

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.....	2
Heroku Database as a Service.....	2
Object–relational mapping	3
Creating two database tables with Prisma Migrate	3
Database tables scheme	4

DOMAIN MODEL CREATION

Heroku Database as a Service

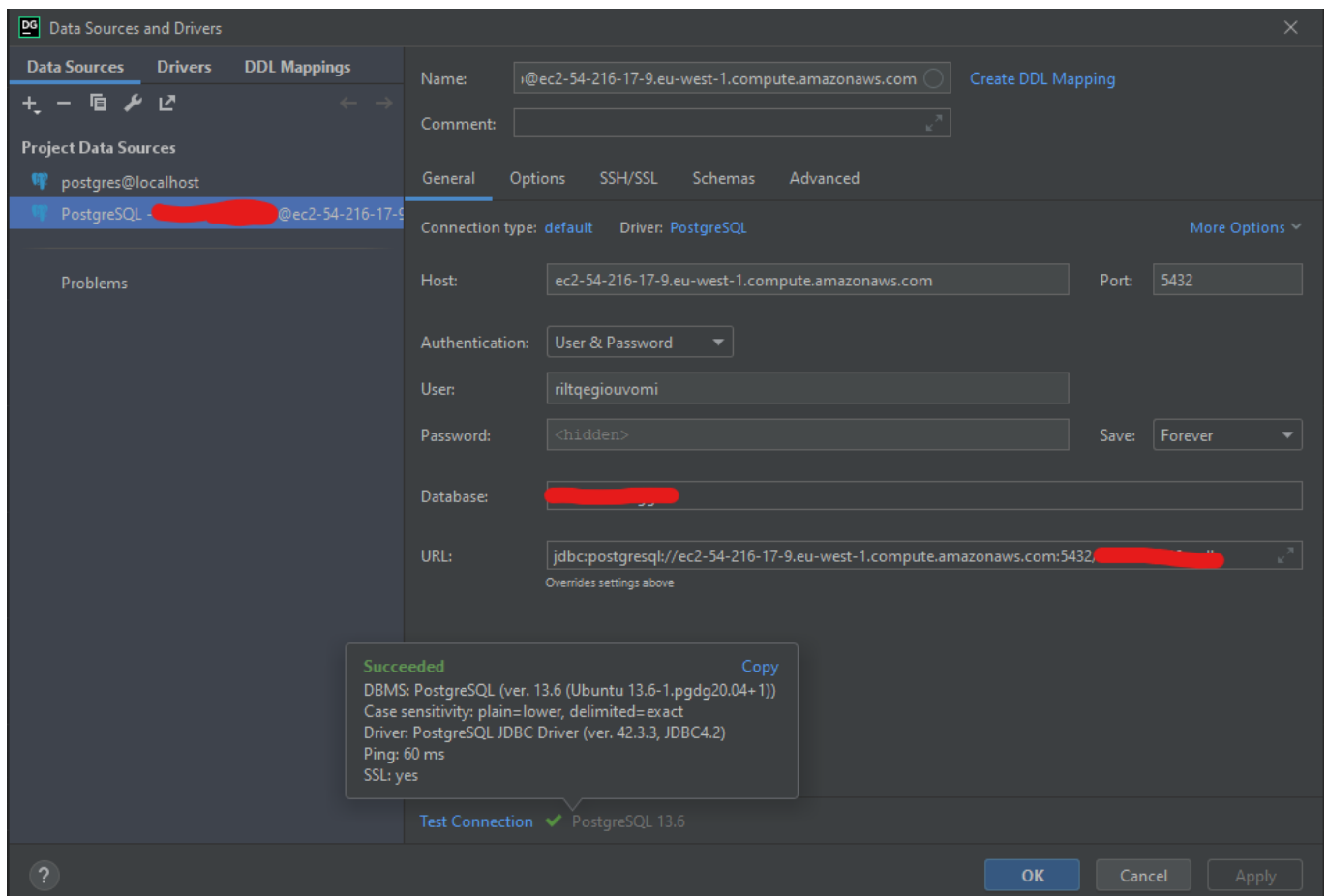
In order to create a domain model we need to create a database, Heroku in terms offers a free hosted database with some limits, which should be enough for our use. To create the database we run the command:

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

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

```
Creating heroku-postgresql:hobby-dev on web-6th-sem... free
Database has been created and is available
! This database is empty. If upgrading, you can transfer
! data from another database with pg:copy
Use heroku addons:docs heroku-postgresql to view documentation
[BLANKDESKTOP-CRUISE] web-6th-sem powershell no config [0] heroku config
Warning: heroku update available from 7.59.2 to 7.59.4.
web-6th-sem Config Vars
DATABASE_URL: postgres://riltqegiouvomi:~@ec2-54-216-17-9.eu-west-1.compute.amazonaws.com:5432/
```

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

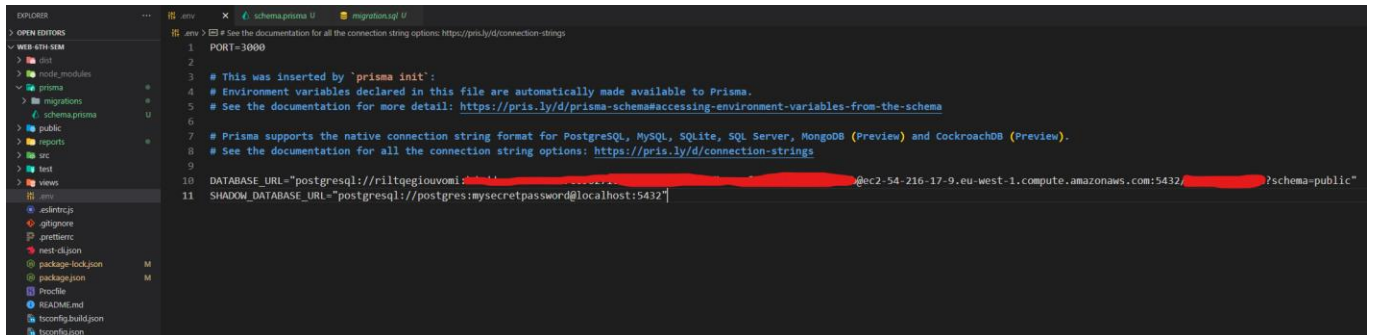
To work with database and send requests to it from our application we need to use an ORM. In this case I'm using Prisma. To install Prisma:

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

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

```
npx prisma init
```

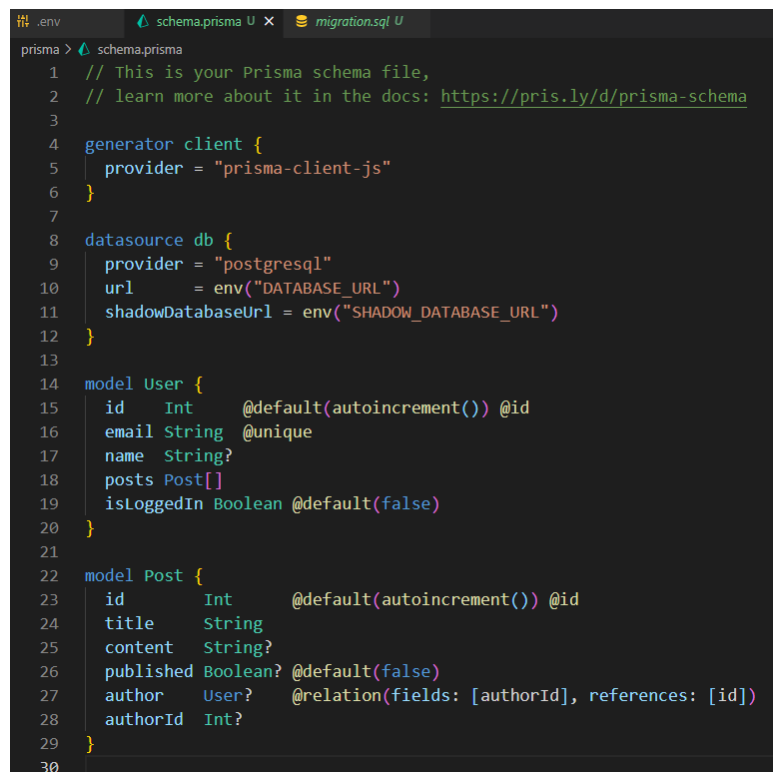
This will create a schema.prisma file and will append some parameters to the .env file



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

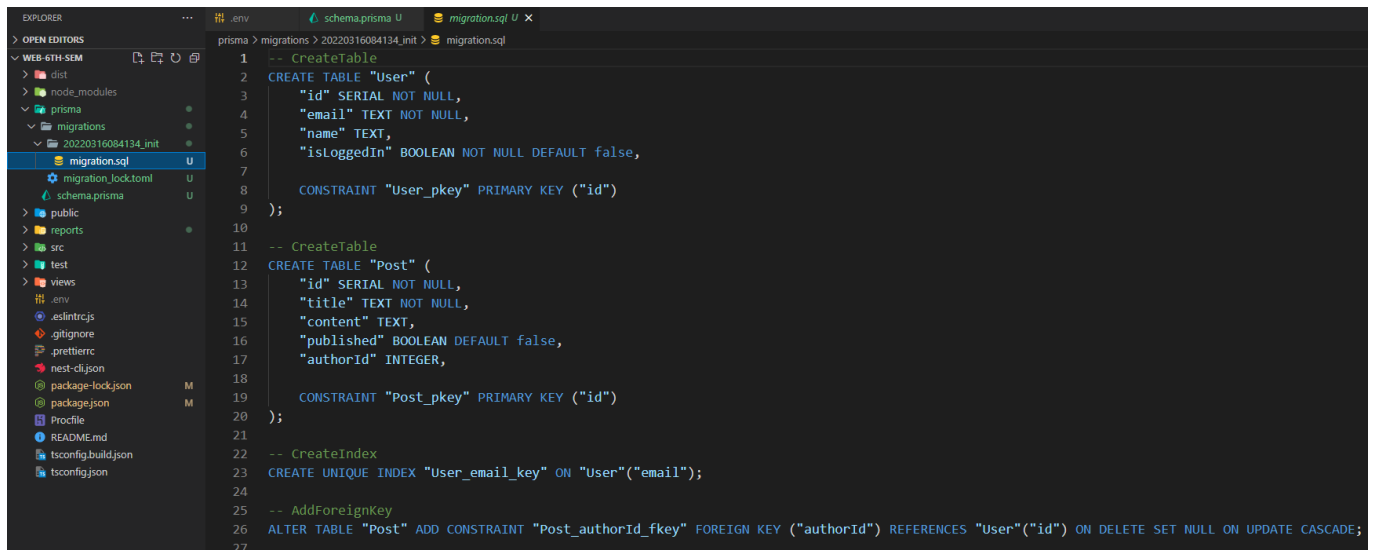
Creating two database tables with Prisma Migrate

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



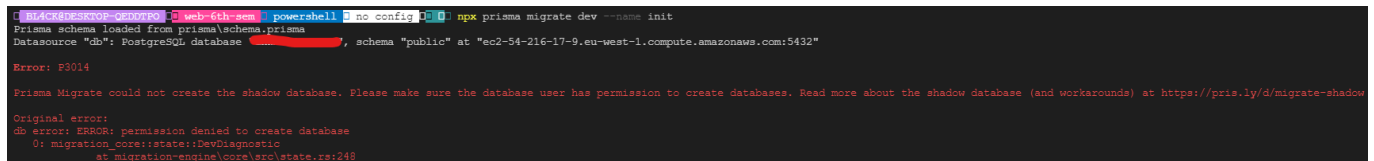
Now we need to generate the SQL migration files against the database:

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



```
1 -- CreateTable
2 CREATE TABLE "User" (
3   "id" SERIAL NOT NULL,
4   "email" TEXT NOT NULL,
5   "name" TEXT,
6   "isLoggedIn" BOOLEAN NOT NULL DEFAULT false,
7
8   CONSTRAINT "User_pkey" PRIMARY KEY ("id")
9 );
10
11 -- CreateTable
12 CREATE TABLE "Post" (
13   "id" SERIAL NOT NULL,
14   "title" TEXT NOT NULL,
15   "content" TEXT,
16   "published" BOOLEAN DEFAULT false,
17   "authorId" INTEGER,
18
19   CONSTRAINT "Post_pkey" PRIMARY KEY ("id")
20 );
21
22 -- CreateIndex
23 CREATE UNIQUE INDEX "User_email_key" ON "User"("email");
24
25 -- AddForeignKey
26 ALTER TABLE "Post" ADD CONSTRAINT "Post_authorId_fkey" FOREIGN KEY ("authorId") REFERENCES "User"("id") ON DELETE SET NULL ON UPDATE CASCADE;
```

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



```
Prisma schema loaded from prisma/schema.prisma
Datasource "db": PostgreSQL database "prisma", schema "public" at "ec2-54-216-17-9.eu-west-1.compute.amazonaws.com:5432"

Error: P3014
Prisma Migrate could not create the shadow database. Please make sure the database user has permission to create databases. Read more about the shadow database (and workarounds) at https://pris.ly/d/migrate-shadow

Original error:
db error: ERROR: permission denied to create database
    at migration-engine\core\src\state.rs:248
```

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:

