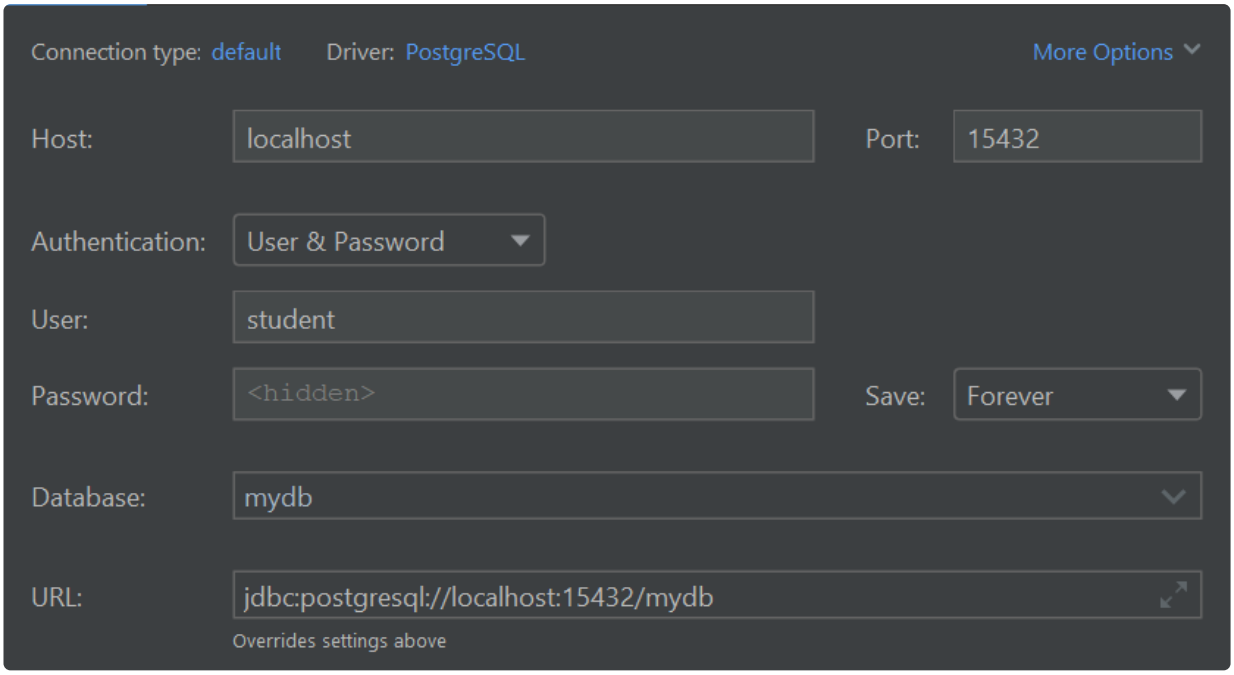
DB-HW4

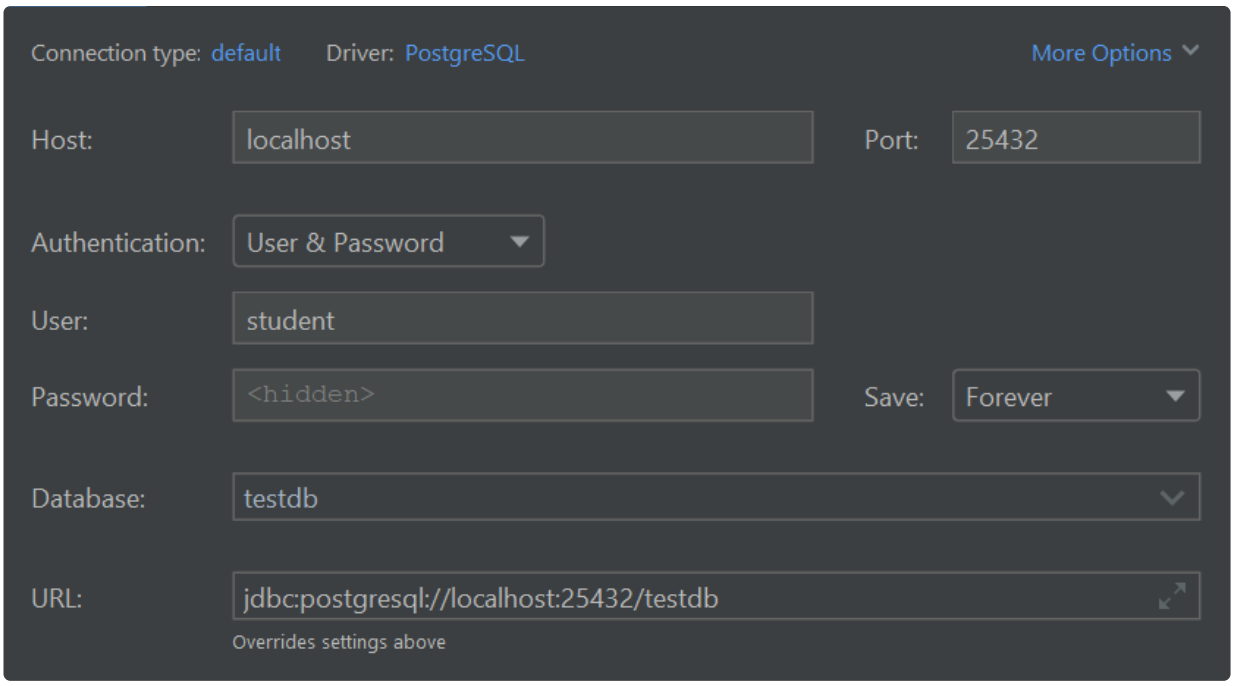
Создаем docker-compose.yml:

```yaml
services:
  postgres1:
    image: postgres:alpine
    container_name: postgres1
    environment:
      POSTGRES_USER: student
      POSTGRES_PASSWORD: student
      POSTGRES_DB: mydb
    ports:
      - "15432:5432"
  postgres2:
    image: postgres:alpine
    container_name: postgres2
    environment:
      POSTGRES_USER: student
      POSTGRES_PASSWORD: student
      POSTGRES_DB: testdb
    ports:
      - "25432:5432"
```

db_less_4		Running (2/2)	0%	
postgres2 151ffcd1464b	postgres:alp	Running	0%	25432:5432
postgres1 45614686bdf1	postgres:alp	Running	0%	15432:5432

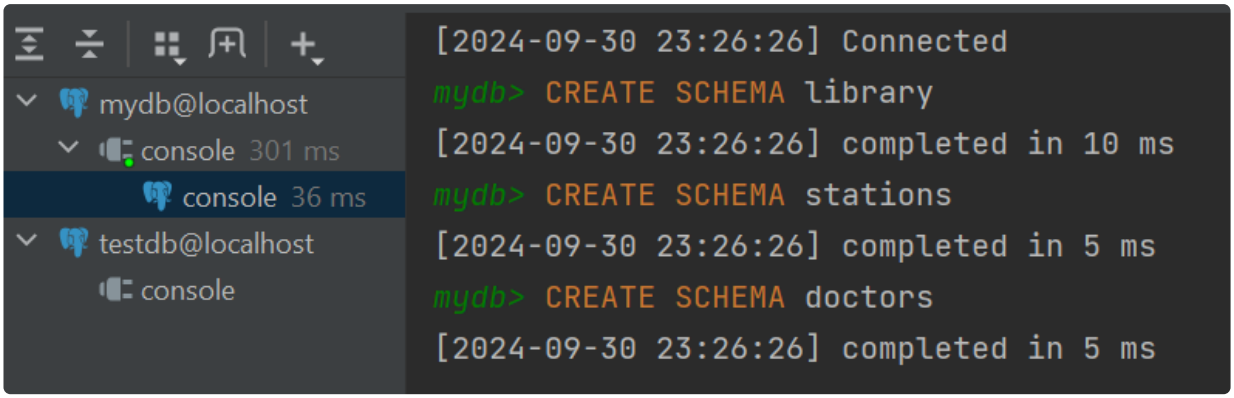
Подключимся к базе с помощью DataGrip:





Создадим отдельные схемы под наши базы данных:

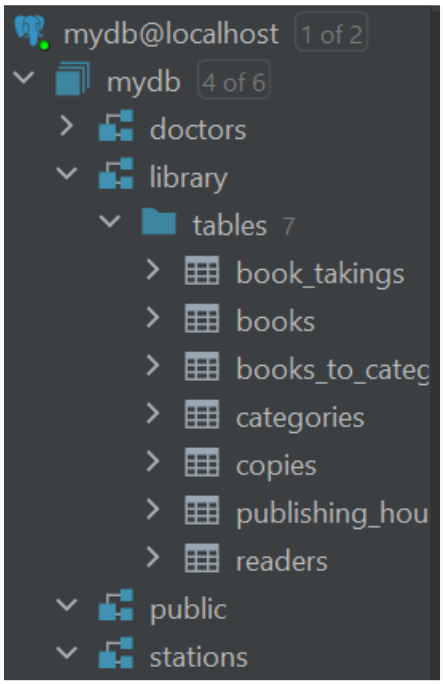
```
CREATE SCHEMA library;
CREATE SCHEMA stations;
CREATE SCHEMA doctors;
```



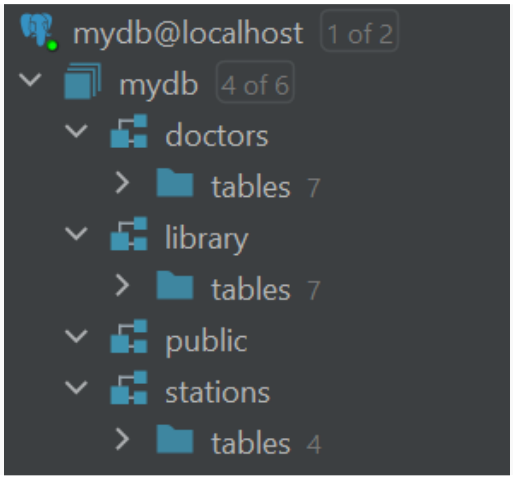
Создаем базы данных:

```
CREATE TABLE library."readers" (
  "id" integer PRIMARY KEY,
  "surname" varchar(255),
  "name" varchar(255),
  "address" varchar(255),
  "birthday" date
);

CREATE TABLE library."book_takings" (
  "reader_id" integer,
  "copy_id" integer,
  "return_date" date
);

CREATE TABLE library."copies" (
  "id" integer PRIMARY KEY,
  "book_isbn" integer,
  "position" integer
);

CREATE TABLE library."books" (
  "isbn" integer PRIMARY KEY,
  "year" integer,
  "name" varchar(255),
  "author" varchar(255),
  "number_of_pages" integer,
  "publishing_house" varchar(255),
  "categories" integer
);

CREATE TABLE library."categories" (
  "name" varchar(255) UNIQUE,
  "parent_category" varchar(255)
);

CREATE TABLE library."publishing_houses" (
  "name" varchar(255) PRIMARY KEY,
```

```
  "address" varchar(255)
);

CREATE TABLE library."books_to_categories" (
  "book_isbn" integer,
  "category_name" varchar(255),
  PRIMARY KEY ("book_isbn", "category_name")
);

ALTER TABLE library."categories" ADD FOREIGN KEY ("parent_category") REFERENCES
library."categories" ("name");

ALTER TABLE library."books_to_categories" ADD FOREIGN KEY ("book_isbn") REFERENCES
library."books" ("isbn");

ALTER TABLE library."books_to_categories" ADD FOREIGN KEY ("category_name")
REFERENCES library."categories" ("name");

ALTER TABLE library."books" ADD FOREIGN KEY ("publishing_house") REFERENCES
library."publishing_houses" ("name");

ALTER TABLE library."copies" ADD FOREIGN KEY ("book_isbn") REFERENCES
library."books" ("isbn");

ALTER TABLE library."book_takings" ADD FOREIGN KEY ("reader_id") REFERENCES
library."readers" ("id");

ALTER TABLE library."book_takings" ADD FOREIGN KEY ("copy_id") REFERENCES
library."copies" ("id");
```



Аналогично, для двух других схем базы данных.



Накидаем в таблицу со станциями каких-нибудь данных:

```
insert into stations.cities (name, region)
select
    md5(random()::text),
    md5(random()::text)
from generate_series(1, 100);
```

console [mydb@localhost] | cities [mydb@localhost]

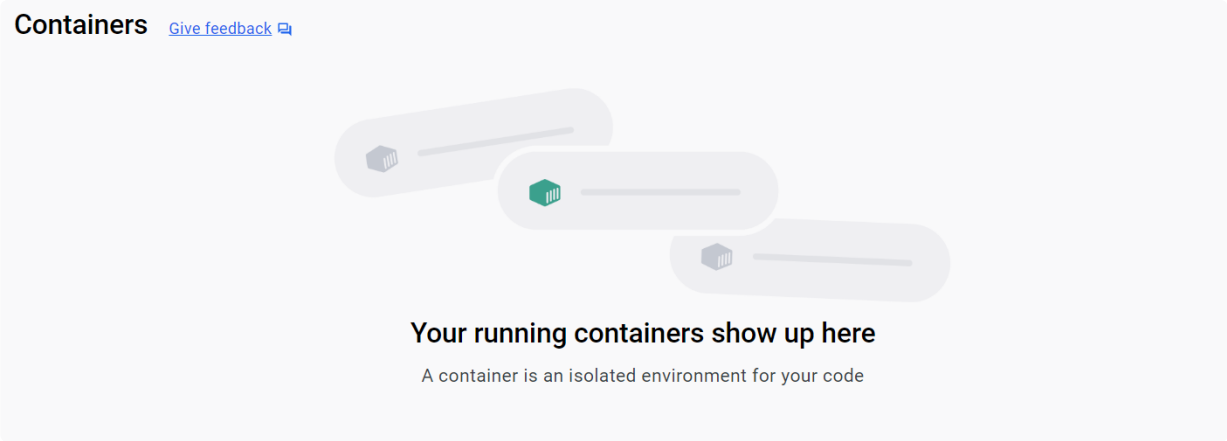
100 rows | Tx: Auto | DDL

94999596537c91214271abf3bafcfe7f | Filter rows

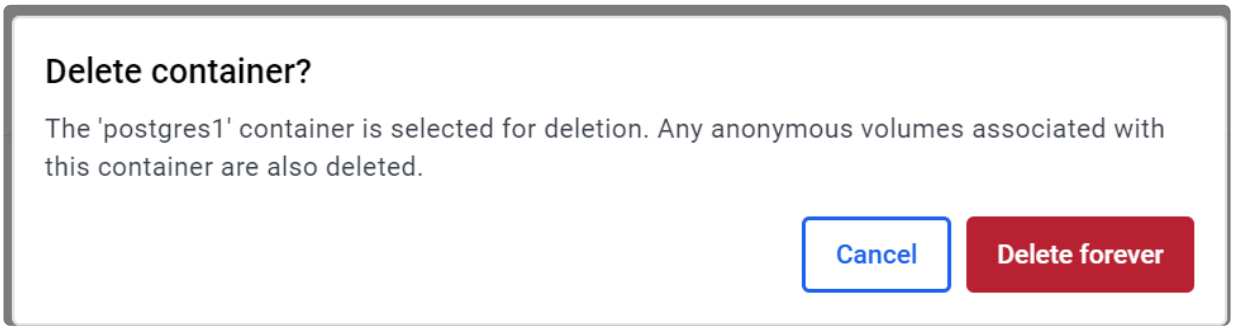
	name	region
1	44245485476e12eaf26fb89f3b2c3060	8ce224a646eb63032d5e8f046ea54ccc
2	5942947728c1cb8aa4b1b74e2ca28350	176d552c38d277a37b23b0264fd9fd15
3	fc1db71d28c43f7baf253816eafb4950	60d2174b4b248a958e4c1b6478de89e9
4	836460dabe0fb6c831e30bf25c1bc92a	fb9cc1f112ac3f6cdbd18172cda2303f
5	2a8ac78411829bbf505e33c28db2af9c	d03761c377d3a1f42464574f92ae9e76
6	859488485b21436e800c013623fa98a7	e3b7f20ea73f46c5dd9b8cbbafd9462d
7	e65e648d2f593ee80e37d5f737923f3a	046e9e937bdbaeea4b6c688f452fd545
8	4aa59bc2f16dc68abf44d3f608067952	ef62a092bc8c09d5263bdf478da4ff7a
9	25124cd8f08917f56bbcfc59587cc660	5ef7501e7900eb8746380b931e48fcef
10	687fd0ac8b95696cb582c51a6b8813a8	ad9aac89b5c74f47664fa220acd7b10e
11	ca6bc8ab3f9eb31db228174daeb35f1f	7721bc8c1a81eebde9c97ffc68fd1ae2
12	31ca12507fece1a2d818cdcdb2055772	8cdb906424c92886a5c8af6faa45a633
13	4f2a289ee31be5697176aa23f67e4baa	68ebf948e79f174dac30168d55fd49dd
14	e283c7fefa52f49051fb4710a486e4d5	ae401e3632c80abae482417c4e2c2fed
15	6c98ebfd8970e054196eb651368eb470	3570c23ab690c8651340e8b45befaa1a
16	c611867693a4be88b9d0998f0e43ae23	366ff62854e9076570a52e78fa559178
17	2beb9efa80ae9fce050cf69d66b8f9cd	5481268c8f2c5274dd66d22f79ada63f
18	4101f8c5bd4a56f8641965ce89f5615e	614362957c69bb00224100d4978ee7e9
19	72090f9b60491eb34b0ff8911fef11d9	e3c6798f9d1b597db3ffc95b192d8b84
20	afb541f1c04257d925c11b23319afc2a	5790998a3d61a7ee20610f063d04808a

Выполним

```
docker-compose down
```

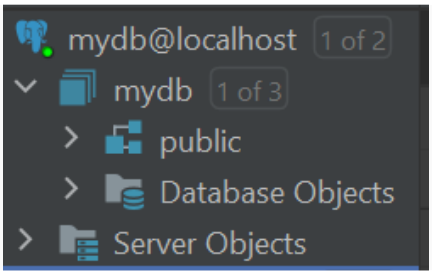


Если же просто удалить контейнер через интерфейс Docker Desktop, увидим следующее:



Так или иначе, запускаем контейнеры заново с помощью

```
docker-compose up -d
```



Данных не осталось :(

Это произошло, потому что мы не создали volume, в котором бы сохранились наши данные, а анонимные volumes, как видно из алерта в Docker Desktop, тоже удаляются.