|  |
| --- |
|  |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| REVISION HISTORY | | | | | |
| Ver. | Description of Change | Author | Date | Approved | |
| Name | Effective Date |
| 1.0 | Initial status | [Kiryl Bucha](mailto:Kiryl_Bucha@epam.com) | 12-JAN-2012 |  |  |
| 2.0 | Updated in accordance with renewed content | [Elias Nema](mailto:Elias_Nema@epam.com) | 20-JAN-2014 |  |  |
| 3.0 | Task review | Hanna Klimovich | 15-NOV-2017 |  |  |

Contents

[1. Hands-On Task 3](#_Toc383292320)

[2. Data Modeling Task 3](#_Toc383292321)

[3. Analytical task 3](#_Toc383292322)

[3.1. Partitioning 3](#_Toc383292323)

[3.2. Business 3](#_Toc383292324)

[4. Results 3](#_Toc383292325)

# Hands-On Task

Create any table with some kind of partitioning to show next concepts:

* Adding Partition.
* Coalescing Partition.
* Dropping Partition.
* Merging Partition.
* Moving Partition.
* Splitting Partition.
* Truncating Partition.

Preparations

Table to try out partitions created (2 and 7 part)

CREATE TABLE dates\_hash

(

date\_id DATE PRIMARY KEY,

dateDayMonth NUMBER,

dateDayOfWeekNameEN VARCHAR2(25),

dateMonth NUMBER,

dateQuarterYear NUMBER,

dateSemesterYear NUMBER,

dateYear NUMBER

)

PARTITION BY HASH

(

dateyear

)

PARTITIONS 4;

Partitions:

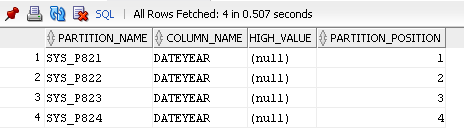


Table with Range Partitioning is used for parts 1, 3, 4, 5, 6

CREATE TABLE DATES

(

date\_id DATE PRIMARY KEY,

dateDayMonth NUMBER,

dateDayOfWeekNameEN VARCHAR2(25),

dateMonth NUMBER,

dateQuarterYear NUMBER,

dateSemesterYear NUMBER,

dateYear NUMBER

)

PARTITION BY RANGE

(

dateyear

)

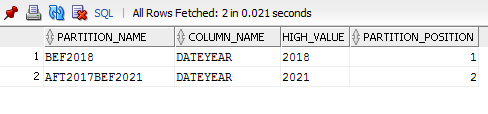
(

PARTITION bef2018 VALUES LESS THAN(2018),

PARTITION aft2017bef2021 VALUES LESS THAN(2021)

);

Partitions:

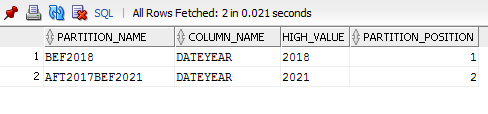


1. **Adding Partition**

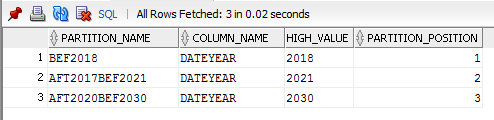
If partition is added, further some new partitions may be added manually.

ALTER TABLE dates ADD PARTITION aft2020bef2030 VALUES LESS THAN ( 2030 );

Before Alter clause:



After Alter clause:



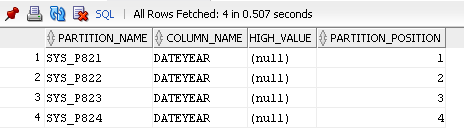
1. **Coalescing Partition**

Coalescing partitions is a way of reducing the number of partitions in a hash-partitioned table.

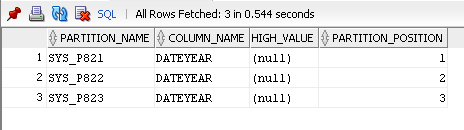
Coalescing Partition can be used only with Hash Partitioning.

ALTER TABLE DATES\_HASH COALESCE PARTITION;

Before Alter clause:



After Alter clause:



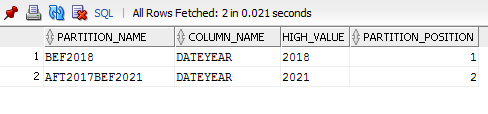
1. **Dropping Partition**

Dropping Partition drops partition with whole data in it.

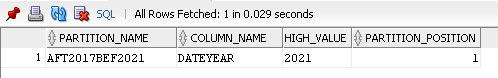
Dropping Partitions can’t be used with Hash Partitioning and Reference Partitioning.

*ALTER TABLE dates DROP PARTITION BEF2018 UPDATE GLOBAL INDEXES;*

Before Alter clause:



After Alter clause:



1. **Merging Partition**

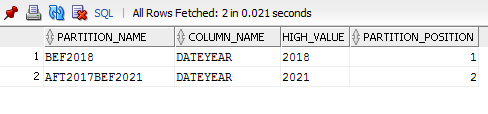
Merging Partitions merges the contents of two partitions into one partition.

Dropping Partitions can’t be used with Hash Partitioning and Reference Partitioning.

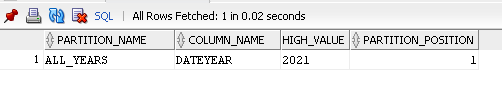
ALTER TABLE dates

MERGE PARTITIONS BEF2018, AFT2017BEF2021 INTO PARTITION ALL\_YEARS;

Before Alter clause:



After Alter clause:



1. **Moving Partition**

MOVE PARTITION clause of the ALTER TABLE statement to can be used to:

* Re-cluster data and reduce fragmentation
* Move a partition to another tablespace
* Modify create-time attributes
* Store the data in compressed format using table compression

ALTER TABLE dates MOVE PARTITION BEF2018 TABLESPACE TBS\_PDB\_TEST NOLOGGING COMPRESS;

1. **Splitting Partition**

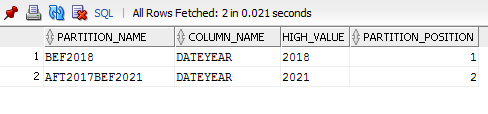
Oracle Database implements a SPLIT PARTITION operation by creating two new partitions and redistributing the rows from the partition being split into the two new partitions. This is a time-consuming operation because it is necessary to scan all the rows of the partition being split and then insert them one-by-one into the new partitions. Further if you do not use the UPDATE INDEXES clause, then both local and global indexes also require rebuilding.

ALTER TABLE dates SPLIT PARTITION

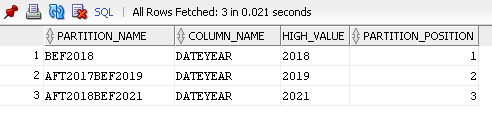
aft2017bef2021 at (2019) INTO ( PARTITION

aft2017bef2019, PARTITION aft2018bef2021);

Before Alter clause:



After Alter clause:



1. **Truncating Partition**

Use the ALTER TABLE TRUNCATE PARTITION statement to remove all rows from a table partition, with or without reclaiming space. Truncating a partition in an interval-partitioned table does not move the transition point. You can truncate partitions and subpartitions in a reference-partitioned table.

ALTER TABLE dates\_hash TRUNCATE PARTITION SYS\_P841;

# Analytical task

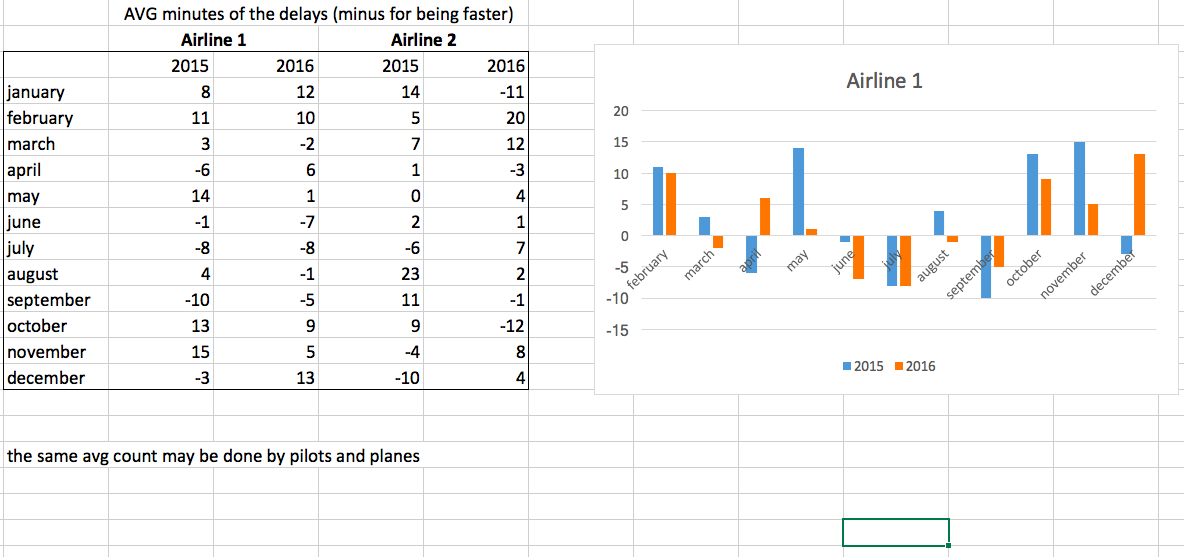
## Partitioning

Add chapter on describing Fact table partitioning strategy, which fields will it be based on and why (use composite partitioning).

Partitioning will be based on the year and month fields. To my mind is better to use range partitioning.

## Business

Create a report layout of the task you are trying to solve with your DWH (e.g. 'I want to analyze my sales on month and customers' region location and product type'). It could be done in Excel (just some headers, colors, dummy numbers, and little description) or any other tool of your preference. This would help to understand what task we are trying to solve.



# Results

Result of this lab work should be:

* Screenshots and description of partitioning maintenance operations.
* Chapter in document about advantages of partitioning the fact table in described way.
* Chapter in a document describing needed reports with possible layouts.