

Национальный исследовательский ядерный
университет МИФИ

Предмет: Безопасность Баз Данных

Лабораторная работа №1
“Построение модели данных”

Выполнил:

Студент: Федоров Алексей Евгеньевич

Группа: Б20-505

Список выполненных шагов настройки

Для установки postgresql я использовал такие технологии как docker, docker-compose

- 1) Нашел образ postgresql на https://hub.docker.com/_/postgres
- 2) Создал docker-compose.yml файл

```
services:
  db:
    image: postgres
    restart: always
    environment:
      POSTGRES_PASSWORD: ullibniss
      POSTGRES_USER: ullibniss
      POSTGRES_DB: ships_management
    ports:
      - 9999:5432
    volumes:
      - "/home/ullibniss/university/db_security/database/data:/var/lib/postgresql/data"
```

POSTGRES_USER - не рутовый юзер

POSTGRES_PASSWORD - пароль для это юзера

POSTGRES_DB - название базы данных

- 3) Запустил docker-compose конфигурацию командой
docker-compose up -d

```
> docker-compose up -d --force-recreate
/usr/lib/python3/dist-packages/requests/__init__.py:89: RequestsDependencyWarning: urllib3 (1.26.14) or chardet (3.0.4) doesn't match a supported version!
  warnings.warn("urllib3 ({}) or chardet ({}) doesn't match a supported version "
/usr/lib/python3/dist-packages/paramiko/transport.py:219: CryptographyDeprecationWarning: Blowfish has been deprecated
  "class": algorithms.Blowfish,
Creating network "database_default" with the default driver
Pulling db (postgres:latest)...
latest: Pulling from library/postgres
f03b40093957: Pull complete
9d674c93414d: Pull complete
de781e8e259a: Pull complete
5ea6efaf51f6: Pull complete
b078d5f4ac82: Pull complete
97f84fb2a918: Pull complete
5a6bf2f43fb8: Pull complete
f1a40e88fea4: Pull complete
4be673794a1a: Pull complete
9d72f84fb861: Pull complete
5d52569da92e: Pull complete
5d48f8e991ff: Pull complete
4ae692d11ad3: Pull complete
Digest: sha256:31c9342603866f29206a06b77c8fed48b3c3f70d710a4be4e8216b134f92d0df
Status: Downloaded newer image for postgres:latest
Creating database_db_1 ... done
```

- 4) Подключился к базе

```
~/university/db_security/labs master ?1
psql -h 127.0.0.1 -p 9999 -U ullibniss -d ships_management -w
```

5) Загнал таблицы с данными из лабы 1-1

```
> ~ / university / db_security / labs > master ? 1  
psql -h 127.0.0.1 -p 9999 -U ullibniss -d ships_management -W < lab1-1/ports_db.sql
```

6) Прогнал скрипты из лабы 1-2

```
> psql -h 127.0.0.1 -p 9999 -U ullibniss -d ships_management -W < lab1-2/ports_db_select_queries.sql  
Password:  
      name  
-----  
Nautilus  
Valhala  
Piece of Ship  
Unsinkable  
Aquaholic  
Pirate-Hookers  
Glory Haul  
Fishsizzle  
Tip Sea  
The Leaky Barrel  
Fortuna  
Ghost in the Tacobell  
(12 rows)  
  
      name  
-----  
Unsinkable  
Pirate-Hookers  
Tip Sea  
(3 rows)  
  
 name | surname | age
```

7) Прогнал скрипты из лабы 1-3

```
> psql -h 127.0.0.1 -p 9999 -U ullibniss -d ships_management -W < lab1-3/ports_db_complex_queries.sql  
Password:  
 id | sourceid | destinationid  
----+-----+-----  
  1 |         1 |           5  
  4 |         4 |           1  
  6 |         2 |           3  
(3 rows)  
  
      name | age  
-----+-----  
Nautilus | 45  
Nautilus | 54  
Nautilus | 10  
Valhala | 90  
Valhala | 50  
Valhala | 10  
Valhala | 19  
Piece of Ship | 36  
Piece of Ship | 33  
Unsinkable | 88  
Unsinkable | 10  
Unsinkable | 23  
Unsinkable | 76  
Unsinkable | 11
```

Листинг использованных инструкций SQL

Листинги взяты из первых трех лаб

- 1) https://github.com/ullibniss/mephi-db-security-2023/blob/master/labs/lab1-1/ports_db.sql
- 2) https://github.com/ullibniss/mephi-db-security-2023/blob/master/labs/lab1-2/ports_db_select_queries.sql
- 3) https://github.com/ullibniss/mephi-db-security-2023/blob/master/labs/lab1-3/ports_db_complex_queries.sql

Заключение

В этой лабораторной работы был изучен способ развертки и настройки postgresql. Было произведен деплой, автоматическая первичная настройка, подключение к базе данных, а также мигрированы данные из прошлых лаб и прогнаны скрипты.