SQL MODULE
LAB – 8

Name: Pravallika Ullikanti

AFID: 0366355

Lab 1:

Use the Student management system Database and table from our previous lab and write a sql query to achieve the below scenario.

Assume you are managing a university database that tracks student enrollments in various courses. You have two tables, "Student" and "Enrollment". The goal is to retrieve information about each student's ID, first name, last name, and their enrollment details, including the enrollment ID and the associated course ID.

Hint: Use inner join to retrieve data.

Submission:

Create an SQL script file containing your solutions for all tasks (queries). Name the file "lab_assignment1.sql" Provide comments above each query to indicate the query's purpose.

ChatGPT Exercise

Using ChatGPT generates SQL queries of the below problem.

Scenario 1:

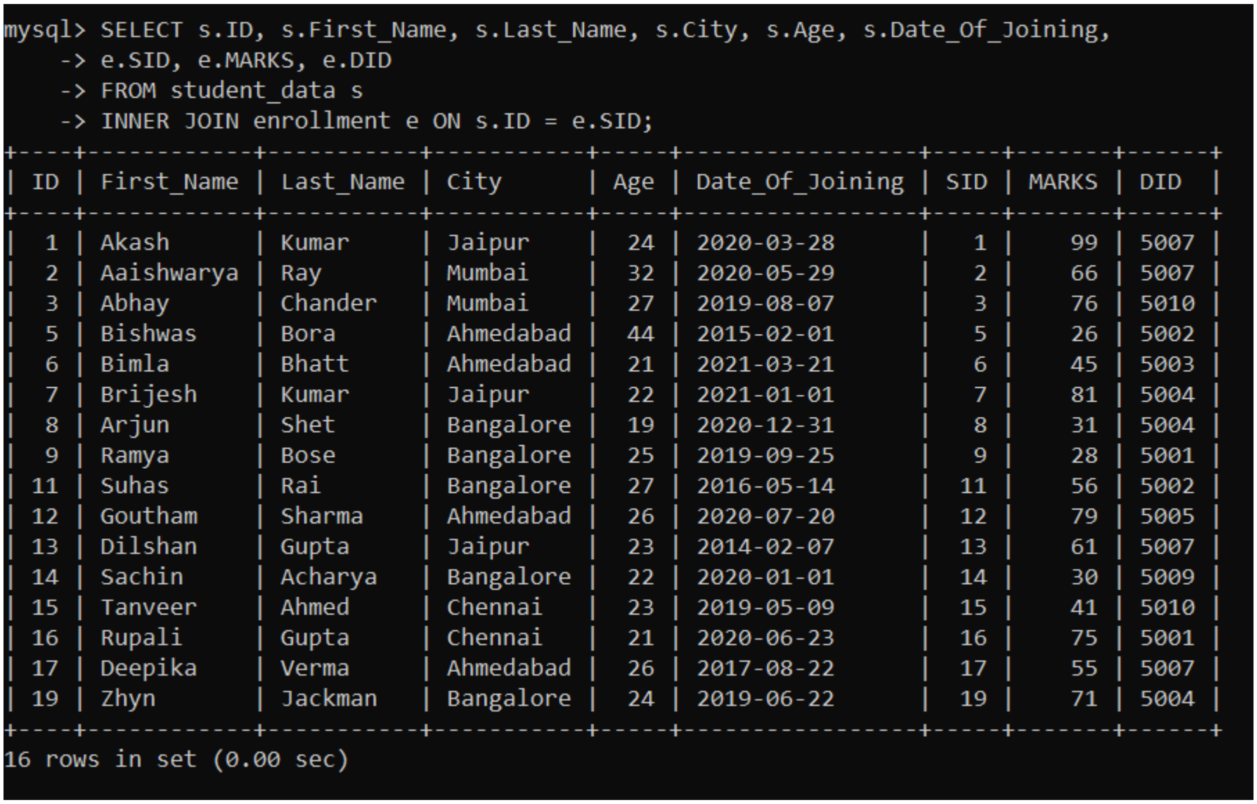
Imagine you have tables for students and courses. Use an inner join to generate a list of all possible student-course combinations, displaying the student name and course name.

We have a "Student" table with the following a columns:StudentId,FirstName,lastName and "Course" table with the following a columns: CourseId,CourseName and Enrollment table with the following a columns:EnrollmentID,StudentID(Foreign key),CourseID(Foreign Key).You want to use inner join to generate a list of all possible student-course combinations.Generate the ChatGPT prompt for the above scenario.

1. Assume you are managing a university database that tracks student enrollments in various courses. You have two tables, "Student" and "Enrollment". The goal is to retrieve information about each student's ID, first name, last name, and their enrollment details, including the enrollment ID and the associated course ID.

Hint: Use inner join to retrieve data.

Output:



```
mysql> SELECT s.ID, s.First_Name, s.Last_Name, s.City, s.Age, s.Date_Of_Joining,
    -> e.SID, e.MARKS, e.DID
    -> FROM student_data s
    -> INNER JOIN enrollment e ON s.ID = e.SID;
+----+------------+-----------+-----------+-----+-----------------+-----+-------+------+
| ID | First_Name | Last_Name | City      | Age | Date_Of_Joining | SID | MARKS | DID  |
+----+------------+-----------+-----------+-----+-----------------+-----+-------+------+
|  1 | Akash      | Kumar     | Jaipur    |  24 | 2020-03-28      |   1 |    99 | 5007 |
|  2 | Aaishwarya | Ray       | Mumbai    |  32 | 2020-05-29      |   2 |    66 | 5007 |
|  3 | Abhay      | Chander   | Mumbai    |  27 | 2019-08-07      |   3 |    76 | 5010 |
|  5 | Bishwas    | Bora      | Ahmedabad |  44 | 2015-02-01      |   5 |    26 | 5002 |
|  6 | Bimla      | Bhatt     | Ahmedabad |  21 | 2021-03-21      |   6 |    45 | 5003 |
|  7 | Brijesh    | Kumar     | Jaipur    |  22 | 2021-01-01      |   7 |    81 | 5004 |
|  8 | Arjun      | Shet      | Bangalore |  19 | 2020-12-31      |   8 |    31 | 5004 |
|  9 | Ramya      | Bose      | Bangalore |  25 | 2019-09-25      |   9 |    28 | 5001 |
| 11 | Suhas      | Rai       | Bangalore |  27 | 2016-05-14      |  11 |    56 | 5002 |
| 12 | Goutham    | Sharma    | Ahmedabad |  26 | 2020-07-20      |  12 |    79 | 5005 |
| 13 | Dilshan    | Gupta     | Jaipur    |  23 | 2014-02-07      |  13 |    61 | 5007 |
| 14 | Sachin     | Acharya   | Bangalore |  22 | 2020-01-01      |  14 |    30 | 5009 |
| 15 | Tanveer    | Ahmed     | Chennai   |  23 | 2019-05-09      |  15 |    41 | 5010 |
| 16 | Rupali     | Gupta     | Chennai   |  21 | 2020-06-23      |  16 |    75 | 5001 |
| 17 | Deepika    | Verma     | Ahmedabad |  26 | 2017-08-22      |  17 |    55 | 5007 |
| 19 | Zhyn       | Jackman   | Bangalore |  24 | 2019-06-22      |  19 |    71 | 5004 |
+----+------------+-----------+-----------+-----+-----------------+-----+-------+------+
16 rows in set (0.00 sec)
```

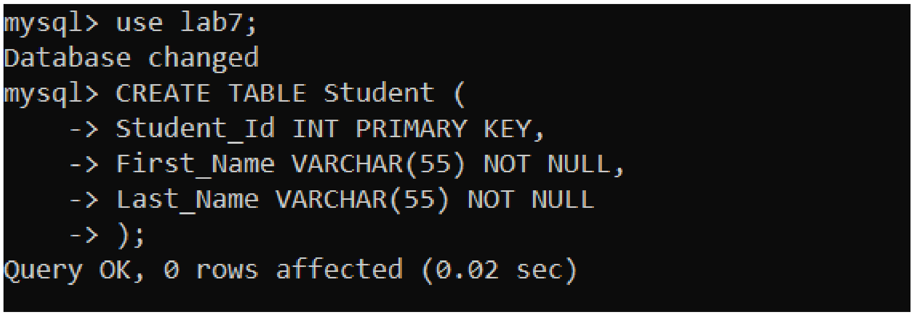
ChatGPT Exercise

Using ChatGPT generates SQL queries of the below problem. Scenario 1: Imagine you have tables for students and courses. Use an inner join to generate a list of all possible student-course combinations, displaying the student name and course name. We have a "Student" table with the following a columns:StudentId,FirstName,lastName and

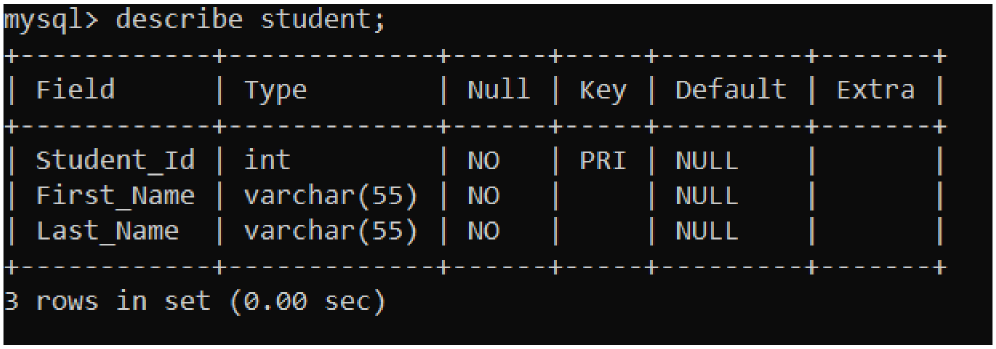
code:

a) CREATE TABLE Student (Student_Id INT PRIMARY KEY, First_Name VARCHAR (55) NOT NULL, Last_Name VARCHAR (55) NOT NULL);

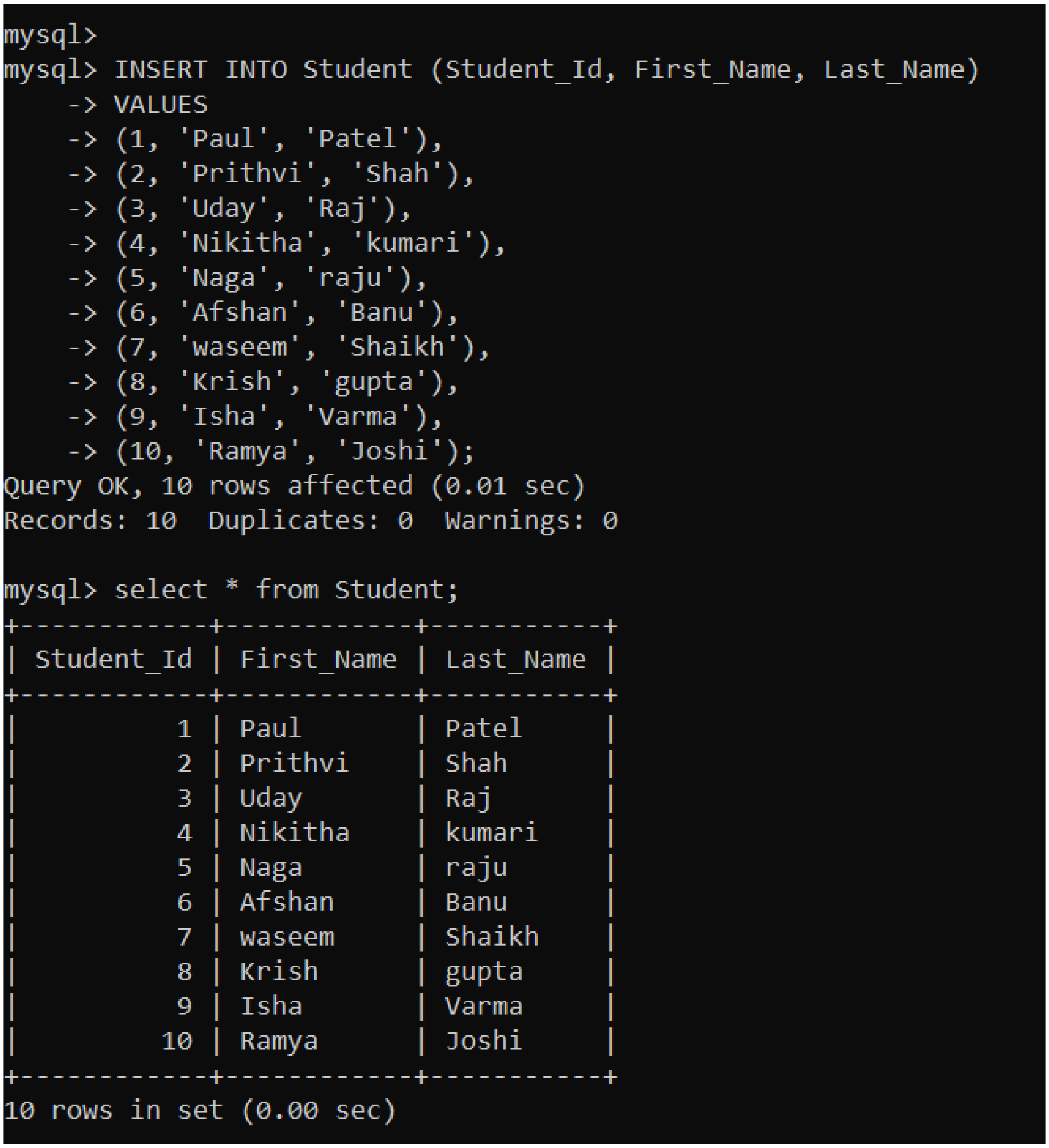
Output:



```
mysql> use lab7;
Database changed
mysql> CREATE TABLE Student (
    -> Student_Id INT PRIMARY KEY,
    -> First_Name VARCHAR(55) NOT NULL,
    -> Last_Name VARCHAR(55) NOT NULL
    -> );
Query OK, 0 rows affected (0.02 sec)
```

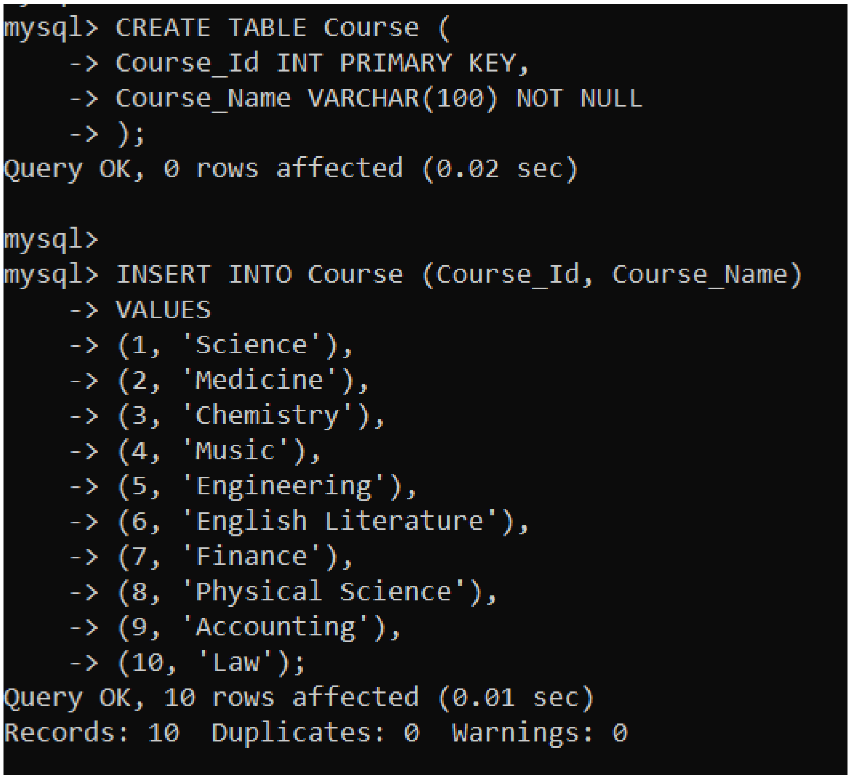


```
mysql> describe student;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| Student_Id | int         | NO   | PRI | NULL    |       |
| First_Name | varchar(55) | NO   |     | NULL    |       |
| Last_Name  | varchar(55) | NO   |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

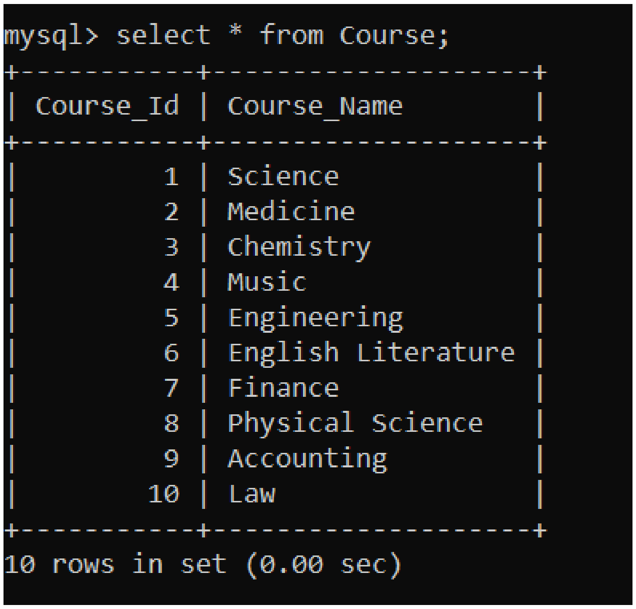


```
mysql>
mysql> INSERT INTO Student (Student_Id, First_Name, Last_Name)
    -> VALUES
    -> (1, 'Paul', 'Patel'),
    -> (2, 'Prithvi', 'Shah'),
    -> (3, 'Uday', 'Raj'),
    -> (4, 'Nikitha', 'kumari'),
    -> (5, 'Naga', 'raju'),
    -> (6, 'Afshan', 'Banu'),
    -> (7, 'waseem', 'Shaikh'),
    -> (8, 'Krish', 'gupta'),
    -> (9, 'Isha', 'Varma'),
    -> (10, 'Ramya', 'Joshi');
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> select * from Student;
+------------+------------+-----------+
| Student_Id | First_Name | Last_Name |
+------------+------------+-----------+
|          1 | Paul       | Patel     |
|          2 | Prithvi    | Shah      |
|          3 | Uday       | Raj       |
|          4 | Nikitha    | kumari    |
|          5 | Naga       | raju      |
|          6 | Afshan     | Banu      |
|          7 | waseem     | Shaikh    |
|          8 | Krish      | gupta     |
|          9 | Isha       | Varma     |
|         10 | Ramya      | Joshi     |
+------------+------------+-----------+
10 rows in set (0.00 sec)
```

b)Course" table with the following a columns: CourseId,CourseName.



```
mysql> CREATE TABLE Course (
    -> Course_Id INT PRIMARY KEY,
    -> Course_Name VARCHAR(100) NOT NULL
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql>
mysql> INSERT INTO Course (Course_Id, Course_Name)
    -> VALUES
    -> (1, 'Science'),
    -> (2, 'Medicine'),
    -> (3, 'Chemistry'),
    -> (4, 'Music'),
    -> (5, 'Engineering'),
    -> (6, 'English Literature'),
    -> (7, 'Finance'),
    -> (8, 'Physical Science'),
    -> (9, 'Accounting'),
    -> (10, 'Law');
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0
```



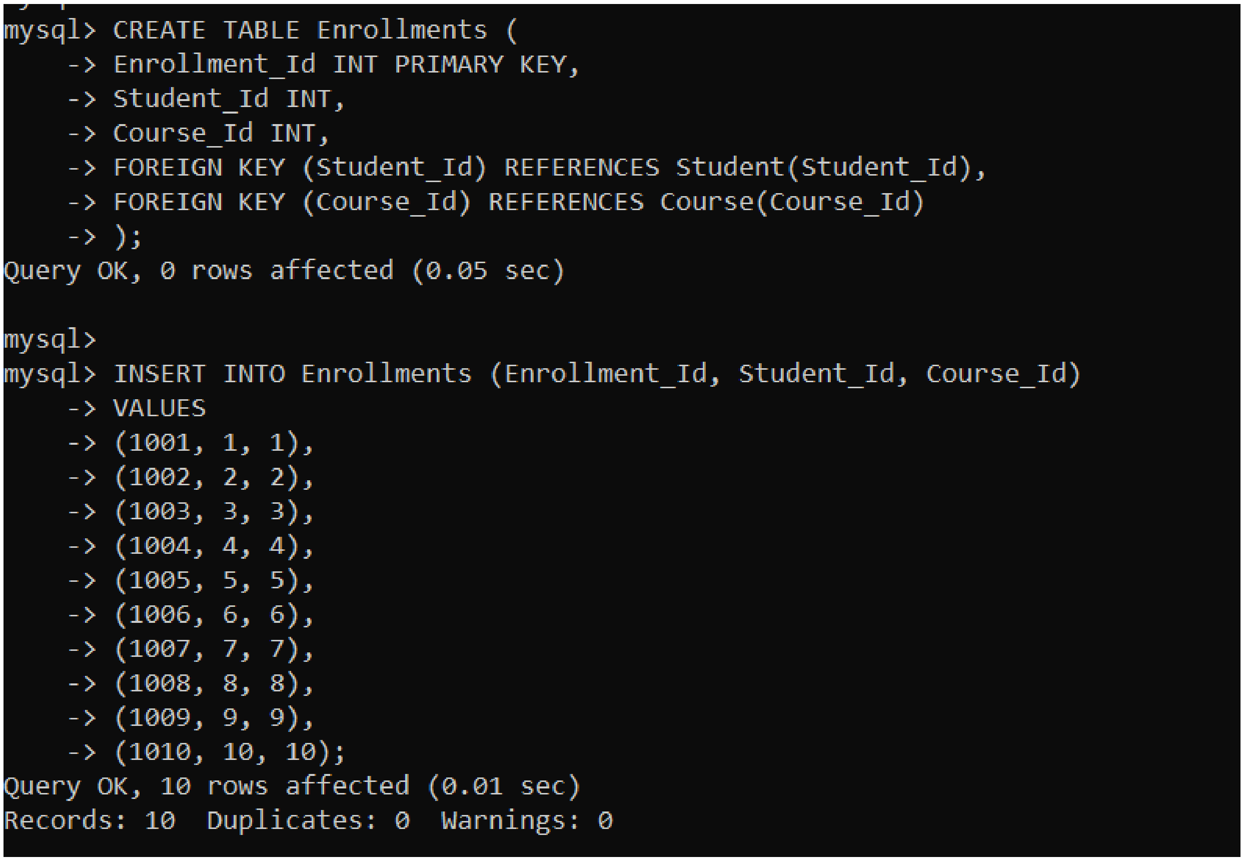
```
mysql> select * from Course;
+-----------+--------------------+
| Course_Id | Course_Name        |
+-----------+--------------------+
|         1 | Science            |
|         2 | Medicine           |
|         3 | Chemistry          |
|         4 | Music              |
|         5 | Engineering        |
|         6 | English Literature |
|         7 | Finance            |
|         8 | Physical Science   |
|         9 | Accounting         |
|        10 | Law                |
+-----------+--------------------+
10 rows in set (0.00 sec)
```

Enrollment table with the following a columns:

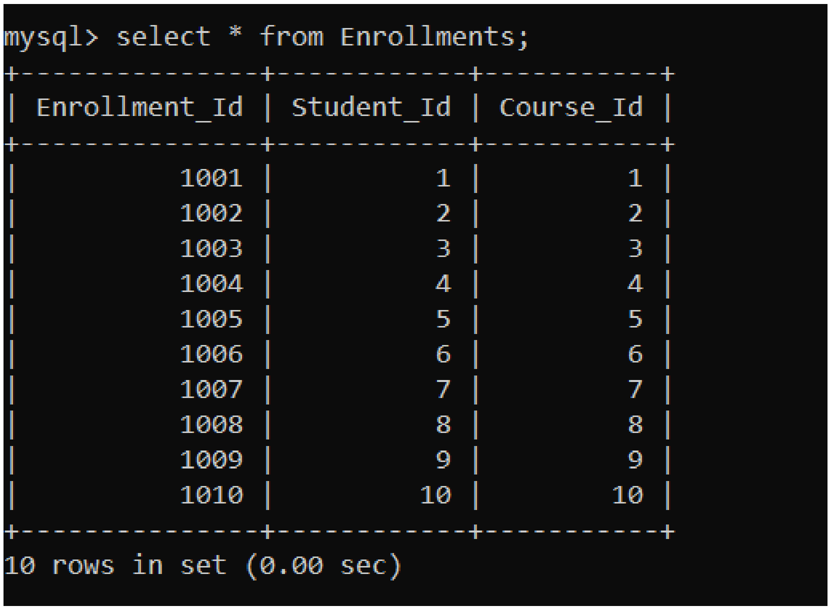
EnrollmentID,StudentID(Foreign key),CourseID(Foreign Key).

Code:

CREATE TABLE Enrollments (Enrollment_Id INT PRIMARY KEY, Student_Id INT, Course_Id INT, FOREIGN KEY (Student_Id) REFERENCES Student (Student_Id), FOREIGN KEY (Course_Id) REFERENCES Course (Course_Id));



```
mysql> CREATE TABLE Enrollments (
    -> Enrollment_Id INT PRIMARY KEY,
    -> Student_Id INT,
    -> Course_Id INT,
    -> FOREIGN KEY (Student_Id) REFERENCES Student(Student_Id),
    -> FOREIGN KEY (Course_Id) REFERENCES Course(Course_Id)
    -> );
Query OK, 0 rows affected (0.05 sec)

mysql>
mysql> INSERT INTO Enrollments (Enrollment_Id, Student_Id, Course_Id)
    -> VALUES
    -> (1001, 1, 1),
    -> (1002, 2, 2),
    -> (1003, 3, 3),
    -> (1004, 4, 4),
    -> (1005, 5, 5),
    -> (1006, 6, 6),
    -> (1007, 7, 7),
    -> (1008, 8, 8),
    -> (1009, 9, 9),
    -> (1010, 10, 10);
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0
```



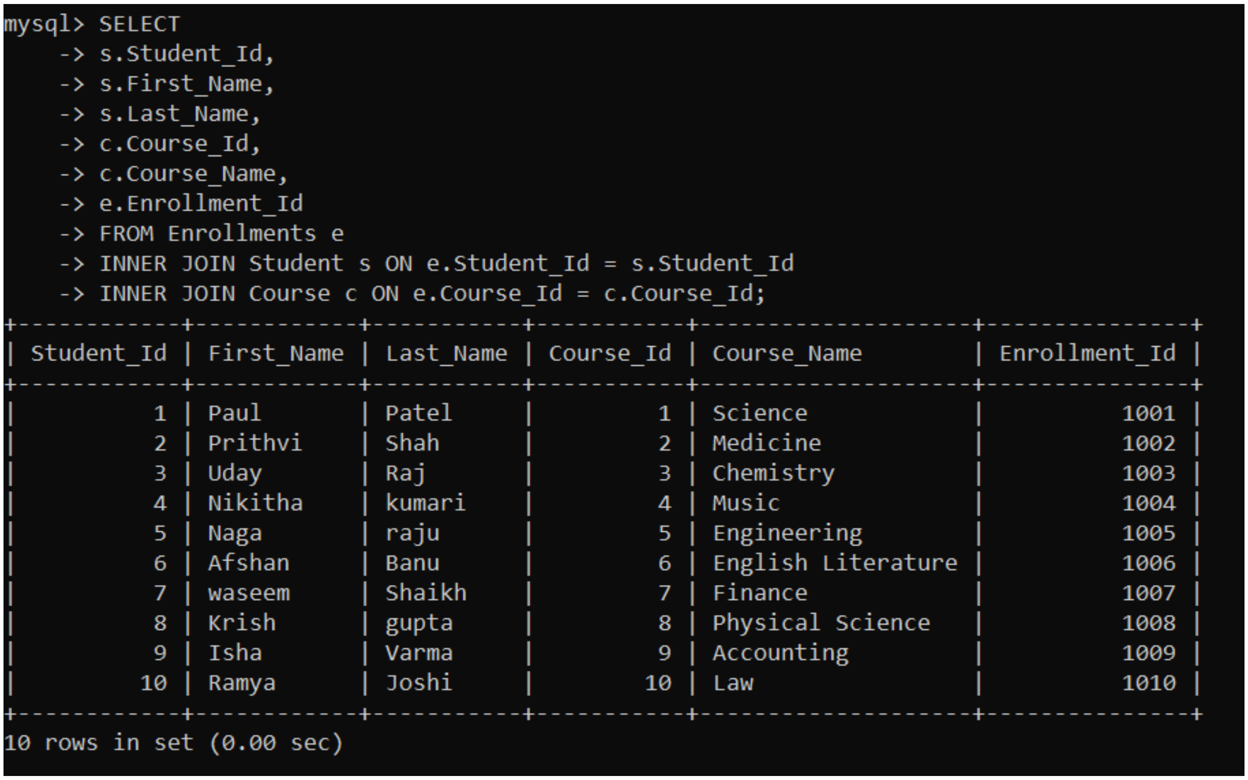
```
mysql> select * from Enrollments;
+---------------+------------+-----------+
| Enrollment_Id | Student_Id | Course_Id |
+---------------+------------+-----------+
|          1001 |          1 |         1 |
|          1002 |          2 |         2 |
|          1003 |          3 |         3 |
|          1004 |          4 |         4 |
|          1005 |          5 |         5 |
|          1006 |          6 |         6 |
|          1007 |          7 |         7 |
|          1008 |          8 |         8 |
|          1009 |          9 |         9 |
|          1010 |         10 |        10 |
+---------------+------------+-----------+
10 rows in set (0.00 sec)
```

You want to use an inner join to generate a list of all possible student-course combinations.

Generate the ChatGPT prompt for the above scenario.

Code:

```
SELECT  s.Student_Id,  s.First_Name,  s.Last_Name,

c.Course_Id, c.Course_Name

e.Enrollment_Id, e.Student_Id, e.Course_Id ;

FROM Enrollments e

 INNER JOIN Student s ON e.Student_Id = s.Student_Id

INNER JOIN Course c ON e.Course_Id = c.Course_Id;
```



```
mysql> SELECT
    -> s.Student_Id,
    -> s.First_Name,
    -> s.Last_Name,
    -> c.Course_Id,
    -> c.Course_Name,
    -> e.Enrollment_Id
    -> FROM Enrollments e
    -> INNER JOIN Student s ON e.Student_Id = s.Student_Id
    -> INNER JOIN Course c ON e.Course_Id = c.Course_Id;
+------------+------------+-----------+-----------+--------------------+---------------+
| Student_Id | First_Name | Last_Name | Course_Id | Course_Name        | Enrollment_Id |
+------------+------------+-----------+-----------+--------------------+---------------+
|          1 | Paul       | Patel     |         1 | Science            |          1001 |
|          2 | Prithvi    | Shah      |         2 | Medicine           |          1002 |
|          3 | Uday       | Raj       |         3 | Chemistry          |          1003 |
|          4 | Nikitha    | kumari    |         4 | Music              |          1004 |
|          5 | Naga       | raju      |         5 | Engineering        |          1005 |
|          6 | Afshan     | Banu      |         6 | English Literature |          1006 |
|          7 | waseem     | Shaikh    |         7 | Finance            |          1007 |
|          8 | Krish      | gupta     |         8 | Physical Science   |          1008 |
|          9 | Isha       | Varma     |         9 | Accounting         |          1009 |
|         10 | Ramya      | Joshi     |        10 | Law                |          1010 |
+------------+------------+-----------+-----------+--------------------+---------------+
10 rows in set (0.00 sec)
```