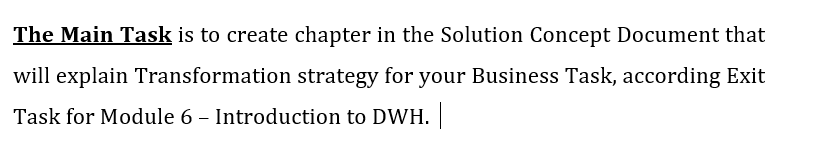
**Лабораторная работа №10**

**Невейков Андрей, 2022**

**All scripts here:**

**https://github.com/AndreyNeveikov/DataMola\_Labs/tree/main/Modul\_2/lab10**

2.1. Task 01: Transformation Description

**Механизмы трансформации**

* Преобразование данных с помощью SQL
* Преобразование данных с помощью PL/SQL
* Преобразование данных с помощью табличных функций

### **Преобразование данных с помощью SQL**

После загрузки данных в базу данных можно выполнять преобразования данных с помощью операций SQL. Существует четыре основных метода реализации преобразований данных SQL:

* CREATE TABLE... AS SELECT И INSERT /\*+APPEND\*/ AS SELECT
* Преобразование данных с помощью UPDATE
* Преобразование данных с помощью MERGE
* Преобразование данных с помощью многотабличной инструкции INSERT

Преобразование с помощью SQL – простой и быстрый способ, обеспечивающий высокую скорость работы. Распространенным типом преобразования данных является подстановка данных. При преобразовании подстановки данных изменяются некоторые или все значения одного столбца.

### **Преобразование данных с помощью PL/SQL**

В среде хранилища данных можно использовать процедурные языки, такие как PL/SQL, для реализации сложных преобразований в базе данных Oracle. В то время как CTAS работает с целыми таблицами и делает упор на параллелизм, PL/SQL предлагает подход, основанный на строках, и может использовать очень сложные правила преобразования. Например, процедура PL/SQL может открывать несколько курсоров и считывать данные из нескольких исходных таблиц, объединять эти данные с помощью сложных бизнес-правил и, наконец, вставлять преобразованные данные в одну или несколько целевых таблиц. Было бы трудно или невозможно выразить ту же последовательность операций, используя стандартные операторы SQL.

### **Преобразование данных с помощью табличных функций**

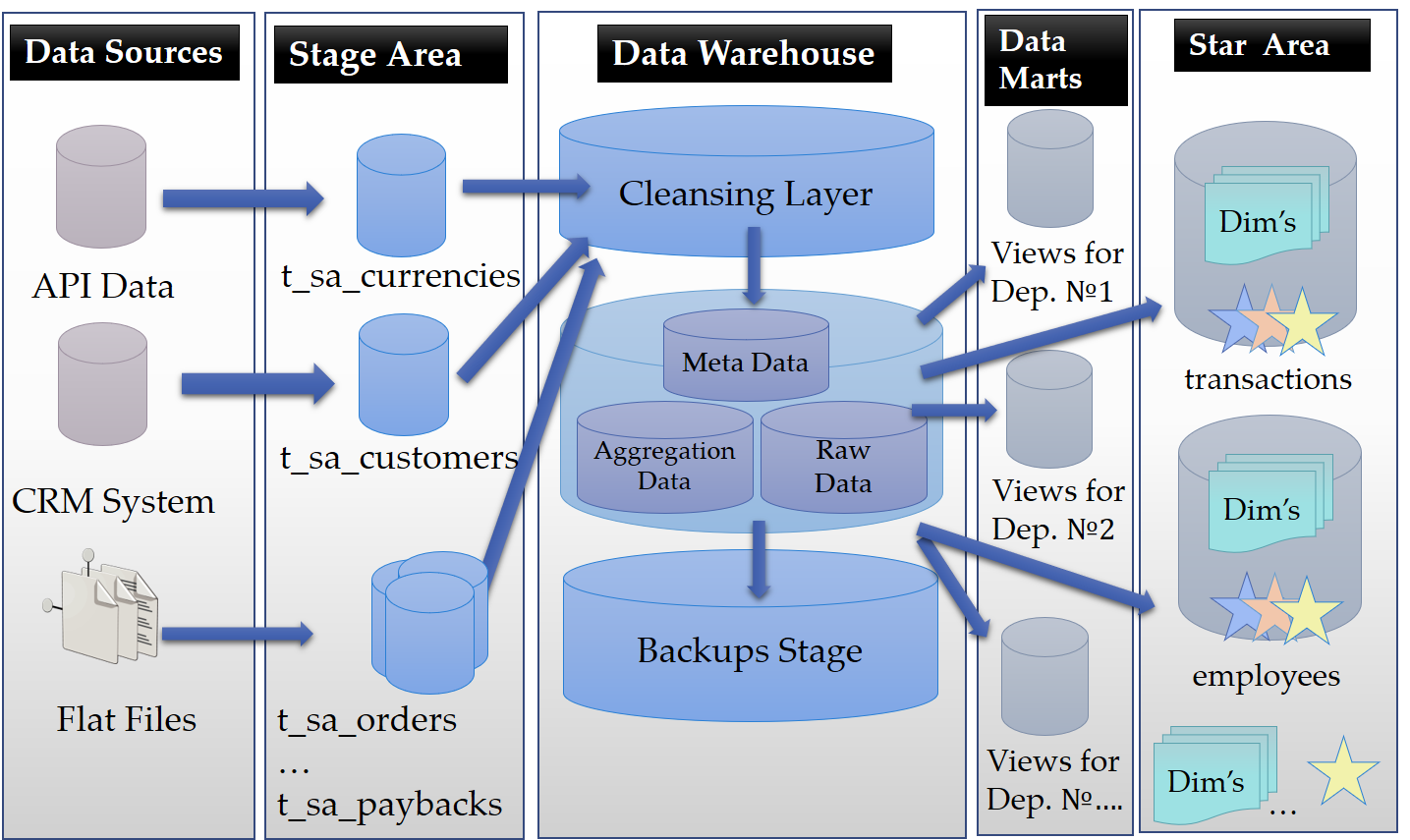
Табличные функции обеспечивают поддержку конвейерного и параллельного выполнения преобразований, реализованных на PL/SQL, C или Java. Сценарии, упомянутые ранее, могут выполняться без использования промежуточных таблиц, которые прерывают поток данных на различных этапах преобразования.

### **Преобразование данных в моей бизнес-модели**

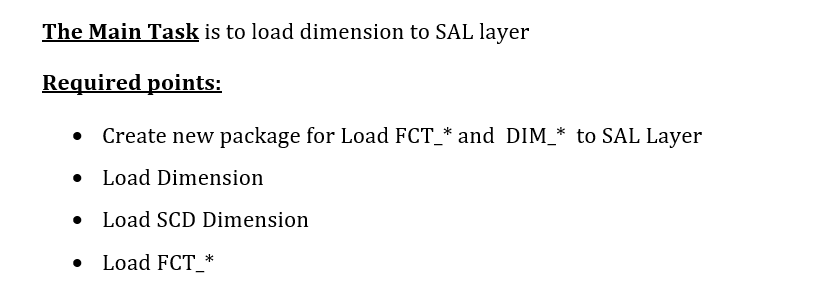
В моем бизнесе лучше использовать совместно преобразование данных с помощью SQL для небольших таблиц, таблиц, которые редко обновляются и таблиц-справочников т.к. их простая структура обеспечит максимальную скорость трансформации. Для факт-таблиц, в которые добавляются id из других таблиц лучше использовать преобразование с помощью PL/SQL т.к. надо объединять несколько таблиц.

Схема трансформации данных:

Из data sources данные попадают в stage area, далее из нее в cleansing layer, где происходит поиск повторяющихся значений, null­-полей, проверка форматов данных и происходит трансформация под единый стандарт, после cleansing layer-а создаются индексы и данные разделяются между таблицами в Data Warehouse. Когда в DW собрана вся информация, создаются Data Marts для каждого отдела (бухгалтеров, продавцов, директоров и т.д.). На star area объединяются данные из нескольких dimentions для составления финальных отчетов или создания информационных продуктов для клиентов на основе проанализированных данных.



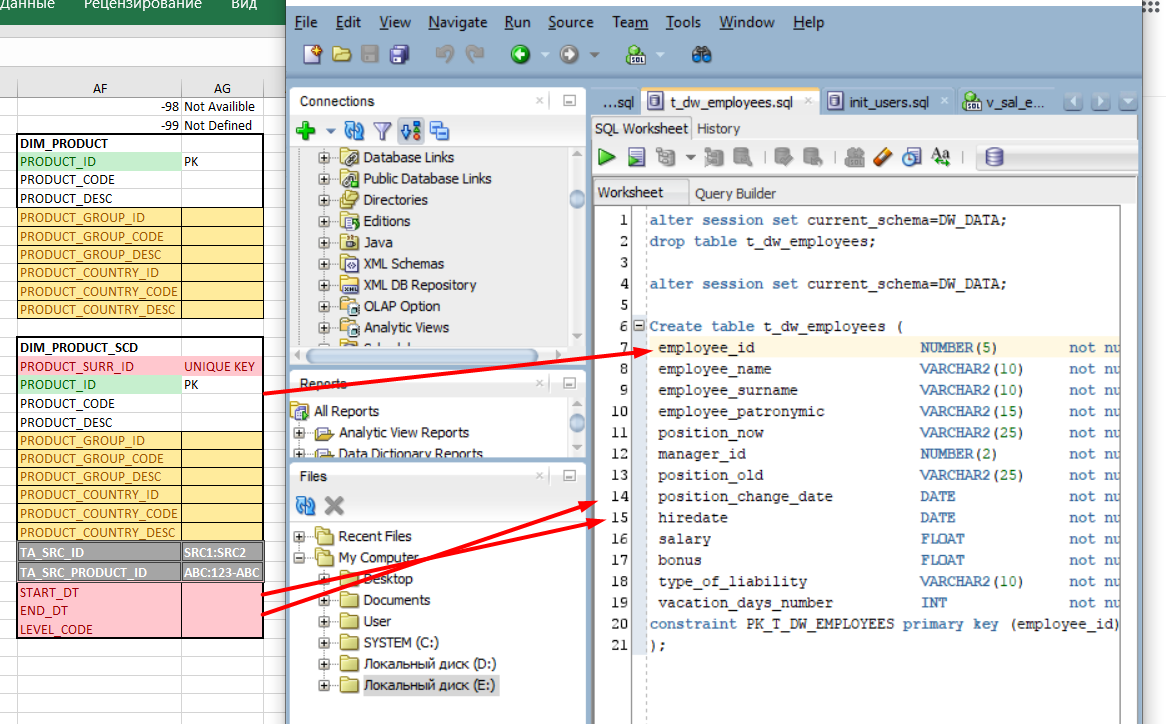
3.1. Task 02: Loading to SAL Layer Data



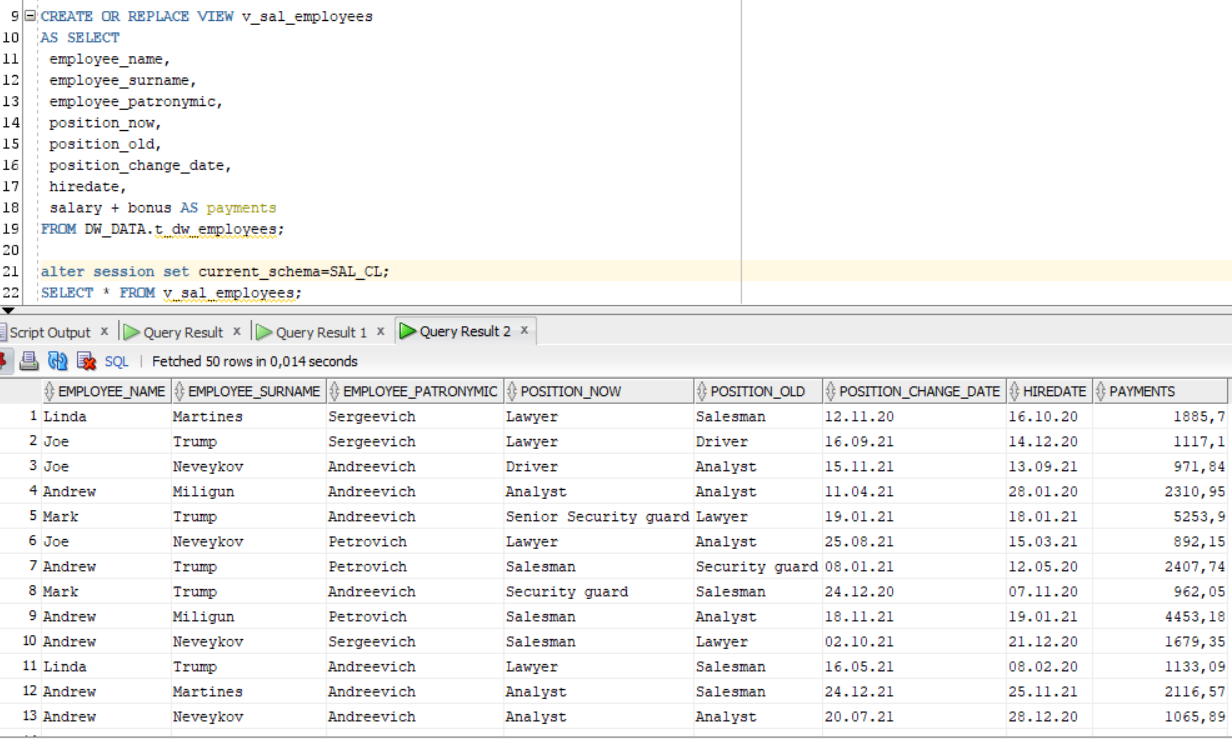
Чтобы SAL-уровень привнес какую-то дополнительную информацию, я добавлял дополнительные поля на основе математических вычислений из существующих полей. Например: «окупаемость» = 1-(расходы/доходы), «Затраты на сотрудника» = зарплата + премия. Также добавлял группировки и агрегирующие функции, чтобы результирующие данные могли быть реально применимы в бизнесе.

Load SCD Dimention:

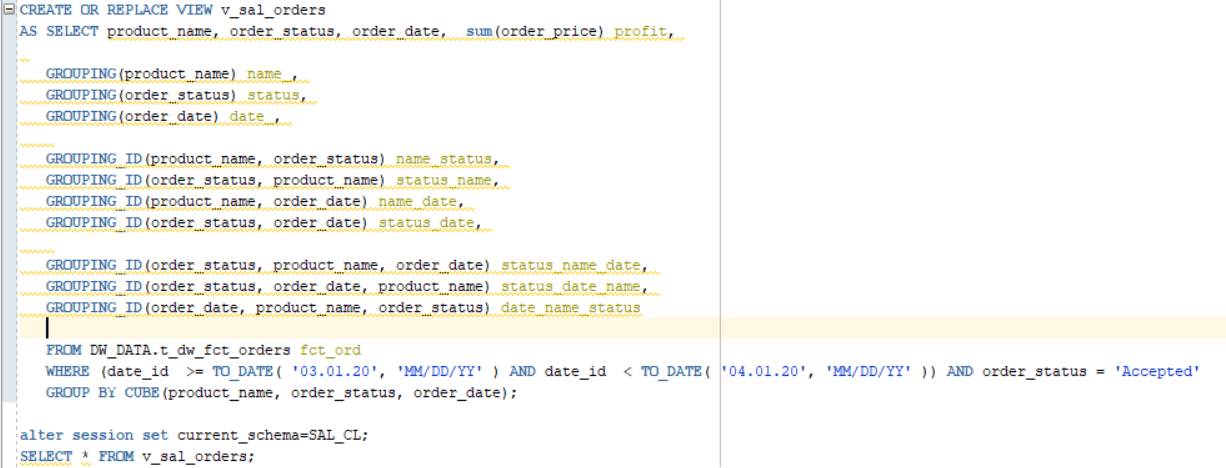
В моей схеме dimention employees относится к SCD-3



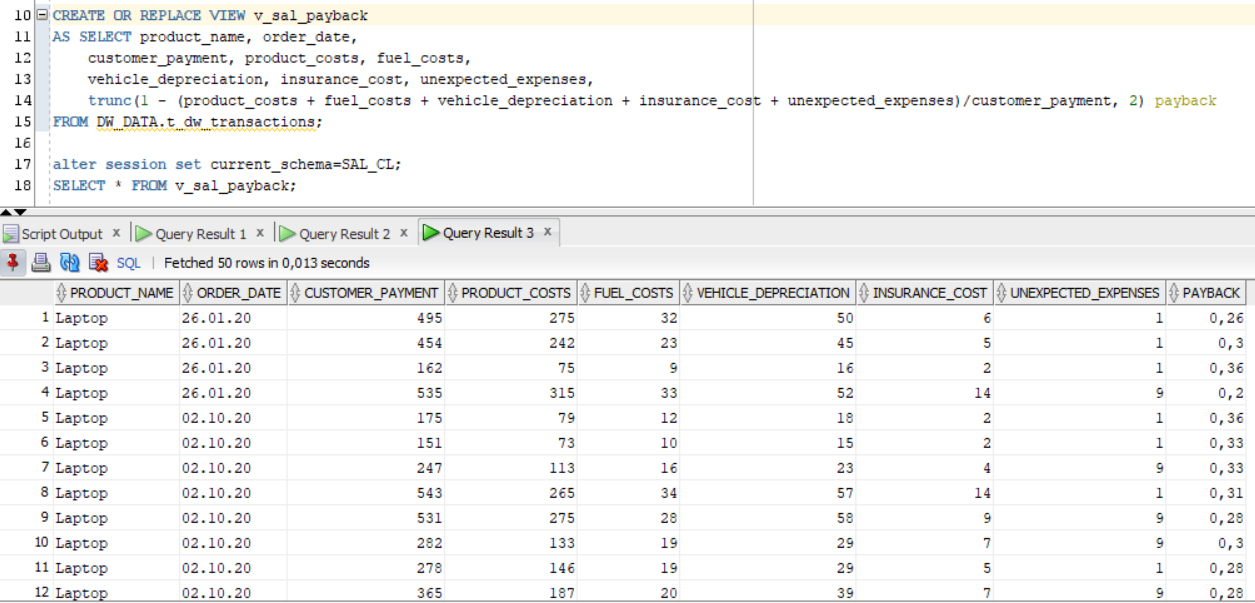
Результат переноса:



Load FCT\_table:

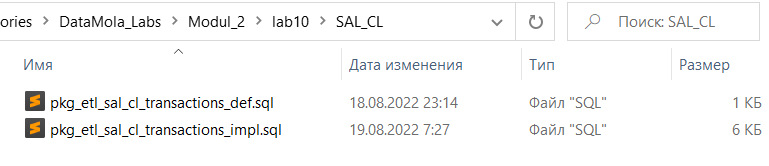


Load dimentions:

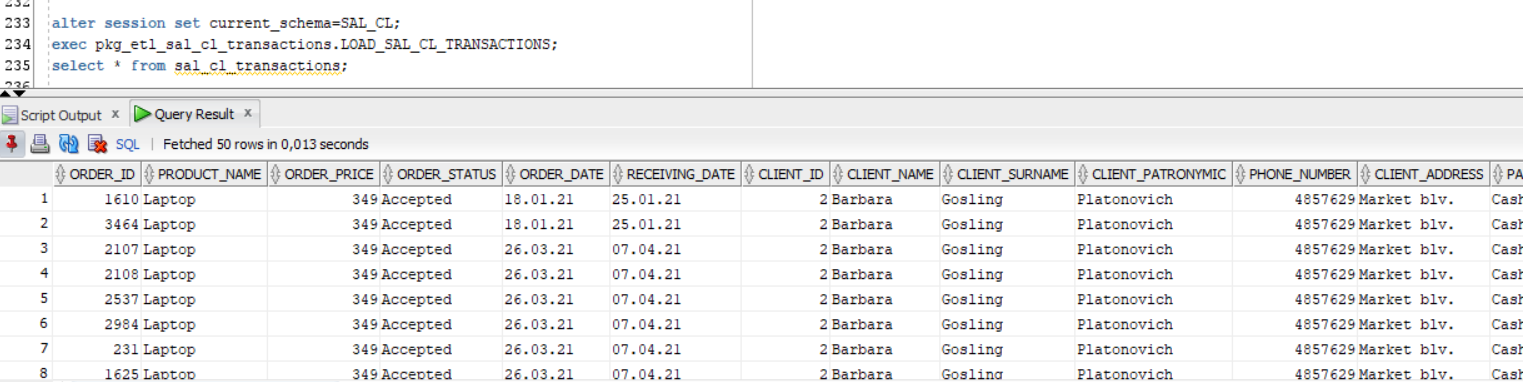


Create new package for load FCT\_\* and DIM\_\* tables:

Чтобы загрузить одним пакетом FCT\_\* и DIM\_\* я собрал их в одну таблицу и перенес всю информацию вместе. Пакет состоит из двух файлов: в def объявляется пакет, в impl прописана его логика.



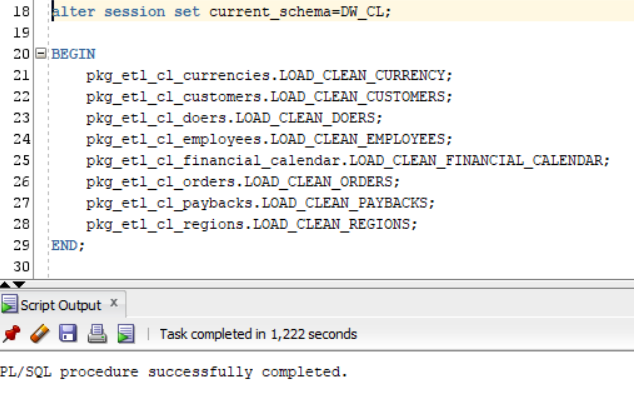
Результат переноса:



### **Задание от Кирилла:**

Объединить прогрузку данных по уровням:

SA-CL:



CL-DW:

