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.

## Creating table

### Range Partition

CREATE TABLE DimDate

(

DATE\_KEY VARCHAR2(8) NOT NULL,

FULL\_DATE DATE NOT NULL,

FULL\_DATE\_DESCR VARCHAR2(40) NOT NULL,

DAY\_OF\_WEEK NUMBER(1,0) NOT NULL,

DAY\_OF\_MONTH NUMBER(2,0) NOT NULL,

DAY\_OF\_YEAR NUMBER(3,0) NOT NULL,

DAY\_NAME VARCHAR2(10) NOT NULL,

WEEKEND\_INDICATOR VARCHAR2(40) NOT NULL,

LAST\_DAY\_OF\_WEEK\_INDICATOR VARCHAR2(40) NOT NULL,

LAST\_DAY\_OF\_MONTH\_INDICATOR VARCHAR2(40) NOT NULL,

WEEK\_OF\_MONTH NUMBER(1,0) NOT NULL,

WEEK\_OF\_YEAR NUMBER(2,0) NOT NULL,

WEEK\_ENDING\_DATE DATE NOT NULL,

MONTH\_NUMBER NUMBER(2,0) NOT NULL,

MONTH\_NAME VARCHAR2(32) NOT NULL,

MONTH\_YEAR CHAR(32) NOT NULL,

QUARTER\_NUMBER NUMBER(1,0) NOT NULL,

QUARTER\_YEAR CHAR(32) NOT NULL,

YEAR\_NUMBER NUMBER(4,0) NOT NULL,

)

PARTITION BY RANGE(date\_key)

(PARTITION sales\_2005\_2020 VALUES LESS THAN ('20200101'), --2005-2020

PARTITION sales\_2020\_2035 VALUES LESS THAN ('20350101'),

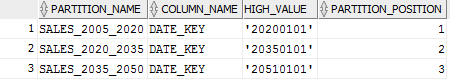
PARTITION sales\_2035\_2050 VALUES LESS THAN ('20510101')

)

;

Here was used a range partition (‘less than’ equals ‘<’).

3 partitions were created, key column – date key (format YYYYMMDD).



### Hash Partiton

CREATE TABLE DimDate\_hash

(

DATE\_KEY VARCHAR2(8) NOT NULL,

FULL\_DATE DATE NOT NULL,

FULL\_DATE\_DESCR VARCHAR2(40) NOT NULL,

DAY\_OF\_WEEK NUMBER(1,0) NOT NULL,

DAY\_OF\_MONTH NUMBER(2,0) NOT NULL,

DAY\_OF\_YEAR NUMBER(3,0) NOT NULL,

DAY\_NAME VARCHAR2(10) NOT NULL,

WEEKEND\_INDICATOR VARCHAR2(20) NOT NULL,

LAST\_DAY\_OF\_WEEK\_INDICATOR VARCHAR2(40) NOT NULL,

LAST\_DAY\_OF\_MONTH\_INDICATOR VARCHAR2(40) NOT NULL,

WEEK\_OF\_MONTH NUMBER(1,0) NOT NULL,

WEEK\_OF\_YEAR NUMBER(2,0) NOT NULL,

WEEK\_ENDING\_DATE DATE NOT NULL,

MONTH\_NUMBER NUMBER(2,0) NOT NULL,

MONTH\_NAME VARCHAR2(32) NOT NULL,

MONTH\_YEAR CHAR(32) NOT NULL,

QUARTER\_NUMBER NUMBER(1,0) NOT NULL,

QUARTER\_YEAR CHAR(32) NOT NULL,

YEAR\_NUMBER NUMBER(4,0) NOT NULL

)

PARTITION BY HASH (date\_key)

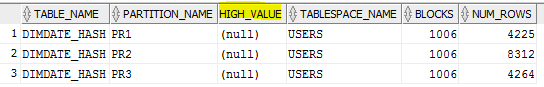
(PARTITION pr1,

PARTITION pr2,

PARTITION pr3

)

;



A row is placed into a partition based on the result of passing the partitioning key into a hashing algorithm. Using this approach, data is randomly distributed across the partitions rather than grouped. Therefore high value is null.

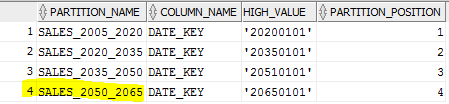
## Adding partition

ALTER TABLE DimDate ADD PARTITION sales\_2050\_2065 VALUES LESS THAN ('20650101');

BEGIN

sp\_DATE\_DIMENSION\_1(2051,2060);

END;



**N.B.**

\*In case of a table partitioned by LIST, the new partition must be specified with key values that are **not already defined**

\*In case of a table partitioned by RANGE with data, the new RANGE partition can be added with a RANGE key value **greater than the existing maximum key value of the table**

If we don’t stick to these rules **the following error** may occur: An attempt was made to insert a record into, a Range or Composite Range object, with a concatenated partition key that is beyond the concatenated partition bound list of the last partition -OR- An attempt was made to insert a record into a List object with a partition key that did not match the literal values specified for any of the partitions.

But we can create partition with a reserve (e.g. High\_value 20650101, but data is inserted till 20600101).

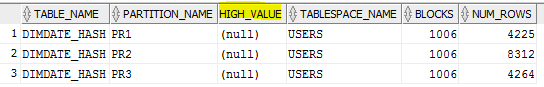
## Coalescing Partition

Coalescing partitions is a way of **reducing the number of partitions** in a hash-partitioned table. When a hash partition is coalesced, its contents are **redistributed into one or more remaining partitions** determined by the hash function. The specific partition that is coalesced is **selected by the database**, and is **dropped** after its contents have been redistributed. PR3 was coalesced.

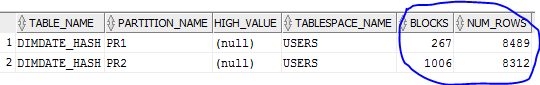
ALTER TABLE dimdate\_hash

COALESCE PARTITION;

Before:



After:

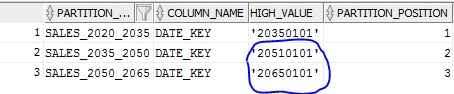


Num\_rows is the same so no data was lost.

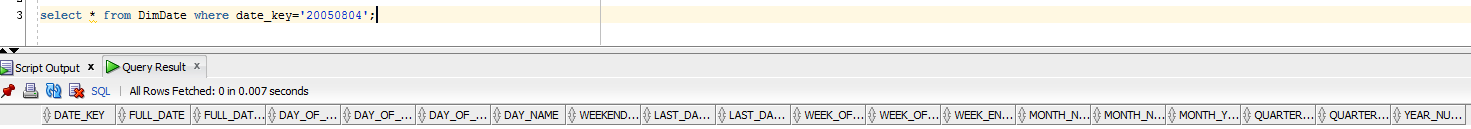
## Drop partition

Far and away, the "drop partition" syntax is the fastest way to remove large volumes of data.  By partitioning the data, dropping the whole partition takes mere seconds, as opposed to hours with traditional "delete" techniques.

ALTER TABLE DimDate DROP PARTITION sales\_2005\_2020;



Drop partition drops **a partition and its data.** So if we want to drop a partition but **keep its data** in the table, we must **merge the partition** into one of the adjacent partitions.



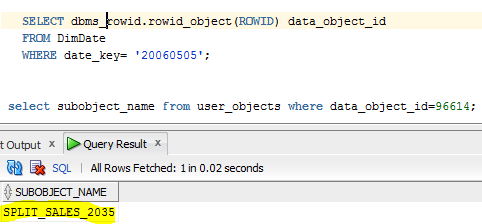
If we drop a partition and later insert a row that would have belonged to the dropped partition, the row will be stored in **the next higher partition**. However, if we drop the highest partition, the insert **will fail** because the range of values represented by the dropped partition is no longer valid for the table.

When a table contains only **one partition**, we cannot drop the partition. The table should be dropped.

1) Row was inserted:

insert into DimDate VALUES ('20060505', '21-JUL-2006', '5','5','5','5','5','5','5','5','5','5','21-JUL-2006','5','5','5','5','5','5');

2) Row was put in the closest partition (split partition from the paragraph 1.7)



## Merging Partition

Here we basically merge 2 or more partitions into one. In the MERGE PARTITION we should specify **both partitions we merge and into which partition we merge.**

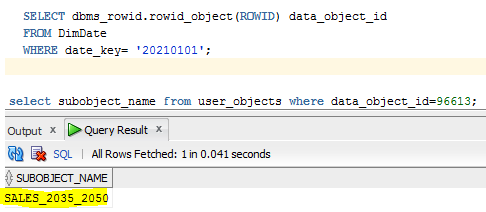
ALTER TABLE DimDate

MERGE PARTITIONS sales\_2020\_2035, sales\_2035\_2050

INTO PARTITION sales\_2035\_2050;



We can check, that date that was in Sales\_2020\_2035 partition now in Sales\_2035\_2050.



Another check:



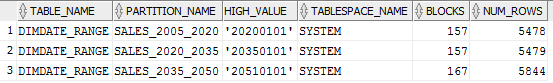
## Moving partition

Here we move a table partition to **another segment**. MOVE PARTITION always drops the partition's old segment and creates a new segment, even if you do not specify a new tablespace.

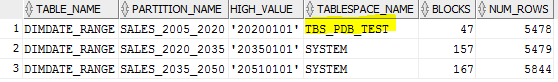
ALTER TABLE DimDate\_RANGE MOVE PARTITION sales\_2005\_2020

TABLESPACE tbs\_pdb\_test NOLOGGING COMPRESS;

Before:



After:



I did it under system, because my user hadn’t had enough rights.

## Splitting partition

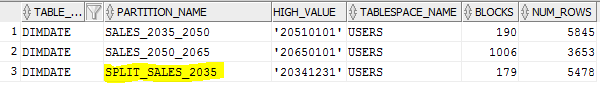
Here we **divide a partition into two partitions**.

It’s possible to split partitions in a way where each has a new segment, new physical attributes, and new initial extents. The segment associated with the old partition will be discarded.

If we **do not specify physical attributes** (PCTFREE, PCTUSED, INITRANS, MAXTRANS, STORAGE) for the new partitions, **the current values** of the partition being split are used as the default values for both partitions.

ALTER TABLE DimDate

SPLIT PARTITION SALES\_2035\_2050 AT ('20341231') INTO (PARTITION SPLIT\_SALES\_2035, PARTITION SALES\_2035\_2050);



## Truncating partition

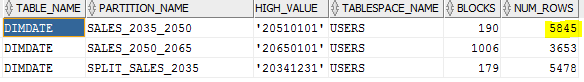
**Removes all rows** from a partition in a table. Freed space is deallocated or reused depending on whether DROP STORAGE or REUSE STORAGE is specified in the clause.

**N.B.** If we want to truncate a partition that contains data, we must first disable any referential integrity constraints on the table. Alternatively, we can delete the rows and then truncate the partition.

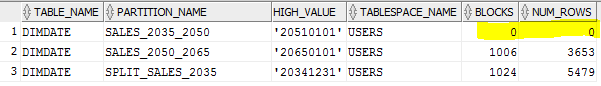
ALTER TABLE DimDate

TRUNCATE PARTITION SALES\_2035\_2050 DROP STORAGE;

Before:



After:



# Analytical task

## Partitioning

Партицирование будет осуществляться по полю Date\_key, так как нас в основном будут интересовать заказы соврешенные в последние 5 лет.

Внутри также будет партицирование (composite partitioning )по Sales\_amount, так как необходимо будет разграничивать крупных, средних и мелких покупателей.

## Business

Цель: Анализ продаж существующих категорий книг (печатные, электронные, аудио) за каждый месяц 2017 года (до октбря включительно, тк сейчас ноябрь) в Минской и Бресткой областях РБ.

* AVG – среднее значени Sales\_amount
* MAX – максимальное значени Sales\_amount
* % of all – процент выручки от общегочисла продаж

Красный цвет: самый низкий уровень продаж.

Зеленый цвет: самый высокий уровень продаж.

