|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| 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:**

***-- 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}")**

**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")**

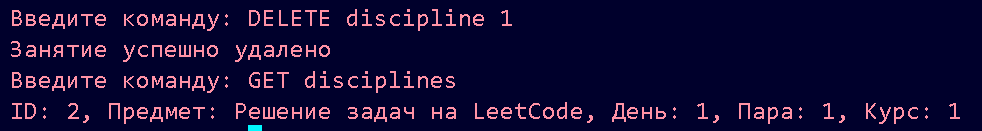
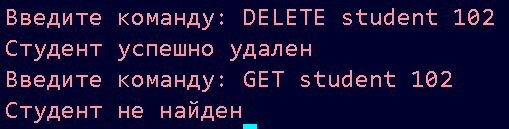
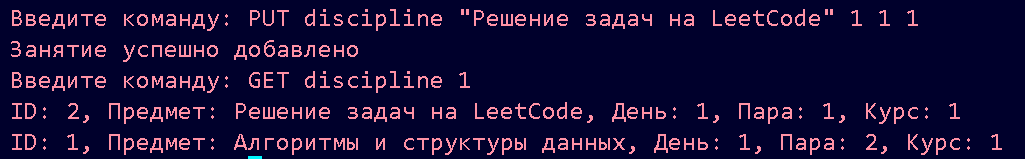
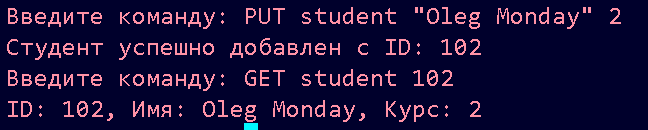
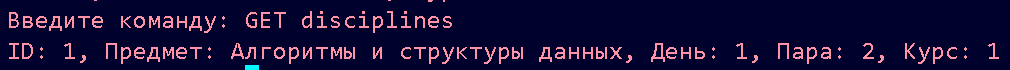
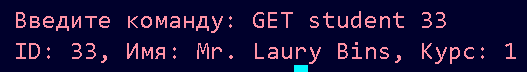
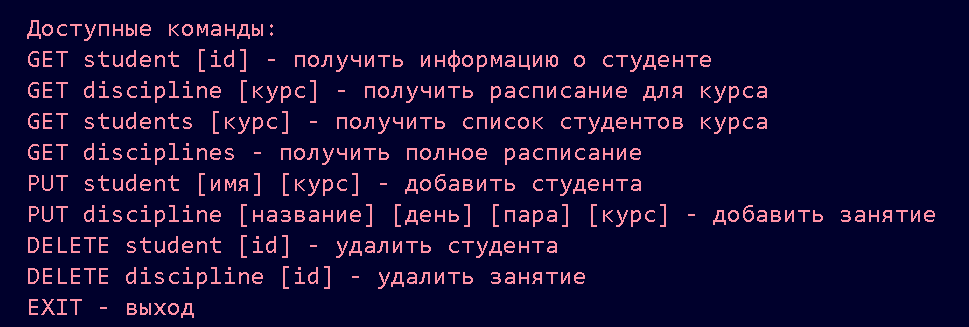
**except Exception as e:**

**print(f"Произошла ошибка: {e}")**

**if \_\_name\_\_ == "\_\_main\_\_":**

**main()**

**Результаты:**



DELIMITER ';'

CSV;