Report

Laboratory Work 4

Dmitry Ladutsko

August 13, 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. Business analyses tasks – Reports

2.1. Task 01: Create Packages for Reload Dimension from SA\_\*

**The Main Task** is to independent packages to reload dimension according your DWH solution concept which was developed on Module 6. Introduction to DWH.

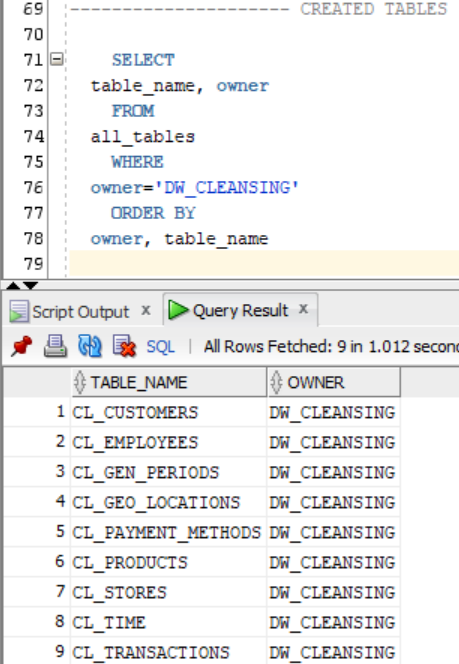
**Required points:**

* Create all required Dim objects on DW Layer
* Grant all required Privileges to DW\_CL (Cleansing Layer)
* Create Packages to reload Dim data (one package = one dimension.)
* Example future SAL.DIM\_GEO\_SCD will store all procedure on pkg\_etl\_dim\_geo\_dw. But this package will store all small sub dims – T\_COUNTRIES, T\_REGIONS etc. )
* Use Explicit Cursor (One package)
* Use Explicit Cursor and FORALL Bulk Insertion (One package)
* Use Variable Cursor and FORALL Bulk Insertion (One package)
* Use Merge (One packages)

**Task Results:**

Create required objects:

* Put objects script to Git.
* Prepare Document with Screenshot of Data on Dimensions
* Test data for consistent
* Test Procedure for Repeatable execution (Nothing should change)

** Firstly, I created** tables on Cleansing level

SELECT

table\_name, owner

FROM

all\_tables

WHERE

owner='DW\_CLEANSING'

ORDER BY

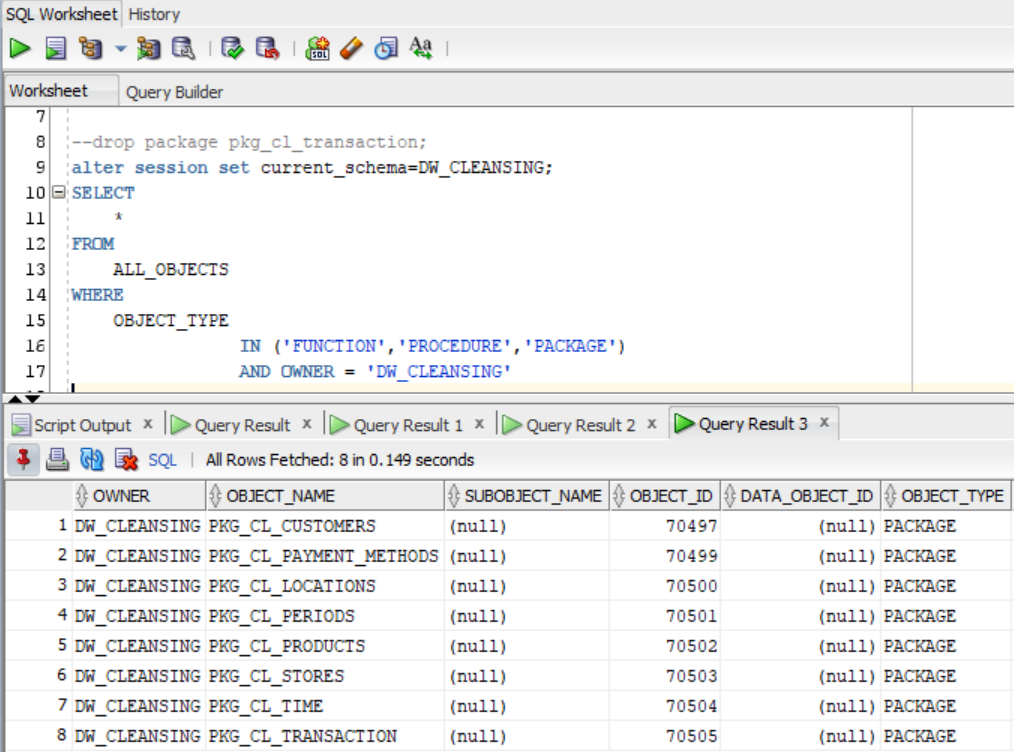
owner, table\_name

* CL\_CUSTOMERS
* CL\_EMPLOYEES
* CL\_GEN\_PERIODS
* CL\_GEO\_LOCATIONS
* CL\_PAYMENT\_METHODS
* CL\_PRODUCTS
* CL\_STORES
* CL\_TIME
* CL\_TRANSACTIONS

***Note.*** All scripts stored in GitHub, exit task folder. But also duplicated in lab4 folder AS CL folder.

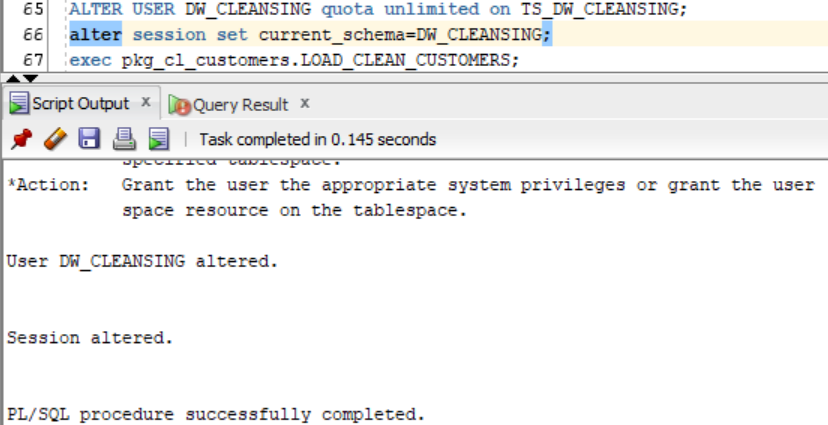
Then I began **creating packages**, which consist of 2 files :

* \*table\_name\*\_define.sql
* \*table\_name\*\_body.sql

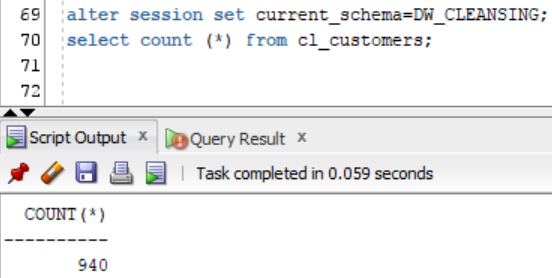
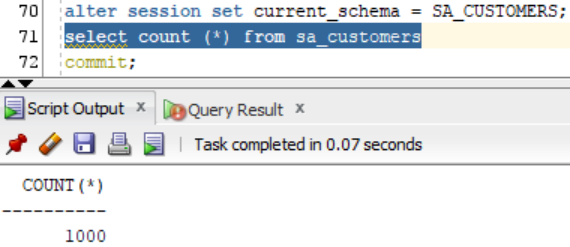


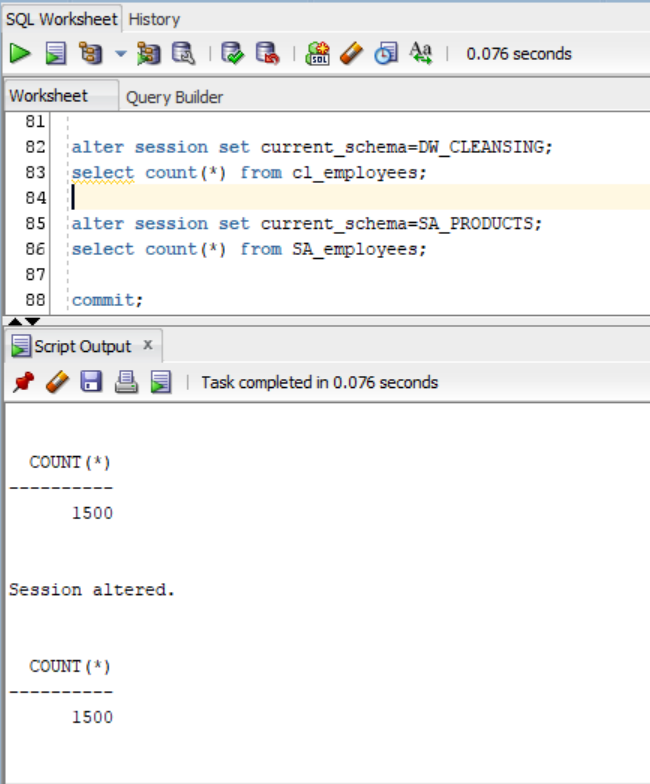
***Note.*** Later I will show the **Data Flow Diagram** to explain how I store files / sub – folders / folders

Now let’s try to create package bodies. And of course execute them to move data from SA level to CL level

****

**\*After almost an hour** I finally understood that it is a great idea to grant my user more space resource :)



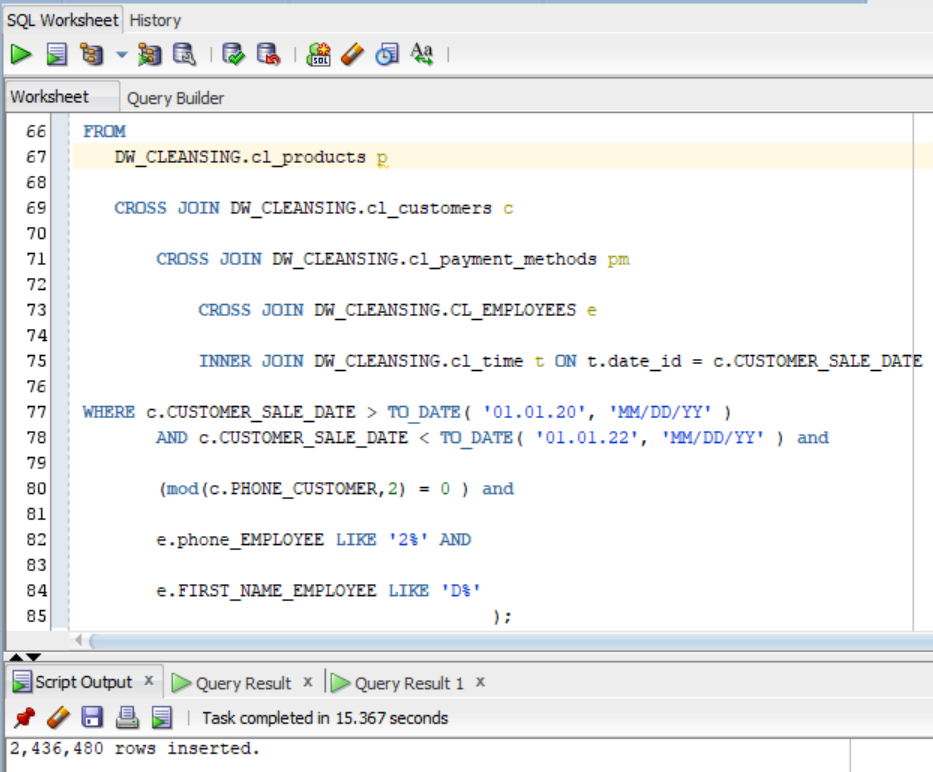


As you can see, if we **count** rows from the same tables (but on different layers(different tablespaces)) thee number of rows differs because in \*.SA table were some rows with NULL values.

***Note.*** Next I am going to make a body statements for every table (except of some( 1 – 3 ), which are not needed to be cleansed)

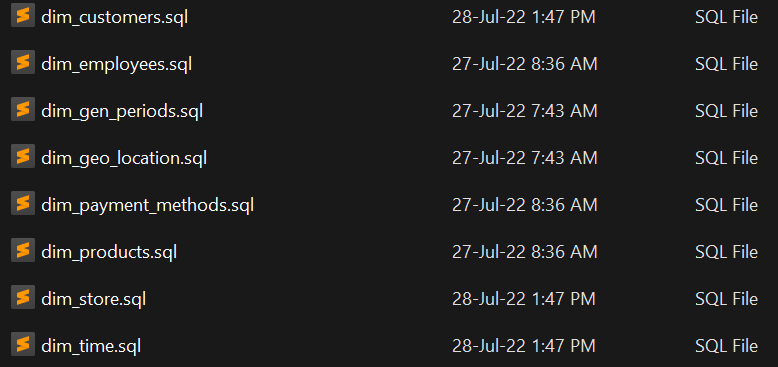
***Note.*** However, employees table **was not cleansed** because of they were created using ***NOT NULL* constraint**. It seems to be useless cleansing such tables but if someone will change this table structure **in future** and try to use *null* values in some rows, then **it will be worthy**.

***Note.*** I coded every package and executed their bodies before moving data from sa\_\* tables to cl\_\* tables, and generating transactions table on cleansing layer.

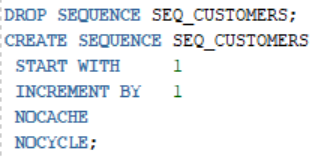


***Note.*** Finally, I got ~2.5 m rows inserted into cl.transaction table fulfilled with fully cleansed data (of course with some artificial constraints)!

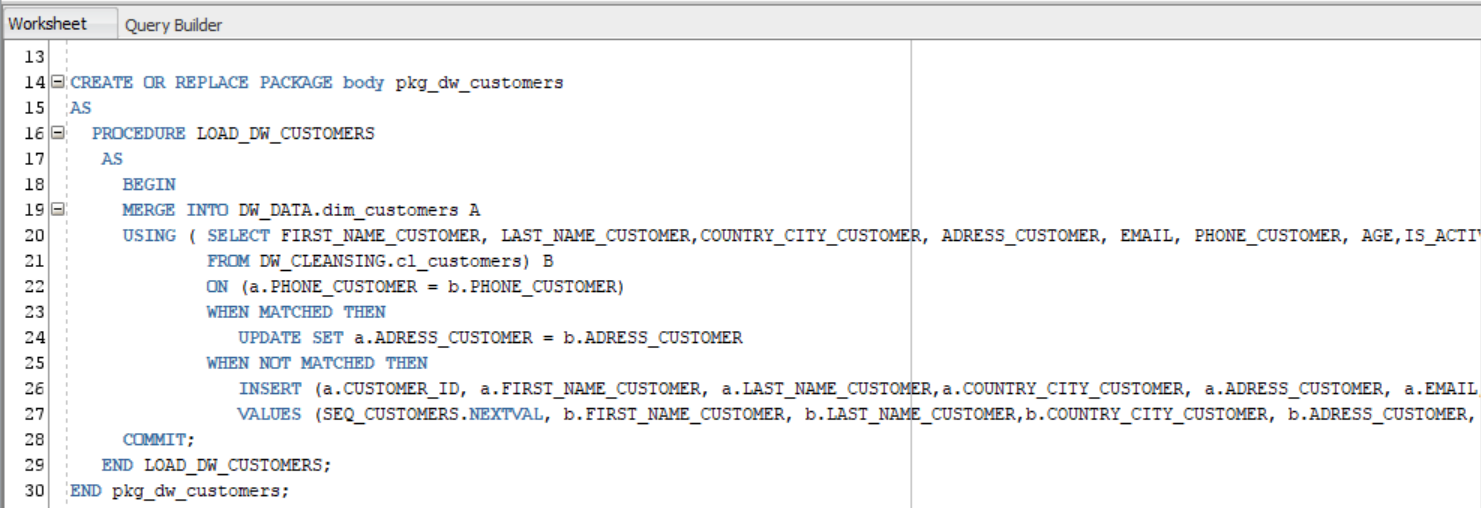
***Note.*** In fact, we need to add only decision – making data. I thought that I could throw away some columns (e.g. customers/employees emails, phones etc.), but then I realised I could use it to, for example, select customers emails to target marketing content. So, anyway if I decide it is not needed, I will alter this tables.

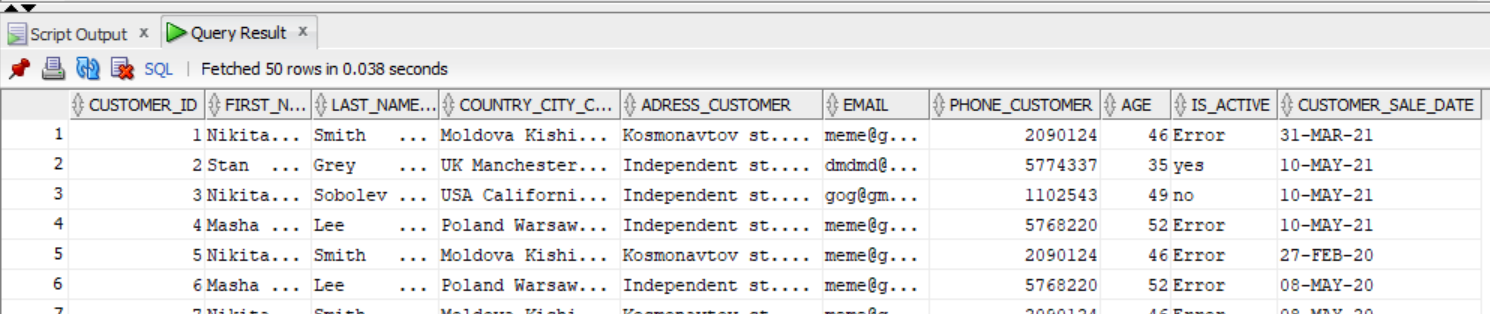


***Note.*** We have created dimension tables before. Copy of dimensions creation scripts duplicated in lab4 folder.

***Note.*** I created such sequences to auto increment primary key id’s. Then again added packages using following structure :

* \*table\_name\*\_define.sql
* \*table\_name\*\_body.sql

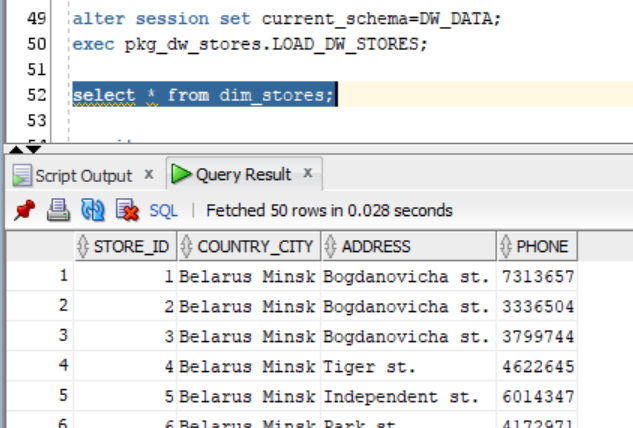


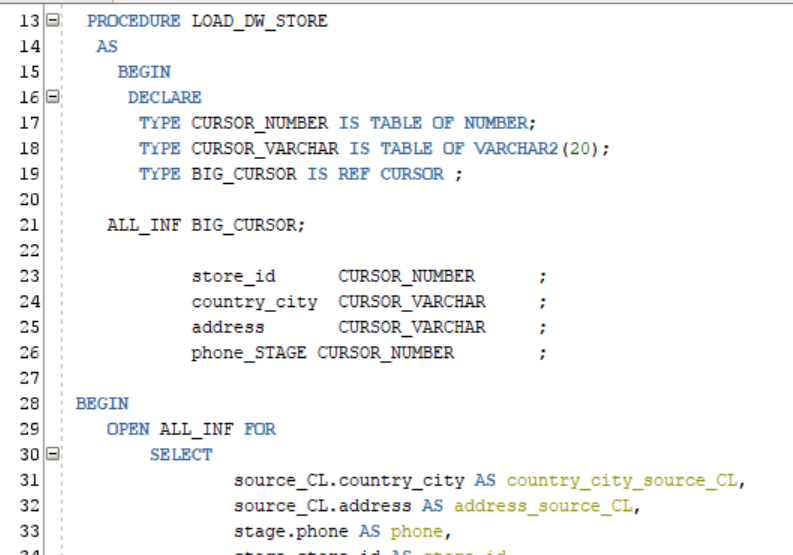
****

I used **Use Explicit Cursor** in pkg\_cl\_employees\_body.sql

**Merge** in pkg\_dw\_employees\_body.sql

**Explicit Cursor and FOR ALL BLUNK Insertion** in pkg\_dw\_stores\_body.sql

****

****

**Laboratory Work Summary:** At this laboratory work we created Cleansing and Data warehouse layers, created tables for them and practiced how using different methods we can move data from one layer to another. I used following methods for data movements between different layers:

* Explicit Cursor
* Explicit Cursor and FORALL Bulk Insertion
* Variable Cursor and FORALL Bulk Insertion
* Merge