МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

Санкт-Петербургский национальный исследовательский университет ИТМО

Мегафакультет трансляционных информационных технологий

Факультет информационных технологий и программирования

Лабораторная работа №3

По дисциплине «Технологии управления данными»

Выполнил студент группы №M3306

Бочагова Екатерина

Проверил

Повышев Владислав Вячеславович

Санкт-Петербург

2025

Условие лабораторной работы:

Витрина данных представляет собой специализированное хранилище, где данные консолидируются по определенному признаку. В данной работе требуется консолидировать результаты продаж по неделям. Это обеспечит уменьшение нагрузки на основное Хранилище данных.

Основными сущностями Витрины будут:

1 Недели;

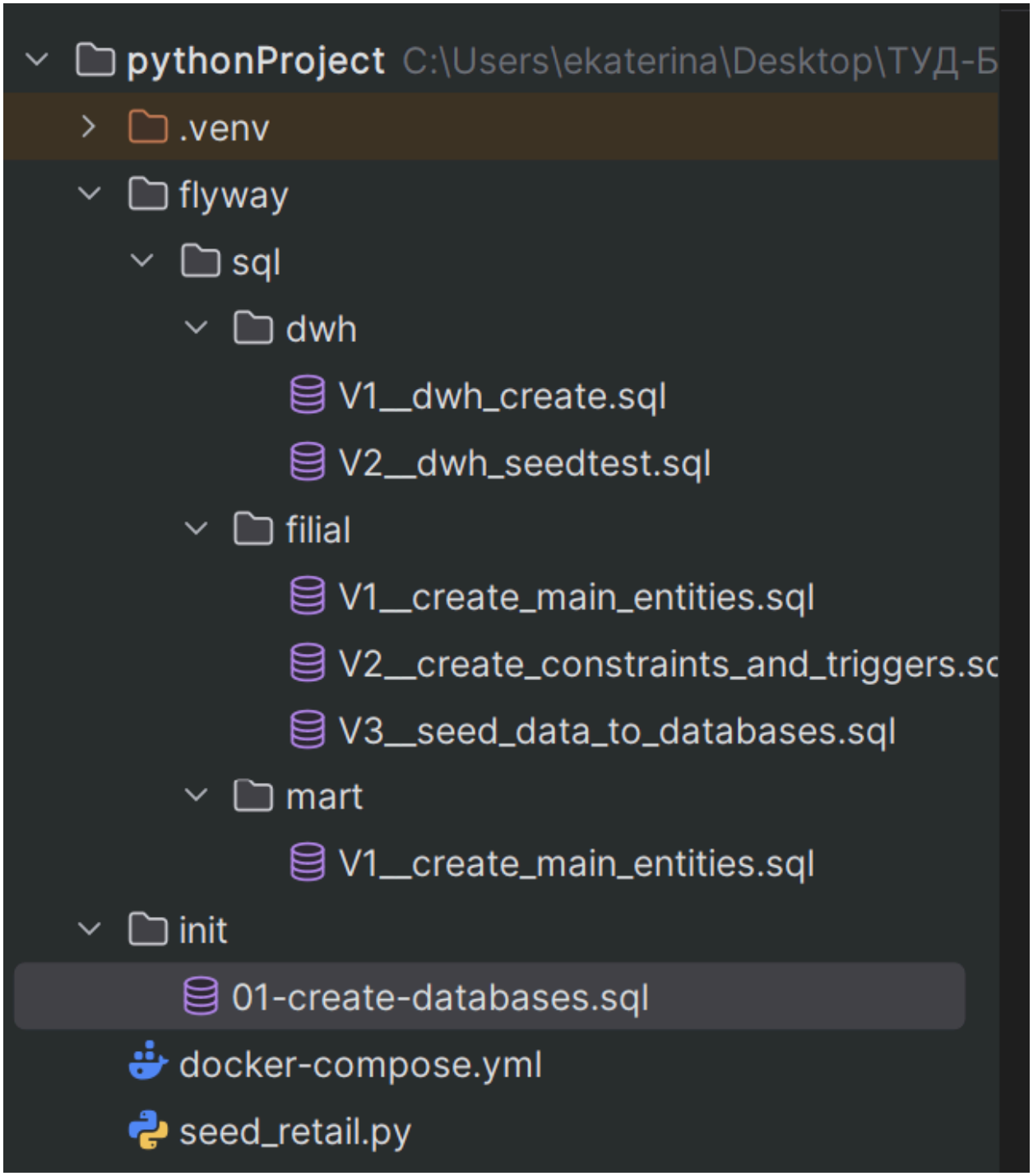
2 Продажи.

Сущность недели хранит информацию о временных промежутках, за которые была произведена консолидация данных. Сущность Продажи обеспечивает хранение информации о результатах продаж в Филиале в соответствующую неделю.

Подготовка к выполнению работы:

1) Немного изменим структуру проекта - в папку с миграциями добавим еще одну папку - mart - для миграций, связанных с витриной хранилища
2) Дополним docker-compose еще одним сервисом для миграций - для витрины

Итак, нынешняя структура проекта будет выглядеть следующим образом:



Дополненный docker-compose.yml (выделю цветом только добавленный кусок кода, писала сама без использования ии)

```
version: '3.9'


services:
```

```yaml
postgres:
  image: postgres:16.9
  restart: unless-stopped
  environment:
    POSTGRES_USER: postgres
    POSTGRES_PASSWORD: password
    POSTGRES_DB: postgres
  ports:
    - "5434:5432"
  volumes:
    - pgdata:/var/lib/postgresql/data
    - ./init:/docker-entrypoint-initdb.d
  networks:
    - filialnet
  healthcheck:
    test: ["CMD-SHELL", "pg_isready -U postgres"]
    interval: 5s
    timeout: 5s
    retries: 5
    start_period: 10s


flyway_west:
  image: flyway/flyway:11.14.1
  depends_on:
    postgres:
```

```
      condition: service_healthy
  environment:
    FLYWAY_URL: jdbc:postgresql://postgres:5432/filial_west
    FLYWAY_USER: postgres
    FLYWAY_PASSWORD: password
    FLYWAY_LOCATIONS: filesystem:/flyway/sql/filial
    FLYWAY_BASELINE_ON_MIGRATE: true
    FLYWAY_SQL_MIGRATION_PREFIX: V
    FLYWAY_SQL_MIGRATION_SEPARATOR: __
    FLYWAY_SQL_MIGRATION_SUFFIXES: .sql
  volumes:
    - ./flyway/sql/filial:/flyway/sql/filial
  networks:
    - filialnet
  command: migrate


flyway_east:
  image: flyway/flyway:11.14.1
  depends_on:
    postgres:
      condition: service_healthy
  environment:
    FLYWAY_URL: jdbc:postgresql://postgres:5432/filial_east
    FLYWAY_USER: postgres
    FLYWAY_PASSWORD: password
```

```
    FLYWAY_LOCATIONS: filesystem:/flyway/sql/filial
    FLYWAY_BASELINE_ON_MIGRATE: true
    FLYWAY_SQL_MIGRATION_PREFIX: V
    FLYWAY_SQL_MIGRATION_SEPARATOR: __
    FLYWAY_SQL_MIGRATION_SUFFIXES: .sql
  volumes:
    - ./flyway/sql/filial:/flyway/sql/filial
  networks:
    - filialnet
  command: migrate


flyway_dwh:
  image: flyway/flyway:11.14.1
  depends_on:
    postgres:
      condition: service_healthy
  environment:
    FLYWAY_URL: jdbc:postgresql://postgres:5432/dw_retail
    FLYWAY_USER: postgres
    FLYWAY_PASSWORD: password
    FLYWAY_LOCATIONS: filesystem:/flyway/sql/dwh
    FLYWAY_BASELINE_ON_MIGRATE: true
    FLYWAY_SQL_MIGRATION_PREFIX: V
    FLYWAY_SQL_MIGRATION_SEPARATOR: __
    FLYWAY_SQL_MIGRATION_SUFFIXES: .sql
```

```
    volumes:
      - ./flyway/sql/dwh:/flyway/sql/dwh
    networks:
      - filialnet
    command: migrate


flyway_mart:
    image: flyway/flyway:11.14.1
    depends_on:
      postgres:
        condition: service_healthy
      flyway_dwh:
        condition: service_completed_successfully
    environment:
      FLYWAY_URL: jdbc:postgresql://postgres:5432/dw_retail
      FLYWAY_USER: postgres
      FLYWAY_PASSWORD: password
      FLYWAY_LOCATIONS: filesystem:/flyway/sql/mart
      FLYWAY_BASELINE_ON_MIGRATE: true
      FLYWAY_VALIDATE_ON_MIGRATE: false
      FLYWAY_CLEAN_DISABLED: false
      FLYWAY_SQL_MIGRATION_PREFIX: V
      FLYWAY_SQL_MIGRATION_SEPARATOR: __
      FLYWAY_SQL_MIGRATION_SUFFIXES: .sql
    volumes:
```

```
      - ./flyway/sql/mart:/flyway/sql/mart

    networks:

      - filialnet

    command: migrate


volumes:

 pgdata:


networks:

 filialnet:
```

Проверим работоспособность - при помощи команд docker-compose down -v (удалим еще все вольюмы из бд, чтобы проверить, не сломали ли мы что-то) и docker-compose up --build

Видим, что процессы завершились с кодом 0 -> все отлично и корректно отработало

```
flyway_east-1 exited with code 0
flyway_dwh-1 exited with code 0
flyway_west-1 exited with code 0
flyway_mart-1 exited with code 0
```

Перейдем к описанию таблиц, которые нам необходимо создать на следующем этапе

1) Недели - dim_week

   Хранит информацию о календарных неделях

Week_key - INT - Уникальный ключ недели в формате ГГГГНН - первичный ключ

week_start_date - DATE - Дата начала недели (понедельник)

week_end_date DATE - Дата окончания недели (воскресенье)

Year - INT - ISO-год

Iso_week - INT - Номер ISO-недели (от 1 до 53).

Load_date - TIMESTAMPTZ - Дата и время загрузки записи (по умолчанию now()).

2) Продажи - fact_weekly_sales

Содержит агрегированные показатели продаж по неделям, филиалам и продуктам

Weekly_sale_key - SERIAL - Уникальный идентификатор строки (первичный ключ)

Week_key - INT - Ссылка на измерение недель (dim_week.week_key)

Branch_key - INT - Идентификатор филиала

Product_key - INT - Идентификатор товара

Total_quantity - BIGINT - Общее количество проданных единиц товара за неделю

Total_revenue - NUMERIC(18,2) - Общая выручка за неделю

Sales_count - BIGINT - Количество чеков за неделю

Last_update - TIMESTAMPTZ - Дата и время последнего обновления записи

Скрипт:

(отредактировала при помощи claude 4.5)

```
CREATE EXTENSION IF NOT EXISTS pgcrypto;



CREATE SCHEMA IF NOT EXISTS mart;

SET search_path TO mart;



CREATE TABLE IF NOT EXISTS dim_week (
```

```
    week_key INT PRIMARY KEY,
    week_start_date DATE NOT NULL UNIQUE,
    week_end_date DATE NOT NULL,
    year INT NOT NULL,
    iso_week INT NOT NULL,
    load_date TIMESTAMPTZ NOT NULL DEFAULT now()
);


CREATE TABLE IF NOT EXISTS fact_weekly_sales (
    weekly_sale_key SERIAL PRIMARY KEY,
    week_key INT NOT NULL,
    branch_key INT NOT NULL,
    product_key INT NOT NULL,
    total_quantity BIGINT NOT NULL DEFAULT 0,
    total_revenue NUMERIC(18,2) NOT NULL DEFAULT 0,
    sales_count BIGINT NOT NULL DEFAULT 0,
    last_update TIMESTAMPTZ NOT NULL DEFAULT now(),
    UNIQUE (week_key, branch_key, product_key)
);


CREATE INDEX IF NOT EXISTS idx_fact_week ON
fact_weekly_sales(week_key);
CREATE INDEX IF NOT EXISTS idx_fact_branch ON
fact_weekly_sales(branch_key);
CREATE INDEX IF NOT EXISTS idx_fact_product ON
fact_weekly_sales(product_key);
CREATE UNIQUE INDEX IF NOT EXISTS idx_fact_week_branch_product
```

```
 ON fact_weekly_sales(week_key, branch_key, product_key);



DO $$

DECLARE

 week_start date := ('2023-01-01'::date - EXTRACT(ISODOW FROM
'2023-01-01'::date)::int + 1)::date;

 last date := '2025-12-31';

 wk_iso int;

 wk_year int;

 wk_key int;

BEGIN

 WHILE week_start <= last LOOP

   wk_iso := EXTRACT(week FROM week_start)::int;

   wk_year := EXTRACT(isoyear FROM week_start)::int;

   wk_key := wk_year * 100 + wk_iso;



   INSERT INTO dim_week (week_key, week_start_date, week_end_date,
year, iso_week)

   VALUES (wk_key, week_start, week_start + INTERVAL '6 days',
wk_year, wk_iso)

   ON CONFLICT (week_key) DO NOTHING;



   week_start := week_start + INTERVAL '1 week';

 END LOOP;

END $$;
```