Report

Laboratory Work 9

Dmitry Ladutsko

July 28, 2022

1. Prerequisites

1.1. Passwords Index

Password Group	Login Name	Password
Operation System	root	"rootadmin"
	oracle	"oracleadmin"
Oracle System	sys	"sysadmin"
	system	"sysadmin"
Oracle Users	All DB users	"%PWD%"

1.2. Folder Paths Index

Path Group	Path Description	Path
Operation System	Oracle RDBMS – BIN	/oracle/app/oracle
	Oracle Inventory	/oracle/app/oraInventory
	Oracle Database Storage	/oracle/oradata
	Oracle Install Directory	/oracle/install
Oracle	ORACLE_BASE	/oracle/app/oracle
	ORACLE_HOME	$ORACLE_BASE/product/11.2
FTP	ftp Incoming Folder	**/ftp/incoming**

2. Oracle Architecture - Partitioning

2.1. Task 01: CREATE Example of Range partitioning

The Main Task is to creating example of range partitioning table. Perform Administration tasks on all partitioning types:

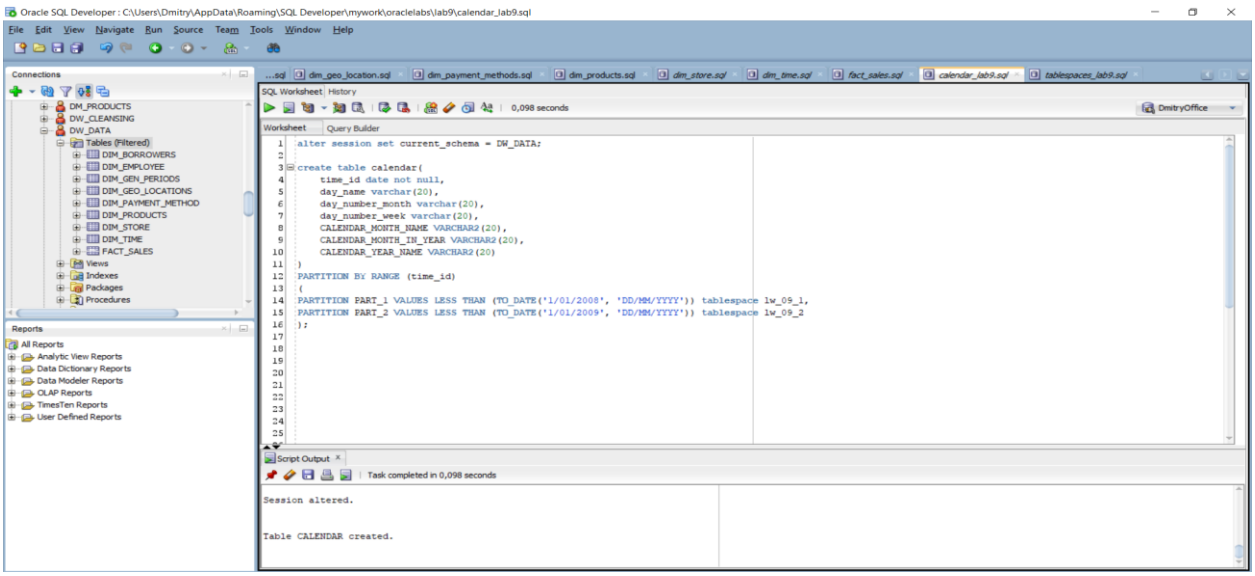
Maintenance Operation	Range Composite Range-*	Interval Composite Interval-*	Hash	List Composite List-*	Reference
Adding Partitions	ADD PARTITION	ADD PARTITION	ADD PARTITION	ADD PARTITION	N/A[Foot 1]
Coalescing Partitions	N/A	N/A	COALESCE PARTITION	N/A	N/A[Footref 1]
Dropping Partitions	DROP PARTITION	DROP PARTITION	N/A	DROP PARTITION	N/A[Footref 1]
Merging Partitions	MERGE PARTITIONS	MERGE PARTITIONS	N/A	MERGE PARTITIONS	N/A[Footref 1]
Moving Partitions	MOVE PARTITION	MOVE PARTITION	MOVE PARTITION	MOVE PARTITION	MOVE PARTITION

Maintenance Operation	Range Composite Range-*	Interval Composite Interval-*	Hash	List Composite List-*	Reference
Splitting Partitions	SPLIT PARTITION	SPLIT PARTITION	N/A	SPLIT PARTITION	N/A[Footref 1]
Truncating Partitions	TRUNCATE PARTITION	TRUNCATE PARTITION	TRUNCATE PARTITION	TRUNCATE PARTITION	TRUNCATE PARTITION

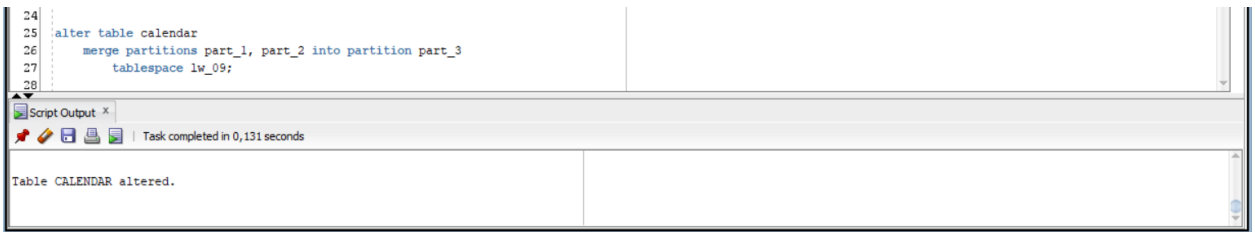
Task Results:

Create document that will store all screenshot about **Maintenance Operations**;

Range Composite



Picture 1 - Creating Partition



Picture 2 – Merging Partition

```
33  alter table calendar
34      move partition part_3
35          tablespace lw_09_1 nologging compress;
```

Script Output | Task completed in 0,089 seconds

```
Table CALENDAR altered.
```

Picture 3 - Moving Partition

```
32  alter table calendar
33      split partition part_3
34          at (to_date('1/01/2005', 'DD/MM/YYYY'))
35              into(partition part_1, partition part_2);
```

Script Output | Task completed in 0,118 seconds

```
Table CALENDAR altered.
```

Picture 4 - Spliting Partition

```
31  alter table calendar
32      truncate partition part_1;
```

Script Output | Task completed in 0,094 seconds

```
Table CALENDAR truncated.
```

Picture 5 - Truncating Partition

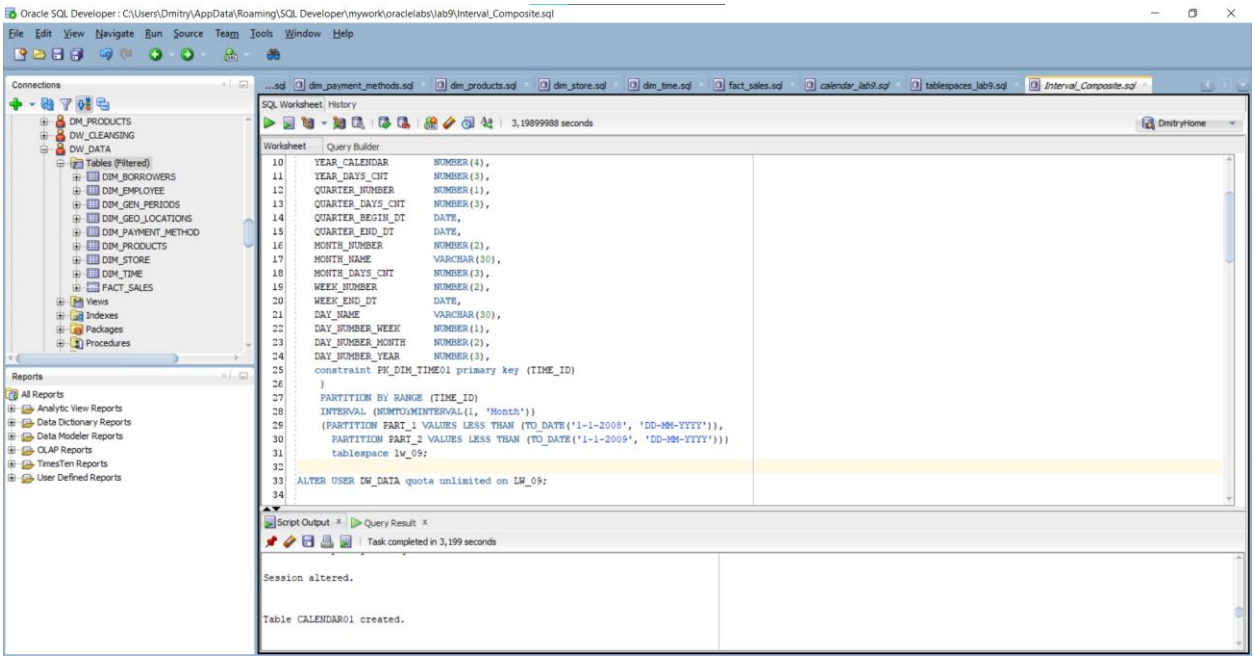
```
34  alter table calendar
35      drop partition part_1;
```

Script Output | Task completed in 0,057 seconds

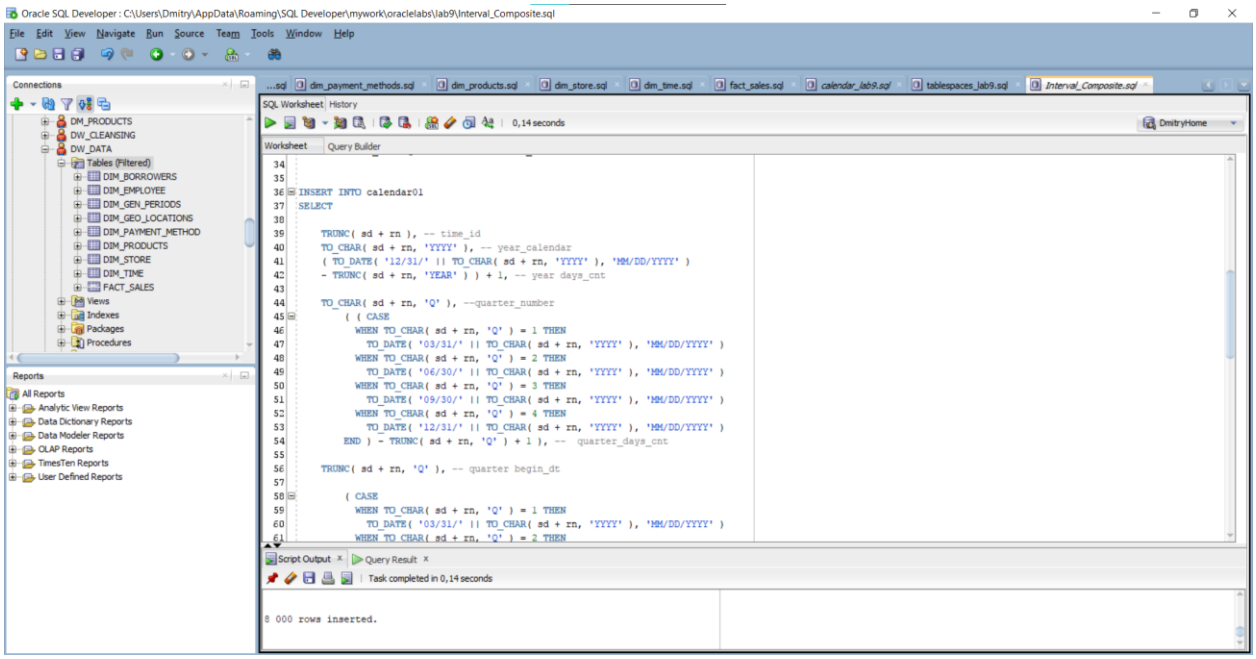
```
Table CALENDAR altered.
```

Picture 6 - Dropping Partition

Interval Composite Interval

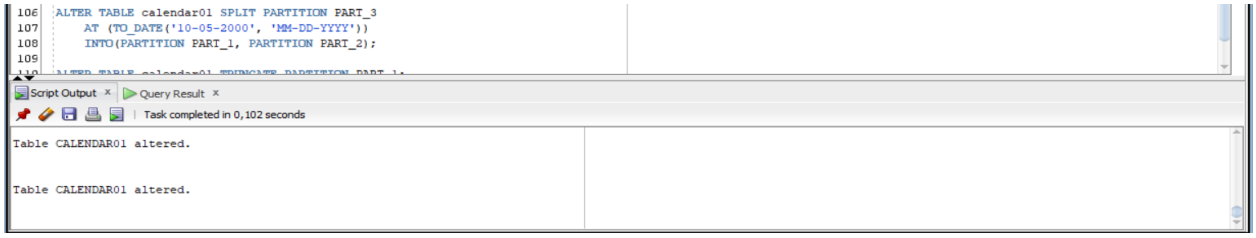


Picture 7 - Creating Partition

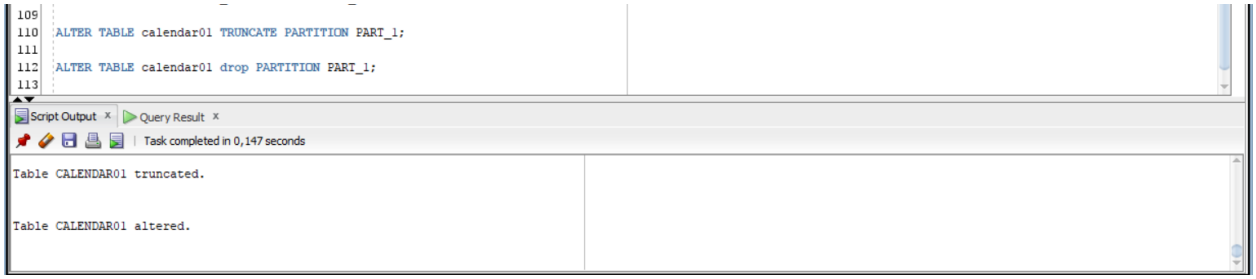


Picture 8 - Inserion and Merging

Picture 9 - Moving Partition

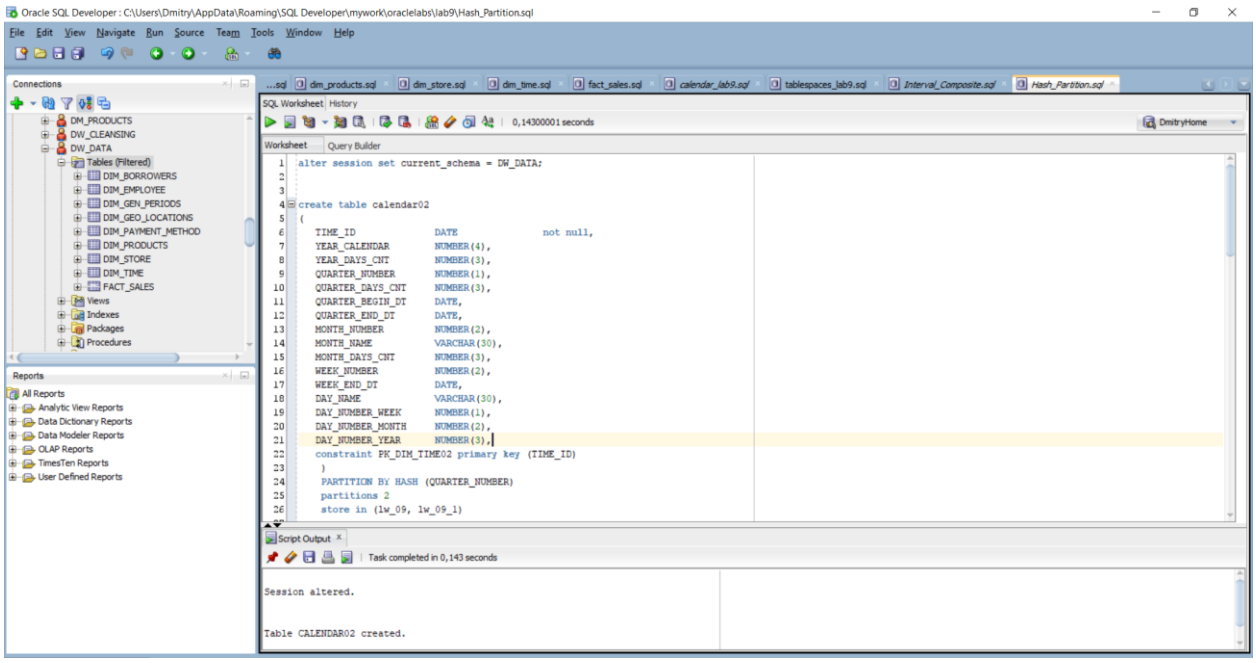


Picture 10 - Splitting Partition



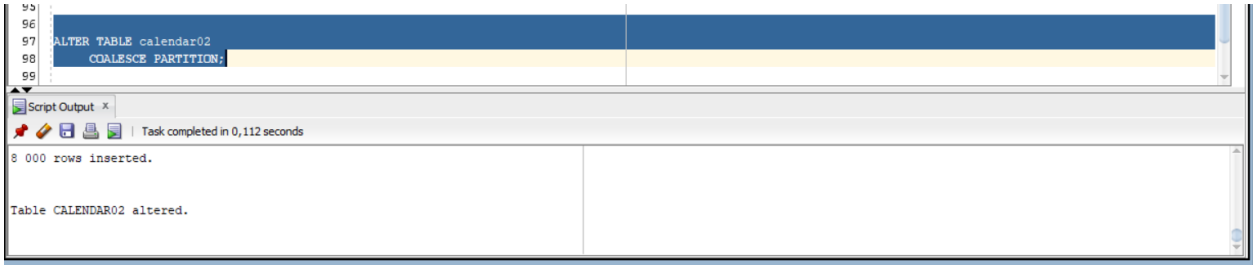
Picture 11 - Dropping And Truncating Partition

Hash



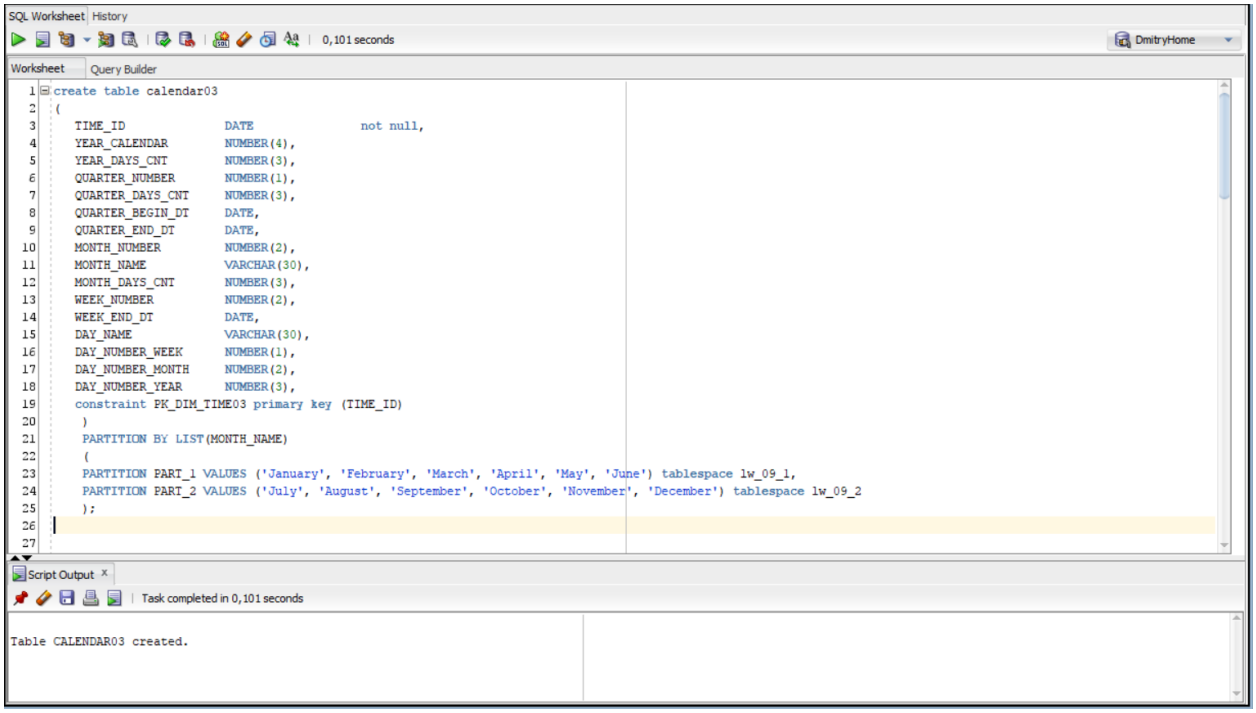
Picture 12 – Creating Partition

Picture 13 - Insering Table

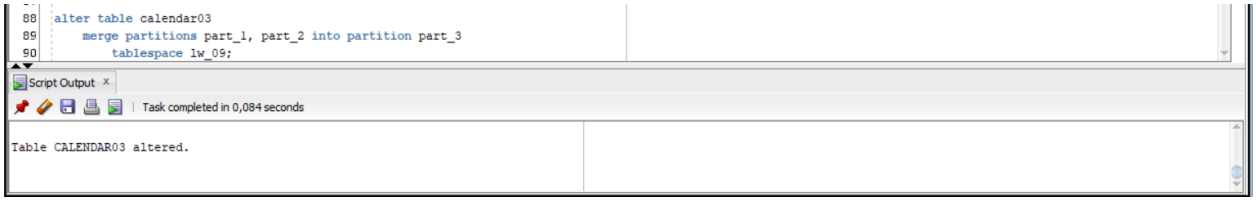


Picture 14 - Coalescing Partition

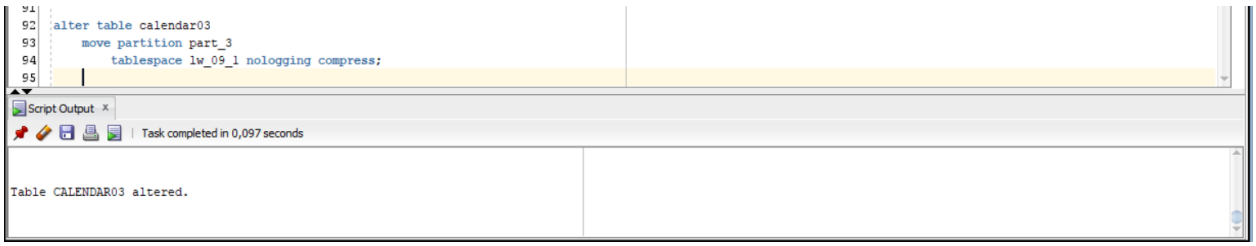
List Composite



Picture 15 - Creating Partition



Picture 16 - Merging Partition



Picture 17 - Moving Partition

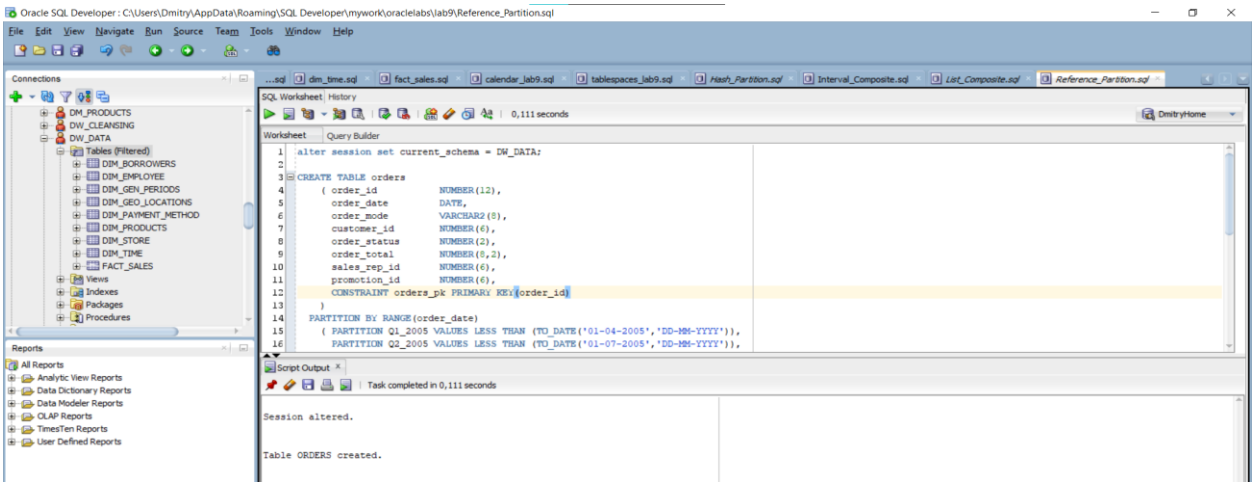
Picture 18 - Splitting Partition

```
alter table calendar03
    truncate partition part_1;

alter table calendar03
    drop partition part_1;
```

```
Table CALENDAR03 truncated.


Table CALENDAR03 altered.
```

Picture 19 - Truncating and Dropping Partition

Reference



Picture 20 - Creating Partition

```
CREATE TABLE order_items
    ( order_id          NUMBER(12) NOT NULL,
      line_item_id      NUMBER(3)  NOT NULL,
      product_id        NUMBER(6)  NOT NULL,
      unit_price        NUMBER(8,2),
      quantity          NUMBER(8),
      CONSTRAINT order_items_fk
      FOREIGN KEY(order_id) REFERENCES orders(order_id)
    )
    PARTITION BY REFERENCE(order_items_fk);
```

```
Session altered.


Table ORDERS created.


Table ORDER_ITEMS created.
```

Picture 21 - Creating Referencing Table

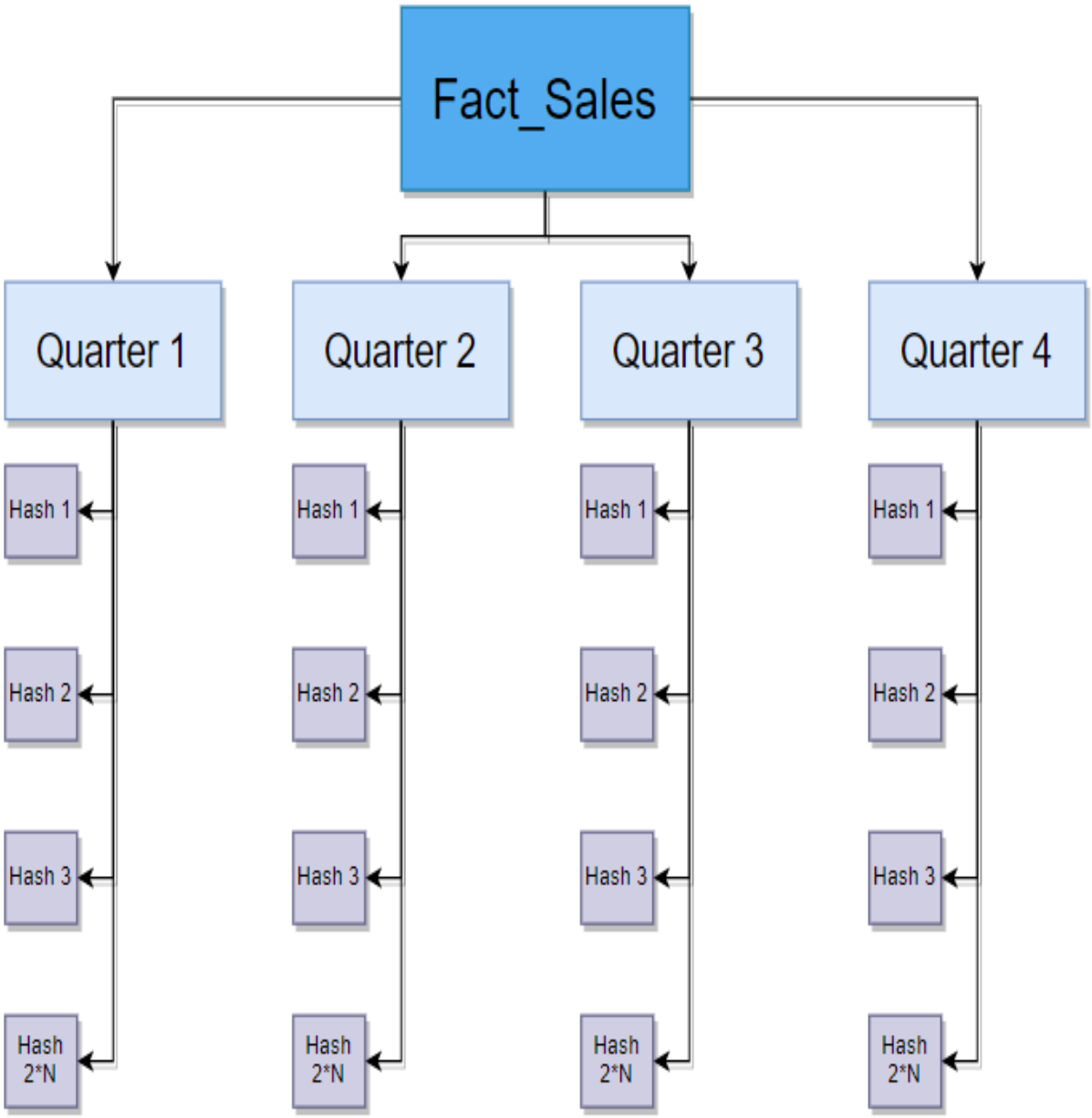
```
alter table order_items
    move partition q1_2005
        tablespace lw_09;

alter table order_items
    truncate partition q1_2005;
```

```
Table ORDER_ITEMS altered.


Table ORDER_ITEMS truncated.
```

Picture 22 - Moving and Truncating Partition

3. Business Task - Partitioning Facts

3.1. Partitioning Facts



Picture 23 - Partitioning Fact Table

```
PARTITION BY RANGE (TIME_ID) INTERVAL (NUMTODSINTERVAL(1,'DAY'))
subpartition by hash(CUSTOMER_ID) subpartitions 4
(
    PARTITION QUARTER_1 VALUES LESS THAN('04')
    (
      subpartition QUARTER_1_sub_1,
      subpartition QUARTER_1_sub_2,
      subpartition QUARTER_1_sub_3,
      subpartition QUARTER_1_sub_4
    ),
    PARTITION QUARTER_2 VALUES LESS THAN('07')
    (
      subpartition QUARTER_2_sub_1,
      subpartition QUARTER_2_sub_2,
      subpartition QUARTER_2_sub_3,
      subpartition QUARTER_2_sub_4
     ),
     PARTITION QUARTER_3 VALUES LESS THAN('10')
    (
       subpartition QUARTER_3_sub_1,
      subpartition QUARTER_3_sub_2,
      subpartition QUARTER_3_sub_3,
      subpartition QUARTER_3_sub_4
    ),
     PARTITION QUARTER_4 VALUES LESS THAN('13')
    (
      subpartition QUARTER_4_sub_1,
      subpartition QUARTER_4_sub_2,
      subpartition QUARTER_4_sub_3,
```

Picture 24 - Example Partitioning Fact Table

Laboratory work summary:

We can **improve query performance** using partitioning by Quarters**.** We need to divide partitions into sub – partitions (4 cause of amount of s – p needed to be in a degree of 2).

We touched principals of Partitions creation. We use several of them to practice. Also described some benefits of using range and hash partitioning e.g. for our business task. All diagrams and scripts are stored in GitHub (link in README file in Labs folder)