresponds to the `chart_of_accounts_id`
- So ordering `gl_code_combinations` by the `chart_of_accounts_id` and then by the relevant `segment##` columns is effective for separating charts of accounts
- But there may be more than one set of books in a chart of accounts