**Traditional data-warehouse on loan service risk analysis**

Introduction:

This project is a personal project, it does not contain any real data for the purpose of data privacy. The aim of this project is to stimulate scenarios on traditional data warehouse, to show my understanding in data engineering using oracle, PL/SQL and ETL tools (Kettle).

**When and why do you need to build a data warehouse?**

When there is a request from operation team asking to show specific information display in the system for particular reasons, in this case they want to analysis the risk of lending loans to borrowers by looking at some set KPIs to identify whether it becomes a bad debt.

Quick Brief:

OLAP (Online Analytical Processing) and OLTP (Online Transaction Processing) serve distinct, complementary roles in modern data architectures. OLTP systems are optimized for real-time, high-volume transactional operations like order processing, banking, and inventory management, crucial for day-to-day business functions. OLAP systems, on the other hand, are designed for complex analytical workloads, enabling data aggregation and multi-dimensional queries, crucial for business intelligence (BI) and decision support.

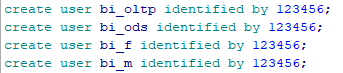
Working process:

Raw data source: see OLTP1.sql and OLTP2.sql

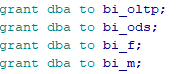
Design stage: see F-mapping.xls and M-mapping.xls

Development stage (main):

To simulate the scenario, we need to build some temporary user accounts in PL/SQL:



And give access:

(can also grant select any table&execute any procedures to users).

After logging into the bi\_oltp account, we can start loading the raw data (imagine we are moving data from one data warehouse to another). :

PL/SQL: tool --import file -- select file destination paths--import

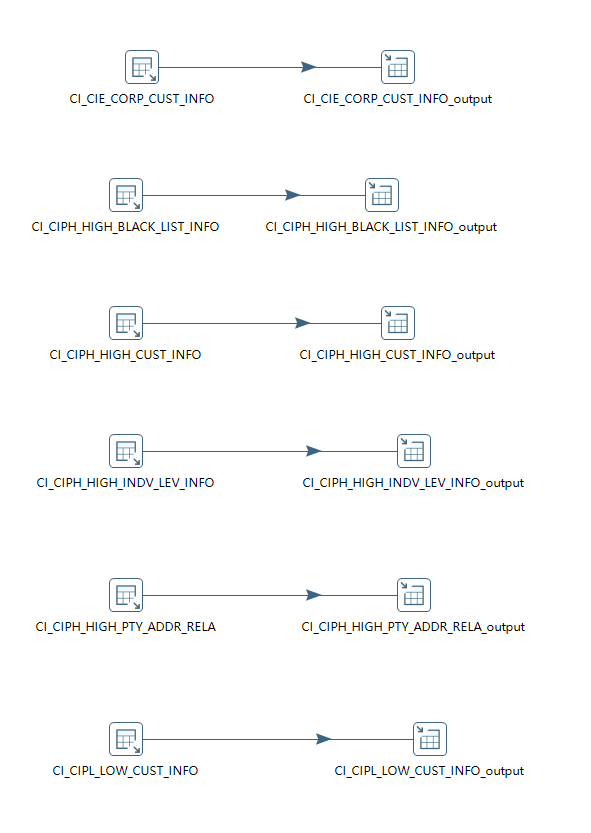
Bear in mind that in every layer (OLTP,ODS,F,M), the table structure needs to be imported into each layer before the ETL process:

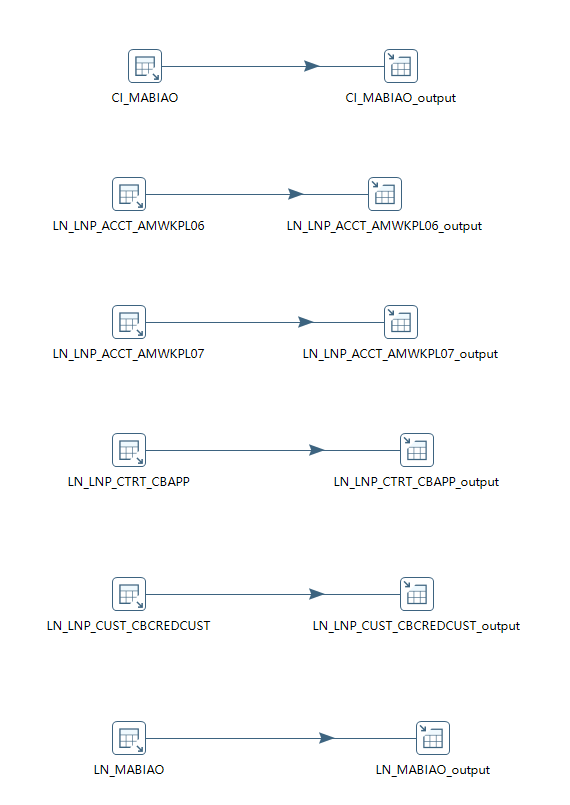
Ods---> ods\_table\_structure file

1. ->F\_customer\_wide\_table & F\_loan\_wide\_table

M---> M\_bad\_loan\_non\_performing\_loan & M\_high\_risk\_customer

In the ODS(Operational Data Store) layer: use **kettle** to ETL relevant data from OLTP layer to ODS (bi\_ods account) check out ‘kettle\_oltp\_dos[2].zip’



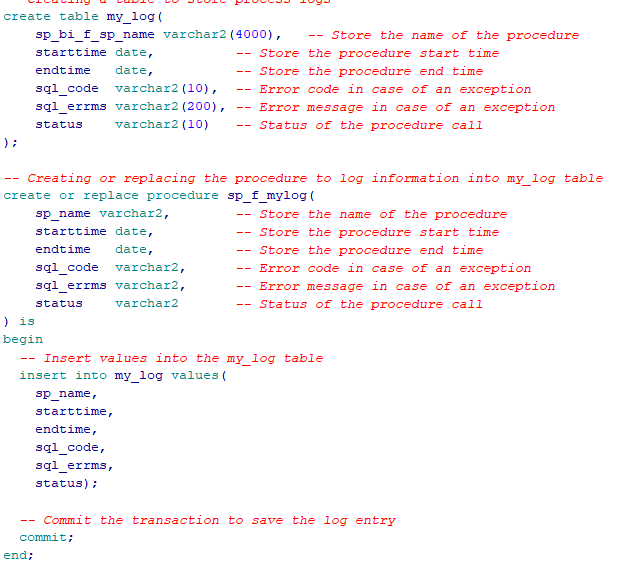


After all data are in the ods layer, we work with F-mapping to ETL the data from bi\_ods to bi\_f (F stands for facts, there are dimension and fact tables where fact tables are like order details which changes frequently and it would be rough for the server to do full updates everyday, therefore we should use incremental update; dimension tables such as sku\_info does not change frequently can do full updates; however for this project we all going to use full update for all just to safe time.)

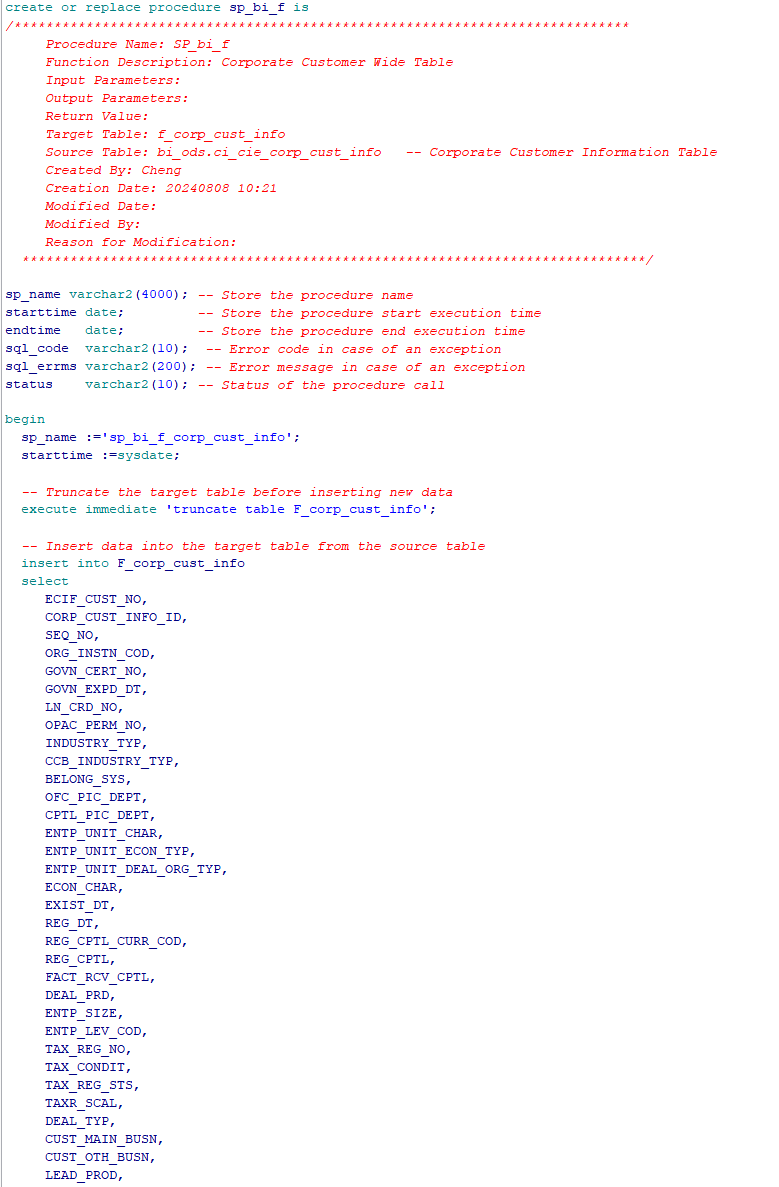
Create a log file **table**, and create a **procedure** called my\_log to insert data when called :

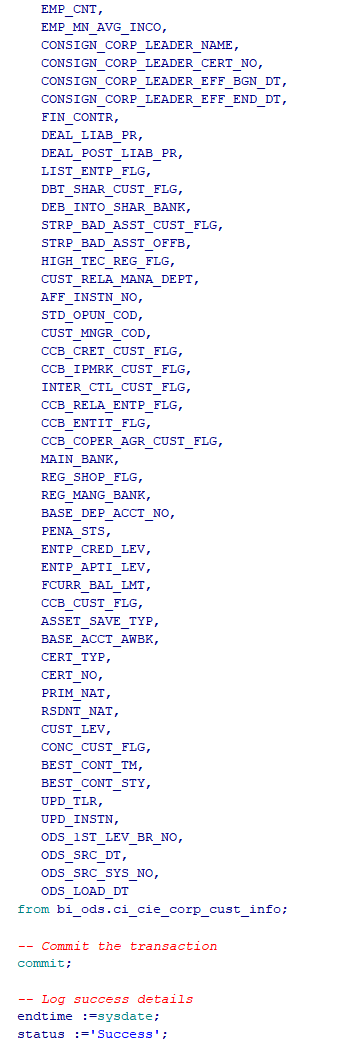
(the purpose is to show historical auto-import of data to see if the outcome:success/fail.)

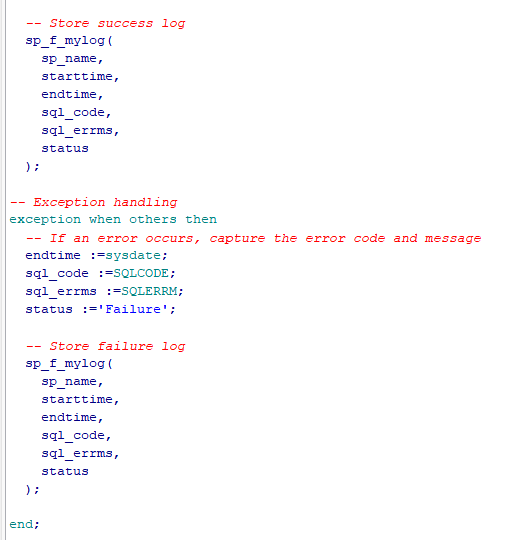
Check out all the codes in PLSQL in ‘traditional\_data\_warehouse\_bi\_bank\_project.sql’.



After creating the log file and log procedure, we need to start the ETL process from ods to f layer, and in the ods-f procedure call the log procedure to make sure it runs with the ETL procedure (1 log file for the layer, 3 fact tables in f layer--look at F mapping.)



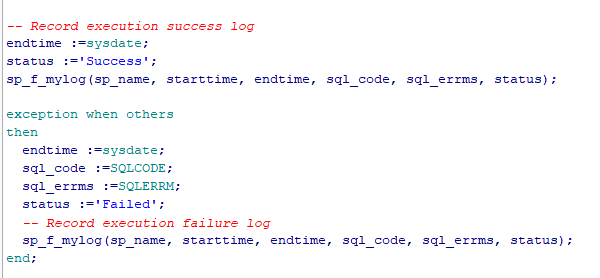






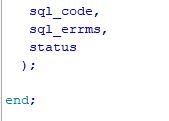




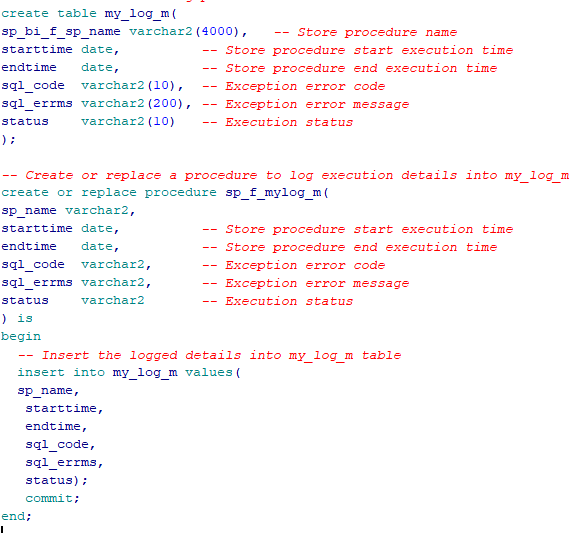




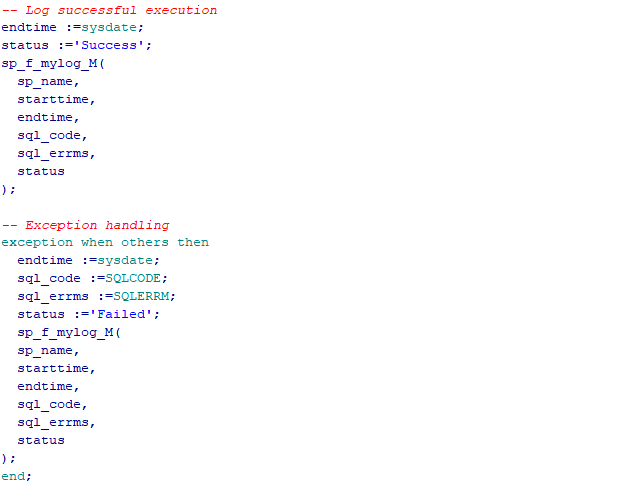




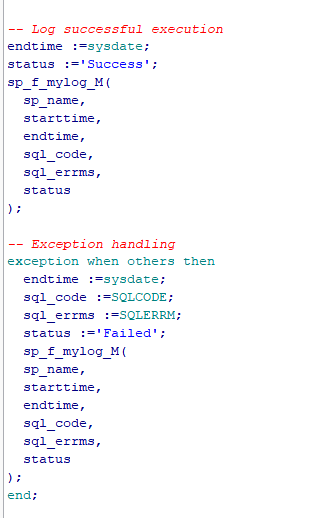
From bi\_f layer to bi\_m layer (look at m-mapping), also create a logfile table, log procedure and fact tables procedures, bear in mind this is the last layer, which means the operation staff will see the KPIs, and usually you need to do UAT (User\_acceptance\_test) with them.











After writing procedures to save work from ods-f-m layer, go back to **kettle** and set a homework-task to automate the process: (check out ETL\_automation.zip)

