МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики
Мегафакультет трансляционных информационных технологий
Факультет информационных технологий и программирования

Лабораторная работа №3
По дисциплине «Web-программирование»
Создание доменной модели

Выполнили студент группы М33081
Аль Даббагх Харит Хуссейн

Проверил
Приискалов Роман Андреевич

САНКТ-ПЕТЕРБУРГ

2022

СОДЕРЖАНИЕ

DOMAIN MODEL CREATION

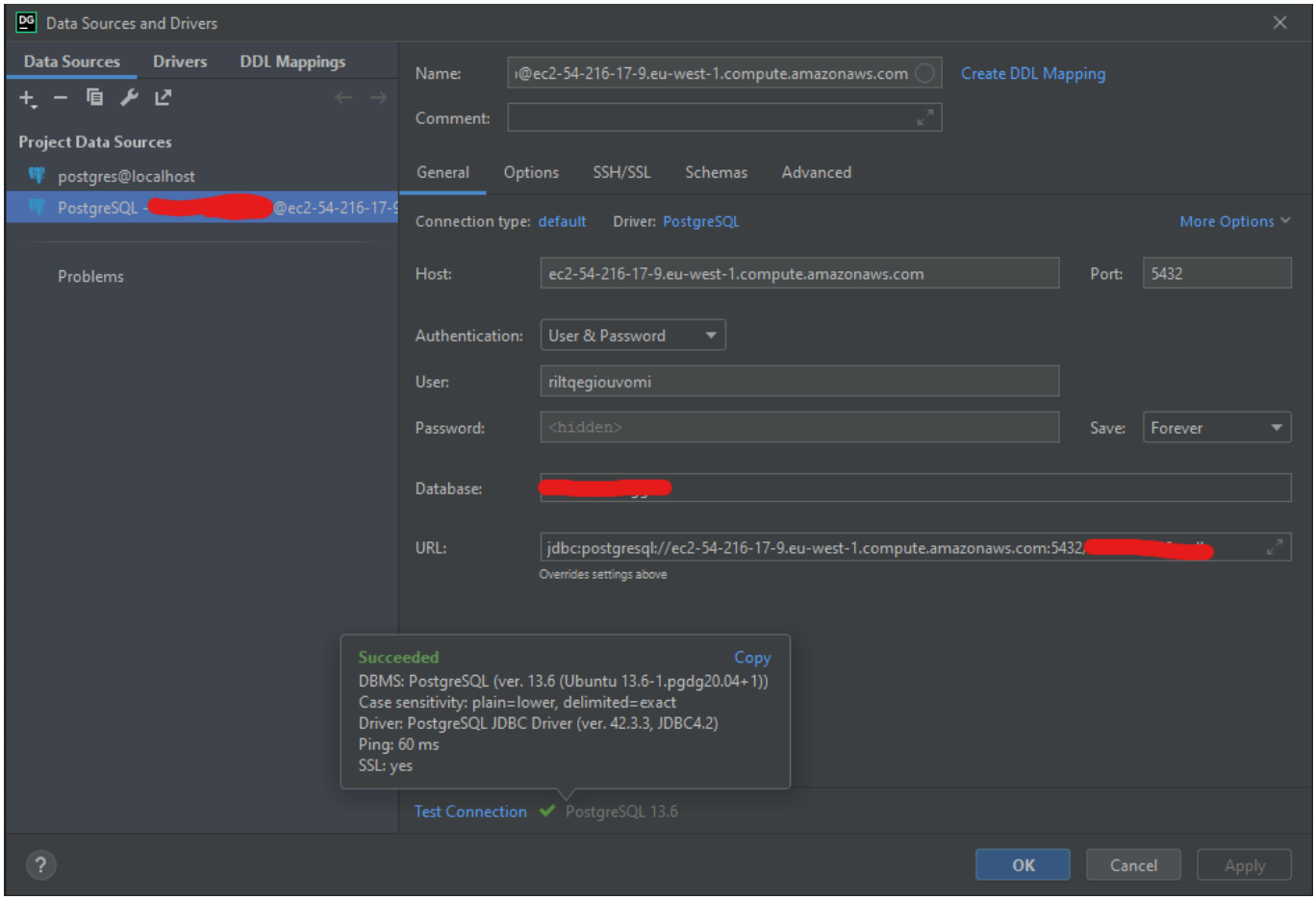
Heroku Database as a Service

In order to create a domain model we need to create a database, Heroku in terms offers a free hosted database with some limits, which should be enough for our use. To create the database we run the command:

```
heroku addons:create heroku-postgresql:hobby-dev
```

where hobby-dev is the free version of the database.

Next we can check our connection to the database using a tool like DataGrip as show below:



Now to work with PostgreSQL using node we have to install a client:

```
npm install pg --save
```

Object–relational mapping

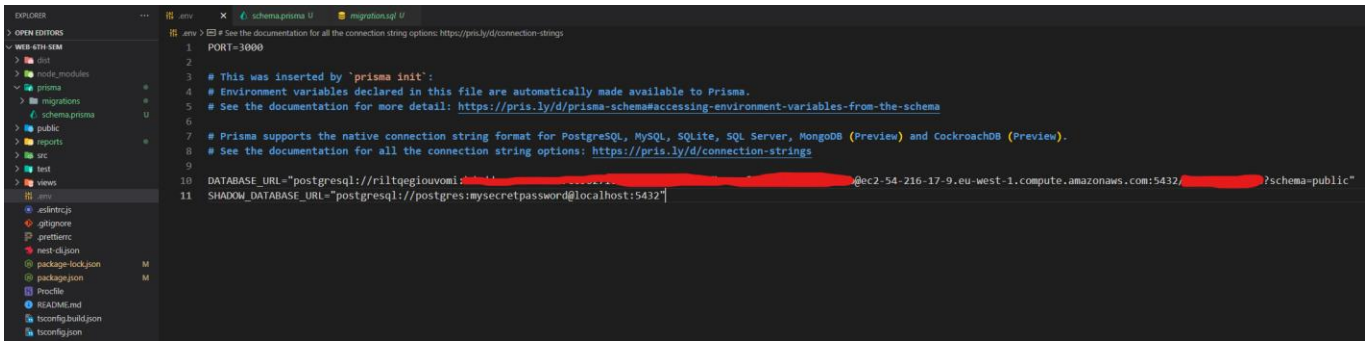
To work with database and send requests to it from our application we need to use an ORM. In this case I'm using Prisma. To install Prisma:

```
npm install prisma --save-dev
```

For the initial setup of Prisma we need to run the command:

```
npx prisma init
```

This will create a schema.prisma file and will append some parameters to the .env file



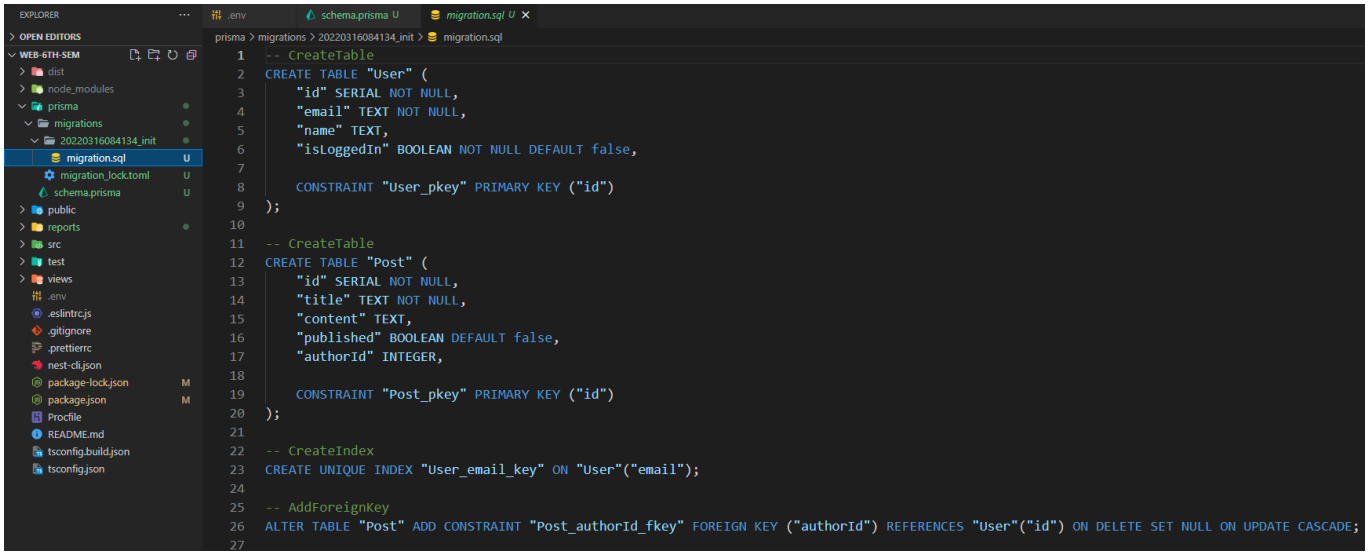
In the .env file we put our connection string to the heroku database.

Creating two database tables with Prisma Migrate

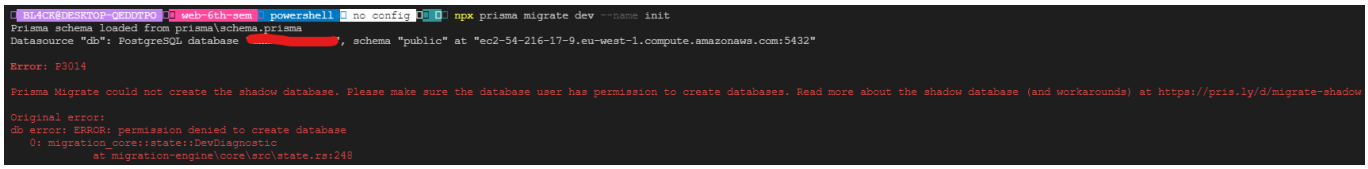
Now let's create our domain models as show in the screenshot on the right.

Now we need to generate the SQL migration files against the database:

```
npx prisma migrate dev --name init
```



Unfortunately before this step we run into a problem with working with cloud databases like this one, which is not having superuser previliges to perform tasks and create a shadow database.



To solve this problem, we need to create the shadow database manually, and the easiest way to do so is using Docker.

```
docker run --name some-postgres -e POSTGRES_PASSWORD=mysecretpassword -d -p 5432:5432 postgres
```

Keep in mind that the shadow database connection string needs to be added to the .env file.

Database tables scheme

We are also required in the lab to display the database relations in a diagram, we can do so using DataGrip for example: