Student Management System

1-

1-What the system for?

The system is dedicated to managing student data, registering them in courses, and determining the grades of each student in each subject.

Objective: To manage students, subjects, lecturers, and study recordings.

2- Who are the user?

Admin: Responsible for system administration.

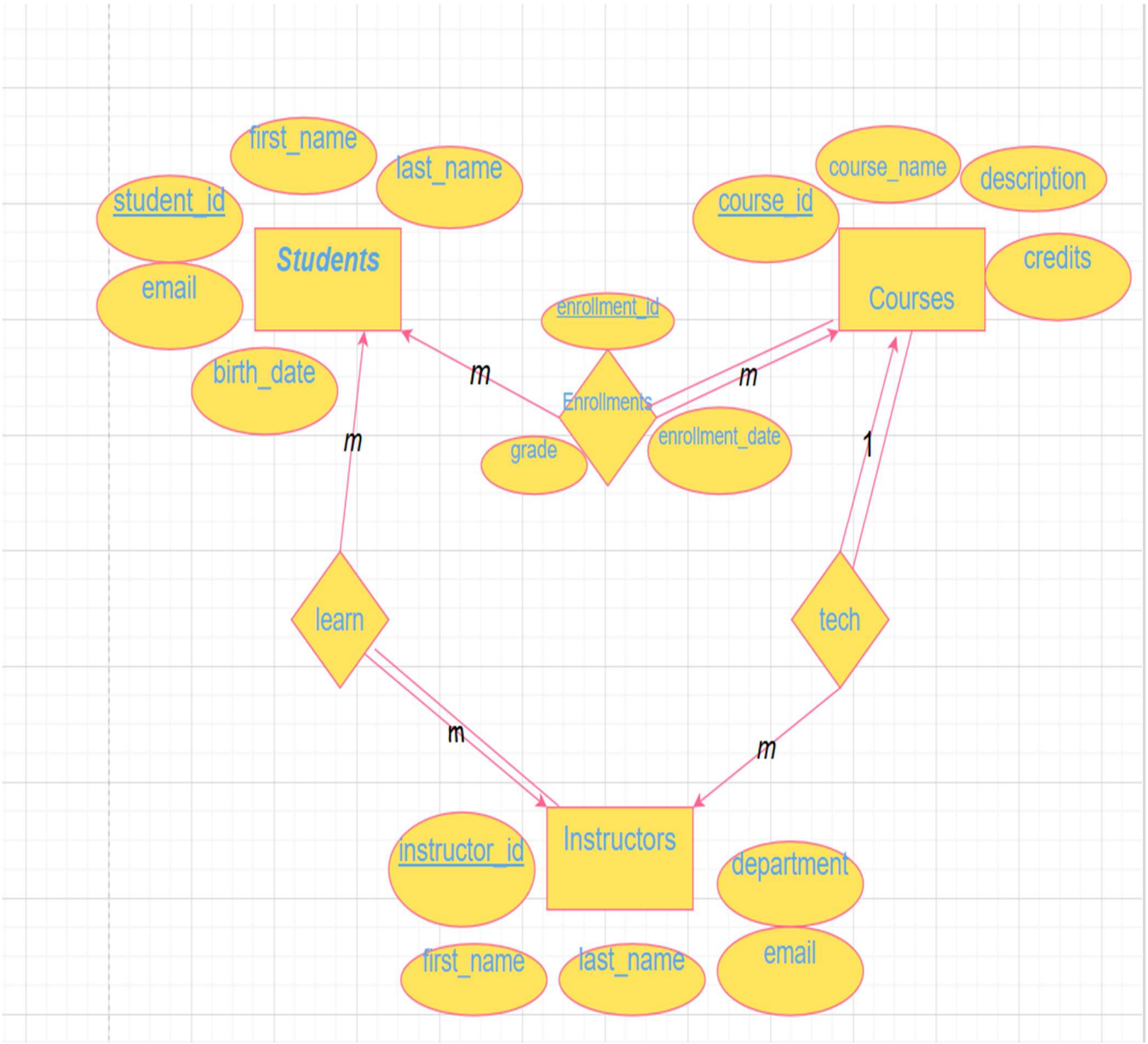
Instructor: The teacher who teaches courses and enters grades.

Student: The student who registers in the courses.

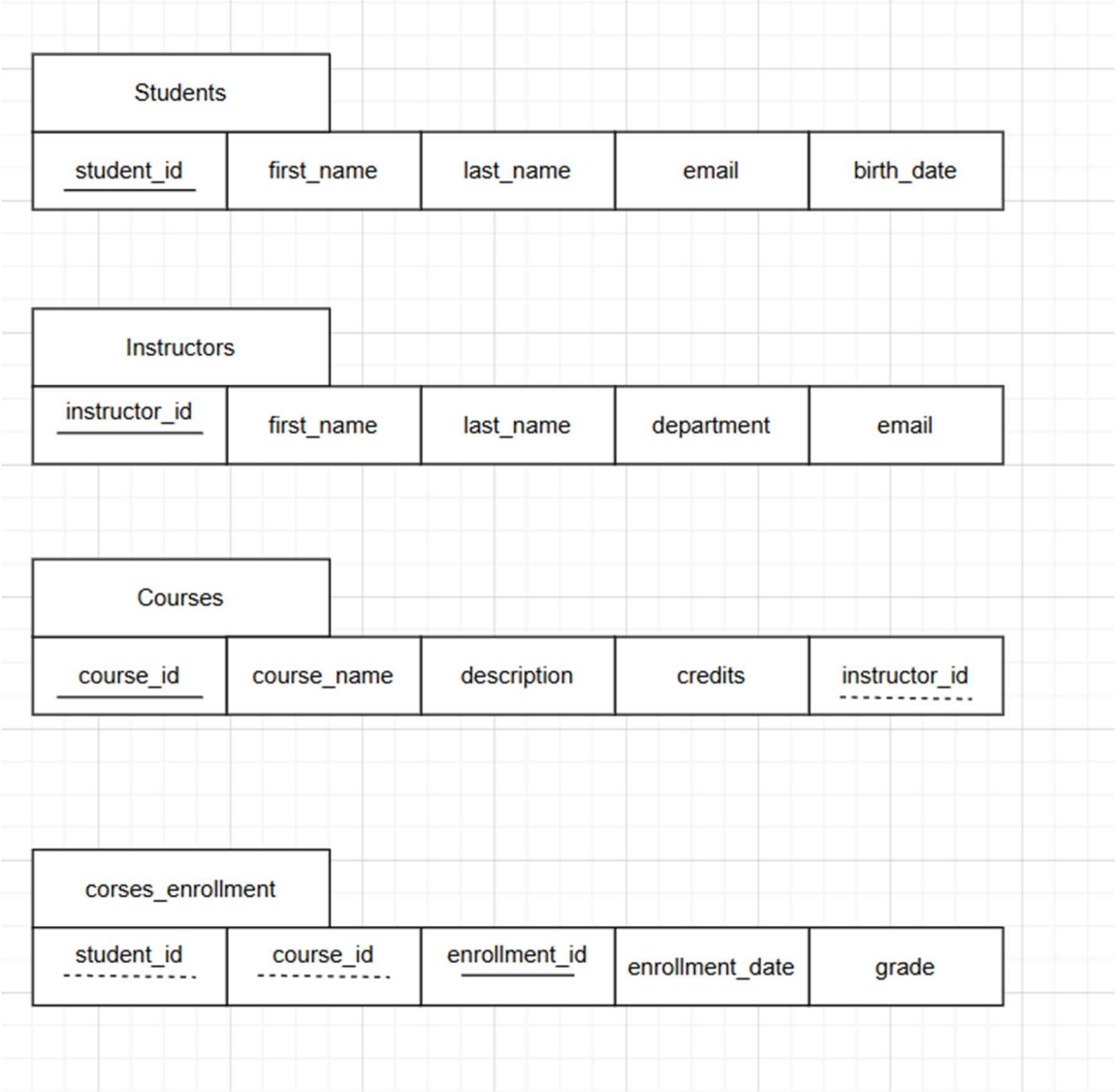
3-

Students	Courses	Instructors
student_id (PK)	course_id (PK)	instructor_id (PK)
first_name	course_name	first_name
last_name	Description	last_name
Email	Credits	Email
birth_date	instructor_id (FK)	department

2-


first_name
last_name
student_id
Students
email
birth_date
course_name
description
course_id
Courses
credits
enrollment_id
Enrollments
grade
enrollment_date
m
m
m
1
learn
tech
m
m
instructor_id
Instructors
department
first_name
last_name
email

*Mapping :



3.

1NF	no repeating group
2NF	No partial
3NF	No transitive dependencies

- All tables are already designed in 3NF (Third Normal Form) format:

  1.There is no repetition.
  2.Each table contains a primary key.
  3.The columns are entirely dependent on the primary key.
  4.Relationships are clear using foreign keys.

3-Suppose a preliminary table in this form:

student_id	student_name	courses	instructor_names	grade
1	Ali Ahmed	Database, Math	Dr. Hany, Dr. Omar	A, B
2	Sara Mohamed	Database	Dr. Hany	A
3	Mona Said	Math, Programming	Dr. Omar, Dr. Tamer	B, C

1NF

student_id	student_name	course_name	instructor_name	grade
1	Ali Ahmed	Database	Dr. Hany	A
1	Ali Ahmed	Math	Dr. Omar	B
2	Sara Mohamed	Database	Dr. Hany	A
3	Mona Said	Math	Dr. Omar	B
3	Mona Said	Programming	Dr. Tamer	C

2NF

Divide into tables:

Students:

student_id	student_name
1	Ali Ahmed
2	Sara Mohamed
3	Mona Said

Courses:

course_id	course_name	instructor_id
101	Database	201
102	Math	202
103	Programming	203

3NF

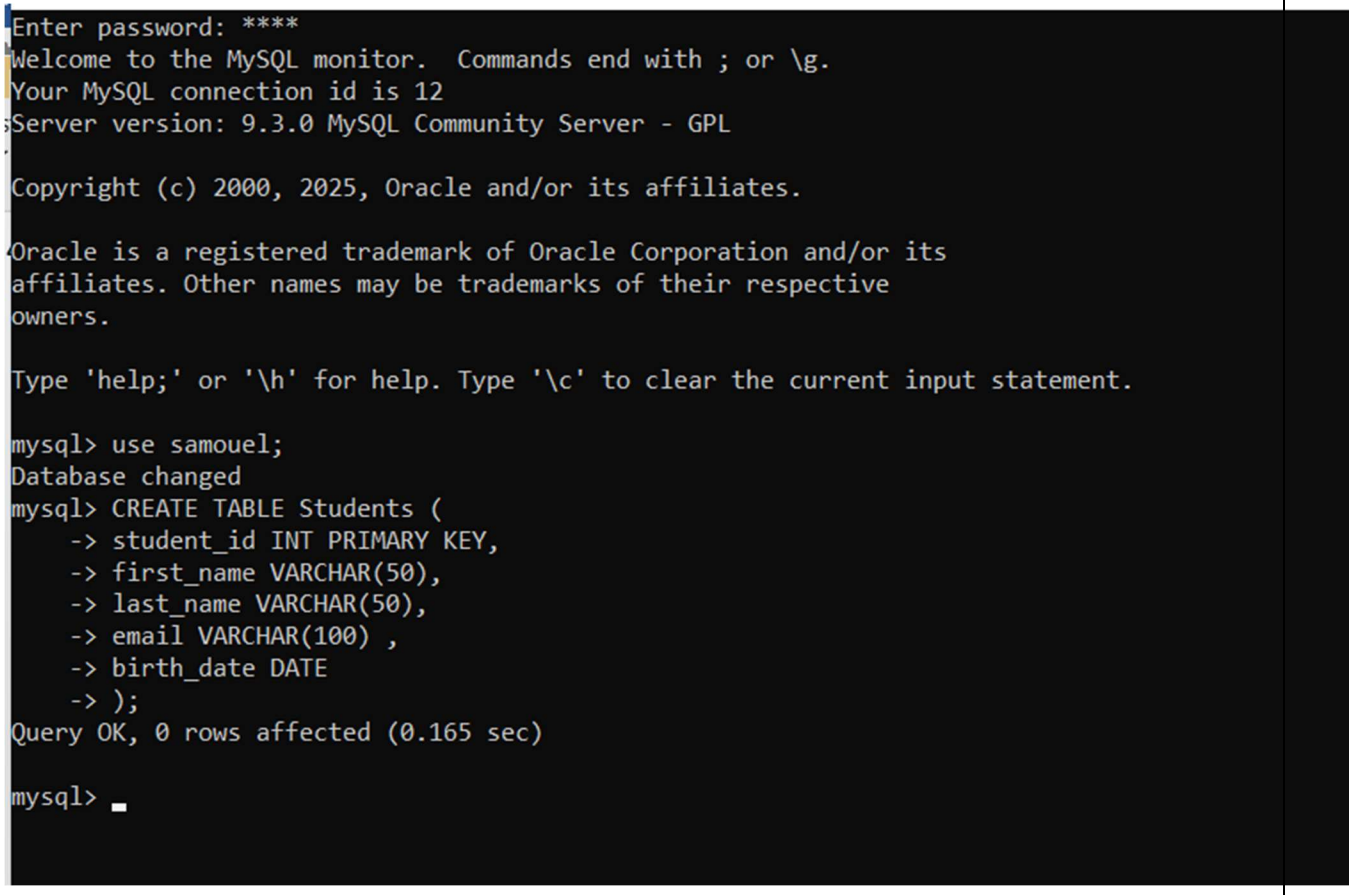
- **Students(student_id, student_name) in**
- **Instructors(structor_id, instructor_name)**
- **Courses(course_id, course_name, instructor_id)**

- **Enrollments(<u>student_id</u>, <u>course_id</u>, grade)**

4- SQL code

4.1

```
CREATE TABLE Students (
student_id INT PRIMARY KEY,
first_name VARCHAR(50),
last_name VARCHAR(50),
email VARCHAR(100) ,
birth_date DATE
);
```




```
CREATE TABLE Instructors (
instructor_id INT PRIMARY KEY,
```

first_name VARCHAR(50),
last_name VARCHAR(50),
email VARCHAR(100) ,
department VARCHAR(100)
);

```
mysql> CREATE TABLE Instructors (
    -> instructor_id INT PRIMARY KEY,
    -> first_name VARCHAR(50),
    -> last_name VARCHAR(50),
    -> email VARCHAR(100) ,
    -> department VARCHAR(100)
    -> );
Query OK, 0 rows affected (0.142 sec)
```

CREATE TABLE Courses (
course_id INT PRIMARY KEY,
course_name VARCHAR(100),
description TEXT,
credits INT,
instructor_id INT,
FOREIGN KEY (instructor_id) REFERENCES Instructors(instructor_id)
);

```
mysql> CREATE TABLE Courses (
    -> course_id INT PRIMARY KEY,
    -> course_name VARCHAR(100),
    -> description TEXT,
    -> credits INT,
    -> instructor_id INT,
    -> FOREIGN KEY (instructor_id) REFERENCES Instructors(instructor_id)
    -> );
Query OK, 0 rows affected (0.288 sec)
```

CREATE TABLE courses_enrollment (
enrollment_id INT PRIMARY KEY,
student_id INT,
course_id INT,
enrollment_date DATE,
grade VARCHAR(2),
FOREIGN KEY (student_id) REFERENCES Students(student_id),
FOREIGN KEY (course_id) REFERENCES Courses(course_id)
);

```
mysql> CREATE TABLE courses_enrollment (
    -> enrollment_id INT PRIMARY KEY,
    -> student_id INT,
    -> course_id INT,
    -> enrollment_date DATE,
    -> grade VARCHAR(2),
    -> FOREIGN KEY (student_id) REFERENCES Students(student_id),
    -> FOREIGN KEY (course_id) REFERENCES Courses(course_id)
    -> );
Query OK, 0 rows affected (0.304 sec)
```

4.2

GRANT SELECT ON Students TO student_user;

GRANT SELECT ON Courses TO student_user;

GRANT SELECT ON Enrollments TO student_user;

GRANT INSERT, UPDATE, DELETE ON Enrollments TO student_user;

GRANT ALL PRIVILEGES ON Courses TO student_user;

REVOKE INSERT, UPDATE ON Enrollments FROM student_user;

5-

INSERT INTO Students VALUES

(1, 'Ali', 'Hassan', 'ali.hassan@example.com', '2001-05-12'),

(2, 'Sara', 'Mahmoud', 'sara.mahmoud@example.com', '2002-08-25'),

(3, 'Yousef', 'Ibrahim', 'yousef.ibrahim@example.com', '2000-01-15'),

(4, 'Mona', 'Adel', 'mona.adel@example.com', '2003-04-20'),

(5, 'Khaled', 'Fathi', 'khaled.fathi@example.com', '1999-12-30'),

(6, 'Laila', 'Omar', 'laila.omar@example.com', '2001-06-10'),

(7, 'Ahmed', 'Zaki', 'ahmed.zaki@example.com', '2002-09-01'),

(8, 'Reem', 'Sami', 'reem.sami@example.com', '2000-11-11'),

(9, 'Nour', 'Ihab', 'nour.ihab@example.com', '2001-03-05'),

(10, 'Tamer', 'Hani', 'tamer.hani@example.com', '2003-07-17');

```
mysql> INSERT INTO Students VALUES
    -> (1, 'Ali', 'Hassan', 'ali.hassan@example.com', '2001-05-12'),
    -> (2, 'Sara', 'Mahmoud', 'sara.mahmoud@example.com', '2002-08-25'),
    -> (3, 'Yousef', 'Ibrahim', 'yousef.ibrahim@example.com', '2000-01-15'),
    -> (4, 'Mona', 'Adel', 'mona.adel@example.com', '2003-04-20'),
    -> (5, 'Khaled', 'Fathi', 'khaled.fathi@example.com', '1999-12-30'),
    -> (6, 'Laila', 'Omar', 'laila.omar@example.com', '2001-06-10'),
    -> (7, 'Ahmed', 'Zaki', 'ahmed.zaki@example.com', '2002-09-01'),
    -> (8, 'Reem', 'Sami', 'reem.sami@example.com', '2000-11-11'),
    -> (9, 'Nour', 'Ihab', 'nour.ihab@example.com', '2001-03-05'),
    -> (10, 'Tamer', 'Hani', 'tamer.hani@example.com', '2003-07-17');
Query OK, 10 rows affected (0.058 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

INSERT INTO Instructors VALUES

(1, 'Dr. Hany', 'Mostafa', 'hany.mostafa@example.com', 'Computer Science'),

(2, 'Dr. Salma', 'Ibrahim', 'salma.ibrahim@example.com', 'Mathematics'),

(3, 'Dr. Omar', 'Ali', 'omar.ali@example.com', 'Engineering'),

(4, 'Dr. Dina', 'Hassan', 'dina.hassan@example.com', 'Physics'),

(5, 'Dr. Nabil', 'Adel', 'nabil.adel@example.com', 'Business'),

(6, 'Dr. Ahmed', 'Tarek', 'ahmed.tarek@example.com', 'Biology'),

(7, 'Dr. Sara', 'Fawzy', 'sara.fawzy@example.com', 'Chemistry'),

(8, 'Dr. Khaled', 'Youssef', 'khaled.youssef@example.com', 'Philosophy'),

(9, 'Dr. Mona', 'Sherif', 'mona.sherif@example.com', 'History'),

(10, 'Dr. Rania', 'Gamal', 'rania.gamal@example.com', 'Literature');

```
mysql> INSERT INTO Instructors VALUES
    -> (1, 'Dr. Hany', 'Mostafa', 'hany.mostafa@example.com', 'Computer Science'),
    -> (2, 'Dr. Salma', 'Ibrahim', 'salma.ibrahim@example.com', 'Mathematics'),
    -> (3, 'Dr. Omar', 'Ali', 'omar.ali@example.com', 'Engineering'),
    -> (4, 'Dr. Dina', 'Hassan', 'dina.hassan@example.com', 'Physics'),
    -> (5, 'Dr. Nabil', 'Adel', 'nabil.adel@example.com', 'Business'),
    -> (6, 'Dr. Ahmed', 'Tarek', 'ahmed.tarek@example.com', 'Biology'),
    -> (7, 'Dr. Sara', 'Fawzy', 'sara.fawzy@example.com', 'Chemistry'),
    -> (8, 'Dr. Khaled', 'Youssef', 'khaled.youssef@example.com', 'Philosophy'),
    -> (9, 'Dr. Mona', 'Sherif', 'mona.sherif@example.com', 'History'),
    -> (10, 'Dr. Rania', 'Gamal', 'rania.gamal@example.com', 'Literature');
Query OK, 10 rows affected (0.042 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

INSERT INTO Courses VALUES

(101, 'Database Systems', 'Intro to relational databases', 3, 1),

(102, 'Calculus I', 'Basic calculus concepts', 4, 2),

(103, 'Physics 101', 'Mechanics and motion', 3, 4),

(104, 'Software Engineering', 'Design and architecture of software', 3, 1),

(105, 'Microeconomics', 'Economic principles', 2, 5),

(106, 'Linear Algebra', 'Matrix operations and vectors', 3, 2),

(107, 'Networks', 'Introduction to networking', 3, 3),

(108, 'Statistics', 'Probability and distributions', 3, 2),

(109, 'Web Development', 'Frontend and backend basics', 3, 1),

(110, 'Project Management', 'Managing software projects', 2, 5);

```
mysql> INSERT INTO Courses VALUES
    -> (101, 'Database Systems', 'Intro to relational databases', 3, 1),
    -> (102, 'Calculus I', 'Basic calculus concepts', 4, 2),
    -> (103, 'Physics 101', 'Mechanics and motion', 3, 4),
    -> (104, 'Software Engineering', 'Design and architecture of software', 3, 1),
    -> (105, 'Microeconomics', 'Economic principles', 2, 5),
    -> (106, 'Linear Algebra', 'Matrix operations and vectors', 3, 2),
    -> (107, 'Networks', 'Introduction to networking', 3, 3),
    -> (108, 'Statistics', 'Probability and distributions', 3, 2),
    -> (109, 'Web Development', 'Frontend and backend basics', 3, 1),
    -> (110, 'Project Management', 'Managing software projects', 2, 5);
Query OK, 10 rows affected (0.045 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

INSERT INTO courses_enrollment VALUES

(1, 1, 101, '2024-01-10', 'A'),

(2, 2, 102, '2024-01-11', 'B'),

(3, 3, 103, '2024-01-12', 'C'),

(4, 4, 104, '2024-01-13', 'A'),

(5, 5, 105, '2024-01-14', 'B'),

(6, 6, 106, '2024-01-15', 'A'),

(7, 7, 107, '2024-01-16', 'B'),

(8, 8, 108, '2024-01-17', 'C'),

(9, 9, 109, '2024-01-18', 'A'),

(10, 10, 110, '2024-01-19', 'B');

```
mysql> INSERT INTO courses_enrollment VALUES
    -> (1, 1, 101, '2024-01-10', 'A'),
    -> (2, 2, 102, '2024-01-11', 'B'),
    -> (3, 3, 103, '2024-01-12', 'C'),
    -> (4, 4, 104, '2024-01-13', 'A'),
    -> (5, 5, 105, '2024-01-14', 'B'),
    -> (6, 6, 106, '2024-01-15', 'A'),
    -> (7, 7, 107, '2024-01-16', 'B'),
    -> (8, 8, 108, '2024-01-17', 'C'),
    -> (9, 9, 109, '2024-01-18', 'A'),
    -> (10, 10, 110, '2024-01-19', 'B');
Query OK, 10 rows affected (0.058 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

6-make SQL queries:

SELECT s.first_name, s.last_name, c.course_name

FROM courses_enrollment e

JOIN Students s ON e.student_id = s.student_id

JOIN Courses c ON e.course_id = c.course_id

WHERE c.course_id = 101;

```
mysql> SELECT s.first_name, s.last_name, c.course_name
    -> FROM courses_enrollment e
    -> JOIN Students s ON e.student_id = s.student_id
    -> JOIN Courses c ON e.course_id = c.course_id
    -> WHERE c.course_id = 101;
+------------+-----------+------------------+
| first_name | last_name | course_name      |
+------------+-----------+------------------+
| Ali        | Hassan    | Database Systems |
+------------+-----------+------------------+
1 row in set (0.004 sec)
```

SELECT *

FROM Students

WHERE first_name LIKE '%Ali%';

```
mysql> SELECT *
    -> FROM Students
    -> WHERE first_name LIKE '%Ali%';
+------------+------------+-----------+-----------------------+------------+
| student_id | first_name | last_name | email                 | birth_date |
+------------+------------+-----------+-----------------------+------------+
|          1 | Ali        | Hassan    | ali.hassan@example.com | 2001-05-12 |
+------------+------------+-----------+-----------------------+------------+
1 row in set (0.007 sec)
```

SELECT c.course_name, COUNT(e.student_id) AS total_students

FROM Courses c

JOIN courses_enrollment e ON c.course_id = e.course_id

GROUP BY c.course_name

ORDER BY total_students DESC;

```
mysql> SELECT c.course_name, COUNT(e.student_id) AS total_students
    -> FROM Courses c
    -> JOIN courses_enrollment e ON c.course_id = e.course_id
    -> GROUP BY c.course_name
    -> ORDER BY total_students DESC;
+----------------------+----------------+
| course_name          | total_students |
+----------------------+----------------+
| Database Systems     |              1 |
| Calculus I           |              1 |
| Physics 101          |              1 |
| Software Engineering |              1 |
| Microeconomics       |              1 |
| Linear Algebra       |              1 |
| Networks             |              1 |
| Statistics           |              1 |
| Web Development      |              1 |
| Project Management   |              1 |
+----------------------+----------------+
10 rows in set (0.016 sec)
```

SELECT course_name, description

FROM Courses

WHERE description LIKE '%database%';

```
mysql> SELECT course_name, description
    -> FROM Courses
    -> WHERE description LIKE '%database%';
+------------------+-------------------------------+
| course_name      | description                   |
+------------------+-------------------------------+
| Database Systems | Intro to relational databases |
+------------------+-------------------------------+
1 row in set (0.003 sec)
```

SELECT s.first_name, s.last_name, e.grade

FROM courses_enrollment e

JOIN Students s ON e.student_id = s.student_id

ORDER BY e.grade DESC;

```
mysql> SELECT s.first_name, s.last_name, e.grade
    -> FROM courses_enrollment e
    -> JOIN Students s ON e.student_id = s.student_id
    -> ORDER BY e.grade DESC;
+------------+-----------+-------+
| first_name | last_name | grade |
+------------+-----------+-------+
| Yousef     | Ibrahim   | C     |
| Reem       | Sami      | C     |
| Sara       | Mahmoud   | B     |
| Khaled     | Fathi     | B     |
| Ahmed      | Zaki      | B     |
| Tamer      | Hani      | B     |
| Ali        | Hassan    | A     |
| Mona       | Adel      | A     |
| Laila      | Omar      | A     |
| Nour       | Ihab      | A     |
+------------+-----------+-------+
10 rows in set (0.004 sec)
```

Conclusion

The Student Management System provides a scalable and efficient solution for educational data management.

Through sound database design principles and normalization (up to 3NF), the system ensures data integrity,

eliminates redundancy, and supports various academic operations.

SQL scripts demonstrate how real-world use cases can be handled securely and effectively.