BIG DATA

Студент группы МПИ-22-2-1
Королев Никита

Отчет по лабораторной работе № 1

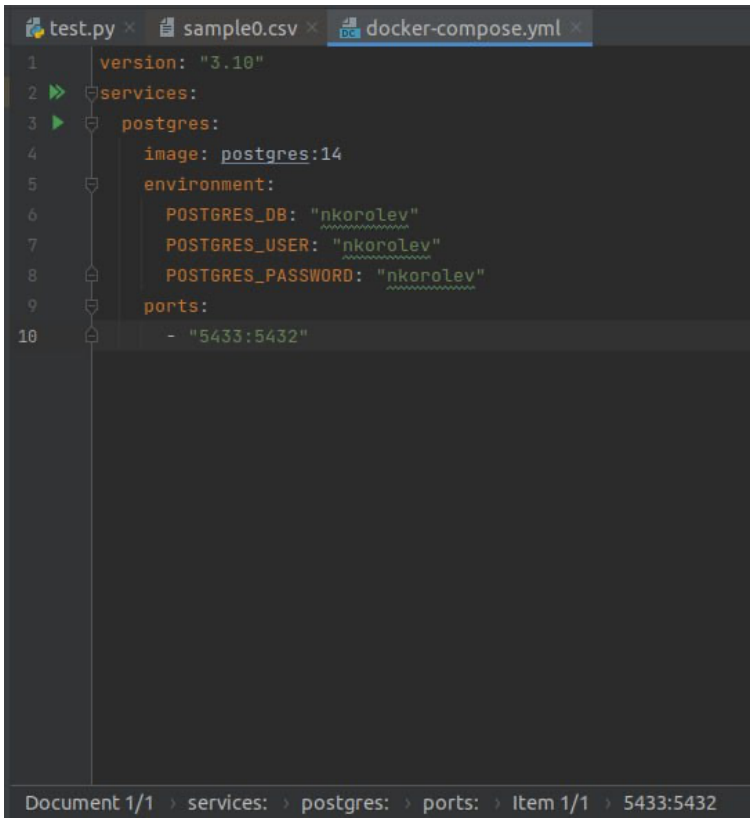

Рисунок 1 – Конфигурация файла docker-compose.yml

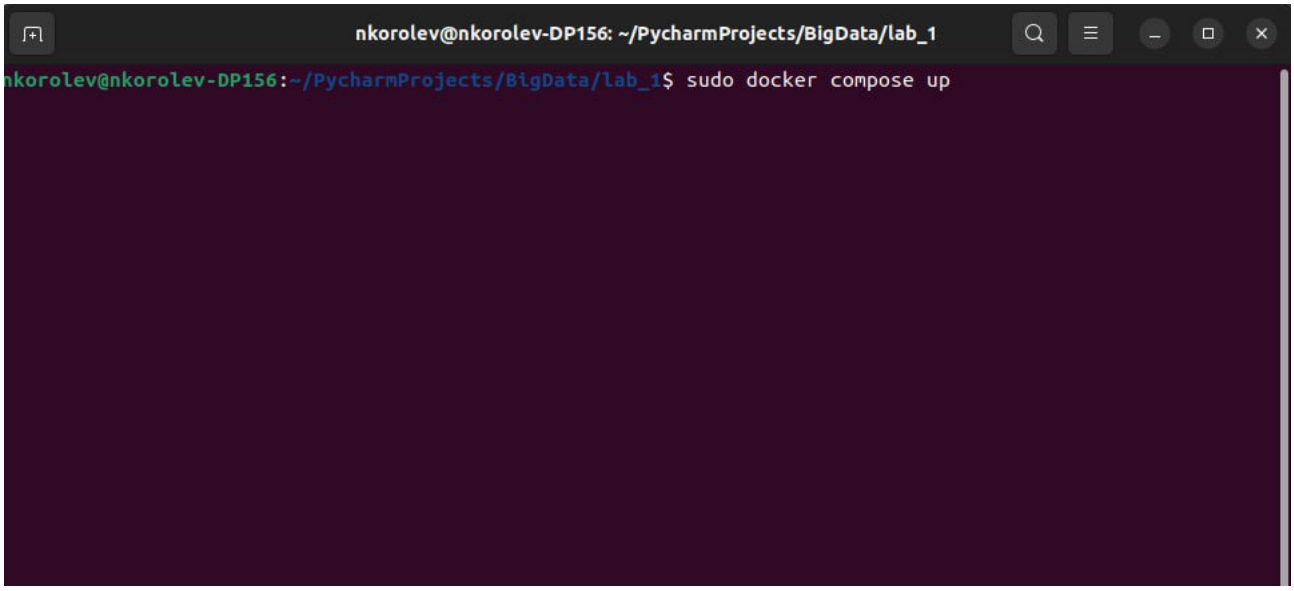

Рисунок 2 – «Поднятие» docker-compose с образом базы данных PostgreSQL

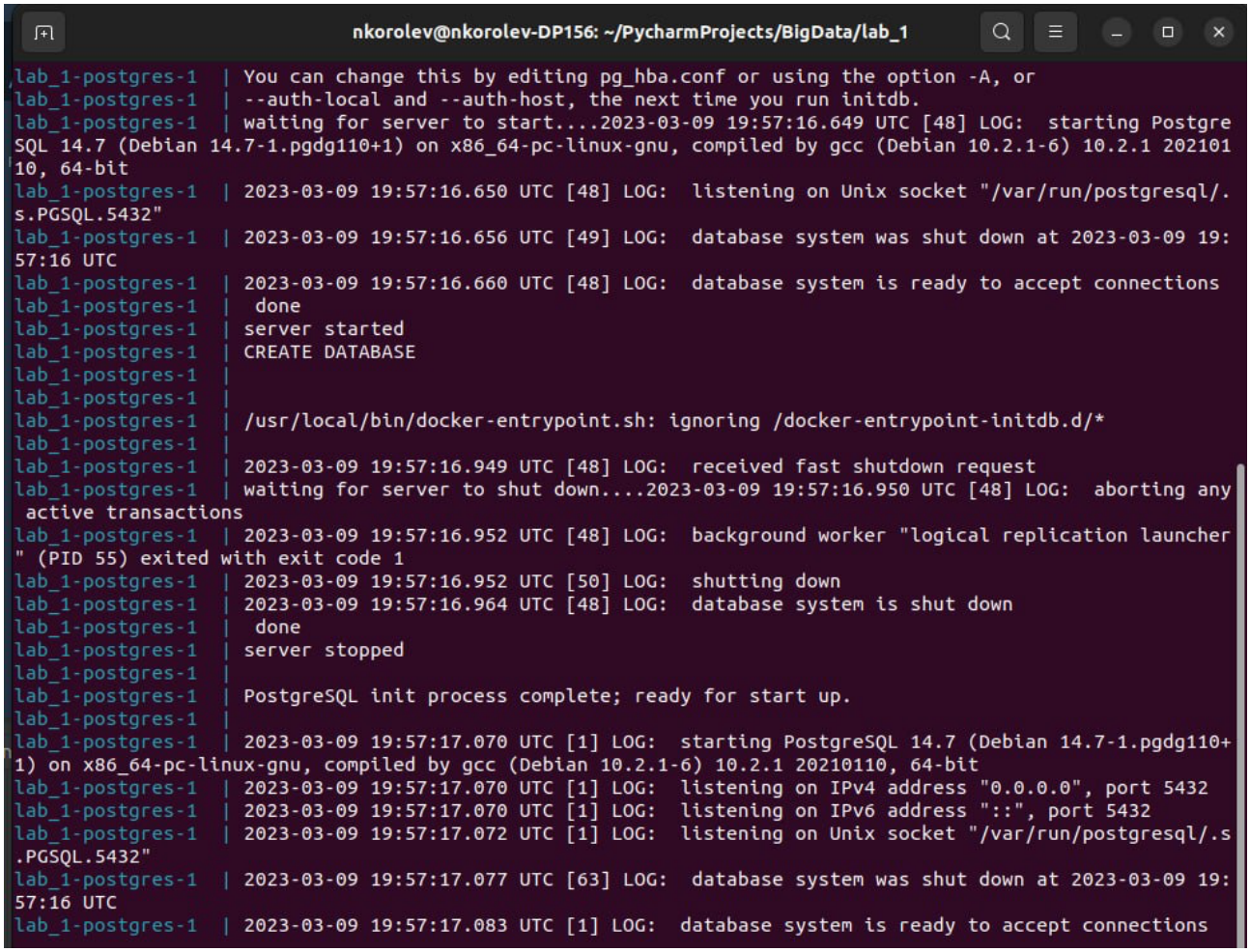

Рисунок 3 – docker-compose с образом PostgreSQL запущен

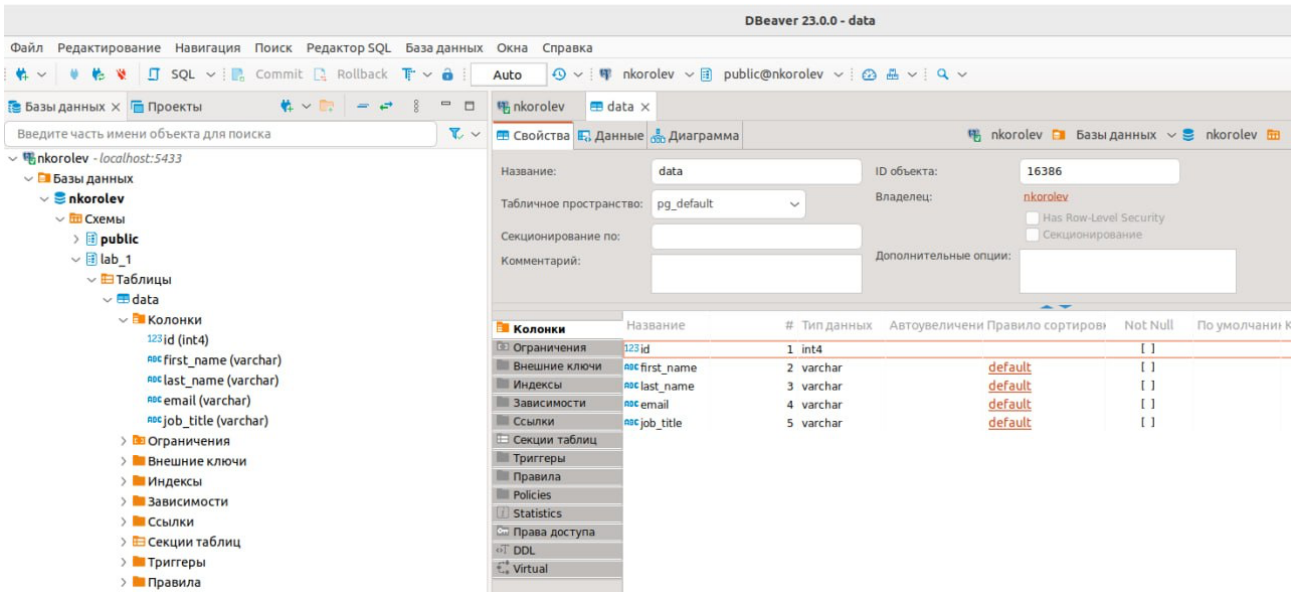

Рисунок 4 – В ПО для управления базами данных DBeaver в базе данных nkorolev создана схема lab_1 и таблица data с колонками id, first_name, last_name, email, job_title

test.py | sample0.csv | docker-compose.yml

```
index,first_name,last_name,email,job_title
1,Shelby,Terrell,elijah57@example.net,Games developer
2,Phillip,Summers,bethany14@example.com,Phytotherapist
3,Kristine,Travis,bthompson@example.com,Homeopath
4,Yesenia,Martinez,kaitlinkaiser@example.com,Market researcher
5,Lori,Todd,buchananmanuel@example.net,Veterinary surgeon
6,Erin,Day,tconner@example.org,Waste management officer
7,Katherine,Buck,conniecowan@example.com,Intelligence analyst
8,Ricardo,Hinton,wyattbishop@example.com,Hydrogeologist
9,Dave,Farrell,nmccann@example.net,Lawyer
10,Isaiah,Downs,virginiaterrell@example.org,Engineer
11,Sheila,Ross,huangcathy@example.com,Advertising account executive
12,Stacy,Newton,rayleroy@example.org,Warden/ranger
13,Mandy,Blake,jefferynoble@example.org,"Scientist, clinical (histocompatibility and immunogenetics)"
14,Bridget,Nash,mercedes44@example.com,Social worker
15,Crystal,Farmer,pmiranda@example.org,Agricultural consultant
16,Thomas,Knight,braunpriscilla@example.net,Sport and exercise psychologist
17,Maurice,Rangel,sheenabanks@example.com,Secretary/administrator
18,Frank,Meadows,gbrewer@example.org,Audiological scientist
19,Alvin,Paul,gilbertdonaldson@example.com,Teacher
20,Jared,Mitchell,jcortez@example.com,Paediatric nurse
21,Jacqueline,Norton,carias@example.net,"Scientist, marine"
```

Рисунок 5 – Файл в формате .csv, данные из которого необходимо перенести в таблицу в базе данных

test.py | sample0.csv | docker-compose.yml

```
import psycopg2


def input_values():
    id = input('Write user_id')
    first_name = input('Write first name')
    last_name = input('Write last name')
    email = input('Write email')
    job_title = input('Write job title')
    return id, first_name, last_name, email, job_title


if __name__ == "__main__":
    conn = psycopg2.connect(host='localhost', port=5433, database='nkorolev', user='nkorolev', password='nkorolev')
    cursor = conn.cursor()
    while True:
        print('1 - add row /n 2 - exit')
        status_code = input()
        if status_code == '1':
            id, first_name, last_name, email, job_title = input_values()
            sql = "INSERT INTO lab_1.data (id, first_name, last_name, email, job_title) VALUES ({}, " \
                  "'{}', '{}', '{}', '{}');".format(id, first_name, last_name, email, job_title)
            cursor.execute(sql)
        elif status_code == '2':
            break
        else:
            print('Error')
    conn.commit()
    conn.close()
```

Рисунок 6 – Python-скрипт для ручного занесения записей в таблицу в базе данных по одной строке за раз

Рисунок 7 – Внесение данных в таблицу в ручном режиме (работает скрипт, представленный на рисунке 6)

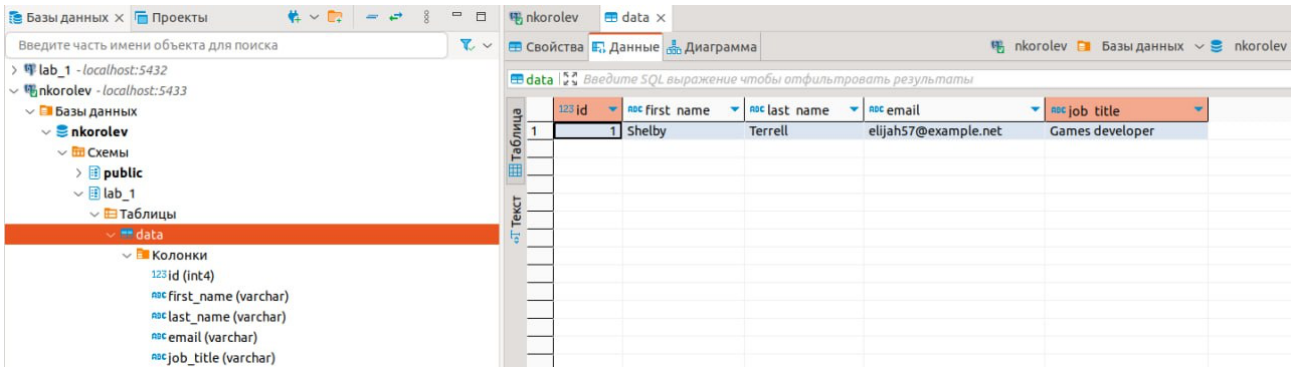

Рисунок 8 – После внесения записи вручную с помощью Python-скрипта в таблице в базе данных при обновлении появляются внесенные нами данные

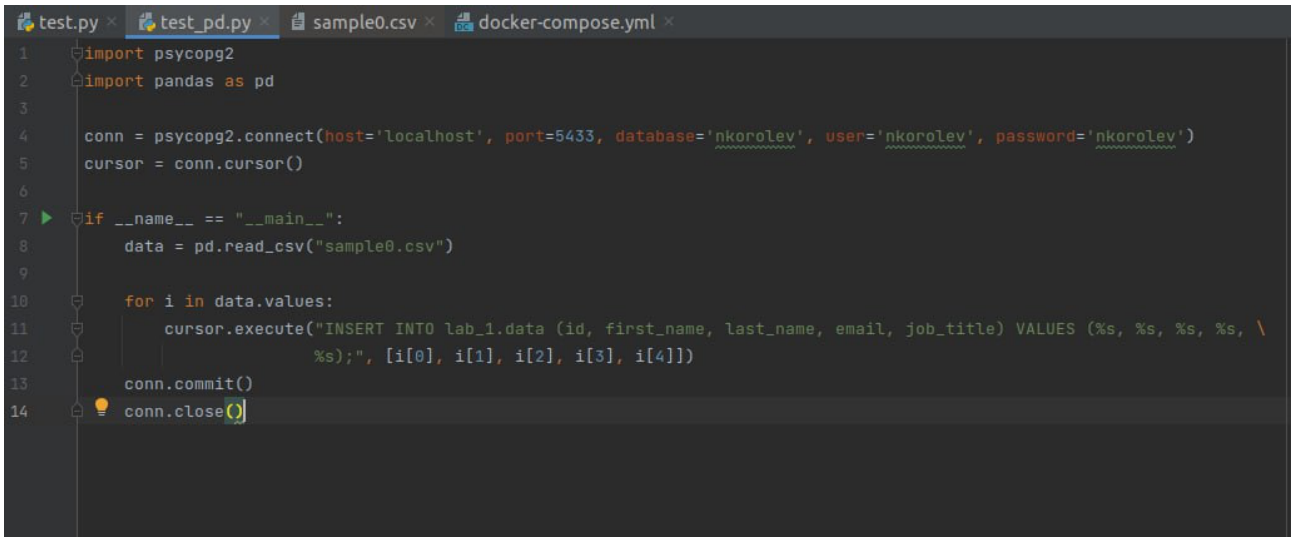

Рисунок 9 – Для ускорения внесения данных из csv файла применим библиотеку для работы с данными Pandas и напишем новый скрипт, позволяющий автоматически переносить все данные в таблицу

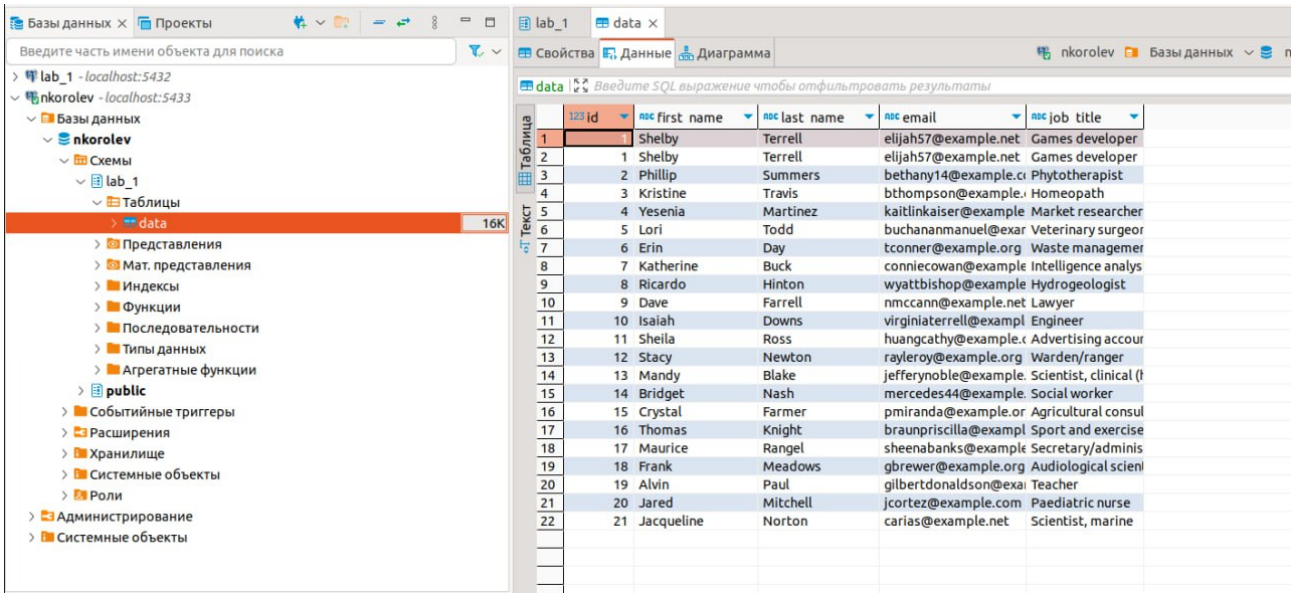

Рисунок 10 – После успешной отработки скрипта, представленного на рисунке 10 видим, что все данные, содержащиеся в csv файле (рисунок 5) были перенесены в таблицу в базе данных