Report

Laboratory Work 10

Dmitry Ladutsko

August 18, 2022

# 

# 1. Prerequisites Task Information

## 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. ETL Transformation - BASIC

## 2.1. Task 01: Transformation Description

**The Main Task** is to create chapter in the Solution Concept Document that will explain Transformation strategy for your Business Task, according Exit Task for Module 6 – Introduction to DWH.

### **SQL Transformation**

The SQL transformation processes SQL queries midstream in a pipeline. The SQL transformation can be an active or passive transformation. You can insert, delete, update, and retrieve rows from a database. You can pass the database connection information to the SQL transformation as input data at run time. The transformation processes external SQL scripts or SQL queries that you create in an SQL editor. The SQL transformation processes the query and returns rows and database errors.

### **PL/SQL Transforming**

In a data warehouse environment, you can use procedural languages such as PL/SQL to implement complex transformations in the Oracle Database. Whereas CTAS operates on entire tables and emphasizes parallelism, PL/SQL provides a row-based approached and can accommodate very sophisticated transformation rules.

For example, a PL/SQL procedure could open multiple cursors and read data from multiple source tables, combine this data using complex business rules, and finally insert the transformed data into one or more target table. It would be difficult or impossible to express the same sequence of operations using standard SQL statements.

### **Transforming Data Using Table Functions**

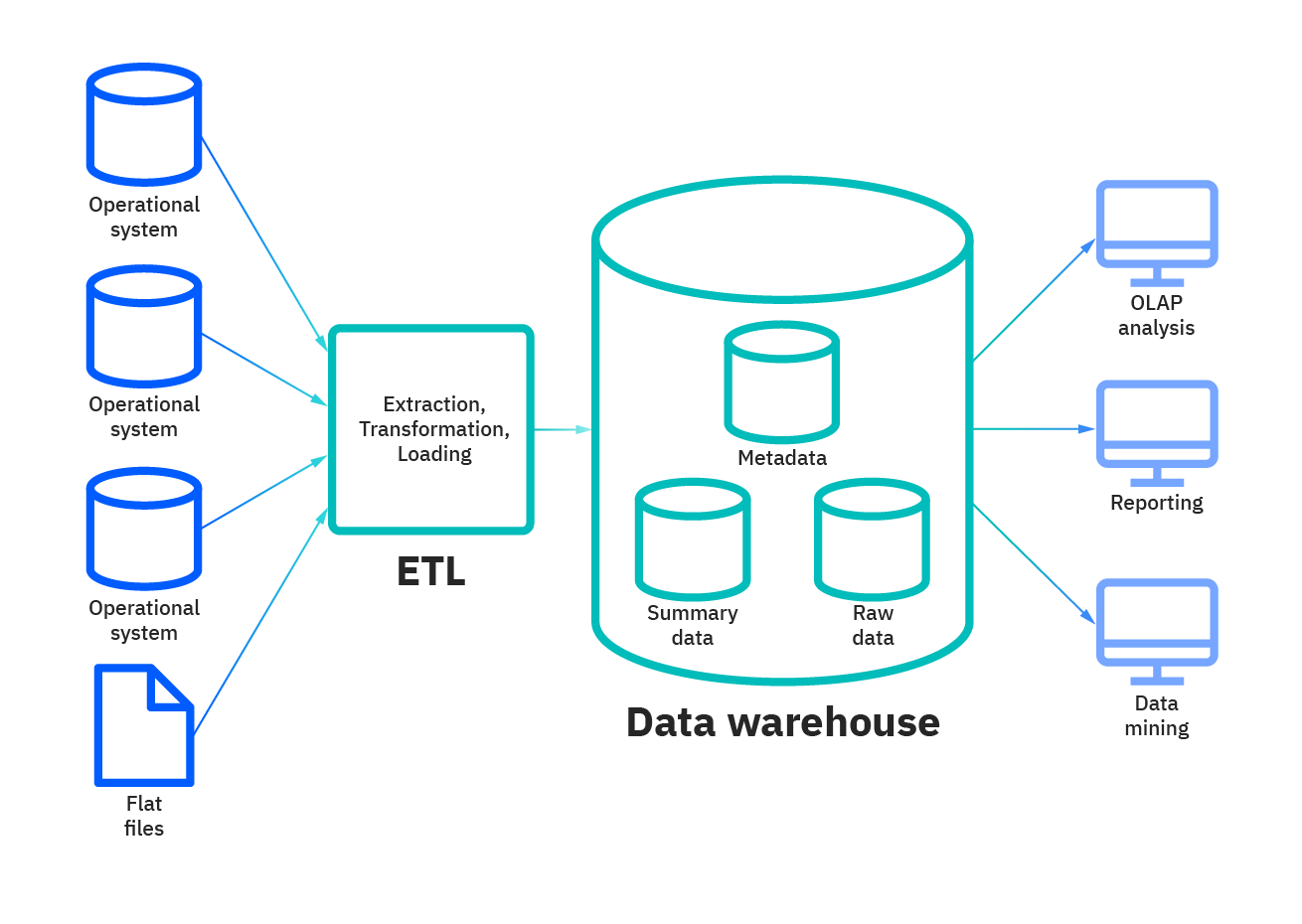
Table functions provide the support for pipelined and parallel execution of transformations implemented in PL/SQL, C, or Java.

Scenarios as mentioned earlier can be done without requiring the use of intermediate staging tables, which interrupt the data flow through various transformations steps.

### **Business solution concept Transformation Strategy**

For my business the right way is to use a combo of Transformation types. Thus different objects need to be transformed in different ways.

Reference tables are not needed to be transformed, for example, using PL/SQL Transformation Strategy as well as Customers and Employees Dimensions and Fact Sales table, for example are needed to be updated or loaded within new data using Multiple Cursors, Functions and Procedures to manipulate with several objects at the same time.



# 3. ETL Transformation – Loading SAL

**Task 01 is common for LabWork 10 (Task 02), 11 (Task 01).**

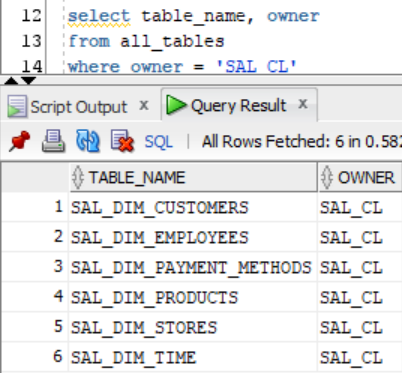
## 3.1. Task 02: Loading to SAL Layer Data

**The Main Task** is to load dimension to SAL layer

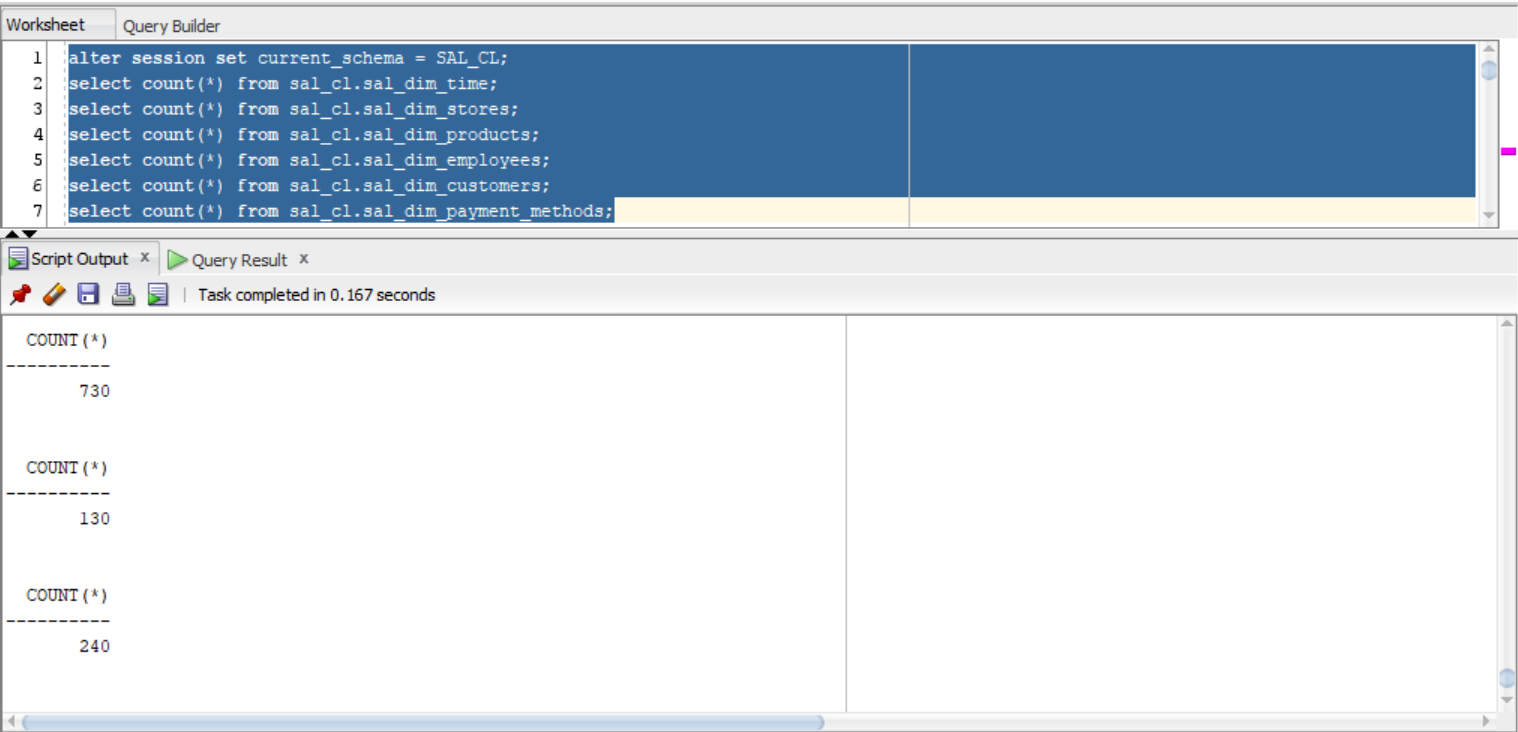
**Required points:**

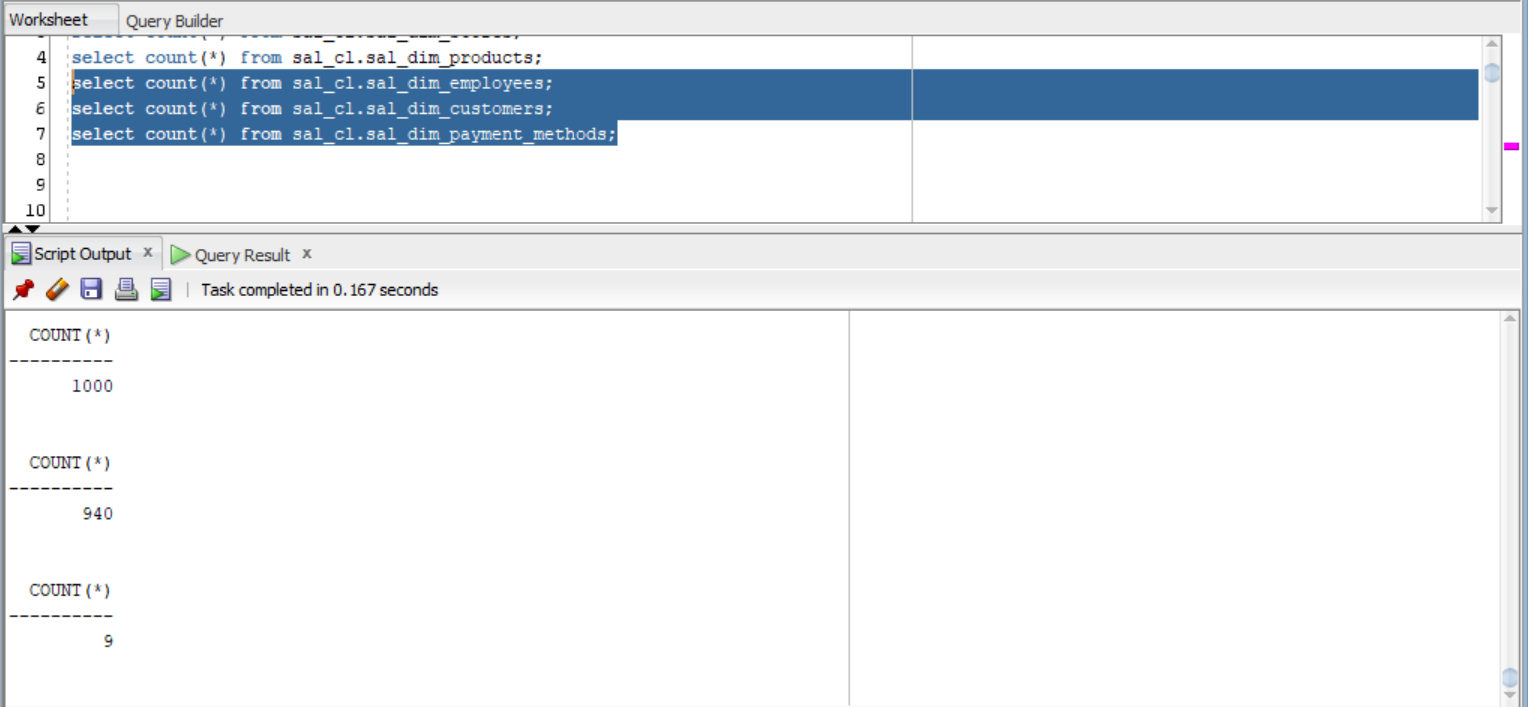
* Create new package for Load FCT\_\* and DIM\_\* to SAL Layer
* Load Dimension
* Load SCD Dimension
* Load FCT\_\*

***Note.*** Let’s create tables on SAL Layer using created before user SAL\_CL (all created tables list ):

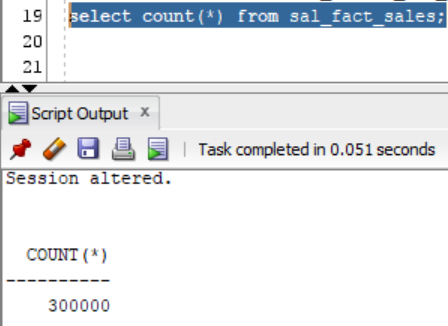


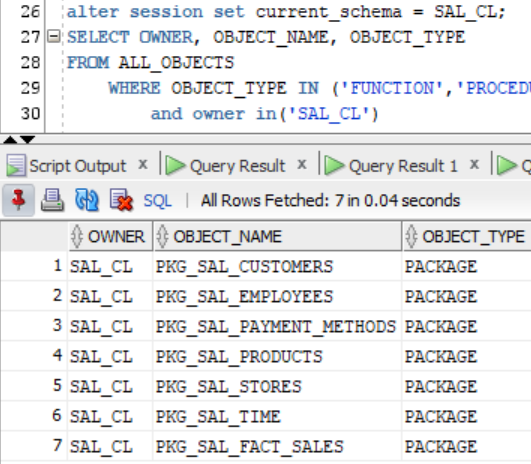
***Note.*** Now we need to move data **from DW layer to SAL layer**. I used procedures with cursor for it. I’ve also created **references** between created tables to later easily **join** them using in **incapsulated into views reports.**

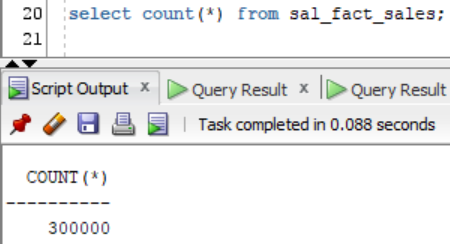


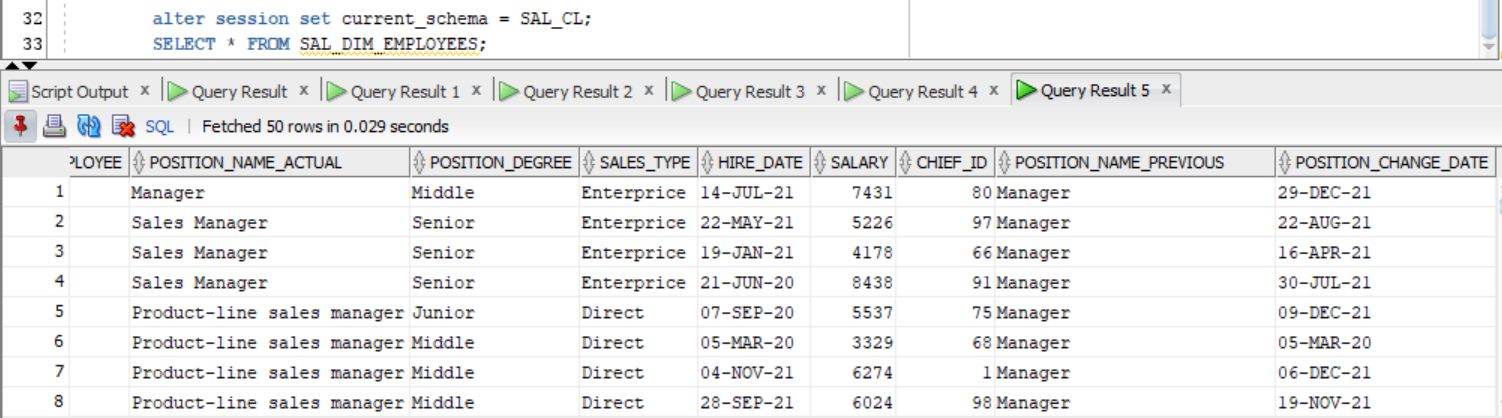


***Note.*** As you can see at the screenshot above, all tables are fulfilled with data from previous layer(DW). So, now we need to make the same step to create **fact table on SAL level** and **fulfill it with data**

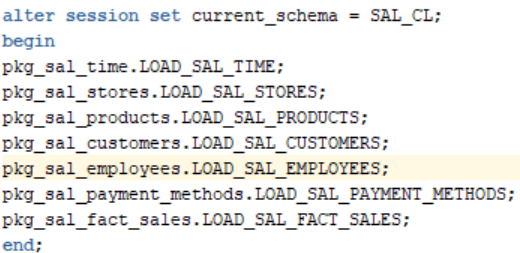








***Note.*** Loaded SCD type 3 table (\*\_employees)



***Note.*** I have summered all packages in one procedure and also putted it on PACKAGES folder.

Also will do the same with other layers load (SA - > CL - > DW - > SAL). File is going to be named as **full load package ( \* ).sql**

***Note.*** (e.g. full load package DW-SAL.sql)

***NOTE.*** New **views with group** statements will be created in **Laboratory Work 11**.

**Laboratory Work Summary**

**At this Laboratory Work** we practiced more about how to create **SAL Layer**, how faster load data from previous layers. How faster we could load it.

**We have learned** more about **Transformation** and it types. Used our gotten knowledge to specify and describe our Business Solution Transformation Strategy.