**Лабораторная работа №9**

**Невейков Андрей, 2022**

**All scripts here:**

## **https://github.com/AndreyNeveikov/DataMola\_Labs/tree/main/Modul\_2/lab9**

## Lab9: 2.1. Task 01: Transportation Description

**Transportation Using Flat Files**

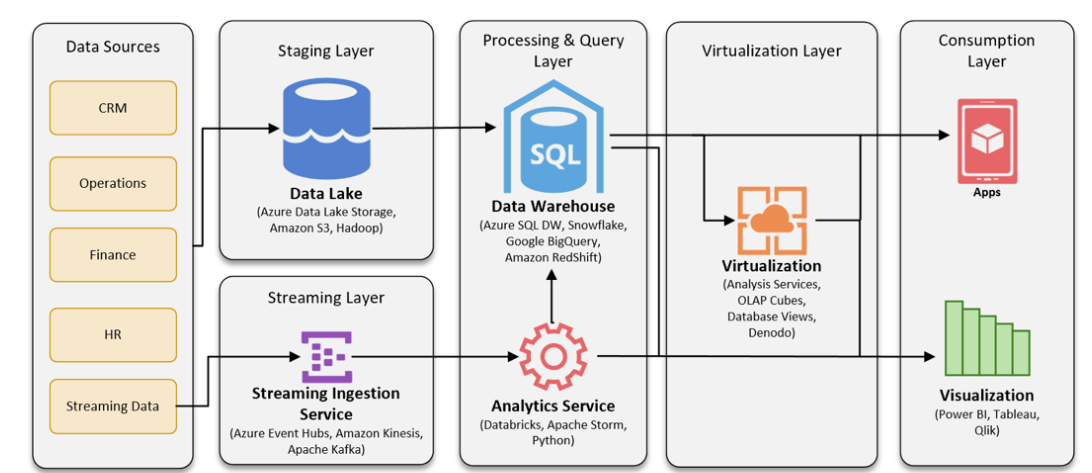
Наиболее распространенным методом передачи данных является передача плоских файлов с использованием таких механизмов, как FTP или другие протоколы доступа к удаленной файловой системе. Данные выгружаются или экспортируются из исходной системы в плоские файлы с использованием методов, а затем передаются на целевую платформу с использованием FTP или аналогичных механизмов. Поскольку исходные системы и хранилища данных часто используют разные операционные системы и системы баз данных, использование плоских файлов часто является самым простым способом обмена данными между разнородными системами с минимальными преобразованиями. Однако даже при передаче данных между однородными системами плоские файлы часто являются наиболее эффективным и простым в управлении механизмом передачи данных.  
 В моем бизнесе лучше всего брат информацию из Flat файлов т.к. каждая таблица привязана к разному источнику (orders и clients с сайтом, currencies с API биржи, employees с компьютеров HR-специалистов). Также Flat файлы легко генерируются бэкэндом сайта.

### **Transportation Using Transportable Tablespaces**

С помощью переносимых табличных пространств файлы данных Oracle (содержащие табличные данные, индексы и почти любой другой объект базы данных Oracle) можно напрямую транспортировать из одной базы данных в другую. Кроме того, подобно импорту и экспорту, переносимые табличные пространства предоставляют механизм для транспортировки метаданных в дополнение к транспортировке данных.  
Переносимые табличные пространства имеют некоторые заметные ограничения: исходная и целевая системы должны работать под управлением Oracle8 i (или выше), должны работать под управлением одной и той же операционной системы, должны использовать один и тот же набор символов и , до Oracle9i , должны использовать одинаковый размер блока.    
 В моем бизнесе применение данного метода не выгодно т.к. объемы данных относительно небольшие, чтобы перемещать из одной бд в другую. Системы используют разные операционные системы. И применение данного способа в данном случае просто вносит дополнительную сложность в продукт.

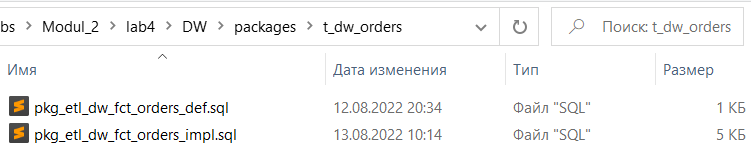
**Architecture**

Для моего бизнеса лучше всего подходит *Lambda* архитектура или смешанная т.к. есть часть Streaming Data (orders и customers), актуальность которых нужно проверять постоянно, остальные обновляются реже, например информация о новых сотрудниках от HR-ов, финансовые операции для квартальных и годовых отчетов, существуют справочники календаря и регионов, которые вообще никогда не обновляются. Так получилось, что идеальная архитектура моего бизнеса полностью совпадает со следующей схемой:

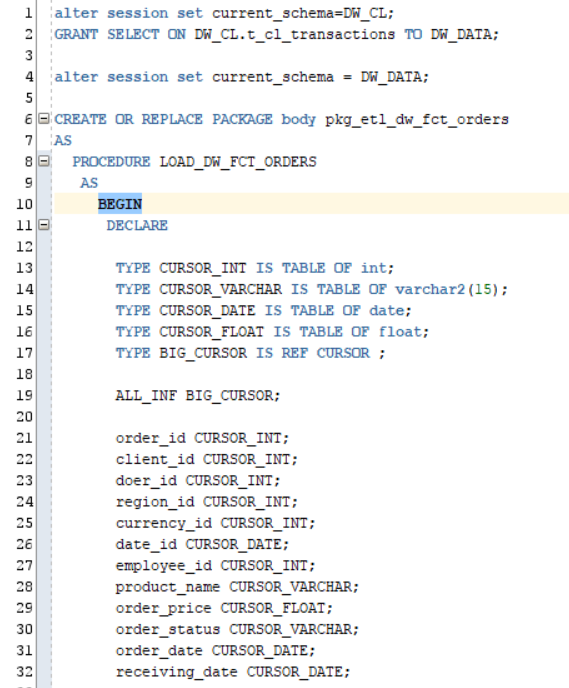
****

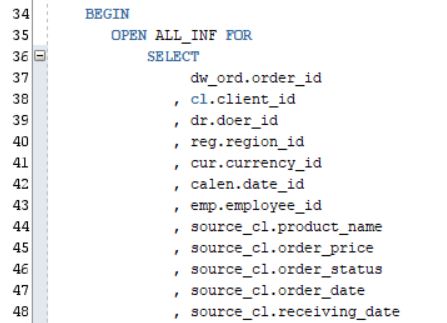
## Lab8-9 3.1. Task 02: Prepare Table of Facts to DW Layer

Пакеты для прогрузки таблицы фактов были закончены еще утром 13.08.2022 для lab4. Пакет состоит из def и impl частей. В def – объявление пакета, в impl – логика.

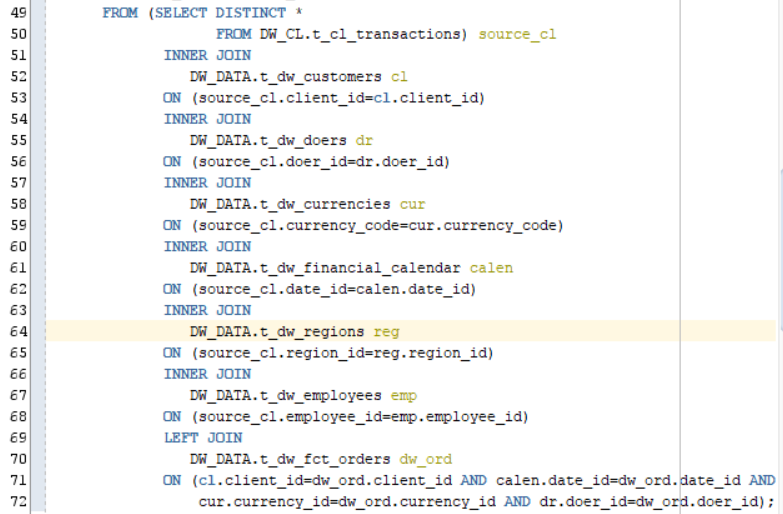


Для перемещения данных используется процедура с курсором:

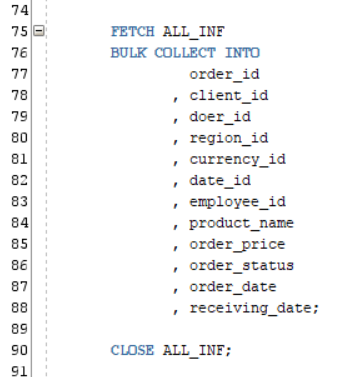




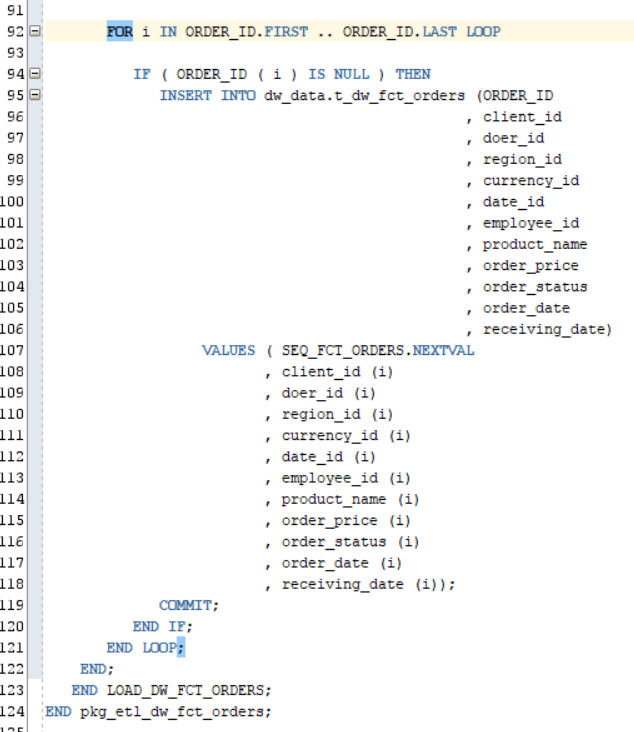
Перемещение данных происходит при помощи соединения с другими таблицами, для сохранения целостности данных т.к. на DW-уровне появляются несколько полей с индексами из других таблиц.



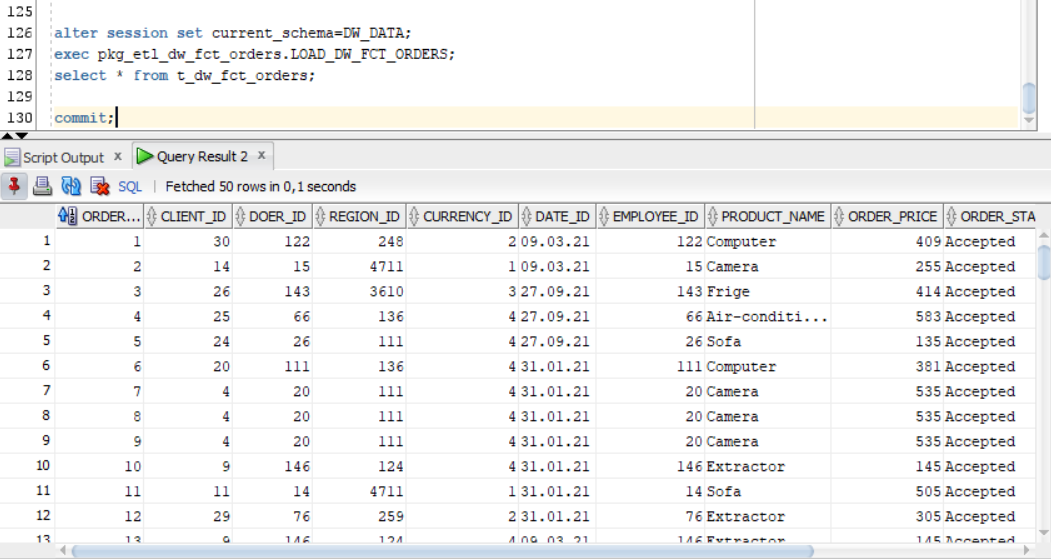
Извлекаются все данные:



Начинается цикл, который заполняет таблицу нужными значениями и присваивает собственный индекс заказа:



Результат работы пакета:



**SUMMARY**

1. **Я научился** переносить данные в факт-таблицу.
2. **Я узнал**, способ сохранения целостности данных при переносе информации в факт-таблицу (связывая ее с другими, предварительно заполненными

таблицами;

1. **Я понял**, как переносить поля из нескольких таблиц в одну одновременно.