The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle.
Agenda

• Oracle partitioning overview
• Oracle 11g new features
• Boost Siebel performance and scalability with Oracle partitioning
• Siebel CRM partitioning strategy
• Real-world partitioning examples
• ILM for partitioned objects and multi-tier storage
• HP – Siebel 8.0 on Oracle partition
• Summary
Oracle Partitioning Overview
Overview

- Table and Index Partitioning
  - Allows data to be broken down into smaller, more manageable pieces called partitions

A nonpartitioned table can have partitioned or nonpartitioned indexes.

A partitioned table can have partitioned or nonpartitioned indexes.

Table 1

Table 2
### Core functionality

<table>
<thead>
<tr>
<th>Version</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle8</td>
<td>Range partitions, global range index</td>
</tr>
<tr>
<td>Oracle8i</td>
<td>Hash and composite range-hash partitioning</td>
</tr>
<tr>
<td>Oracle9i</td>
<td>List partitioning</td>
</tr>
<tr>
<td>Oracle9i R2</td>
<td>Composite range-list partitioning</td>
</tr>
<tr>
<td>Oracle 10g</td>
<td>Global hash indexes</td>
</tr>
<tr>
<td>Oracle 10g R2</td>
<td>1M partitions per table</td>
</tr>
</tbody>
</table>

#### Oracle 11g
- Partitioning by reference
- Virtual column partitioning
- Automatic interval partitioning
- New composite partitioning:
  - range-range, list-range, list-list, list-hash
- Partition Advisor
Oracle 11g New Features

• ILM Assistant
  • Freely available tool on OTN to enable the information lifecycle management of partitioned objects

• Oracle Database 11g Partitioning
  • Complete the basic partitioning strategies
    • New composite partitioning methods
      • Range-range, list-range, list-list, list-hash
  • Broaden and simplify partitioning usage
    • Virtual column based partitioning
    • REF Partitioning
    • Interval Partitioning
    • Partition Advisor
Virtual Columns Based Partitioning

Business Problem
- Extended Schema attributes are fully derived and dependent on existing common data
- Redundant storage or extended view definitions are solving this problem today
  • Requires additional maintenance and creates overhead

Solution
- Oracle Database 11g introduces virtual columns
  • Purely virtual, meta-data only
- Treated as real columns except no DML
  • Virtual columns can have statistics
  • Virtual columns are eligible as partitioning key
- Enhanced performance and manageability
Partitioning in Oracle Database 11g
Virtual Column-Based Partitioning for Siebel Tables

ORDERS

<table>
<thead>
<tr>
<th>ORDER_ID</th>
<th>ORDER_DATE</th>
<th>CUSTOMER_ID...</th>
<th>REGION AS (SUBSTR(ORDER_ID, 6, 2))</th>
</tr>
</thead>
<tbody>
<tr>
<td>9834-US-14</td>
<td>12-JAN-2007</td>
<td>65920</td>
<td>US</td>
</tr>
<tr>
<td>8300-EU-97</td>
<td>14-FEB-2007</td>
<td>39654</td>
<td>EU</td>
</tr>
<tr>
<td>3886-EU-02</td>
<td>16-JAN-2007</td>
<td>4529</td>
<td>EU</td>
</tr>
<tr>
<td>3699-US-63</td>
<td>02-FEB-2007</td>
<td>18733</td>
<td>US</td>
</tr>
</tbody>
</table>

- **REGION** requires no storage
- Partition by ORDER_DATE, REGION
Interval Partitioning

• Business Problem
Need to improve manageability of partitioned tables by creating partitions automatically

• Solution
  • Interval partitioning allows to extend partitions automatically when data is inserted first time

• Available Techniques
  • Interval
  • Interval-List
  • Interval-Hash
  • Interval-Range
Partitioning in Oracle Database 11g
Interval Partitioning for Siebel Tables

- Derivative of RANGE partitioning
- Creates partitions with constant width (e.g., “month”)
  - For DATE and NUMBER columns
- Partitions created on the fly, when 1st row inserted
- Existing RANGE partitions can be converted to INTERVAL

CREATE TABLE S_OPTY
  PARTITION BY RANGE (created)
  INTERVAL(‘MONTH’) ....
Ref Partitioning

- Equi-partitions a foreign key table with a primary key table
  - Foreign key table need not contain the partitioning key

**Partition Order_lines by Order_Date in Orders table**

- Foreign keys will be placed in same relative partition as parent primary keys
- Siebel Applications are not able to take advantage of this partitioning technique because Siebel tables don’t have PK and FK relationships.
Boost Siebel Performance and Scalability with Oracle Partitioning
Support Statement of Siebel on Oracle Partition

• Partitioning support status
  • Oracle/Siebel does not have an out-of-box partitioning support in the current releases
  • Oracle/Siebel considers adding an out-of-box partitioning support in the next major release

• Partitioning support statement
  • Oracle partitioning can be supported in Siebel 7.7, 7.8 and 8.0 with Oracle CBO via customization
    • Partitioning is supported in Oracle AES’ Siebel/Oracle partitioning offering as an independent service
    • Partitioning is also supported via customization approved by Oracle AES through a partitioning design/review process

CBO – Cost-Based Optimizer
Architecting a Partitioning Strategy To Boost Siebel Performance

• Rule of thumb #1:
  Partitioning is NOT a silver bullet for one’s performance problems

• Rule of thumb #2:
  Understand the performance bottleneck before determining the partitioning strategy

• Rule of thumb #3:
  Plan a proof of concept

• Rule of thumb #4:
  Plan performance / scalability benchmark and tuning

• Rule of thumb #5:
  Plan SOP (Standard Operational Procedure) changes for go-live
Partitioning Design Considerations

- An effective Oracle partitioning with Siebel application suggests
  - Address the business requirements correctly
  - Design an easy-to-maintain partitioning strategy
  - Architect a well performed partitioning on desired objects without regression on others
  - Architect multi-tier storage to deal with ever growing data volume with optimal performance
- Preparation of performance / scalability benchmark with the real-world data volume and workload
- Preparation of standard operational procedure change after table/index partitioning
- Making partitioning design easy to upgrade to future Siebel releases
Customization of Siebel Schema for Oracle Partitioning

- Replacing LONG data type with CLOB data type for all partitioned tables
- Evaluating de-normalized columns if they are parts of partition keys *
- Evaluating extension columns in extension tables if they are parts of partition keys **

* Make sure the partitioning strategy is consistent among base tables and intersection tables
** Sometimes an extension column in an extension table may have to be moved to its base table to facilitate a partitioning strategy.
Partitioning Design Examples

• For a Siebel Remote implementation
  ▪ Partition S_DOCK_TXN_LOG to reduce the lock contentions caused by heavy-duty Siebel transaction logging
    ○ Step 1 - Any long column needs to be altered to a CLOB data type. This requires Siebel repository change
    ○ Step 2 – Create 16 tablespaces to hold the 16 hash partitions as following:
      ○ TABLESPACE S_DOCK_TXN_LOG_TBL_PAR01
      ○ TABLESPACE S_DOCK_TXN_LOG_TBL_PAR02
      ○ … …
      ○ TABLESPACE S_DOCK_TXN_LOG_TBL_PAR16
    ○ Step 3 – Hash partition S_DOCK_TXN_LOG table with the partition key on TXN_ID
    ○ Step 4 – Hash partition S_DOCK_TXN_LOG_P1 index with the partition key on TXN_ID
Partitioning Design Examples (cont.)

• For a Siebel Remote implementation
  ▪ Benchmark tool
    ○ EIM ongoing batches with transaction logging turned on
  ▪ Experience gained from this benchmark
    ○ Before partition, Oracle database shows high % of waits caused by
      ○ enq: TX - index contention
      ○ enq: TX – row lock contention
    ○ DML performance degrades sharply when the # of parallel batches increases
    ○ After partition, the wait % is reduced
      ○ DML performance is stabilized when the # of parallel batches increases
    ○ With 32 or more partitions, the wait % does not drop significantly with the transaction volume under test
For a Siebel Call Center application

- Partition `S_EVT_ACT` to resolve the View performance problem
  - Step 1 – What business people want
    - Activity data is aged out by `CREATED` dates
    - Activity data aged more than 1 year will be archived but may be required for reporting purpose
    - Any activity aged more than 3 years will be purged from the table after being archived
  - Step 2 – Understand the requirements after further analysis
    - `S_EVT_ACT` table will only hold 3 years of data
    - An user may query 3 years of data occasionally
    - An user will query 1 year of data frequently
    - An activity normally is closed within a couple of weeks (less than 3 months)
    - An user normally query on open activities with other filters such as `type = 'EMAIL REQUEST'`
    - There are more than fixed list of type values for all activities
Partitioning Design Examples (cont.)

- Step 3 – Analyze the data shape
  - More than 40% open activities are aged within the last 3 months
  - Around 30% open activities are aged between 3 and 6 months
  - Around 10% open activities are aged between 6 and 12 months
  - Around 20% open activities are aged more than 1 year

- Step 4 – Design a partition strategy
  - Range partition on CREATED date column with the following two options
    - Partition on even partitions for each quarter
    - Partition on uneven partitions for last_3_month, last_3_to_6_month, last_6_to_12_month, last_12_to_36_months
Partition Design Examples (cont.)

- Step 5 – Alter Siebel configuration to take advantage of this partition strategy
  - Add filter on CREATED date for major queries
  - Design the following PDQs
    - Open_last_3mo
    - Open_last_6mo
    - Open_last_yr
    - Open_more_than_yr

- Step 6 – Benchmark and tuning
  - Comparing the even partition per quarter and uneven partition strategies
    - No obvious performance difference for queries under test
    - Concluded taking an easy maintainable partition strategy by using even partition per quarter

- Step 7 – Query tuning with partitioned table and indexes
  - Creating indexes with the leading column on CREATED to help pruning
Partition Design Examples (cont.)

• Benchmark ACTIVITY VIEW performance through UI
  • Before partition
    • Average 2.6s
  • After partition
    • Querying last 3 months open
      • Mostly less than 2s with many less than 1s
    • Querying last 6 months open
      • Similar to above
    • Querying last 1 year open
      • Around 2s
    • Querying last 3 years open
      • Around 3.2s
Partitioning Design Examples (cont.)

• For a data archive requirement – use Virtual Column Partition
  ▪ Business requires to archive historical data from S_EVT_ACT table based on the following criteria

    ( TODO_CD IN
      ( 'Opened',
        'Rejected',
        'Closed',
        'Attrition Risk'
      ) AND
      TODO_ACTL_END_DT < add_months(sysdate, -5) and
      PERFRM_BY_PER_ID is NULL and
      EVT_STAT_CD = '3 - Closed'
    ) OR
    ( LAST_UPD < add_months(sysdate, -1) and
      EVT_STAT_CD = '2 - Cancelled' and
      PERFRM_BY_PER_ID is NULL
    )
Partitioning Design Examples (cont.)

• Step 1 – customize S_EVT_ACT by adding a virtual column V_ARCHIVE_FLG as

   (case
    (TODO_CD IN ('Opened', 'Rejected', 'Closed', 'Attrition Risk') and
     TODO_ACTL_END_DT < add_months(sysdate, -5) and
     PERFRM_BY_PER_ID is NULL and EVT_STAT_CD = '3 - Closed' when TRUE then 'Y' else 'N'
    )
   or
   (LAST_UPD < add_months(sysdate, -1) and
    EVT_STAT_CD = '2 - Cancelled' and
    PERFRM_BY_PER_ID is NULL
   )
   when TRUE then 'Y'
   else 'N'
end) VIRTUAL
Partitioning Design Examples (cont.)

- Step 2 – create S_EVT_ACT table with range partitioning
  
  ```sql
  create table SIEBEL.S_EVT_ACT
  (   ... ...
    V_ARCHIVE_FLG as
    (case
      (TODO_CD IN ('Opened', 'Rejected', 'Closed', 'Attrition Risk') and
       TODO_ACTL_END_DT < add_months(sysdate, -5) and
       PERFRM_BY_PER_ID is NULL and EVT_STAT_CD = '3 - Closed'
     )
    or
    (LAST_UPD < add_months(sysdate, -1) and
     EVT_STAT_CD = '2 - Cancelled' and
     PERFRM_BY_PER_ID is NULL
    )
    when TRUE then 'Y'
    else 'N'
    end) VIRTUAL
  )
  Partition by range (CREATED, V_ARCHIVE_FLG) ... ...
  ```
Benchmark and Performance Tuning

- Tuning tips
  - Upgrade to Oracle 11g to take advantage of new features
  - Set the following parameter on Oracle instance
    - `_hash_join_enabled = TRUE`
  - Delete statistics on any table with less than 15 rows
  - CBO sometimes prefers to use the index on a partitioned column instead of a better global index
    - Check whether the partition statistics is skewed
    - Delete partition statistics as a workaround
  - Put each partition on its own tablespace
  - Design partitioning with the consideration of corresponding application configuration change
Oracle Partitioning Enables ILM

- **Facts**
  - Disk gets cheaper but cannot catch up with the data volume growth
  - Large tables normally hold many less frequently accessed data
  - Customer considers RAID 5 to lower the cost
  - RAID 5 performs worse than RAID 10 (or RAID 01)
  - Siebel does not have an out-of-box archiving/purging mechanism

- **Advantage of Oracle partitioning**
  - Large tables are candidates for partitioning
  - Architecting multi-tier storage such that Tier 1 storage in RAID 10, Tier 2 storage in RAID 5, etc.
  - Allocating most frequently accessed partitions into Tier 1 storage and less frequently accessed partitions into Tiers 2, 3, ...
  - Gaining performance boost while maintaining data accessibility instead of an archiving/purging strategy
HP – Siebel 8.0 on Oracle Partition
HP – Siebel 8.0 on Oracle Partition

- HP CRM implementation
  - Upgrade from Siebel 7.5.3 to Siebel 8.0
  - Approximately 35K users worldwide
  - Approximately 150 countries
  - Three regions have own implementation by using the same Siebel repository

- Challenges
  - Large data volume
  - Different data shape
  - Same repository

- Solution
  - Oracle partitioning on common objects with different data shape
• HP has performance issues with Siebel OPTY related Views
  • Different regions have very different data volume, different org structure and data shape
  • Different regions have different policies on retaining historical data
  • Partitioning OPTY based on time period shape the data consistently within a fixed duration
Implementation of Oracle partitioning on Siebel OPTY object as following:

- Step 1: analyze business requirement
  - Opportunity are frequently queries by Date
- Step 2: design Siebel schema change
  - Recommendation made to Siebel configuration team to make necessary changes to support partitioning Key.
- Step 3: physical schema design
  - Range partition on Date
• Step 4: performance benchmark and tuning
  • Performance comparison before/after implementation on Oracle partitioning
    • Before (some PDQs have performance issues in one region while no issue on others because of different org structure and data shapes in the region.
    • After (no performance issue on OPTY across all regions)
      • Some PDQs perform better in one region without partitioning
      • However the same PDQs perform reasonably well in all regions with partitioning
  • With partitioning, we gain the performance for all regions with the same Siebel repository. As data grows we get the ability to archive the data from partitions.
    • Significantly reduced maintenance cost to make ad hoc fixes for a specific region
Summary

• Siebel CRM application benefits from Oracle partitioning
• To enable Oracle partitioning for Siebel, adjustment on Siebel schema may be required
• To make Oracle partitioning perform well, benchmark and tuning are required
• Oracle partitioning tuning consists
  ▪ Generic tuning on the database backend
  ▪ Application specific tuning to make partitioning work with the application configuration
• Oracle Expert Services’ Siebel/Oracle partitioning offering
  ▪ Workshop to help customers design Siebel/Oracle partitioning
  ▪ Technical review on customer’s own Siebel/Oracle partitioning to assure conformance of Siebel best practice
For More Information

search.oracle.com

Demo at Oracle Database demo grounds:

Siebel and PeopleSoft Applications
with Oracle Database Technologies