Дипломный проект по профессии Инженер данных

1.Цель проекта.

Разработка системы извлечения, трансформации и загрузки данных (ETL) в хранилище данных с последующим построением аналитической витрины и визуализации ключевых метрик.

2. Используемые технологии

Язык программирования: Python (pandas, sqlalchemy)

СУБД: PosgreSQL

Оркестрация ETL: Apache Airflow

BI-инструмент: Tableau

IDE (набор ПО для создания кода): Dbeaver, VSCode, Jupyter Notebook

3. Структура хранилища данных.

Основной источник данных — CSV-файл, содержащий информацию о продажах: транзакции, клиенты, товары, филиалы, даты и пр. Всего 1000 строк, структура таблицы включает такие поля, как invoice_id, branch, city, customer_type, gender, product_line, unit_price, quantity', tax_5_percent, total, date, time, payment_method, cogs, gross_margin_percentage, gross_income, rating.

3. Описание слоёв хранилища

3.1. Слой NDS (Normalized Data Store) нормализованное хранилище

Созданы следующие нормализованные таблицы:

nds_customers — информация о клиентах (тип и пол);

customer_id-первичный ключ с автоинкриментом,

customer_type – тип клиента,

gender-пол клиента.

nds_products — товары и цены;

product_id (PK) -первичный ключ с автоинкриментом,

product_line – категории товаров,

unit_price-стоимость товара за единицу.

nds_branches — филиалы и города;

branch_id- первичный ключ с автоинкриментом,

branch_code-номер филиала,

city-город расположение филиала.

nds_dates — дата и календарные показатели;

full_date-полная дата,

day-день,

month-месяц,

year-год,

weekday-день недели.

nds_sales — факты продаж со связями на остальные таблицы через внешние ключи;

invoice_id- первичный ключ, номер счет-фактуры,

customer_id-внешний ключ из таблицы nds_customers,

products-внешний ключ из таблицы nds_products,

branch_id-внешний ключ из таблицы nds_branches,

date_id-внешний ключ из таблицы nds_dates,

quantity-количество товаров,

tax_5_percent-налог 5%,

total-сумма покупки,

payment_method-метод оплаты,

cogs-себестоимость,

gross_margin_percentage-процент маржи,

gross_income-валовая прибыль,

rating-Оценка.

3.2. Слой DDS (Dimensional Data Store) схема звезда

Слои построенные по звёздной схеме:

dim_customer — измерение по клиенту;

customer_id-первичный ключ с автоинкриментом,

customer_type – тип клиента,

gender-пол клиента.

dim_product — измерение по продукту;

product_id (PK) -первичный ключ с автоинкриментом,

product_line – категории товаров,

unit_price-стоимость товара за единицу.

dim_branch — измерение по филиалу;

branch_id- первичный ключ с автоинкриментом,

branch_code-номер филиала,

city-город расположение филиала.

dim_date — измерение по дате;

full_date-полная дата,

day-день,

month-месяц,

year-год,

weekday-день недели.

fact_sales — таблица фактов продаж с расчётными полями:

invoice_id- первичный ключ, номер счет-фактуры,

customer_id-внешний ключ из таблицы dim_customers,

products-внешний ключ из таблицы dim_products,

branch_id-внешний ключ из таблицы dim_branch,

date_id-внешний ключ из таблицы dim_date,

quantity-количество товаров,

tax_5_percent-налог 5%,

total-сумма покупки,

payment_method-метод оплаты,

cogs-себестоимость,

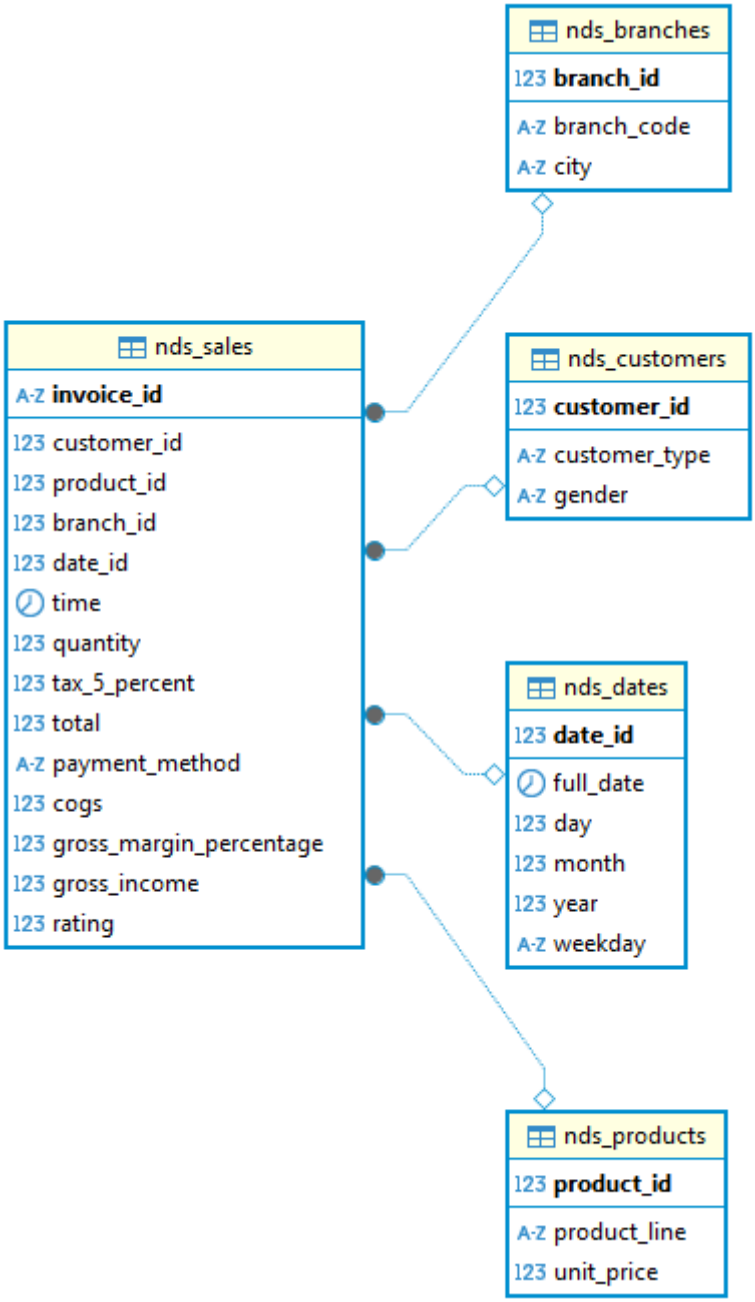
gross_income-валовая прибыль,

rating-Оценка.

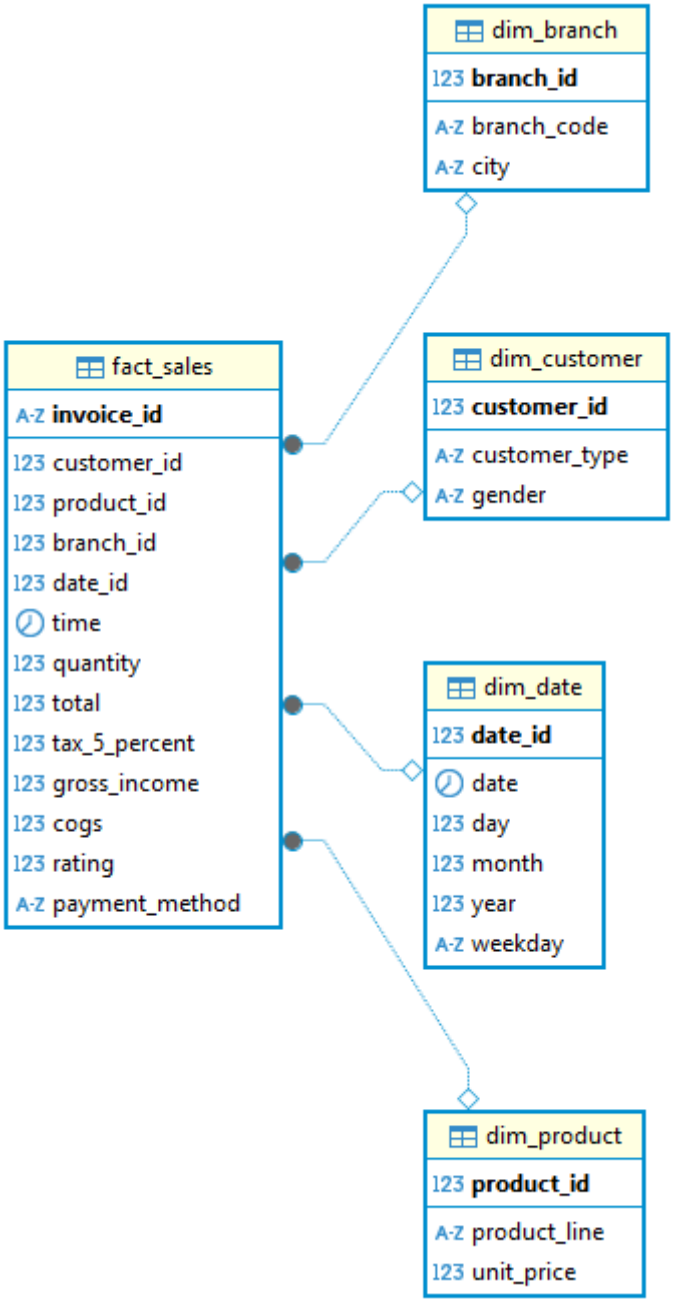
4. ER-диаграммы

В качестве ER-диаграмм использованы схемы, сгенерированные в DBeaver. Они показывают связи между таблицами NDS и DDS и визуально иллюстрируют структуру хранилища.

ER-диаграмма схемы NDS.



ER-диаграмма схемы DDS.



5. ETL-процессы

Реализованы с помощью Python (pandas + psycopg2) и SQL:

Загрузка и очистка данных из CSV;

Проверка и удаление дубликатов по invoice_id;

Заполнение нормализованных таблиц (NDS);

Формирование витрин данных (DDS) с использованием surrogate keys и агрегаций.

6. Оркестрация

Оркестрация процессов реализована с использованием Apache Airflow. Настроены DAG` и, выполняющие:

create_all_tables - создание схем и таблиц;

load_sales_data – трасформация и загрузка из CSV-файла в на ненормализованную таблицу sales_data;

etl_sales_to_nds - заполнение ND –таблиц с surrogate keys;

etl_nds_to_dds – построение DDS на основе NDS

master_dag – код для запуска dag` ов по очереди загрузку и трансформацию данных.

7. Качество данных

Проведены проверки на Python-скрипте и показаны выводы в отдельно созданной таблице по проверке качества данных:

Удаление дубликатов по invoice_id;

Обработка пропущенных значений;

Валидация типов и диапазонов значений.

8. Визуализация в Tableau

Созданы дашборды на основе данных из DDS в Tableau:

Анализ среднего чека по месяцам;

Анализ продаж категории продуктов по клиенту;

Анализ продаж по способу оплаты по городам и филиалам;

Средний чек по дням каждого месяца;

Средний рейтинг по категориям товаров;

Создан параметр для фильтрации по типу и полу клиента.

9.Особенности реализации

Все таблицы создаются программно через SQLAlchemy.

DAG полностью автоматизирует цепочку от загрузки до DDS.

Уникальные surrogate keys с автоинкрементом.

Структура данных документирована с помощью ER-диаграмм (DBeaver).

Возможность масштабирования DAG`ов при добавлении новых источников.

10. Выводы

Реализована полноценная ETL-система с оркестрацией.

Данные хранятся в PostgreSQL, витрина построена в DDS.

Построены BI-дашборды в Tableau на основе fact-таблицы.

Подготовлена финальная документация для защиты диплома.

Дженишбеков Нурбек

DEG-34

18.04.2025