Name: JACKSON KIMOTHO

Id: 24/06455

DATABASE DESIGN AND DEVELOPMENT CLASS ASSIGNMENT 3

Design a University Course Registration System (10 marks). Use any OOP Language of your choice, such as C++ or Python.

Objective:
Implement an object-oriented data model to simulate a Course Registration System, where students can register for courses, view their enrolled courses, and update personal or course information.

Assignment Requirements:
ENTITIES (CLASSES):

1.STUDENT

1. Attributes:roll_no, department
2. Methods:registerCourse(), viewCouerses(), updateProfile()

2.COURSE

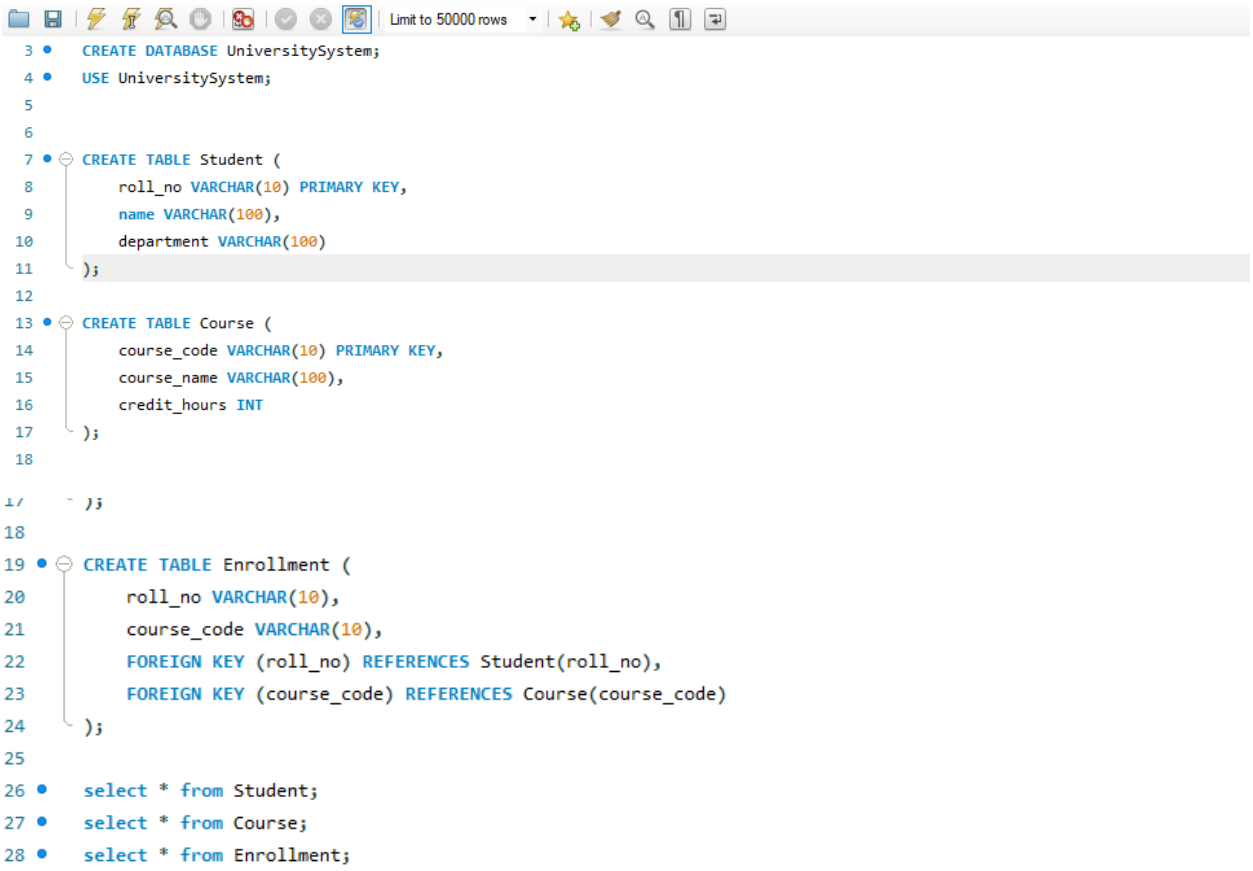
1. Attributes:course_code, course_name, credit_hours
2. Methods:display()

1.ENROLLMENT

1. Attributes:roll_no, course_code
2. Acts as a bridge class (like a junction table in DB)

Tasks:
1. Design the Object Model:
1. Define classes and their attributes/methods.
2. Create relationships (e.g., one student → many courses).
2. Simulate the Database with Arrays:
1. Allow students to input 3–5 students and 3–5 courses.
2. Use arrays to store objects.
3. Implement Operations:
1. Register a student for a course
2. Display all courses a student is enrolled in
3. Update student details (e.g., change name or department)
4. Search for students or courses by ID

Below is the mysql code for creating the database in mysql workbench:



Polite Note:

Below is the python (I used jupyter notebook as my coding environment).This gave me an advantage as I was able to link my database in my mysql to the python code. My python oop is input based so I have input the data manually using python but I can still do the same in my mysql and the details are pushed to the database directly.

```
            )
            db.commit()
            print("✅ Enrollment successful!\n")
```

[30]:
```
# ---------------------- SEARCH FUNCTIONS ----------------------
def search_student():
    roll_no = input("Enter student roll number to search: ")
    cursor.execute("SELECT * FROM Student WHERE roll_no = %s", (roll_no,))
    student = cursor.fetchone()

    if student:
        print(f"🔍 Found: Roll No: {student[0]}, Name: {student[1]}, Department: {student[2]}")
    else:
        print("❌ Student not found.")

def search_course():
    course_code = input("Enter course code to search: ")
    cursor.execute("SELECT * FROM Course WHERE course_code = %s", (course_code,))
    course = cursor.fetchone()

    if course:
        print(f"🔍 Found: Course Code: {course[0]}, Name: {course[1]}, Credit Hours: {course[2]}")
    else:
        print("❌ Course not found.")
```

[*]:
```
# ---------------------- MAIN MENU ----------------------
def main():
```

♦ Step 0: Set Up Database Connection

[22]:
```
import mysql.connector

# Connect to the MySQL database
db = mysql.connector.connect(
    host='localhost',
    user='root',
    password='Jack@11181',
    database="UniversitySystem"
)
cursor = db.cursor()
```

[24]:
```
# ---------------------- STUDENT CLASS ----------------------
class Student:
    def __init__(self):
        self.roll_no = input("Enter student roll number: ")
        self.name = input("Enter student name: ")
        self.department = input("Enter student department: ")

    def register(self):
        cursor.execute(
            "INSERT INTO Student (roll_no, name, department) VALUES (%s, %s, %s)",
            (self.roll_no, self.name, self.department)
        )
        db.commit()
        print("✅ Student registered successfully!\n")

    @staticmethod
    def update_profile():
        roll_no = input("Enter roll number of student to update: ")
        new_name = input("Enter new name (or press Enter to skip): ")
        new_department = input("Enter new department (or press Enter to skip): ")

        if new_name:
            cursor.execute("UPDATE Student SET name = %s WHERE roll_no = %s", (new_name, roll_no))

        if new_department:
            cursor.execute("UPDATE Student SET department = %s WHERE roll_no = %s", (new_department, roll_no))

        db.commit()
        print("✅ Student profile updated.\n")
```

```
            )
            db.commit()
            print("✅ Enrollment successful!\n")
```

[30]:
```
# ---------------------- SEARCH FUNCTIONS ----------------------
def search_student():
    roll_no = input("Enter student roll number to search: ")
    cursor.execute("SELECT * FROM Student WHERE roll_no = %s", (roll_no,))
    student = cursor.fetchone()

    if student:
        print(f"🔍 Found: Roll No: {student[0]}, Name: {student[1]}, Department: {student[2]}")
    else:
        print("❌ Student not found.")

def search_course():
    course_code = input("Enter course code to search: ")
    cursor.execute("SELECT * FROM Course WHERE course_code = %s", (course_code,))
    course = cursor.fetchone()

    if course:
        print(f"🔍 Found: Course Code: {course[0]}, Name: {course[1]}, Credit Hours: {course[2]}")
    else:
        print("❌ Course not found.")
```

[*]:
```
# ---------------------- MAIN MENU ----------------------
def main():
    while True:
        print("\n--- 🎓 University Course Registration System ---")
        print("1. Register a new student")
        print("2. Add a new course")
        print("3. Enroll a student in a course")
        print("4. View student's enrolled courses")
        print("5. Update student profile")
        print("6. Search student by roll number")
        print("7. Search course by course code")
        print("8. Exit")

        choice = input("Choose an option (1-8): ")

        if choice == '1':
            student = Student()
```

```
            student.register()
```

```
        elif choice == '2':
            course = Course()
            course.register()
        elif choice == '3':
            enrollment = Enrollment()
            enrollment.enroll()
        elif choice == '4':
            Student.view_courses()
        elif choice == '5':
            Student.update_profile()
        elif choice == '6':
            search_student()
        elif choice == '7':
            search_course()
        elif choice == '8':
            print("👋 Goodbye!")
            break
        else:
            print("❌ Invalid choice. Please try again.")

# Run the program
main()

# Close connection after exiting
cursor.close()
db.close()
```

```
--- 🎓 University Course Registration System ---
1. Register a new student
2. Add a new course
3. Enroll a student in a course
4. View student's enrolled courses
5. Update student profile
6. Search student by roll number
7. Search course by course code
8. Exit
Choose an option (1-8):
```

--- 🎓 University Course Registration System ---

1. Register a new student
2. Add a new course
3. Enroll a student in a course
4. View student's enrolled courses
5. Update student profile
6. Search student by roll number
7. Search course by course code
8. Exit Choose an option (1-8): 1 Enter student roll number: S001 Enter student name: Alice Johnson Enter student department: Computer science ✅ Student registered successfully!

outputs of inputs I made using python environment :

```
24  );
25
26  select * from Student;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

roll_no	name	department
S001	Alice Johnson	Computer science
S002	Brian Mwangi	Electrical Engineering
S003	Cynthia Wangari	Business Administration
S004	David Kamau	Information Technology
S005	Esther Njeri	Mechanical Engineering
NULL	NULL	NULL

```
27  select * from Course;
28  select * from Enrollment;
29
30
31
32
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

course_code	course_name	credit_hours
BA305	Business Ethics	2
CS101	Introduction to Programming	3
EE202	Circuit Analysis	4
IT210	Networking Fundamentals	3
ME301	Thermodynamics	4
NULL	NULL	NULL

```
28  select * from Enrollment;
29
30
31
32
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

roll_no	course_code
S001	CS101
S001	IT210
S002	EE202
S003	BA305
S004	CS101
S004	IT210
S005	ME301

Below is my ERD diagram: