

Государственное автономное образовательное учреждение города Москвы
“Московский городской педагогический университет”

Лабораторная работа №7
«Работа с внешними приложениями»

Выполнил:
Студент 2-го курса Гавриков В. И.
Группа АБП-231

Проверил:
Доцент департамента информатики
Босенко Т.М.

Москва
2025

Цель работы:

Научиться импортировать и экспортировать данные в базу данных SQL, работать с внешними данными, преобразовывать их в нужный формат и интегрировать с существующими таблицами в базе данных. Работа включает в себя загрузку данных из внешних источников в таблицы базы данных, а также экспорт данных из базы данных в различные форматы.

Примечание:

Каждое задание выполнялось в отдельном блоке. При выполнении каждого задания в начале выполнялось подключение к БД, затем в конце блока соединение закрывалось. Erd-диаграмма сделана на сайте <https://dbdiagram.io/>. Также для выполнения заданий использованы приложения pgAdmin 4 и Visual Studio Code.

Ход работы, результаты выполнения блоков (вариант st_06):

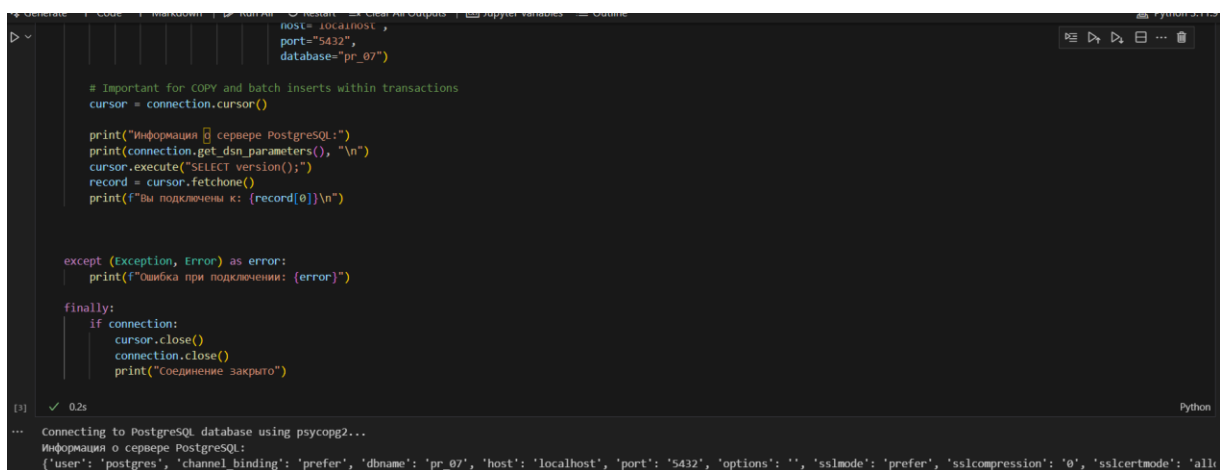
Для начала была создана база данных "pr_07" для выполнения работы. Далее через Visual Studio подключился к этой базе данных, используя psycopg2 (для проверки выведена информация о сервере PostgreSQL). Потом, используя Python, создал таблицы с данными hospital, doctor, и сразу же их заполнил.



```
[1] ✓ 4.5s Python
... Collecting psycopg2
  Using cached psycopg2-2.9.10-cp311-cp311-win_amd64.whl.metadata (5.0 kB)
  Using cached psycopg2-2.9.10-cp311-cp311-win_amd64.whl (1.2 MB)
  Installing collected packages: psycopg2
  Successfully installed psycopg2-2.9.10

[notice] A new release of pip is available: 24.0 -> 25.1.1
[notice] To update, run: C:\Users\gavri\AppData\Local\Microsoft\WindowsApps\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\python.exe -m pip install --upgrade pip

D >
... import psycopg2
... from psycopg2 import Error
[2] ✓ 0.2s Python
```



```
host='localhost',
port='5432',
database='pr_07')

# Important for COPY and batch inserts within transactions
cursor = connection.cursor()

print("Информация о сервере PostgreSQL:")
print(connection.get_dsn_parameters(), "\n")
cursor.execute("SELECT version();")
record = cursor.fetchone()
print(f"Вы подключены к: {record[0]}\n")

except (Exception, Error) as error:
    print(f"Ошибка при подключении: {error}")

finally:
    if connection:
        cursor.close()
        connection.close()
        print("Соединение закрыто")

[3] ✓ 0.2s Python
... Connecting to PostgreSQL database using psycopg2...
Информация о сервере PostgreSQL:
{'user': 'postgres', 'channel_binding': 'prefer', 'dbname': 'pr_07', 'host': 'localhost', 'port': '5432', 'options': '', 'sslmode': 'prefer', 'sslcompression': '0', 'sslcrtmode': 'all'
```

```

cursor = connection.cursor()

# Создание таблицы Hospital
create_table_query = '''
CREATE TABLE Hospital (
    Hospital_Id serial NOT NULL PRIMARY KEY,
    Hospital_Name VARCHAR(100) NOT NULL,
    Bed_Count serial
);
'''

cursor.execute(create_table_query)
connection.commit()
print("Таблица 'Hospital' успешно создана")

# Вставка данных в таблицу Hospital
insert_query = '''
INSERT INTO Hospital (Hospital_Id, Hospital_Name, Bed_Count)
VALUES
(1, 'Mayo Clinic', 200),
(2, 'Cleveland Clinic', 400),
(3, 'Johns Hopkins', 1000),
(4, 'UCLA Medical Center', 1500);
'''

cursor.execute(insert_query)
connection.commit()
print("Данные успешно вставлены в таблицу 'Hospital'")

except (Exception, psycopg2.Error) as error:
    print("Ошибка при подключении или работе с PostgreSQL:", error)

# Закрытие подключения к базе данных
finally:

```

```

# Создание таблицы Doctor
create_table_query = '''
CREATE TABLE Doctor (
    Doctor_Id serial NOT NULL PRIMARY KEY,
    Doctor_Name VARCHAR (100) NOT NULL,
    Hospital_Id serial NOT NULL,
    Joining_Date DATE NOT NULL,
    Speciality VARCHAR (100) NOT NULL,
    Salary INTEGER NOT NULL,
    Experience SMALLINT
);'''
cursor.execute(create_table_query)
connection.commit()
print("Таблица успешно создана")

# Вставка данных в таблицу Doctor
insert_query = '''
INSERT INTO Doctor (Doctor_Id, Doctor_Name, Hospital_Id, Joining_Date, Speciality, Salary, Experience)
VALUES
('101', 'David', '1', '2005-02-10', 'Pediatric', 40000, NULL),
('102', 'Michael', '1', '2018-07-23', 'Oncologist', 20000, NULL),
('103', 'Susan', '2', '2016-05-19', 'Gynecologist', 25000, NULL),
('104', 'Robert', '2', '2017-12-28', 'Pediatric', 28000, NULL),
('105', 'Linda', '3', '2004-06-04', 'Gynecologist', 42000, NULL),
('106', 'William', '3', '2012-09-11', 'Dermatologist', 30000, NULL),
('107', 'Richard', '4', '2014-08-21', 'Gynecologist', 32000, NULL),
('108', 'Karen', '4', '2011-10-17', 'Radiologist', 30000, NULL),
('109', 'James', '1', '2022-01-15', 'Cardiologist', 45000, 5),
('110', 'Emily', '1', '2023-04-10', 'Orthopedic Surgeon', 50000, 3),
('111', 'Olivia', '2', '2021-09-05', 'Neurologist', 42000, 4),
('112', 'John', '2', '2024-02-18', 'Surgeon', 60000, 2),
('113', 'Sophia', '3', '2022-07-30', 'Urologist', 38000, 6),
('114', 'Daniel', '3', '2025-03-22', 'Pulmonologist', 47000, 1),
'''

```

Индивидуальное задание 1: Создайте таблицу "appointments" с полями "ID", "DOCTOR_ID", "DATE".

При выполнении блока создается таблица "appointments". В задании не сказано ее заполнять, поэтому остается пустой.

```
#Задание 1. Создайте таблицу "appointments" с полями "ID", "DOCTOR_ID", "DATE".
try:
    # Подключение к базе данных с помощью psycopg2
    print("Connecting to PostgreSQL database using psycopg2...")
    connection = psycopg2.connect(user="postgres",
                                   password="gvi20052005",
                                   host="localhost",
                                   port="5432",
                                   database="pr_07")

    cursor = connection.cursor()

    #Запрос для создания таблицы
    create_table_query = '''CREATE TABLE appointments
    (ID INT PRIMARY KEY NOT NULL,
    DOCTOR_ID INT NOT NULL,
    DATE DATE);'''
    cursor.execute(create_table_query)
    connection.commit()
    print("Таблица успешно создана в PostgreSQL")

except (Exception, Error) as error:
    print(f"Ошибка при подключении: {error}")

finally:
    if connection:
        cursor.close()
        connection.close()
        print("Соединение закрыто")
```

Индивидуальное задание 2: Выполните SQL-запрос для обновления зарплаты врача с ID=105.

При выполнении блока обновляется зарплата врача с идентификатором №105. Была установлена сумма 50000.

```
#Задание 2. Выполните SQL-запрос для обновления зарплаты врача с ID=105.
try:
    # Подключение к базе данных с помощью psycopg2
    print("Connecting to PostgreSQL database using psycopg2...")
    connection = psycopg2.connect(user="postgres",
                                   password="gvi20052005",
                                   host="localhost",
                                   port="5432",
                                   database="pr_07")

    cursor = connection.cursor()

    #Обновление зарплаты доктора с ID=105
    update_salary_query = '''UPDATE doctor
    SET salary = 50000
    WHERE doctor_id = 105;
    '''
    cursor.execute(update_salary_query)
    connection.commit()
    print("Зарплата ID=105 успешно изменена")

except (Exception, psycopg2.Error) as error:
    print("Ошибка при подключении или работе PostgreSQL:", error)

# Закрытие подключения к базе данных
finally:
    if connection:
        cursor.close()
        connection.close()
        print("Соединение закрыто")
```

Индивидуальное задание 3: Получите все записи о больнице с ID=3.

Задание было выполнено с использованием создания функций def.

команда records = cursor.fetchall() возвращает все записи в python после

выполнения команды select_query = "SELECT

* from hospital WHERE hospital_id = 3". С помощью команды row

выгружаются данные в список.

```
#Задание 3. Получите все записи о больнице с ID=3.
#Функция подключения
def get_connection(database_name):

    print("Connecting to PostgreSQL database using psycopg2...")
    connection = psycopg2.connect(user="postgres",
                                   password="gvi20052005",
                                   host="localhost",
                                   port="5432",
                                   database=database_name)

    return connection

#Функция закрытия подключения
def close_connection():
    if connection:
        connection.close()
        print("Соединение закрыто")

database_name = 'pr_07'
#Функция по поиску записей о больнице с id=3
```

```
#Функция по поиску записей о больнице с id=3
def hospital_num3():
    try:
        connection = get_connection(database_name)
        cursor = connection.cursor()

        #sql-запрос по поиску записей
        sql_select = ''' SELECT * from hospital WHERE hospital_id = 3
        ...

        cursor.execute(sql_select)
        records = cursor.fetchall()
        print(f"Записи о больнице с ID=3:")

        for row in records:
            print(f"Идентификатор больницы: {row[0]}")
            print(f"Наименование: {row[1]}")
            print(f"Количество мест: {row[2]}")

    except (Exception, psycopg2.Error) as error:
        print("Ошибка при подключении или работе PostgreSQL:", error)

#Вывод результата и закрытие соединения
hospital_num3()
close_connection()
```

```
Connecting to PostgreSQL database using psycopg2...
Записи о больнице с ID=3:
Идентификатор больницы: 3
Наименование: Johns Hopkins
Количество мест: 1000
Соединение закрыто
```

Индивидуальное задание 4: Обновите стаж врача с ID=106

Задание выполнено аналогично заданию 2. Для id 106 был установлен стаж 4 года.

```
#Обновите стаж врача с ID=106.
try:
    # Подключение к базе данных с помощью psycopg2
    print("Connecting to PostgreSQL database using psycopg2...")
    connection = psycopg2.connect(user="postgres",
                                   password="gvi20052005",
                                   host="localhost",
                                   port="5432",
                                   database="pr_07")

    cursor = connection.cursor()

    #Обновление зарплаты доктора с ID=105
    update_salary_query = '''UPDATE doctor
SET experience = 4
WHERE doctor_id = 106;
'''

    cursor.execute(update_salary_query)
    connection.commit()
    print("Стаж у ID=106 успешно изменен")

except (Exception, psycopg2.Error) as error:
    print("Ошибка при подключении или работе PostgreSQL:", error)

# Закрытие подключения к базе данных

# Закрытие подключения к базе данных
finally:
    if connection:
        cursor.close()
        connection.close()
        print("Соединение закрыто")
```

```
Connecting to PostgreSQL database using psycopg2...
Стаж у ID=106 успешно изменен
Соединение закрыто
```

Индивидуальное задание 5: Постройте график зависимости зарплаты от опыта врачей.

Для построения графика обращаемся к библиотекам `psycopg2`, `pandas`, `matplotlib.pyplot`. Создаем функцию закрытия соединения с БД. При выполнении запроса с помощью команды `""" SELECT Salary, Experience FROM Doctor WHERE Experience IS NOT NULL; """` выгружаются необходимые данные, которые после загружаются в `dataframe` командой `df = pd.read_sql(select_query, connection)`. Далее по заданным параметрам выстраивается график зависимости зарплаты от опыта врачей.

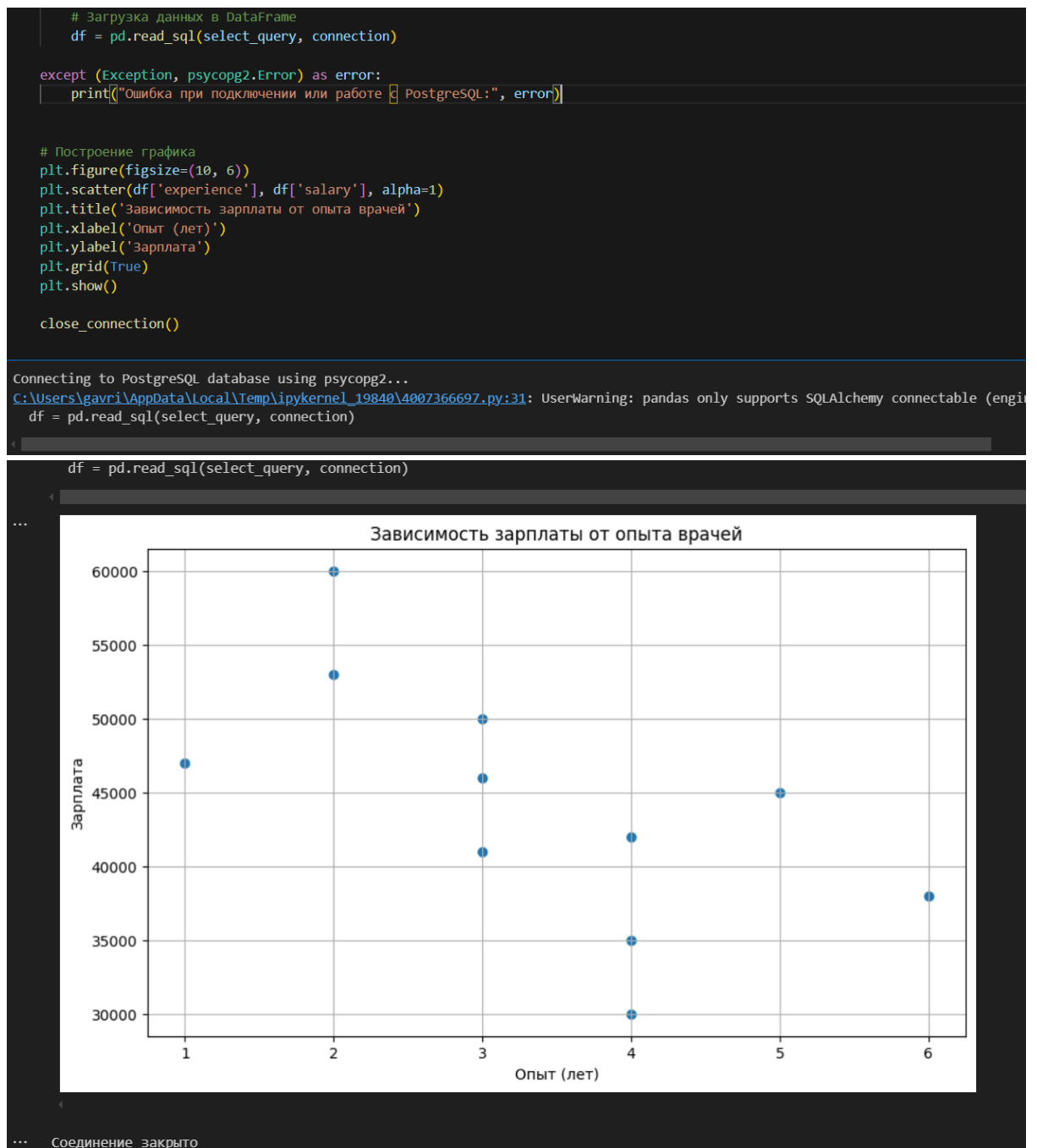
```
#Задание 5. Постройте график зависимости зарплаты от опыта врачей.
import psycopg2
import pandas as pd
import matplotlib.pyplot as plt

#Функция закрытия соединения
def close_connection():
    if connection:
        connection.close()
        print("Соединение закрыто")
# Подключение к PostgreSQL
try:
    # Подключение к базе данных с помощью psycopg2
    print("Connecting to PostgreSQL database using psycopg2...")
    connection = psycopg2.connect(user="postgres",
                                   password="gvi20052005",
                                   host="localhost",
                                   port="5432",
                                   database="pr_07")

    cursor = connection.cursor()

    # SQL-запрос для извлечения данных
    select_query = """
    SELECT Salary, Experience
    FROM Doctor
    WHERE Experience IS NOT NULL;
    """

    cursor.execute(select_query)
    connection.commit()
    # Загрузка данных в DataFrame
```



Выводы:

Научился импортировать и экспортировать данные в базу данных SQL, работать с внешними данными, преобразовывать их в нужный формат и интегрировать с существующими таблицами в базе данных