**Домашнее задание №4**

Шалаев Алексей Дмитриевич БПИ222

**1 Задача**

1. Создается docker-compose.yml с двумя сервисами: PostgreSQL

Написал ради интереса 2 инстанса постгры: master и slave, где slave – реплика master, ее, например, можно исп. для чтения, а master для read & write операций.

version: "3.9"

x-postgres-common:

  &postgres-common

  image: postgres:15

  user: postgres

  healthcheck:

      test: [ "CMD-SHELL", "pg\_isready", "-d", "production"]

      interval: 10s

      timeout: 3s

      retries: 3

  restart: on-failure:3

  networks:

    - db-network

services:

  postgres-master:

    <<: \*postgres-common

    container\_name: db-postgres-master

    environment:

      POSTGRES\_USER: avito

      POSTGRES\_PASSWORD: hackme

      POSTGRES\_DB: production

      REPLICATION\_USER: replicator

      REPLICATION\_PASSWORD: aboba

      POSTGRES\_HOST\_AUTH\_METHOD: "scram-sha-256\nhost replication all 0.0.0.0/0 md5"

      POSTGRES\_INITDB\_ARGS: "--auth-host=scram-sha-256"

    volumes:

      - db-postgres-master-data:/var/lib/postgresql/data

      - ./init-scripts:/docker-entrypoint-initdb.d/

    ports:

      - "15432:5432"

    command: |

      postgres

      -c wal\_level=replica

      -c hot\_standby=on

      -c max\_wal\_senders=10

      -c max\_replication\_slots=10

      -c hot\_standby\_feedback=on

  postgres-slave:

    <<: \*postgres-common

    container\_name: db-postgres-slave

    environment:

      PGUSER: replicator

      PGPASSWORD: replicator-password

    depends\_on:

      - postgres-master

    ports:

      - "25432:5432"

    command: |

      bash -c "

      rm -rf /var/lib/postgresql/data/\*

      until pg\_basebackup --pgdata=/var/lib/postgresql/data -R --slot=replication\_slot --host=postgres-master --port=5432

      do

      echo 'Waiting for primary to connect...'

      sleep 1s

      done

      echo 'Backup done, starting replica...'

      chmod 0700 /var/lib/postgresql/data

      postgres

      "

networks:

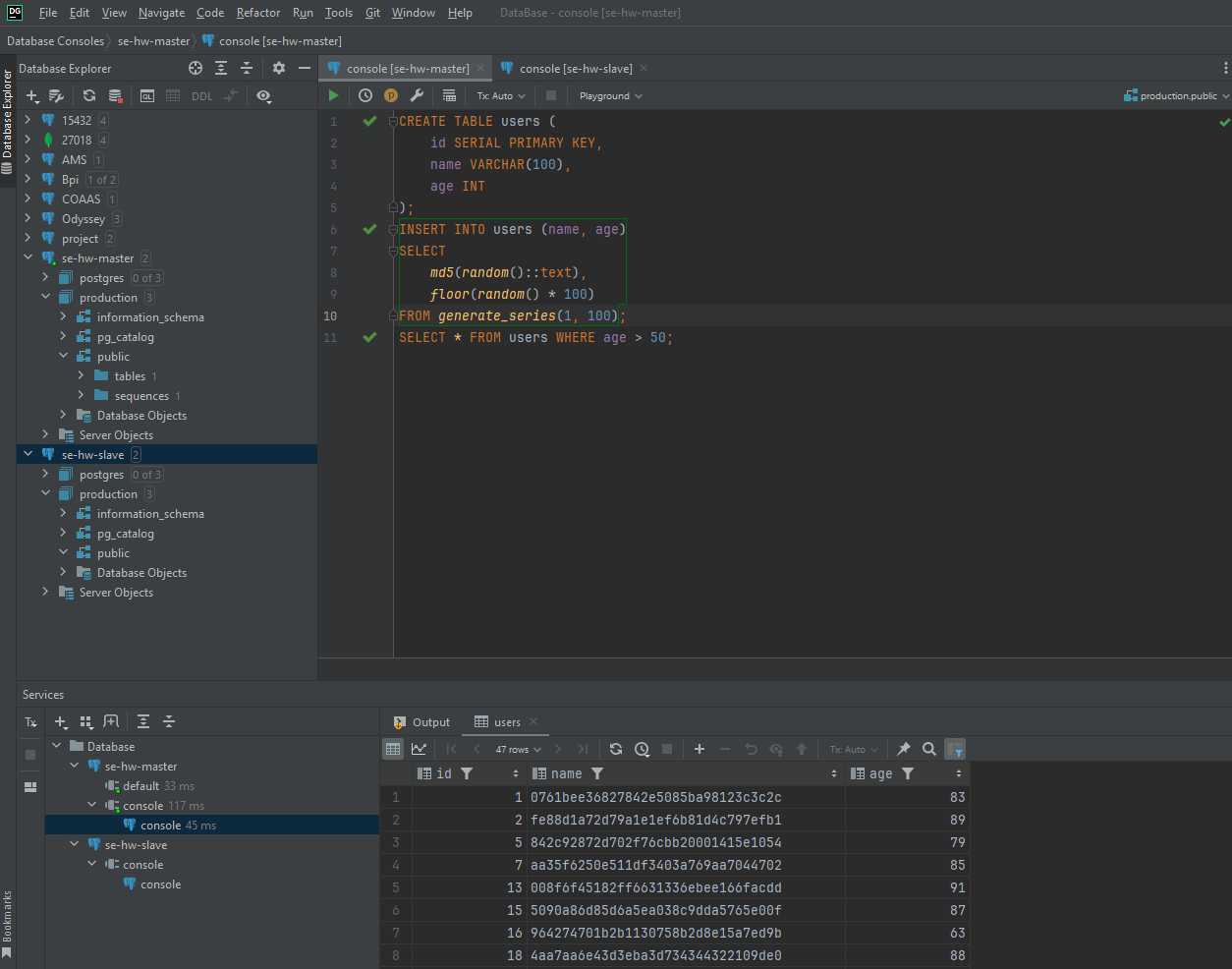
  db-network:

volumes:

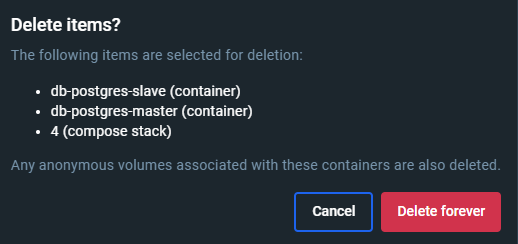
  db-postgres-master-data:

    name: db-postgres-master-data

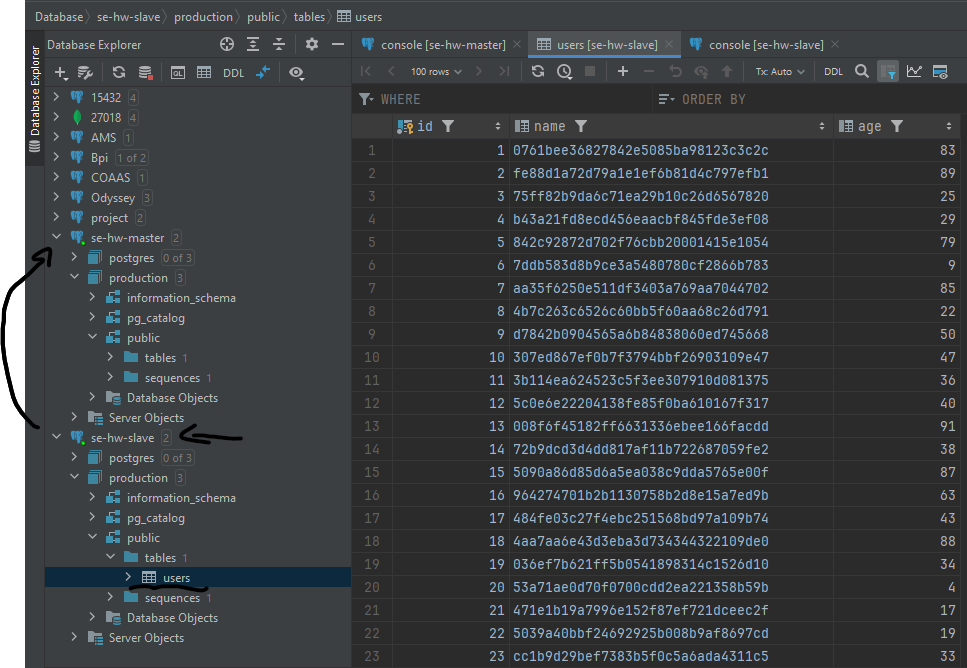
1. Загрузите туда данные
2. Подключитесь к базе данных через DataGrip
3. Выполните тестовые SQL-запросы



1. Удалите контейнеры
2. Сделайте скриншоты и опишите результаты



1. Запустите контейнеры снова и проверьте, что данные остались или нет
2. Сделайте скриншоты и опишите результаты



Так как я создал volume для master postgres, то данные остались, более того из настроенной репликации, они есть и в slave db.

**2 Задача**

В папке init-scripts есть 3 скрипта, которые создают таблицы в master:

01\_library.sql

CREATE TABLE "books" (

  "ISBN" varchar PRIMARY KEY,

  "title" varchar NOT NULL,

  "author" varchar NOT NULL,

  "year" int NOT NULL,

  "pages" int NOT NULL,

  "publisher\_id" int NOT NULL,

  "created\_at" timestamp NOT NULL

);

CREATE TABLE "categories" (

  "id" SERIAL PRIMARY KEY,

  "name" varchar UNIQUE NOT NULL,

  "parent\_category\_id" int

);

CREATE TABLE "book\_categories" (

  "book\_ISBN" varchar NOT NULL,

  "category\_id" int NOT NULL,

  PRIMARY KEY ("book\_ISBN", "category\_id")

);

CREATE TABLE "book\_copies" (

  "copy\_number" SERIAL PRIMARY KEY,

  "ISBN" varchar NOT NULL,

  "shelf\_location" varchar NOT NULL,

  "created\_at" timestamp NOT NULL

);

CREATE TABLE "publishers" (

  "id" SERIAL PRIMARY KEY,

  "name" varchar NOT NULL,

  "address" varchar NOT NULL,

  "created\_at" timestamp NOT NULL

);

CREATE TABLE "readers" (

  "id" SERIAL PRIMARY KEY,

  "first\_name" varchar NOT NULL,

  "last\_name" varchar NOT NULL,

  "address" varchar NOT NULL,

  "birth\_date" date NOT NULL,

  "created\_at" timestamp NOT NULL

);

CREATE TABLE "borrows" (

  "id" SERIAL PRIMARY KEY,

  "reader\_id" int NOT NULL,

  "copy\_number" int NOT NULL,

  "borrow\_date" timestamp NOT NULL,

  "return\_date" timestamp

);

ALTER TABLE "books" ADD FOREIGN KEY ("publisher\_id") REFERENCES "publishers" ("id");

ALTER TABLE "categories" ADD FOREIGN KEY ("parent\_category\_id") REFERENCES "categories" ("id");

ALTER TABLE "book\_categories" ADD FOREIGN KEY ("book\_ISBN") REFERENCES "books" ("ISBN");

ALTER TABLE "book\_categories" ADD FOREIGN KEY ("category\_id") REFERENCES "categories" ("id");

ALTER TABLE "book\_copies" ADD FOREIGN KEY ("ISBN") REFERENCES "books" ("ISBN");

ALTER TABLE "borrows" ADD FOREIGN KEY ("reader\_id") REFERENCES "readers" ("id");

ALTER TABLE "borrows" ADD FOREIGN KEY ("copy\_number") REFERENCES "book\_copies" ("copy\_number");

02\_stations.sql

CREATE TABLE "train\_stations" (

  "name" varchar PRIMARY KEY,

  "num\_tracks" int NOT NULL,

  "city\_name" varchar NOT NULL

);

CREATE TABLE "cities" (

  "name" varchar PRIMARY KEY,

  "region" varchar NOT NULL

);

CREATE TABLE "trains" (

  "train\_number" varchar PRIMARY KEY,

  "length" int NOT NULL

);

CREATE TABLE "journeys" (

  "id" SERIAL PRIMARY KEY,

  "train\_number" varchar NOT NULL,

  "departure\_station\_name" varchar NOT NULL,

  "arrival\_station\_name" varchar NOT NULL,

  "departure\_time" timestamp NOT NULL,

  "arrival\_time" timestamp NOT NULL

);

ALTER TABLE "train\_stations" ADD FOREIGN KEY ("city\_name") REFERENCES "cities" ("name");

ALTER TABLE "journeys" ADD FOREIGN KEY ("train\_number") REFERENCES "trains" ("train\_number");

ALTER TABLE "journeys" ADD FOREIGN KEY ("departure\_station\_name") REFERENCES "train\_stations" ("name");

ALTER TABLE "journeys" ADD FOREIGN KEY ("arrival\_station\_name") REFERENCES "train\_stations" ("name");

03\_hospital.sql

CREATE TABLE "stations" (

  "station\_number" int PRIMARY KEY,

  "name" varchar NOT NULL

);

CREATE TABLE "rooms" (

  "room\_number" int PRIMARY KEY,

  "number\_of\_beds" int NOT NULL,

  "station\_number" int NOT NULL

);

CREATE TABLE "patients" (

  "patient\_number" int PRIMARY KEY,

  "name" varchar NOT NULL,

  "disease" varchar NOT NULL,

  "doctor\_personnel\_number" int NOT NULL

);

CREATE TABLE "station\_personnel" (

  "personnel\_number" int PRIMARY KEY,

  "name" varchar NOT NULL,

  "station\_number" int NOT NULL

);

CREATE TABLE "doctors" (

  "personnel\_number" int PRIMARY KEY,

  "rank" varchar NOT NULL,

  "area" varchar NOT NULL

);

CREATE TABLE "caregivers" (

  "personnel\_number" int PRIMARY KEY,

  "qualification" varchar NOT NULL

);

CREATE TABLE "admissions" (

  "patient\_number" int NOT NULL,

  "room\_number" int NOT NULL,

  "admission\_from" timestamp NOT NULL,

  "admission\_to" timestamp,

  PRIMARY KEY ("patient\_number", "room\_number")

);

ALTER TABLE "rooms" ADD FOREIGN KEY ("station\_number") REFERENCES "stations" ("station\_number");

ALTER TABLE "patients" ADD FOREIGN KEY ("doctor\_personnel\_number") REFERENCES "doctors" ("personnel\_number");

ALTER TABLE "station\_personnel" ADD FOREIGN KEY ("station\_number") REFERENCES "stations" ("station\_number");

ALTER TABLE "doctors" ADD FOREIGN KEY ("personnel\_number") REFERENCES "station\_personnel" ("personnel\_number");

ALTER TABLE "caregivers" ADD FOREIGN KEY ("personnel\_number") REFERENCES "station\_personnel" ("personnel\_number");

ALTER TABLE "admissions" ADD FOREIGN KEY ("patient\_number") REFERENCES "patients" ("patient\_number");

ALTER TABLE "admissions" ADD FOREIGN KEY ("room\_number") REFERENCES "rooms" ("room\_number");

Конечно, стоило эти таблица создавать в своих бд или хотя бы в разных схемах:

