Report

Laboratory Work 1

Dmitry Ladutsko

August 02, 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

Data Warehouse Architecture – Storage Layers 2.1. Task 01: CREATE Storage Objects

The Main Task is to create Physical Objects according yours Solution Proposal that was developed on Module 6 – Oracle DB. Introduction to DWH.

Now I created DDL's for all tables, except of fact table, of course:)

```
Worksheet Query Builder
 1 drop table sa_time;
 3 alter session set current_schema=SA_DATE;
 5 Create table sa_time (
 6 date_id
                   DATE,
                DATE ,
 7 beg_of_year
  beg_of_year
end_of_year
                              5 = create table SA CUSTOMERS (
CHAR (20),
                                                    CHAR (20),
                              8 | COUNTRY CITY CUSTOMER
                                                    CHAR (20),
                             9 ADRESS_CUSTOMER
                                                    CHAR (50),
                                                    CHAR (30),
                                                    NUMBER,
NUMBER.
                                                    VARCHAR2 (6),
                                  CUSTOMER SALE DATE
                                                    DATE
22 );
```

Picture 1 - Creating calendar and customer tables

```
5 CREATE TABLE SA EMPLOYEES (
    first name EMPLOYEE VARCHAR2 (40) NOT NULL,
 6
     last name EMPLOYEE VARCHAR2 (40) NOT NULL,
                                              1 DROP TABLE SA PAYMENT METHODS;
                        VARCHAR2 (40) NOT NULL,
 8
    email EMPLOYEE
                     VARCHAR2 (40) NOT NULL,
9
    phone EMPLOYEE
                                             3 alter session set current schema = SA CUSTOMERS;
    POSITION DEGREE VARCHAR2 (40) NOT NULL,
    POSITION NAME VARCHAR2 (40) NOT NULL,
10
                                             4
11
                                            5 CREATE TABLE SA PAYMENT METHODS (
12
      SALES TYPE VARCHAR2 (40) NOT NULL,
                                             6 PAYMENT_METHOD_NAME VARCHAR2(40) NOT NULL,
13
      HIRE DATE DATE NOT NULL
                                              7
                                                     BANK NAME VARCHAR2 (40) NOT NULL
14 );
                                              8 );
```

Picture 2 - Creating employees and payment methods tables

```
drop table SA_PRODUCTS;
2
3
   alter session set current schema = SA PRODUCTS;
4
5 create table SA_PRODUCTS (
                                                  1 DROP TABLE SA STORES;
6
     PRODUCT NAME VARCHAR2 (30),
                                                  2
7
     MODEL NAME
                        VARCHAR2 (20),
                                                 3 alter session set current schema = SA DATE;
     MEMORY_AMOUNT NUMBER(22,0),
COLOR_NAME VARCHAR2(25),
8
                                                 4
9
                                                 5 CREATE TABLE SA STORES (
      PRICE
10
                         NUMBER (10),
                                                  6 country_city VARCHAR2(20) NOT NULL,
                                                                   VARCHAR2 (20) NOT NULL,
     INSERT_DT
                       DATE,
11
                                                  7
                                                       address
12
     UPDATE DT
                        DATE
                                                  8
                                                        phone
                                                                   NUMBER NOT NULL
13 );
                                                  9 );
```

Picture 3 - Creating products and stores tables

```
-----PAYMENT METHOD
                                                               33
       create table sa_transactions(
                                                                       PAYMENT_METHOD_NAME VARCHAR2 (40) NOT NULL,
                                                               34
14
         --CUSTOMER
      FIRST_NAME_CUSTOMER CHAR(20),
LAST_NAME_CUSTOMER CHAR(20),
COUNTRY_CITY_CUSTOMER CHAR(20),
ADRESS_CUSTOMER CHAR(50),
                                                               35
                                                                      BANK_NAME VARCHAR2 (40) NOT NULL,
15
                                                                                ----PRODUCT
                                                               36
16
                                                              37
                                                              37
38 MODEL_NAME VARCHARZ (22,0),
39 MEMORY_AMOUNT NUMBER (22,0),
COLOR_NAME VARCHAR2 (25),
NUMBER (10),
                                                                        PRODUCT NAME
                                                                                                 VARCHAR2 (30),
17
18
19
       EMAIL
                                       CHAR (30),
       PHONE_CUSTOMER
20
21
       AGE
                                       NUMBER,
                                                                     INSERT_DT
UPDATE_DT
                                                              42
                                       VARCHAR2 (6),
                                                                                             DATE.
22
       IS ACTIVE
                                   DATE,
                                                              43
23
      CUSTOMER_SALE_DATE
                                                                                             DATE.
24
                ---EMPLOYEE
                                                               44
                                                                               ----CALENDAR
       first_name_EMPLOYEE VARCHAR2(40) NOT NULL, 45 date_id
                                                                                                     DATE,
                                                             46 beg_of_year
47 end_of_year
26
                                                                                              DATE
27
                                                                                                VARCHAR2 (44)
VARCHAR2 (1)
28
                                                       49 day_number_in_week
50 day_number_in_month
51 day_number_in_year
52 calendar week number
29
                                                                                                 VARCHAR2 (2)
30
        POSITION DEGREE VARCHAR2 (40) NOT NULL,
        SALES_TYPE VARCHAR2 (40) NOT NULL,
HIRE_DATE DATE NOT NULL
31
                                                                                                    VARCHAR2 (3)
                                                                                                    VARCHAR2(1)
```

Picture 4 – DML

```
alter session set current_schema = SA_CUSTOMERS;
63
        INSERT INTO sa_transactions
65
       (
        SELECT /*+ parallel(SA_CUSTOMERS.sa_CUSTOMERS, 4)*/ *
66
67
           SA_CUSTOMERS.sa_CUSTOMERS
       CROSS JOIN (SELECT * FROM SA_CUSTOMERS.sa_customers
69
          WHERE PHONE CUSTOMER > 9900000)
       CROSS JOIN (SELECT * FROM SA_PRODUCTS.sa_products)
71
       INNER JOIN SA_DATE.sa_time ON date_id = UPDATE_DT
72
       WHERE UPDATE_DT > TO DATE( '01.01.19', 'MM/DD/YY') AND UPDATE_DT < TO DATE( '04.01.22', 'MM/DD/YY')
73
74
```

Picture 5 - Cross Joins on transaction with other tables

2.2. Task 02: Generate Test Data in Storage Layers

The Main Task is to generate test data on Storage layers objects, that was created on task 01.

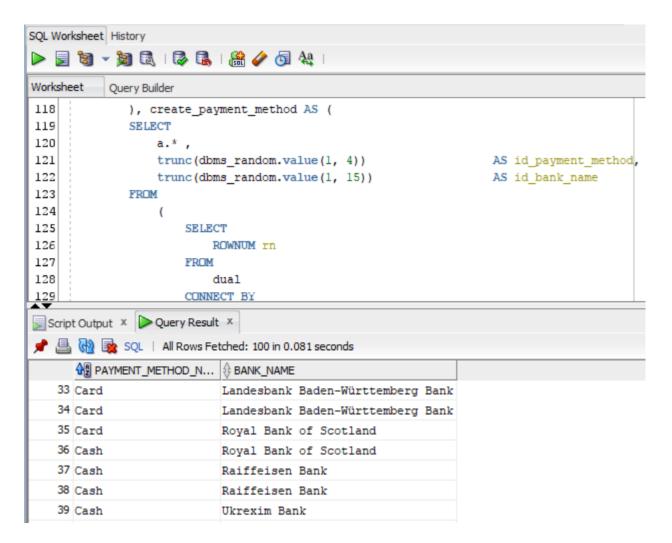
Now I decided to generate more values data, which is also going to be more relevant, apart from previous DML scripts... (All DDL/DML Scripts stored on GitHub)

```
SQL Worksheet History
Worksheet
         Query Builder
                 a.* ,
 361
 362
                 trunc (dbms random.value(1, 11))
                                                               AS id first name,
 363
                 trunc(dbms_random.value(1, 11))
                                                               AS id_last_name,
                trunc (dbms_random.value(1, 11))
 364
                                                               AS id country,
 365
                trunc (dbms random.value(1, 11))
                                                               AS id address,
 366
                trunc (dbms_random.value(1, 11))
                                                               AS id email,
                trunc (dbms_random.value(1000001, 9999999))
                                                              AS phone_number,
 367
 368
                trunc(dbms_random.value(16, 65))
                                                               AS age,
                --????????? trunc(dbms_random.value(1, 1))
                                                                          AS valid from date
 369
                                                              AS id_active,
370
                trunc(dbms_random.value(1, 4))
371
                trunc(dbms_random.value(1, 1))
                                                              AS id_s_date
372
                 , trunc(dbms_random.value(1, 3))
                                                                AS id_customer_payment_method*/
373
             FROM
374
Script Output X Query Result X
🎤 🥔 🔒 📕 | Task completed in 0.269 seconds
Session altered.
6,000 rows inserted.
```

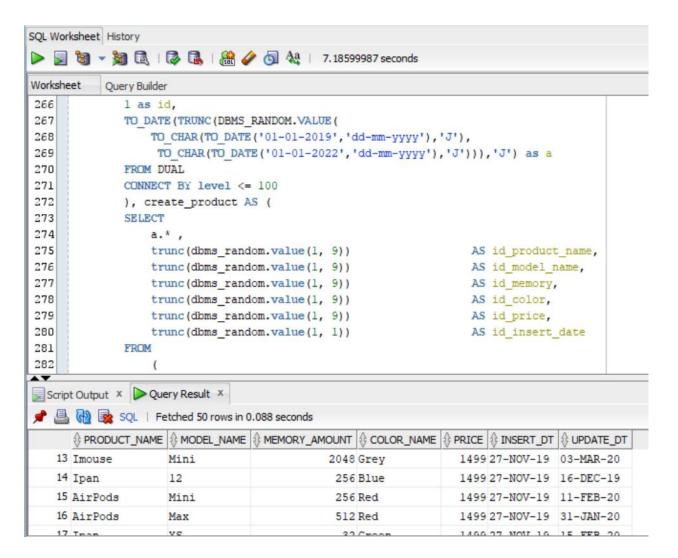
Picture 6 - Customers DML

```
SQL Worksheet History
Worksheet
          Query Builder
413
             ), create_employee AS (
414
             SELECT
415
                a.* ,
                                                            AS id_first_name,
                trunc(dbms_random.value(1, 21))
416
                trunc(dbms_random.value(1, 14))
417
                                                             AS id_last_name,
418
                trunc(dbms random.value(1, 21))
                                                             AS id email,
419
                trunc(dbms_random.value(1000001, 9999999))
                                                             AS phone_number,
420
                trunc(dbms random.value(1, 5))
                                                              AS id position,
421
                trunc(dbms random.value(1, 5))
                                                             AS id degree,
422
                trunc(dbms_random.value(1, 5))
                                                              AS id_type
423
            FROM
424
425
                    SELECT
426
                        ROWNUM rn
Script Output X Query Result X
📌 🤌 🔡 🖺 🔋 | Task completed in 0.361 seconds
Session altered.
1,500 rows inserted.
```

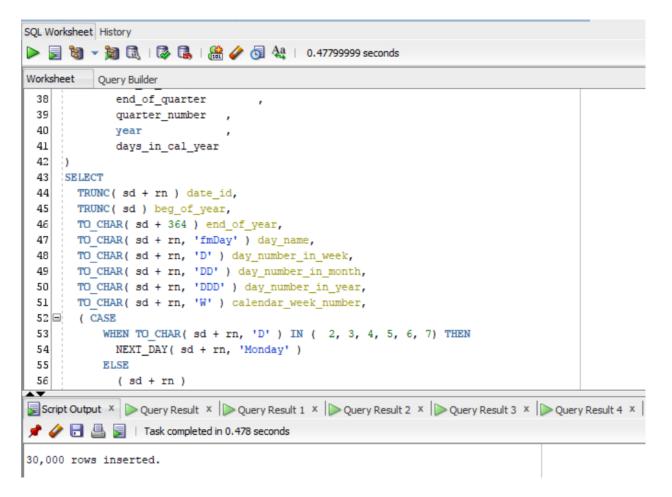
Picture 7 - Employees DML



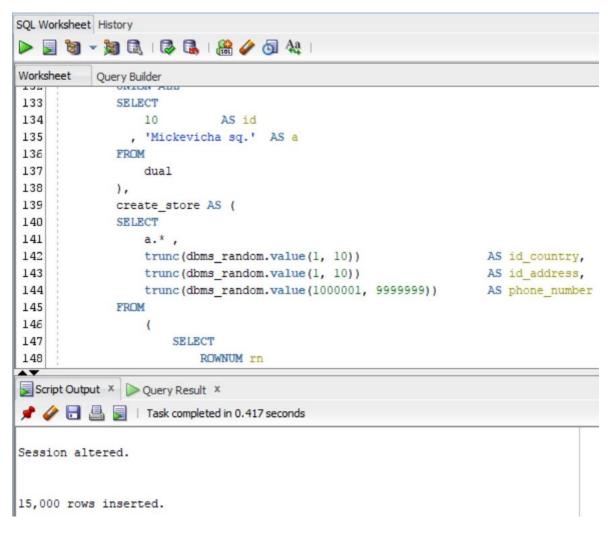
Picture 8 - Payment Method DML



Picture 9 - Produsts DML



Picture 10 - Calendar DML



Picture 11 - Store DML

```
63
         alter session set current_schema = SA_CUSTOMERS;
 64 🖃
         INSERT INTO sa_transactions
 65
 66
          SELECT /*+ parallel(SA CUSTOMERS.sa CUSTOMERS, 4)*/ *
 67
          FROM
            SA_CUSTOMERS.sa_CUSTOMERS
 68
          CROSS JOIN (SELECT * FROM SA_CUSTOMERS.sa_customers
 70
           WHERE PHONE_CUSTOMER > 9900000) -- AND FIRST_NAME_CUSTOMER = 'Dmitry')
          CROSS JOIN (SELECT * FROM SA PRODUCTS.sa products)
 71
          INNER JOIN SA_DATE.sa_time ON date_id = UPDATE_DT
         WHERE UPDATE_DT > TO DATE( '01.01.19', 'MM/DD/YY') AND UPDATE_DT < TO DATE( '04.01.22', 'MM/DD/YY')
 73
 74
          );
 75
Script Output X
📌 🧼 🖥 📕 📗 Task completed in 130.742 seconds
Error report -
ORA-01653: unable to extend table SA_CUSTOMERS.SA_TRANSACTIONS by 2048 in tablespace TS_SA_CUSTOMERS_DATA_001
```

Picture 12 - Transaction DML

Note: Unluckily, at the moment of DML exec it did not executed cause of lack of server local memory amount. But the script is working correctly.

It Is going to show who and what bought in specific store in which date as you can see.

Laboratory work summary:

At this lab we have learned how (and which opportunities) gives as Oracle in generating Data for test e.g. It is absolutely clear that generating so much data is not needed every time we have to test SA layer(but also can use back – end p.l. to generate them easily). Nevertheless, this type is also quite acceptable if we have to test smth not to use some Back techs. All diagrams and scripts are stored in GitHub (link in README file in Labs folder)