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

• Concepts
• Evolution of Oracle Partitioning
  • Proven functionality in 7th generation
• New functionality of Oracle database 11g in detail
• Partitioning and ILM
• Q&A
Oracle Partitioning
Transparent to applications

Large Table
Difficult to Manage

Partition
Divide and Conquer
Easier to Manage
Improve Performance

Composite Partition
Better Performance
More flexibility to match business needs
What is Oracle Partitioning?

It is

- Powerful functionality to logically partition objects into smaller pieces
- Driven by business requirements
- Partitioning for Performance, Manageability, and Availability

It is not

- A way to physically divide – or clump - any large data set into smaller buckets
- A pre-requisite to support a specific hardware/software design
  - Hash mandatory for shared nothing systems
Agenda

- Concepts
- Evolution of Oracle Partitioning
  - Proven functionality in 7th generation
- New functionality of Oracle database 11g in detail
- Partitioning and ILM
- Q&A
## Oracle Partitioning:
### Over Ten Years of Development

<table>
<thead>
<tr>
<th></th>
<th>Core functionality</th>
<th>Performance</th>
<th>Manageability</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Oracle8</strong></td>
<td>Range partitioning</td>
<td>“Static” partition pruning</td>
<td>Basic maintenance operations: add, drop, exchange</td>
</tr>
<tr>
<td></td>
<td>Global range indexes</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Oracle8i</strong></td>
<td>Hash and composite range-hash partitioning</td>
<td>Partition-wise joins “Dynamic” pruning</td>
<td>Merge operation</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Oracle9i</strong></td>
<td>List partitioning</td>
<td></td>
<td>Global index maintenance</td>
</tr>
<tr>
<td><strong>Oracle9i R2</strong></td>
<td>Composite range-list partitioning</td>
<td>Fast partition split</td>
<td></td>
</tr>
<tr>
<td><strong>Oracle10g</strong></td>
<td>Global hash indexes</td>
<td></td>
<td>Local Index maintenance</td>
</tr>
<tr>
<td><strong>Oracle10g R2</strong></td>
<td>1M partitions per table</td>
<td>“Multi-dimensional” pruning</td>
<td>Fast drop table</td>
</tr>
</tbody>
</table>
## Oracle Partitioning:

*Over Ten Years of Development*

<table>
<thead>
<tr>
<th></th>
<th>Core functionality</th>
<th>Performance</th>
<th>Manageability</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Oracle8</strong></td>
<td>Range partitioning</td>
<td>“Static” partition pruning</td>
<td>Basic maintenance operations: add, drop, exchange</td>
</tr>
<tr>
<td></td>
<td>Global range indexes</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Oracle8i</strong></td>
<td>Hash and composite range-hash partitioning</td>
<td>Partition-wise joins</td>
<td>Merge operation</td>
</tr>
<tr>
<td></td>
<td>“Dynamic” pruning</td>
<td>“Dynamic” pruning</td>
<td></td>
</tr>
<tr>
<td><strong>Oracle9i</strong></td>
<td>List partitioning</td>
<td></td>
<td>Global index maintenance</td>
</tr>
<tr>
<td><strong>Oracle9i R2</strong></td>
<td>Composite range-list partitioning</td>
<td>Fast partition split</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Oracle10g</strong></td>
<td>Global hash indexes</td>
<td></td>
<td>Local Index maintenance</td>
</tr>
<tr>
<td><strong>Oracle10g R2</strong></td>
<td>1M partitions per table</td>
<td>“Multi-dimensional” pruning</td>
<td>Fast drop table</td>
</tr>
<tr>
<td><strong>Oracle Database 11g</strong></td>
<td>More composite choices</td>
<td></td>
<td>Interval Partitioning Partition Advisor</td>
</tr>
</tbody>
</table>
Complete the basic partitioning strategies
  • New composite partitioning methods
    • Range-range, list-range, list-list, list-hash

Enhanced Partitioning
  • Virtual column based partitioning
  • REF Partitioning
  • Interval Partitioning

Enhanced Manageability
  • Partition Advisor
Composite Partitioning in Oracle Database 11g
Composite Partitioning - Concept

Table SALES
RANGE(order_date)-RANGE(ship_date)

Jan 2006
Feb 2006
Jan 2007

Composite Partitioning - Concept

Table SALES
RANGE(order_date)-RANGE(ship_date)

- All records with order_date in March 2006 AND ship_date in May 2006
Partitioning in Oracle Database 11g
Complete Composite Partitioning

- Range – range
- List – list
- List – hash
- List – range

ORDERES

**RANGE-RANGE**
Order Date by Order Value

ORDERES

**LIST-RANGE**
Region by Order Value

ORDERES

**LIST-LIST**
Region by Customer Type
Interval Partitioning
Interval Partitioning

- Interval Partitioning
  - Extension to Range Partitioning
  - Full automation for equi-sized range partitions

- Partitions are created as metadata information only
  - Start Partition is made persistent

- Segments are allocated as soon as new data arrives
  - No need to create new partitions
  - Local indexes are created and maintained as well

No need for any partition management
Interval Partitioning

- Range partitioned tables can be extended into interval partitioned tables
  - Simple metadata command
  - Investment protection

Table SALES

Automate the partition management
Interval Partitioning

- Range partitioned tables can be extended into interval partitioned tables
  - Simple metadata command
  - Investment protection

Table SALES

```
ALTER TABLE sales (order_date DATE, ...)
SET INTERVAL(NUMTOYMINTERVAL(1,'month'));
```
REF Partitioning
REF Partitioning

Business Problem
• Related tables benefit from same partitioning strategy
  • Sample 3NF order entry data model
• Redundant storage of the same information solves this problem
  • Data overhead
  • Maintenance overhead

Solution
• Oracle Database 11g introduces REF Partitioning
  • Child table inherits the partitioning strategy of parent table through PK-FK relationship
  • Intuitive modelling
• Enhanced Performance and Manageability
Before REF Partitioning

Table ORDERS
- RANGE(order_date)
- Primary key order_id
- Redundant storage of order_date
- Redundant maintenance

Table LINEITEMS
- RANGE(order_date)
- Foreign key order_id
REF Partitioning

Table ORDERS
- RANGE(order_date)
- Primary key order_id

Table LINEITEMS
- RANGE(order_date)
- Foreign key order_id

PARTITION BY REFERENCE
- Partitioning key inherited through PK-FK relationship
Partitioning in Oracle Database 11g
Reference Partitioning

- Inherit partitioning strategy
Virtual Column based Partitioning
Virtual Columns

**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

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
Partitioning Advisor

- Considers entire query workload to improve query performance
- Advises on partitioning methods
  - Range (equal-interval), range key and interval
  - Hash, hash key
- Integrated, non-conflicting advice with Indexes, MVs

Partition Analysis

Well-tuned SQL & Schema
Oracle Partitioning

• Optimized performance

• One consistent way to manage all your data

• Reduced total cost of ownership
Information Lifecycle Management
Information Lifecycle Management
Match Lifecycle to Storage to Optimize Cost

Data Lifecycle

Active
Less Active
Historical
Archive

High Performance Storage Tier
Low Cost Storage Tier
Historical Storage Tier

ORACLE
What is ILM?

- Information Lifecycle Management encompasses the following:
  - Policies which define how to manage the data
  - Processes which actually manage the data
  - Software which implements the policies & processes
  - Hardware where the data is stored
Why is ILM Important

- Regulatory requirements are driving large increases in Retention of Historical Data
  - New types of data to retain
    - Email, voicemail, medical
  - Longer Retention Period
    - 7 to 30 years
- Very low cost retention is needed to prevent costs from skyrocketing
Information Lifecycle Management

Reduce storage costs accordingly

High Performance Storage Tier = $72 per Gb
Low cost Storage Tier = $14 per Gb
Read only Storage Tier = $7 per Gb

5% Active 35% Less Active 60% Historical

$49,800 $67,700 $58,000
Oracle has unique capabilities that make it ideal for business ILM:

- **Fine grained ILM**
  - Oracle manages the lifecycle of groups of business data down to the level of individual rows
- **Application Transparent ILM**
  - Oracle classifies business data transparently to the application
- **Low Cost ILM**
  - Oracle can use low cost storage to greatly reduce the cost of retaining data

Optimize the Cost of Retaining Data
Implementing Oracle ILM

How Do You Implement Information Lifecycle Management?
Oracle ILM Assistant

- Tool download from OTN
  - http://www.oracle.com/goto/ilm

- Requirements
  - Oracle Application Express 2.2 (formerly HTML Db)
  - Oracle 9i or greater
3 Steps to Business ILM

1. Define Data Classes

2. Create Storage Tiers for the Data Classes

3. Manage Access and Migration of Data by Class

DIGITAL DATA STORAGE
ASM Disk Groups per Storage Tier & Partitions

- Each Tier uses ASM for load balancing within the tier
- Partitions are in different disk groups
- Data is moved between disk groups using
  - Partition Move Operation, or
  - Online Reorganization of tables, or
  - Tablespace Copy followed by “rename”

High Performance Storage Tier

- Disk Group P
  - Current
  - Month
  - Last 11 months

Low Cost Storage Tier

- Disk Group L
  - Year 2002
  - Year 2001
  - 2000

Historical Tier

- Disk Group H
  - Years 1995-1999
The preceding 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.
For More Information

search.oracle.com

or

oracle.com