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

<u>The Main Task</u> 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

```
drop table sa_time;

alter session set current_schema=SA_DATE;

Create table sa_time (
date_id                     DATE,
beg_of_year             DATE        ,
end_of_year             DATE        ,
day_name                    VARCHAR2(44) ,
day_number_in_week          VARCHAR2(1)  ,
day_number_in_month         VARCHAR2(2)  ,
day_number_in_year          VARCHAR2(3)  ,
calendar_week_number        VARCHAR2(1)  ,
week_ending_date            DATE        ,
calendar_month_number       VARCHAR2(2)  ,
days_in_month           VARCHAR2(2)  ,
end_of_month            DATE        ,
end_of_quarter          VARCHAR2(32) ,
quarter_number          VARCHAR2(1)  ,
year                VARCHAR2(4)  ,
days_in_cal_year            NUMBER
);
```

```
create table SA_CUSTOMERS(
    FIRST_NAME_CUSTOMER         CHAR(20),
    LAST_NAME_CUSTOMER          CHAR(20),
    COUNTRY_CITY_CUSTOMER       CHAR(20),
    ADRESS_CUSTOMER             CHAR(50),
    EMAIL                       CHAR(30),
    PHONE_CUSTOMER              NUMBER,
    AGE                         NUMBER,
    IS_ACTIVE                   VARCHAR2(6),
    CUSTOMER_SALE_DATE          DATE
    );
```

Picture 1 - Creating calendar and customer tables

```
CREATE TABLE SA_EMPLOYEES (
    first_name_EMPLOYEE     VARCHAR2(40) NOT NULL,
    last_name_EMPLOYEE      VARCHAR2(40) NOT NULL,
    email_EMPLOYEE          VARCHAR2(40) NOT NULL,
    phone_EMPLOYEE          VARCHAR2(40) NOT NULL,
    POSITION_NAME   VARCHAR2(40) NOT NULL,
    POSITION_DEGREE   VARCHAR2(40) NOT NULL,
    SALES_TYPE      VARCHAR2(40) NOT NULL,
    HIRE_DATE      DATE NOT NULL
);
```

```
DROP TABLE SA_PAYMENT_METHODS;

alter session set current_schema = SA_CUSTOMERS;

CREATE TABLE SA_PAYMENT_METHODS(
    PAYMENT_METHOD_NAME VARCHAR2(40) NOT NULL,
    BANK_NAME    VARCHAR2(40) NOT NULL
);
```

Picture 2 - Creating employees and payment methods tables

```
drop table SA_PRODUCTS;

alter session set current_schema = SA_PRODUCTS;

create table SA_PRODUCTS (
    PRODUCT_NAME        VARCHAR2(30),
    MODEL_NAME          VARCHAR2(20),
    MEMORY_AMOUNT       NUMBER(22,0),
    COLOR_NAME          VARCHAR2(25),
    PRICE               NUMBER(10),
    INSERT_DT           DATE,
    UPDATE_DT           DATE
);
```

```
DROP TABLE SA_STORES;

alter session set current_schema = SA_DATE;

CREATE TABLE SA_STORES (
    country_city  VARCHAR2(20) NOT NULL,
    address       VARCHAR2(20) NOT NULL,
    phone         NUMBER NOT NULL
);
```

Picture 3 – Creating products and stores tables

```
13  create table sa_transactions(
14  --CUSTOMER
15  FIRST_NAME_CUSTOMER          CHAR(20),
16  LAST_NAME_CUSTOMER           CHAR(20),
17  COUNTRY_CITY_CUSTOMER        CHAR(20),
18  ADRESS_CUSTOMER              CHAR(50),
19  EMAIL                        CHAR(30),
20  PHONE_CUSTOMER               NUMBER,
21  AGE                          NUMBER,
22  IS_ACTIVE                    VARCHAR2(6),
23  CUSTOMER_SALE_DATE           DATE,
24  ----------EMPLOYEE
25  first_name_EMPLOYEE      VARCHAR2(40) NOT NULL,
26  last_name_EMPLOYEE       VARCHAR2(40) NOT NULL,
27  email_EMPLOYEE           VARCHAR2(40) NOT NULL,
28  phone_EMPLOYEE           VARCHAR2(60) NOT NULL,
29  POSITION_NAME   VARCHAR2(40) NOT NULL,
30  POSITION_DEGREE   VARCHAR2(40) NOT NULL,
31  SALES_TYPE      VARCHAR2(40) NOT NULL,
32  HIRE_DATE      DATE NOT NULL
```

```
33  ----------PAYMENT METHOD
34  PAYMENT_METHOD_NAME VARCHAR2(40) NOT NULL,
35  BANK_NAME   VARCHAR2(40) NOT NULL,
36  -----------PRODUCT
37  PRODUCT_NAME         VARCHAR2(30),
38  MODEL_NAME           VARCHAR2(20),
39  MEMORY_AMOUNT        NUMBER(22,0),
40  COLOR_NAME           VARCHAR2(25),
41  PRICE                NUMBER(10),
42  INSERT_DT            DATE,
43  UPDATE_DT            DATE,
44  ----------CALENDAR
45  date_id                      DATE,
46  beg_of_year              DATE        ,
47  end_of_year              DATE        ,
48  day_name                     VARCHAR2(44) ,
49  day_number_in_week           VARCHAR2(1)  ,
50  day_number_in_month          VARCHAR2(2)  ,
51  day_number_in_year           VARCHAR2(3)  ,
52  calendar week number         VARCHAR2(1)  ,
```

Picture 4 – 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
68      SA_CUSTOMERS.sa_CUSTOMERS
69   CROSS JOIN (SELECT * FROM SA_CUSTOMERS.sa_customers
70      WHERE PHONE_CUSTOMER > 9900000)
71   CROSS JOIN (SELECT * FROM SA_PRODUCTS.sa_products)
72   INNER JOIN SA_DATE.sa_time ON date_id = UPDATE_DT
73  WHERE UPDATE_DT > TO_DATE( '01.01.19', 'MM/DD/YY' ) AND UPDATE_DT < TO_DATE( '04.01.22', 'MM/DD/YY' )
74  );
```

Picture 5 - Cross Joins on transaction with other tables

2.2. Task 02: Generate Test Data in Storage Layers

<u>The Main Task</u> 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

0.26899999 seconds

Worksheet Query Builder

```
361            a.* ,
362            trunc(dbms_random.value(1, 11))                AS id_first_name,
363            trunc(dbms_random.value(1, 11))                AS id_last_name,
364            trunc(dbms_random.value(1, 11))                AS id_country,
365            trunc(dbms_random.value(1, 11))                AS id_address,
366            trunc(dbms_random.value(1, 11))                AS id_email,
367            trunc(dbms_random.value(1000001, 9999999))     AS phone_number,
368            trunc(dbms_random.value(16, 65))               AS age,
369           --????????? trunc(dbms_random.value(1, 1))                AS valid_from_date
370            trunc(dbms_random.value(1, 4))                 AS id_active,
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 Query Result

Task completed in 0.269 seconds

```
Session altered.

6,000 rows inserted.
```

Picture 6 – Customers DML

SQL Worksheet History

0.361 seconds

Worksheet Query Builder

```
        ), create_employee AS (
        SELECT
            a.* ,
            trunc(dbms_random.value(1, 21))                AS id_first_name,
            trunc(dbms_random.value(1, 14))                AS id_last_name,
            trunc(dbms_random.value(1, 21))                AS id_email,
            trunc(dbms_random.value(1000001, 9999999))     AS phone_number,
            trunc(dbms_random.value(1, 5))                 AS id_position,
            trunc(dbms_random.value(1, 5))                 AS id_degree,
            trunc(dbms_random.value(1, 5))                 AS id_type
        FROM
            (
                SELECT
                    ROWNUM rn
```

Script Output Query Result

Task completed in 0.361 seconds

```
Session altered.

1,500 rows inserted.
```

Picture 7 - Employees DML

SQL Worksheet | History

Worksheet | Query Builder

```
118         ), create_payment_method AS (
119         SELECT
120             a.* ,
121             trunc(dbms_random.value(1, 4))                  AS id_payment_method,
122             trunc(dbms_random.value(1, 15))                 AS id_bank_name
123         FROM
124             (
125                 SELECT
126                     ROWNUM rn
127                 FROM
128                     dual
129                 CONNECT BY
```

Script Output | Query Result

SQL | All Rows Fetched: 100 in 0.081 seconds

	PAYMENT_METHOD_N...	BANK_NAME
33	Card	Landesbank Baden-Württemberg Bank
34	Card	Landesbank Baden-Württemberg Bank
35	Card	Royal Bank of Scotland
36	Cash	Royal Bank of Scotland
37	Cash	Raiffeisen Bank
38	Cash	Raiffeisen Bank
39	Cash	Ukrexim Bank

Picture 8 - Payment Method DML

SQL Worksheet | History

7.18599987 seconds

Worksheet | Query Builder

```
        1 as id,
        TO_DATE(TRUNC(DBMS_RANDOM.VALUE(
            TO_CHAR(TO_DATE('01-01-2019','dd-mm-yyyy'),'J'),
             TO_CHAR(TO_DATE('01-01-2022','dd-mm-yyyy'),'J'))),'J') as a
        FROM DUAL
        CONNECT BY level <= 100
        ), create_product AS (
        SELECT
            a.* ,
            trunc(dbms_random.value(1, 9))                AS id_product_name,
            trunc(dbms_random.value(1, 9))                AS id_model_name,
            trunc(dbms_random.value(1, 9))                AS id_memory,
            trunc(dbms_random.value(1, 9))                AS id_color,
            trunc(dbms_random.value(1, 9))                AS id_price,
            trunc(dbms_random.value(1, 1))                AS id_insert_date
        FROM
            (
```

Script Output | Query Result

SQL | Fetched 50 rows in 0.088 seconds

	PRODUCT_NAME	MODEL_NAME	MEMORY_AMOUNT	COLOR_NAME	PRICE	INSERT_DT	UPDATE_DT
13	Imouse	Mini	2048	Grey	1499	27-NOV-19	03-MAR-20
14	Ipan	12	256	Blue	1499	27-NOV-19	16-DEC-19
15	AirPods	Mini	256	Red	1499	27-NOV-19	11-FEB-20
16	AirPods	Max	512	Red	1499	27-NOV-19	31-JAN-20
17	Ipan	XS	32	Green	1499	27-NOV-19	15-FEB-20

Picture 9 - Produsts DML

SQL Worksheet History

0.47799999 seconds

Worksheet Query Builder

```
        end_of_quarter        ,
        quarter_number    ,
        year              ,
        days_in_cal_year
)
SELECT
  TRUNC( sd + rn ) date_id,
  TRUNC( sd ) beg_of_year,
  TO_CHAR( sd + 364 ) end_of_year,
  TO_CHAR( sd + rn, 'fmDay' ) day_name,
  TO_CHAR( sd + rn, 'D' ) day_number_in_week,
  TO_CHAR( sd + rn, 'DD' ) day_number_in_month,
  TO_CHAR( sd + rn, 'DDD' ) day_number_in_year,
  TO_CHAR( sd + rn, 'W' ) calendar_week_number,
  ( CASE
      WHEN TO_CHAR( sd + rn, 'D' ) IN (  2, 3, 4, 5, 6, 7) THEN
        NEXT_DAY( sd + rn, 'Monday' )
      ELSE
        ( sd + rn )
```

Script Output | Query Result | Query Result 1 | Query Result 2 | Query Result 3 | Query Result 4

Task completed in 0.478 seconds

```
30,000 rows inserted.
```

Picture 10 - Calendar DML

```
SELECT
    10          AS id
  , 'Mickevicha sq.'  AS a
FROM
    dual
),
create_store AS (
SELECT
    a.* ,
    trunc(dbms_random.value(1, 10))                  AS id_country,
    trunc(dbms_random.value(1, 10))                  AS id_address,
    trunc(dbms_random.value(1000001, 9999999))       AS phone_number
FROM
    (
        SELECT
            ROWNUM rn
```

```
Session altered.

15,000 rows inserted.
```

Picture 11 - Store DML

```
alter session set current_schema = SA_CUSTOMERS;
INSERT INTO sa_transactions
(
 SELECT /*+ parallel(SA_CUSTOMERS.sa_CUSTOMERS, 4)*/ *
 FROM
    SA_CUSTOMERS.sa_CUSTOMERS
 CROSS JOIN (SELECT * FROM SA_CUSTOMERS.sa_customers
    WHERE PHONE_CUSTOMER > 9900000)-- AND FIRST_NAME_CUSTOMER = 'Dmitry')
 CROSS JOIN (SELECT * FROM SA_PRODUCTS.sa_products)
 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' )
);
```

```
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)