МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«МИРЭА – РОССИЙСКИЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»
РТУ МИРЭА

Институт Институт кибербезопасности и цифровых технологий

Специальность (направление): 09.03.02 Информационные системы и технологии

Кафедра: КБ-3 «Разработка программных решений и системного программирования»

Дисциплина: «Технологии и инструменты систем управления данными»

Практическая работа
на тему:
Лабораторная работа 1-2

Студент: _____ 18.02.2025 Крашенинников М.В.
подпись *дата* *инициалы и фамилия*

Группа: БСБО-16-23 Шифр: _____

Преподаватель: _____ 18.02.2025 Нурматова Е.В.
подпись *дата* *инициалы и фамилия*

Москва 2024 г.

Инициализация БД:

```sql
-- Создание базы данных
CREATE DATABASE study;

-- Подключение к базе данных
\c study;

-- Создание таблицы students
CREATE TABLE students (
    ids INTEGER PRIMARY KEY,
    name_st VARCHAR(100) NOT NULL,
    nomk INTEGER CHECK (nomk BETWEEN 1 AND 8) NOT NULL
);

-- Создание таблицы disciplines
CREATE TABLE disciplines (
    id SERIAL PRIMARY KEY,
    name_dis VARCHAR(100) NOT NULL,
    dweek INTEGER CHECK (dweek BETWEEN 1 AND 7) NOT NULL,
    nomp INTEGER CHECK (nomp BETWEEN 1 AND 8) NOT NULL,
    nomk INTEGER CHECK (nomk BETWEEN 1 AND 8) NOT NULL,
    UNIQUE (dweek, nomp, nomk)
);

-- Копирование данных из CSV-файла в таблицу students
COPY students(ids, name_st, nomk)
FROM '/mnt/c/Users/boltf/OneDrive/Рабочий стол/nurmatova/1/students.csv'
DELIMITER ';'
CSV;
```

API:

```
import psycopg2
from psycopg2 import Error
import shlex


def connect_to_db():
    return psycopg2.connect(
        user="postgres",
        password="postgres",
        host="127.0.0.1",
        port="5432",
        database="study"
    )


def get_student(student_id):
    try:
        with connect_to_db() as conn:
            with conn.cursor() as cur:
                cur.execute('SELECT * FROM students WHERE ids
= %s', (student_id,))
                result = cur.fetchone()
                if result:
                    print(f"ID: {result[0]}, Имя: {result[1]},
Курс: {result[2]}")
                else:
                    print("Студент не найден")
    except Error as e:
        print(f"Ошибка при получении данных студента: {e}")
```

```python
def get_discipline(course):
    try:
        with connect_to_db() as conn:
            with conn.cursor() as cur:
                cur.execute('''
                    SELECT * FROM disciplines
                    WHERE nomk = %s
                    ORDER BY dweek, nomp
                ''', (course,))
                results = cur.fetchall()
                if results:
                    for r in results:
                        print(f"ID: {r[0]}, Предмет: {r[1]}, День: {r[2]}, Пара: {r[3]}, Курс: {r[4]}")
                else:
                    print("Занятия не найдены")
    except Error as e:
        print(f"Ошибка при получении расписания: {e}")


def get_students(course):
    try:
        with connect_to_db() as conn:
            with conn.cursor() as cur:
                cur.execute('''
                    SELECT * FROM students
                    WHERE nomk = %s
                    ORDER BY name_st
                ''', (course,))
                results = cur.fetchall()
                if results:
```

```
                    for r in results:
                        print(f"ID: {r[0]}, Имя: {r[1]}, Курс:
{r[2]}")
                else:
                    print("Студенты не найдены")
    except Error as e:
        print(f"Ошибка при получении списка студентов: {e}")


def get_disciplines():
    try:
        with connect_to_db() as conn:
            with conn.cursor() as cur:
                cur.execute('SELECT * FROM disciplines ORDER BY
dweek, nomp, nomk')
                results = cur.fetchall()
                if results:
                    for r in results:
                        print(f"ID: {r[0]}, Предмет: {r[1]}, День:
{r[2]}, Пара: {r[3]}, Курс: {r[4]}")
                else:
                    print("Расписание пусто")
    except Error as e:
        print(f"Ошибка при получении расписания: {e}")


def put_student(name, course):
    try:
        with connect_to_db() as conn:
            with conn.cursor() as cur:
                cur.execute('SELECT MAX(ids) FROM students')
                max_id = cur.fetchone()[0]
```

```
                new_id = 1 if max_id is None else max_id + 1

                cur.execute('''
                    INSERT INTO students (ids, name_st, nomk)
                    VALUES (%s, %s, %s)
                ''', (new_id, name, course))
                conn.commit()
                print(f"Студент успешно добавлен с ID: {new_id}")
    except Error as e:
        print(f"Ошибка при добавлении студента: {e}")


def put_discipline(name, day, pair, course):
    try:
        with connect_to_db() as conn:
            with conn.cursor() as cur:
                cur.execute('''
                    INSERT INTO disciplines (name_dis, dweek,
nomp, nomk)
                    VALUES (%s, %s, %s, %s)
                ''', (name, day, pair, course))
                conn.commit()
                print("Занятие успешно добавлено")
    except Error as e:
        print(f"Ошибка при добавлении занятия: {e}")


def delete_student(student_id):
    try:
        with connect_to_db() as conn:
            with conn.cursor() as cur:
```

```
                cur.execute('DELETE FROM students WHERE ids = %s',
(student_id,))
                if cur.rowcount > 0:
                    conn.commit()
                    print("Студент успешно удален")
                else:
                    print("Студент не найден")
    except Error as e:
        print(f"Ошибка при удалении студента: {e}")


def delete_discipline(discipline_id):
    try:
        with connect_to_db() as conn:
            with conn.cursor() as cur:
                cur.execute('DELETE FROM disciplines WHERE id
= %s', (discipline_id,))
                if cur.rowcount > 0:
                    conn.commit()
                    print("Занятие успешно удалено")
                else:
                    print("Занятие не найдено")
    except Error as e:
        print(f"Ошибка при удалении занятия: {e}")


def main():
    print('''
    Доступные команды:
    GET student [id] - получить информацию о студенте
    GET discipline [курс] - получить расписание для курса
    GET students [курс] - получить список студентов курса
```

```
    GET disciplines - получить полное расписание
    PUT student [имя] [курс] - добавить студента
    PUT discipline [название] [день] [пара] [курс] - добавить
занятие
    DELETE student [id] - удалить студента
    DELETE discipline [id] - удалить занятие
    EXIT - выход
    ''')

    while True:
        try:
            command = input('Введите команду: ')
            if not command:
                continue

            # Разбиваем строку, сохраняя текст в кавычках как один
элемент
            command = shlex.split(command)

            if command[0] == 'EXIT':
                break

            if len(command) < 2:
                print("Неверный формат команды")
                continue

            action, entity = command[0], command[1]

            if action == 'GET':
                if entity == 'student' and len(command) == 3:
                    get_student(int(command[2]))
```

```
                elif entity == 'discipline' and len(command) == 3:
                    get_discipline(int(command[2]))
                elif entity == 'students' and len(command) == 3:
                    get_students(int(command[2]))
                elif entity == 'disciplines' and len(command) ==
2:
                    get_disciplines()
                else:
                    print("Неверный формат команды GET")

            elif action == 'PUT':
                if entity == 'student' and len(command) == 4:
                    put_student(command[2], int(command[3]))
                elif entity == 'discipline' and len(command) == 6:
                    put_discipline(command[2], int(command[3]),
int(command[4]), int(command[5]))
                else:
                    print("Неверный формат команды PUT")

            elif action == 'DELETE':
                if entity == 'student' and len(command) == 3:
                    delete_student(int(command[2]))
                elif entity == 'discipline' and len(command) == 3:
                    delete_discipline(int(command[2]))
                else:
                    print("Неверный формат команды DELETE")

            else:
                print("Неизвестная команда")

        except (ValueError, IndexError):
```

```
            print("Ошибка в формате команды")
        except Exception as e:
            print(f"Произошла ошибка: {e}")


if __name__ == "__main__":
    main()
```

Результаты:

```
Доступные команды:
GET student [id] - получить информацию о студенте
GET discipline [курс] - получить расписание для курса
GET students [курс] - получить список студентов курса
GET disciplines - получить полное расписание
PUT student [имя] [курс] - добавить студента
PUT discipline [название] [день] [пара] [курс] - добавить занятие
DELETE student [id] - удалить студента
DELETE discipline [id] - удалить занятие
EXIT - выход
```

```
Введите команду: GET student 33
ID: 33, Имя: Mr. Laury Bins, Курс: 1
```

```
Введите команду: GET discipline 3
Занятия не найдены
```

```
Введите команду: GET students 3
ID: 91, Имя: Billy Haag, Курс: 3
ID: 19, Имя: Christopher Hilpert, Курс: 3
ID: 35, Имя: Dr. Shana Schuster, Курс: 3
ID: 3, Имя: Elta Okuneva, Курс: 3
ID: 51, Имя: Emmet Cruickshank, Курс: 3
ID: 27, Имя: Geovanni Stamm, Курс: 3
ID: 75, Имя: Miss Sid Little, Курс: 3
ID: 11, Имя: Modesta Ritchie, Курс: 3
ID: 83, Имя: Santina Larson, Курс: 3
ID: 59, Имя: Shany Kohler, Курс: 3
ID: 43, Имя: Stanley Mohr, Курс: 3
ID: 67, Имя: Terrell Oberbrunner, Курс: 3
ID: 99, Имя: Trevor Kilback, Курс: 3
```

```
Введите команду: GET disciplines
ID: 1, Предмет: Алгоритмы и структуры данных, День: 1, Пара: 2, Курс: 1
```

```
Введите команду: PUT student "Oleg Monday" 2
Студент успешно добавлен с ID: 102
Введите команду: GET student 102
ID: 102, Имя: Oleg Monday, Курс: 2
```

```
Введите команду: PUT discipline "Решение задач на LeetCode" 1 1 1
Занятие успешно добавлено
Введите команду: GET discipline 1
ID: 2, Предмет: Решение задач на LeetCode, День: 1, Пара: 1, Курс: 1
ID: 1, Предмет: Алгоритмы и структуры данных, День: 1, Пара: 2, Курс: 1
```

```
Введите команду: DELETE student 102
Студент успешно удален
Введите команду: GET student 102
Студент не найден
```

```
Введите команду: DELETE discipline 1
Занятие успешно удалено
Введите команду: GET disciplines
ID: 2, Предмет: Решение задач на LeetCode, День: 1, Пара: 1, Курс: 1
```

;