

Student Course Management System (MySQL Mini Project)

This is a small MySQL project I created to practice basic database concepts like tables, relationships, joins, and queries. It simulates a simple college system where students can enroll in different courses.

Project Overview

The project contains three main tables:

- **students** – stores student details
- **courses** – stores course information
- **enrollments** – connects students to the courses they take

This structure helped me understand how databases use **foreign keys**, **relationships**, and **joins**.

How I Built This (Step-by-Step)

1. Created a Database

```
CREATE DATABASE college_db;  
USE college_db;
```

2. Created Tables

I created three tables and linked them using foreign keys.

Students Table

```
CREATE TABLE students (  
    student_id INT PRIMARY KEY AUTO_INCREMENT,  
    name VARCHAR(50),  
    department VARCHAR(30)  
);
```

3. Courses Table

```
CREATE TABLE courses (
    course_id INT PRIMARY KEY AUTO_INCREMENT,
    course_name VARCHAR(50),
    credits INT
);
```

4. Enrollments Table

```
CREATE TABLE enrollments (
    enroll_id INT PRIMARY KEY AUTO_INCREMENT,
    student_id INT,
    course_id INT,
    semester VARCHAR(10),
    FOREIGN KEY (student_id) REFERENCES students(student_id),
    FOREIGN KEY (course_id) REFERENCES courses(course_id)
);
```

Sample Data Inserted

Students

```
INSERT INTO students (name, department) VALUES
('Ananya', 'CSE'),
('Rahul', 'ECE'),
('Priya', 'IT');
```

Courses

```
INSERT INTO courses (course_name, credits) VALUES
('Database Systems', 3),
('Python Programming', 4),
('Data Structures', 3);
```

Enrollments

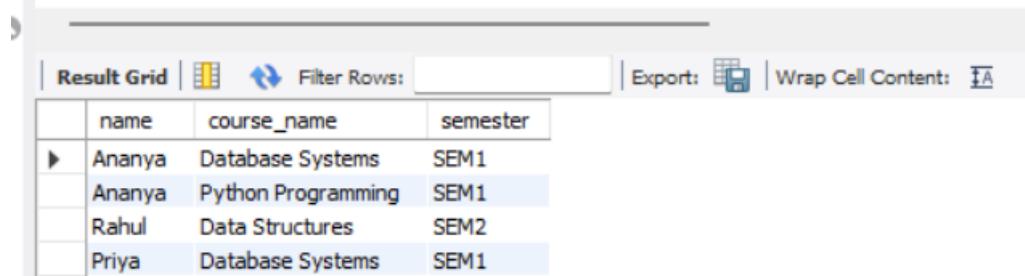
```
INSERT INTO enrollments (student_id, course_id, semester) VALUES  
(1, 1, 'SEM1'),  
(1, 2, 'SEM1'),  
(2, 3, 'SEM2'),  
(3, 1, 'SEM1');
```

🔍 Queries I Used

1. Students with their courses

```
SELECT s.name, c.course_name, e.semester  
FROM students s  
JOIN enrollments e ON s.student_id = e.student_id  
JOIN courses c ON c.course_id = e.course_id;
```

```
1 •  SELECT s.name, c.course_name, e.semester  
2   FROM students s  
3   JOIN enrollments e ON s.student_id = e.student_id  
4   JOIN courses c ON c.course_id = e.course_id;  
5  
6  |
```



The screenshot shows a database query results grid. At the top, there are buttons for 'Result Grid' (selected), 'Filter Rows:', 'Export:', and 'Wrap Cell Content:'. The grid itself has columns for 'name', 'course_name', and 'semester'. The data is as follows:

	name	course_name	semester
▶	Ananya	Database Systems	SEM1
	Ananya	Python Programming	SEM1
	Rahul	Data Structures	SEM2
	Priya	Database Systems	SEM1

2. Number of students per course

```
SELECT c.course_name, COUNT(e.student_id) AS total_students  
FROM courses c  
LEFT JOIN enrollments e ON c.course_id = e.course_id  
GROUP BY c.course_name;
```

```
1 •  SELECT c.course_name, COUNT(e.student_id) AS total_students  
2    FROM courses c  
3    LEFT JOIN enrollments e ON c.course_id = e.course_id  
4    GROUP BY c.course_name;
```

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

	course_name	total_students
▶	Database Systems	2
	Python Programming	1
	Data Structures	1

3. Filter example

```
SELECT * FROM students WHERE department = 'CSE';
```

```
1 •  SELECT * FROM students WHERE department = 'CSE';  
2  |
```

Result Grid | Filter Rows: | Edit: | Export/

	student_id	name	department
▶	1	Ananya	CSE
*	NULL	NULL	NULL

SQL Console output:

Action Output			Message
#	Time	Action	
1	20:36:08	SELECT 1 LIMIT 0, 1000	1 row(s) returned
2	20:36:25	SELECT * LIMIT 0, 1000	Error Code: 1096. No tables used
3	20:36:58	CREATE TABLE students (student_id INT PRIMARY KEY AUTO_INCREMENT, name VARCHAR(50), ...)	0 row(s) affected
4	20:37:56	CREATE TABLE students (student_id INT PRIMARY KEY AUTO_INCREMENT, name VARCHAR(50), ...)	Error Code: 1050. Table 'students' already exists
5	20:37:59	CREATE TABLE students (student_id INT PRIMARY KEY AUTO_INCREMENT, name VARCHAR(50), ...)	Error Code: 1050. Table 'students' already exists
6	20:38:35	CREATE TABLE courses (course_id INT PRIMARY KEY AUTO_INCREMENT, course_name VARCHAR(...), ...)	0 row(s) affected
7	20:38:35	CREATE TABLE enrollments (enroll_id INT PRIMARY KEY AUTO_INCREMENT, student_id INT, cours...)	0 row(s) affected
8	20:38:35	INSERT INTO students (name, department) VALUES ('Ananya', 'CSE'), ('Rahul', 'ECE'), ('Prya', 'IT')	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0
9	20:38:35	INSERT INTO courses (course_name, credits) VALUES ('Database Systems', 3), ('Python Programming', 4), ('Dat...	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0
10	20:38:35	INSERT INTO enrollments (student_id, course_id, semester) VALUES (1, 1, 'SEM1'), (1, 2, 'SEM1'), (2, 3, 'SEM2')...	4 row(s) affected Records: 4 Duplicates: 0 Warnings: 0
11	20:41:06	SELECT s.name, c.course_name, e.semester FROM students s JOIN enrollments e ON s.student_id = e.student...	4 row(s) returned
12	20:41:06	SELECT c.course_name, COUNT(e.student_id) AS total_students FROM courses c LEFT JOIN enrollments e O...	3 row(s) returned
13	20:41:07	SELECT * FROM students WHERE department = 'CSE' LIMIT 0, 1000	1 row(s) returned
14	20:42:11	SELECT s.name, c.course_name, e.semester FROM students s JOIN enrollments e ON s.student_id = e.student...	4 row(s) returned
15	20:43:18	SELECT c.course_name, COUNT(e.student_id) AS total_students FROM courses c LEFT JOIN enrollments e O...	3 row(s) returned
16	20:46:13	SELECT * FROM students WHERE department = 'CSE' LIMIT 0, 1000	1 row(s) returned

What I Learned

- How to create databases and tables in MySQL
- Importance of **primary keys** and **foreign keys**
- How to insert and retrieve data
- How **JOIN**, **GROUP BY**, and **filters** work
- Designing a mini relational database