Report

Laboratory Work 8 - 9

Dmitry Ladutsko

August 16 - 17, 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 Extraction - BASIC

## 2.1. Task 01: Extraction Description

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

**Task Results:**

Create required objects:

* Prepare Document with new Chapter.

***Note.*** From my point of view the **offline extraction** can be assessed as followed: as data is not directly taken from the source, it already has an existing structure, namely it can be tablespaces and logs or in another case the structure may be a result of the extraction method.

Another type of extraction, namely, **full extraction** may be described as complete pulling of data directly from the source. This may be beneficial in a sense that there is no need to track the source system as the extraction process reflects all the available data in the system.

***Note.*** New Chapter “Extraction Description” will be added to Business task solution concept

# 3. ETL Transportation – Example of Loading FCT\_\*

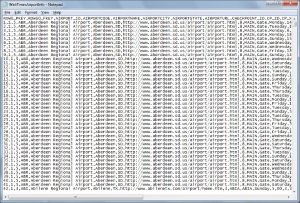
## Task 01: Transportation Description

**Transportation Using Flat Files**

The most common method for transporting data is by the transfer of flat files, using mechanisms such as FTP or other remote file system access protocols. Data is unloaded or exported from the source system into flat files, and is then transported to the target platform using FTP or similar mechanisms.

Source systems and DWH often use different operating systems and database systems, using flat files is often the simplest way to exchange data between heterogeneous systems with minimal transformations. However, even when transporting data between homogeneous systems, flat files are often the most efficient and most easy-to-manage mechanism for data transfer.

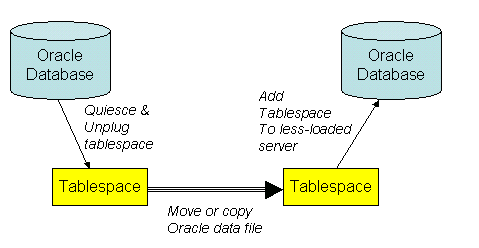
In my business, I guess the best way is working with flat files cause they are easy to generate, use quite a little memory, easy to parse and easy to generate from back – end algorithms.



**Transportation Using Transportable Tablespaces**

Transportable tablespaces is a feature of the Oracle database, introduced with Oracle 8i, that allows DBAs to copy or move tablespaces between databases. It is especially helpful when tablespace size is in huge, in Terrabytes and tradition ways are not as quicker as scp..ing the datafiles. In other words, Transportable Tablespaces allow Oracle data files to be unplugged from a database and copied to another Oracle instance (on another server) and then added into that Oracle instance (refer to Figure below).

The most common applications of transportable tablespaces in data warehouses are in moving data from a staging database to a data warehouse, or in moving data from a data warehouse to a data mart.

****

**Task 02 is common for LabWork 08, 09.**

## Task 02: Prepare Table of Facts to DW Layer

**The Main Task** is to create required objects on DW layers and realize load process for Facts

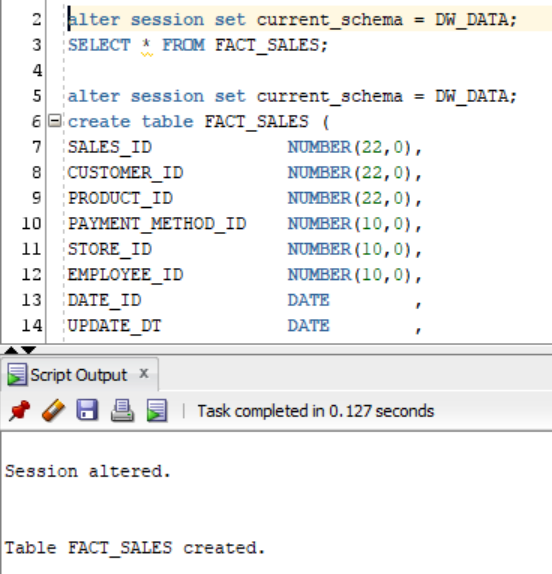
**Required points:**

* Create new package for Load FCT\_\* to DW Layer

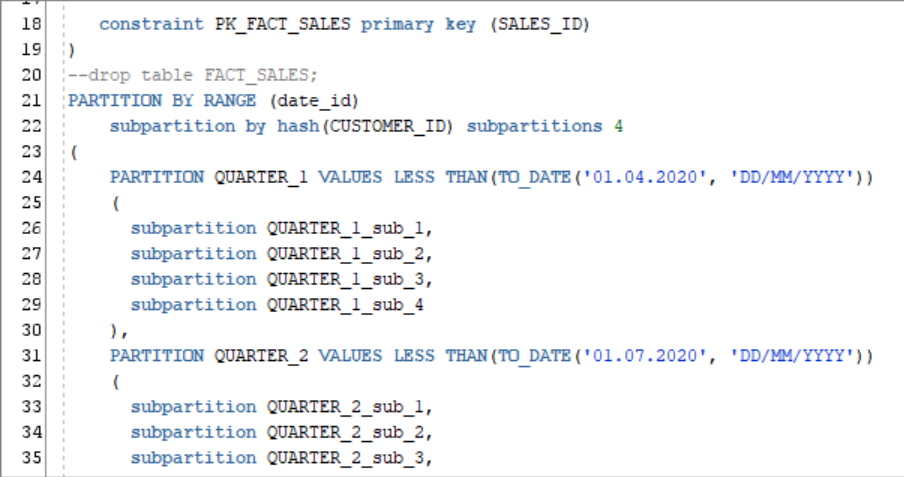
**Task Results:**

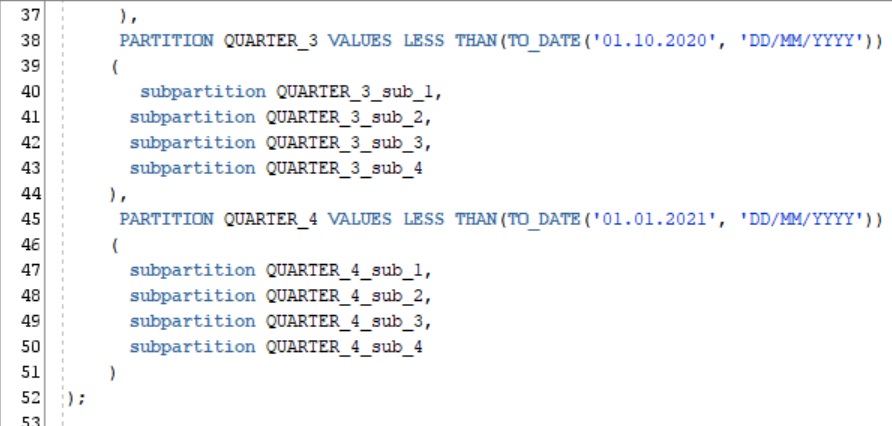
Create required objects:

* Put objects script to Git.
* Prepare Document with Screenshot of Data

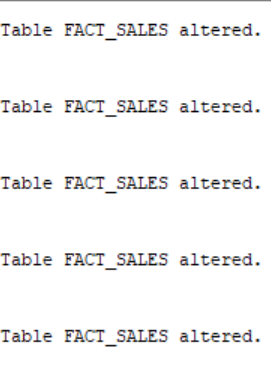
****

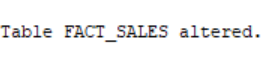
***Note.*** Fact table created

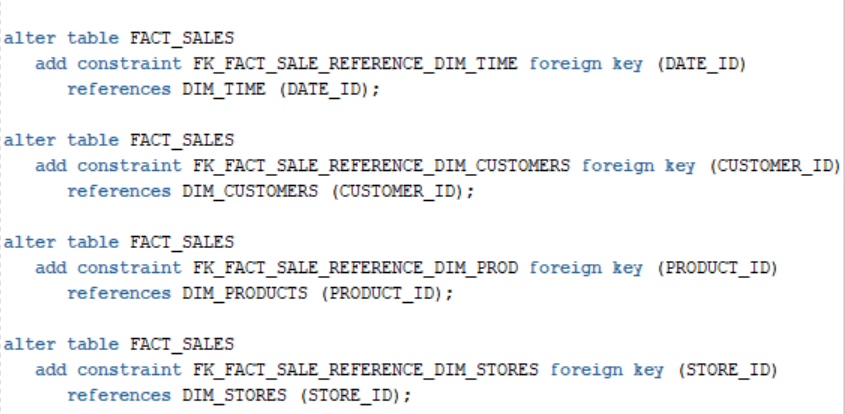


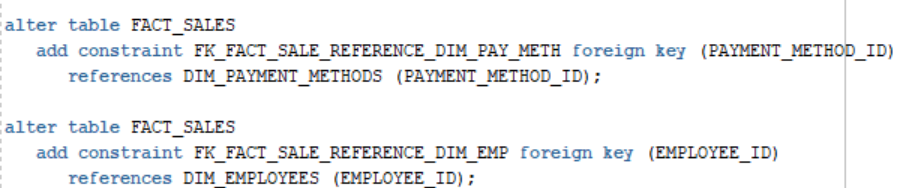
****

***Note.*** It’s Primary key and partitions (with sub – partitions)

****

****

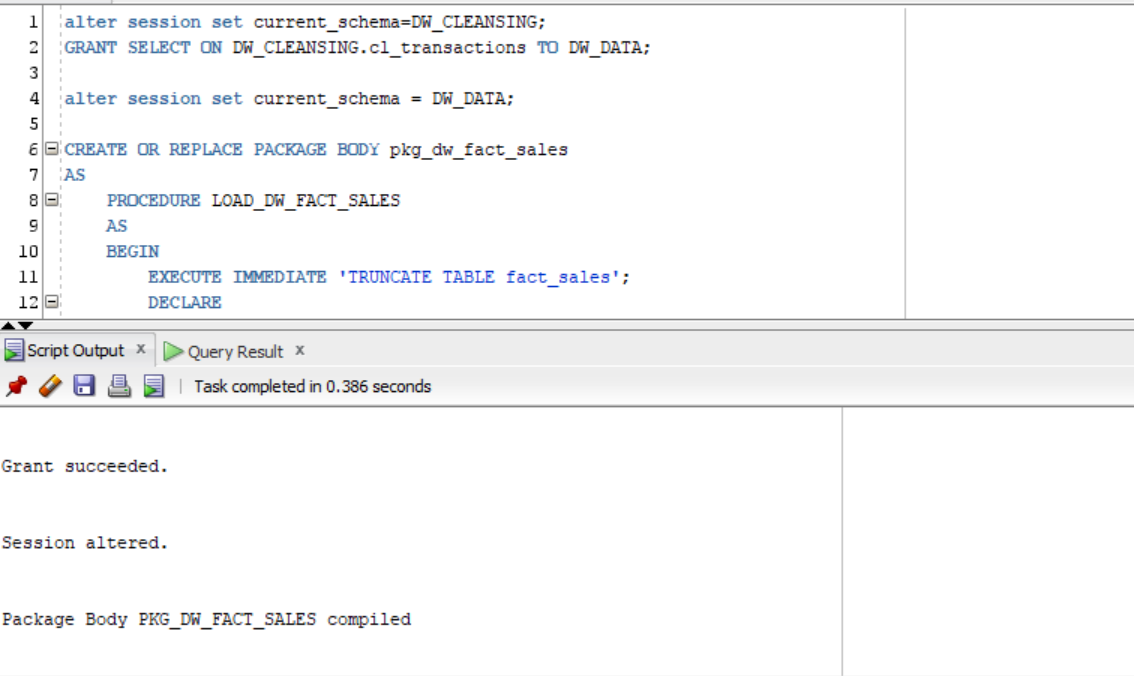
****

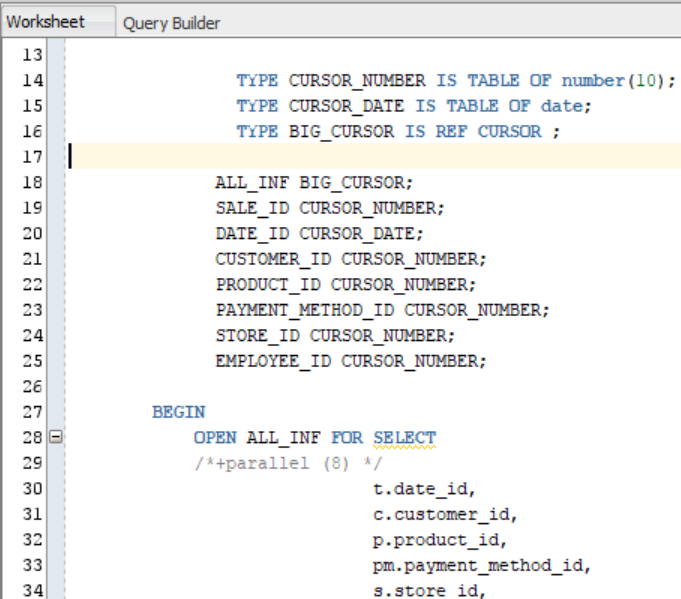
******

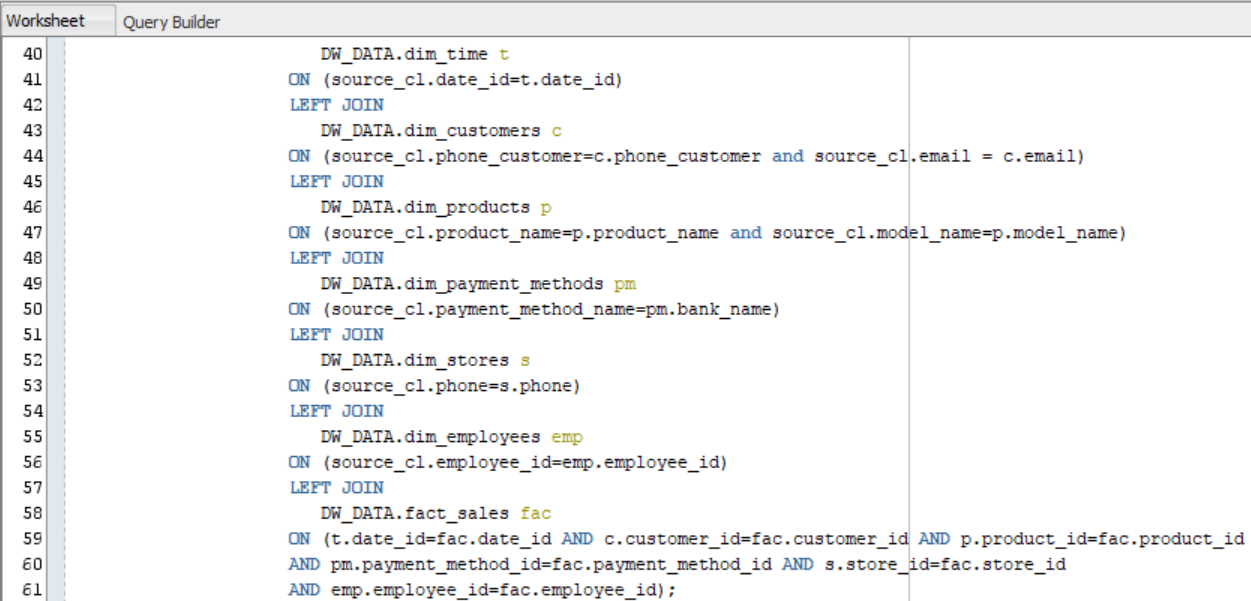
***Note.*** Foreign key constraints. FACT\_SALES TABLE CREATED BEFORE, IN MODULE 1, but now a little bit modified. DDL script added to Laboratory Work 8 – 9 folder and exit task folder.

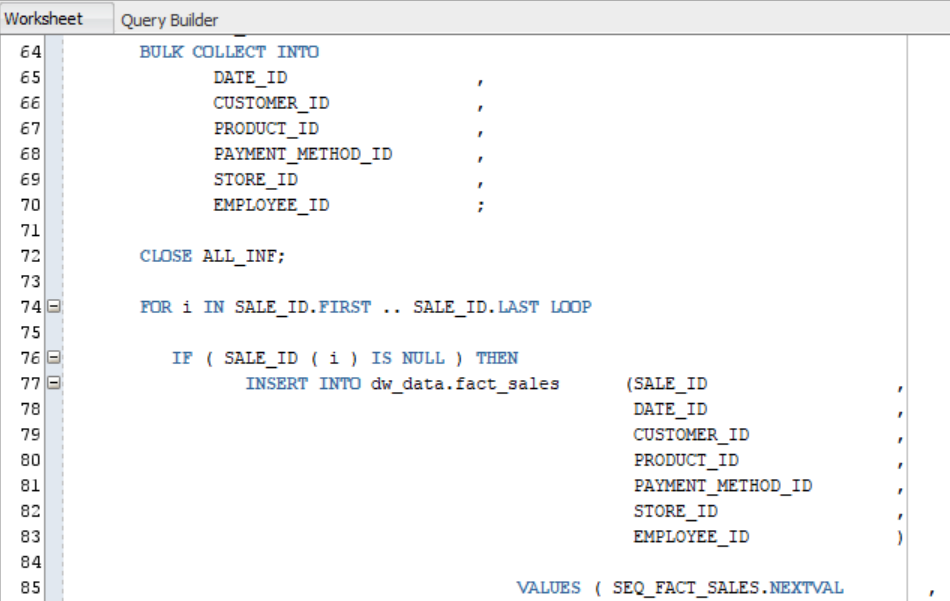
***Note.*** I created DDL script to create **table of facts** as you have seen. Now I created packages to load identificators into Fact Table.

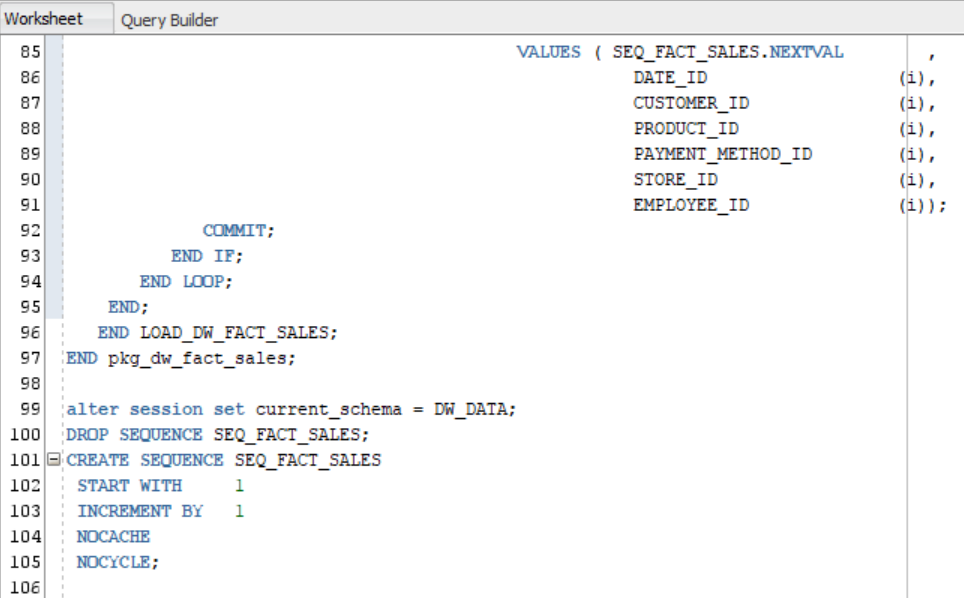
***Note.*** Table of facts may contain other data such as some simple aggregations, but I left it in conservative way, anyway simple aggregations will be easily added later (or use Data Marts for complex reporting). Anyways I will add it :)











Note. I used **Procedure with Cursor** to move id’s.

**Laboratory Work Summary**

**At this Laboratory Work** we practiced more how to create fact table, reference it with dimensions, use partitions and sub – partitions etc.

We learned more about Transportation and Extraction principals and about their types. Learned such types of architectures as Lambda and Kappa One more time used Cursors, so also practised more and more.