Exploring Oracle Database 11g/12c
Partitioning New Features
and Best Practices

Ami Aharonovich
Oracle ACE & OCP
Ami@DBAces.com

About Me

- Oracle ACE
- Oracle Certified Professional DBA (OCP)
- Founder and CEO, DBAces
- President, Israel Oracle User Group
- Oracle DBA consultant and instructor, dealing with Oracle database core technologies
- Frequent speaker at Oracle Open World annual event and various user group conferences around the globe
About DBAces

- Providing complete end-to-end 24x7 expert remote database services and leading solutions for customer’s database and big data environments
- Committed to providing the highest quality technical services delivered by the industry’s top experts
- Implementing advanced security tools and solutions
- Professional expertise in delivering on-site customized and vendor official professional trainings
- Unique winning team led by highly skilled professional world renown experts of Oracle ACE’s

Agenda

- Oracle Partitioning:
  - Basics
  - Strategies
  - Benefits
  - Best Practices
- Oracle 11g Partitioning New Features
- Oracle 12c Partitioning New Features
- Live Demo
Oracle Partitioning

- Enhances database manageability, performance, and availability for a wide variety of applications
- Allows tables and indexes to be subdivided into smaller more manageable pieces called partitions or even sub-partitions
- Each piece is a different database segment and can be managed individually and function independently of others
- One of the most important functionalities of the Oracle database, a key tool for building large systems with high performance and availability requirements

Partitioning Strategies – Single Level

- Range (Oracle 8): maps data to partitions based on ranges of partition key values for each partition
- Hash (Oracle 8i): maps data to partitions by using a hashing algorithm applied to a partitioning key
- List (Oracle 9i): maps data to partitions by using a list of discrete values for the partitioning column
- Interval (Oracle 11g): maps data to partitions or ranges that are automatically created following a specified interval
- System (Oracle 11g): allows the application to explicitly map rows to arbitrary partitions
Composite Partitioning Techniques

<table>
<thead>
<tr>
<th>1st Level Partitioning</th>
<th>2nd Level Partitioning</th>
<th>Oracle Database Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range</td>
<td>Hash</td>
<td>Oracle 8i</td>
</tr>
<tr>
<td></td>
<td>List</td>
<td>Oracle 9i</td>
</tr>
<tr>
<td></td>
<td>Range</td>
<td>Oracle 11g</td>
</tr>
<tr>
<td>List</td>
<td>Range</td>
<td>Oracle 11g</td>
</tr>
<tr>
<td></td>
<td>List</td>
<td>Oracle 11g</td>
</tr>
<tr>
<td></td>
<td>Hash</td>
<td>Oracle 11g</td>
</tr>
<tr>
<td>Hash</td>
<td>List</td>
<td>Oracle 11g</td>
</tr>
<tr>
<td></td>
<td>Range</td>
<td>Oracle 11g</td>
</tr>
<tr>
<td>Interval</td>
<td>Range</td>
<td>Oracle 11g</td>
</tr>
<tr>
<td></td>
<td>List</td>
<td>Oracle 11g</td>
</tr>
<tr>
<td></td>
<td>Hash</td>
<td>Oracle 11g</td>
</tr>
</tbody>
</table>

Partitioning Benefits

- Different partitions that belong to the same table/index can:
  - Reside in different tablespaces
  - Have distinct storage clauses
  - Be maintained by granular commands

- Transparent to existing applications

- Optimizer eliminates partitions that do not need to be scanned (Partition Pruning)

- Join operations can be optimized to perform the join “by the partition” (Partition-wise Joins)
Partitioning Best Practices

- **EXCHANGE PARTITION:**
  - One of the best features in partitioning tables
  - Swap-out standard tables and partitioned tables
  - Ability to load data quickly and easily with minimal impact on current users
    
    ```sql
    ALTER TABLE ... EXCHANGE PARTITION ... WITH TABLE ...;
    ```

- **Using Compression:**
  - Compress some or all partitions using table compression (defined at either tablespace, table or partition level)
  - Compress some or all partitions of a B-tree index using key compression to eliminate repeated occurrences of key column prefix values

---

Partitioning Best Practices

- **Copy Statistics:**
  - Available since 10.2.0.4
  - Can be used to copy statistics of the source [sub] partition to the destination [sub] partition

- **Sub-Partition Templates:**
  - Used for composite partitioned table
  - Simplifies the specification of sub-partitions by not requiring that a sub-partition descriptor be specified for every partition in the table
Oracle 11g Partitioning New Features

- Interval Partitioning
- Virtual Column-Based Partitioning
- Reference Partitioning
- System Partitioning

- Interval Partitioning
  - Automates the creation of range partitions
  - Oracle will create any partition automatically as needed whenever data for a partition is inserted for the very first time
  - Greatly improves the manageability of a ranged partitioned table
  - Available techniques are interval, interval-list, interval-hash, interval-range and interval-reference (12c only)
  - You must specify at least one range partition
  - Partitioning key column must be of NUMBER or DATE type
Interval Partitioning Example

```
CREATE TABLE INTERVAL_PARTITIONING_TBL
(serial_num NUMBER,
 name VARCHAR2(32))
PARTITION BY RANGE (serial_num) INTERVAL (10000)
( PARTITION p1 VALUES LESS THAN (10000),
 PARTITION p2 VALUES LESS THAN (20000),
 PARTITION p3 VALUES LESS THAN (30000));
```

Oracle 11g Partitioning New Features

- Virtual Column-Based Partitioning
  - Allows partitioning key to be defined by an expression, using one or more existing columns of a table and storing the expression as metadata only
  - Enables a more comprehensive match various business requirements
  - Supported with all basic partitioning strategies
  - Can also be used with interval partitioning as well as the partitioning key for reference partitioned tables
  - Virtual columns are treated as regular real columns except no DML operations are allowed
Virtual Column-Based Partitioning Example

CREATE TABLE SALES
    ( PROD_ID        NUMBER       NOT NULL,
      CUST_ID        NUMBER       NOT NULL,
      TIME_ID        DATE         NOT NULL,
      CHANNEL_ID     NUMBER       NOT NULL,
      PROMO_ID       NUMBER       NOT NULL,
      QUANTITY_SOLD  NUMBER(10,2) NOT NULL,
      AMOUNT_SOLD    NUMBER(10,2) NOT NULL,
      PROD_TYPE      NUMBER(1) AS (TO_NUMBER(SUBSTR(TO_CHAR(PROD_ID),1,1)))
    ) TABLESPACE USERS
PARTITION BY RANGE (PROD_TYPE) INTERVAL (1)
    (PARTITION p1 VALUES LESS THAN (1))

Oracle 11g Partitioning New Features

- Reference Partitioning
  - Allow to partition a table by leveraging an existing parent-child relationship
  - Partitioning strategy of parent table is inherited to its child table without the necessity to store the parent’s partitioning key column in the child table
  - Transparently inherits all partitioning maintenance operations that change the logical shape of a table from the parent table to the child table
  - Automatically enables partition-wise joins for the equal-partitions of the parent and child tables
  - Perfect for star schemas in data warehouses; partition the fact table according to the dimension table
Reference Partitioning

Without using reference partitioning

Reference partitioning

Table ORDERS

Table ORDER_ITEMS

Range (ORDER_DATE)

Primary key (ORDER_ID)

Range (ORDER_DATE)

Foreign key (ORDER_ID)

Partition key inherited through PK/FK relationship

Redundant storage/maintenance of ORDER_DATE

Oracle 11g Partitioning New Features

- System Partitioning
  - Enables application-controlled partitioning
  - Allows the application to explicitly map rows to arbitrary partitions
  - Provides the ability to break down a table into meaningless partitions
  - All aspects of partitioning are controlled by the application
  - Common performance benefits of partitioned tables are not available (there is no partitioning key)
  - No support for traditional partition pruning, partition wise joins, and so on
Oracle 12c Partitioning New Features

- Partial Indexes for Partitioned Tables
- ONLINE Move Partitions
- Partition Maintenance Operations on Multiple Partitions
- Asynchronous Global Index Maintenance for DROP and TRUNCATE Partition
- Interval Reference Partitioning

Oracle white paper: Partitioning with Oracle Database 12c

**Partial Indexes for Partitioned Tables**
- New index attribute only applicable to indexes on partitioned tables
- Indexes can be created on a subset of the partitions of a table
- Provide more flexibility in index creation for partitioned tables
- For example, you can choose not to index the most recent partition to avoid any index maintenance work at data insertion time, therefore maximizing data load speed

Partial_Indexes.sql
Oracle 12c Partitioning New Features

- **ONLINE Move Partitions**
  - Partition maintenance operations can be done in a completely online fashion, allowing DML operations to occur while the data maintenance operation is in process.
  - ALTER TABLE ... MOVE PARTITION becomes non-blocking online DDL while DML operations continue to run uninterrupted on the partition that is being moved.
  - Global indexes are maintained during the move partition, so a manual index rebuild is no longer required.
  - With the ONLINE clause, the move operation waits for the transaction to be finished (commit or rollback) and is then executed.

- **Partition Maintenance Operations on Multiple Partitions**
  - Partition maintenance operations can be performed on multiple partitions as part of a single partition maintenance operation (one atomic operation).
  - Simplifies application development and leads to more efficient partition maintenance using less system resources.
  - For example:
    ```sql
    ALTER TABLE table_name ADD
    PARTITION partition_name,...,
    PARTITION partition_name,...,
    PARTITION partition_name...;
    ```
Oracle 12c Partitioning New Features

- Asynchronous Global Index Maintenance for DROP and TRUNCATE Partition
  - Global index maintenance is decoupled from DROP and TRUNCATE partition maintenance operations without rendering a global index unusable
  - Index maintenance is done asynchronously and can be delayed to a later point in time

- Interval Reference Partitioning
  - Referenced partitioned table leverages interval partitioning as the top partitioning strategy
  - Enhances Oracle’s partitioning capabilities to model the database schema according to real business needs

```
CREATE TABLE parent_table_name
  (column_name..., column_name..., ...
  PARTITION BY RANGE (column_name) INTERVAL(n)...;

CREATE TABLE child_table_name
  (column_name..., column_name..., FOREIGN KEY...)
PARTITION BY REFERENCE (fk_constraint_name)...
```
Thank You!
Exploring Oracle Database 11g/12c
Partitioning New Features
and Best Practices

Ami Aharonovich
Oracle ACE & OCP
Ami@DBAces.com