SQL LAB -9
(Inner Join,Outer Join,Left Outer Join)

Submitted by: Pravallika Ullikanti

AF ID : AF0366355

Batch ID : ANP-C7281

Lab: Use the Student Management System Database and table from previous lab.Perform the following commands on the table Student and Enrollment.

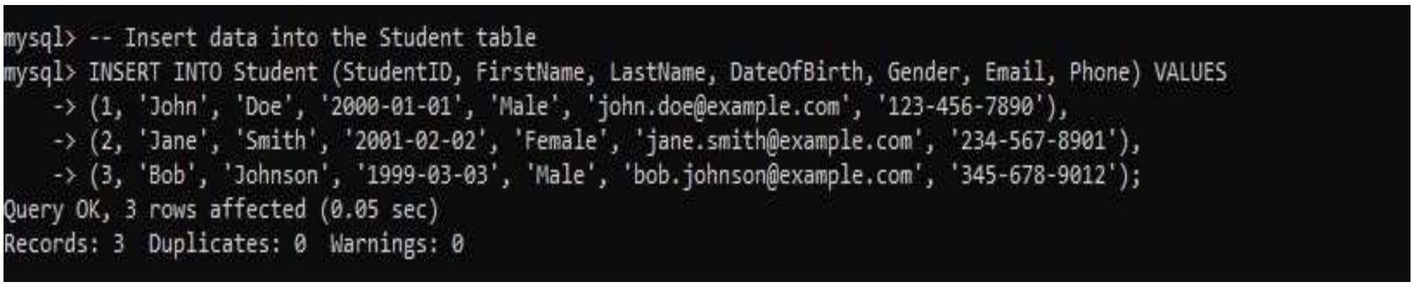
1. Let's consider a scenario where you have a database tracking student enrollments and some students may not be enrolled in any courses.

John Doe (StudentID: 1) is enrolled in courses with EnrollmentIDs 101 and 102.

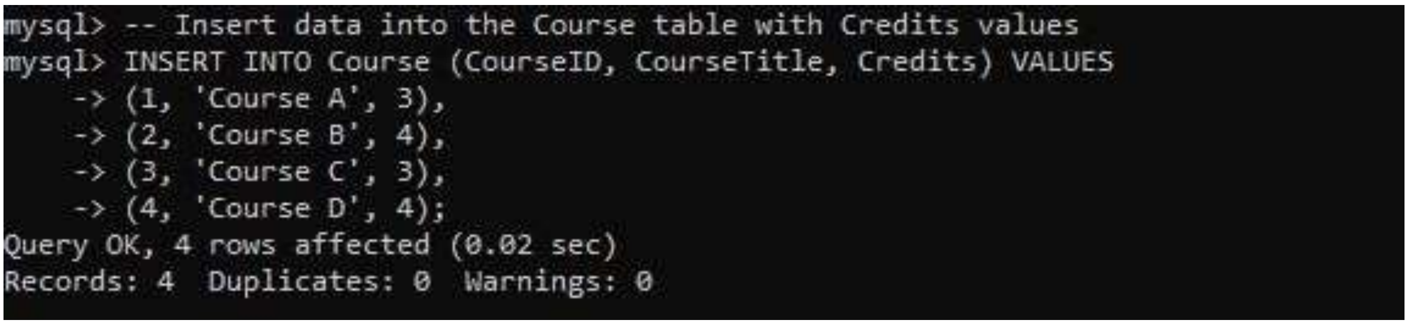
Jane Smith (StudentID: 2) is enrolled in courses with EnrollmentIDs 103 and 104.

Bob Johnson (StudentID: 3) is not enrolled in any courses.

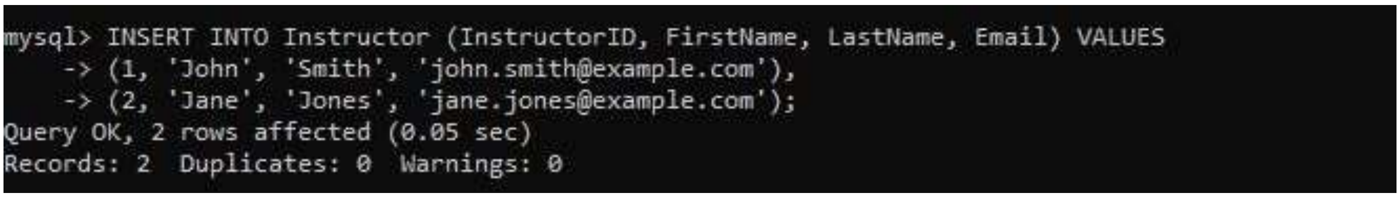
Now, run RIGHT OUTER JOIN query to retrieve data.



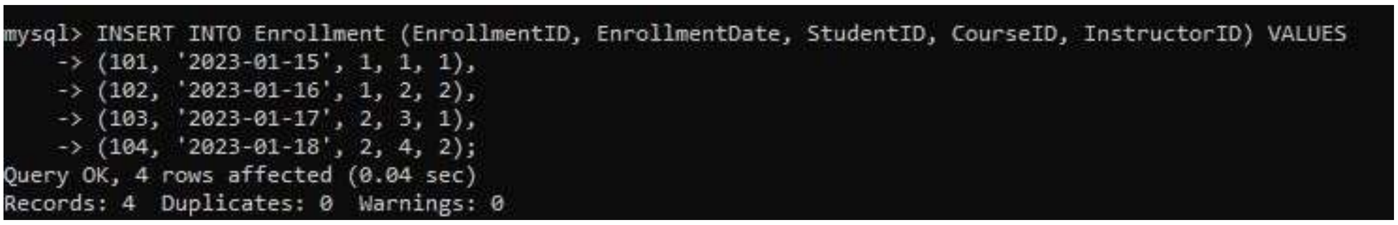
```
mysql> -- Insert data into the Student table
mysql> INSERT INTO Student (StudentID, FirstName, LastName, DateOfBirth, Gender, Email, Phone) VALUES
    -> (1, 'John', 'Doe', '2000-01-01', 'Male', 'john.doe@example.com', '123-456-7890'),
    -> (2, 'Jane', 'Smith', '2001-02-02', 'Female', 'jane.smith@example.com', '234-567-8901'),
    -> (3, 'Bob', 'Johnson', '1999-03-03', 'Male', 'bob.johnson@example.com', '345-678-9012');
Query OK, 3 rows affected (0.05 sec)
Records: 3  Duplicates: 0  Warnings: 0
```



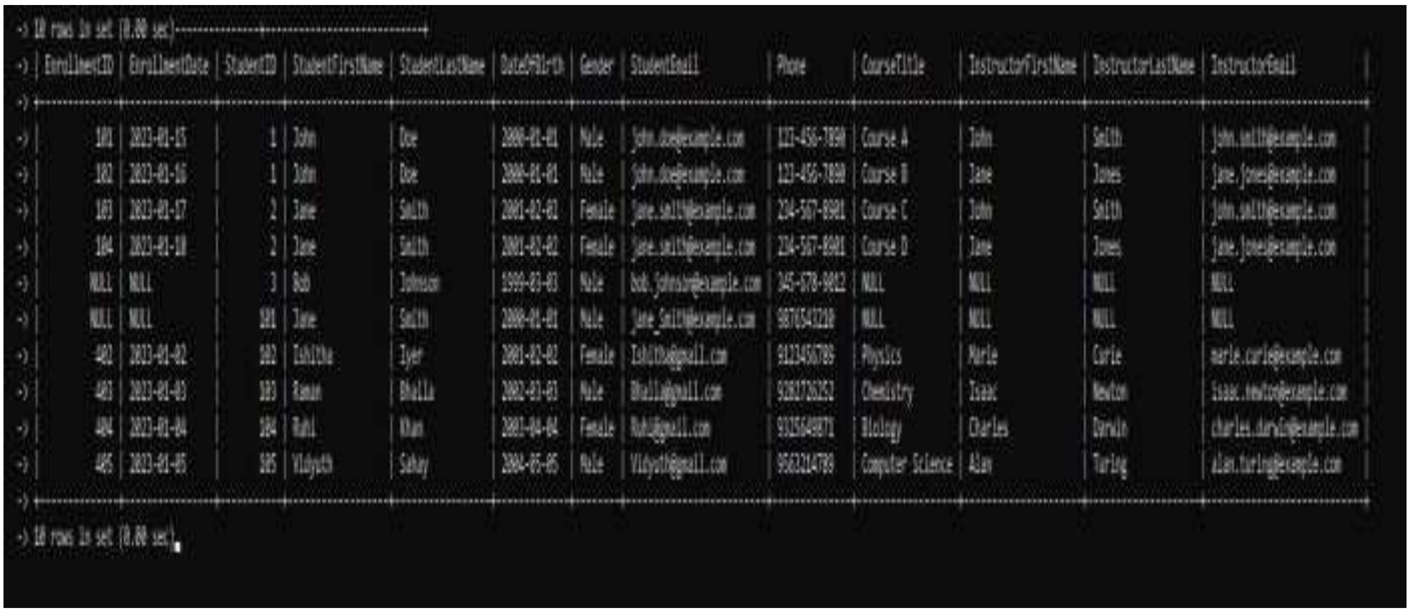
```
mysql> -- Insert data into the Course table with Credits values
mysql> INSERT INTO Course (CourseID, CourseTitle, Credits) VALUES
    -> (1, 'Course A', 3),
    -> (2, 'Course B', 4),
    -> (3, 'Course C', 3),
    -> (4, 'Course D', 4);
Query OK, 4 rows affected (0.02 sec)
Records: 4  Duplicates: 0  Warnings: 0
```



```
mysql> INSERT INTO Instructor (InstructorID, FirstName, LastName, Email) VALUES
    -> (1, 'John', 'Smith', 'john.smith@example.com'),
    -> (2, 'Jane', 'Jones', 'jane.jones@example.com');
Query OK, 2 rows affected (0.05 sec)
Records: 2  Duplicates: 0  Warnings: 0
```



```
mysql> INSERT INTO Enrollment (EnrollmentID, EnrollmentDate, StudentID, CourseID, InstructorID) VALUES
    -> (101, '2023-01-15', 1, 1, 1),
    -> (102, '2023-01-16', 1, 2, 2),
    -> (103, '2023-01-17', 2, 3, 1),
    -> (104, '2023-01-18', 2, 4, 2);
Query OK, 4 rows affected (0.04 sec)
Records: 4  Duplicates: 0  Warnings: 0
```



```
-> 10 rows in set (0.00 sec)+-------------+-----------------------+
-> | EnrollmentID | EnrollmentDate | StudentID | StudentFirstName | StudentLastName | DateOfBirth | Gender | StudentEmail           | Phone        | CourseTitle      | InstructorFirstName | InstructorLastName | InstructorEmail            |
-> +--------------+----------------+-----------+------------------+-----------------+-------------+--------+------------------------+--------------+------------------+---------------------+--------------------+----------------------------+
-> |          101 | 2023-01-15     |         1 | John             | Doe             | 2000-01-01  | Male   | john.doe@example.com   | 123-456-7890 | Course A         | John                | Smith              | john.smith@example.com     |
-> |          102 | 2023-01-16     |         1 | John             | Doe             | 2000-01-01  | Male   | john.doe@example.com   | 123-456-7890 | Course B         | Jane                | Jones              | jane.jones@example.com     |
-> |          103 | 2023-01-17     |         2 | Jane             | Smith           | 2001-02-02  | Female | jane.smith@example.com | 234-567-8901 | Course C         | John                | Smith              | john.smith@example.com     |
-> |          104 | 2023-01-18     |         2 | Jane             | Smith           | 2001-02-02  | Female | jane.smith@example.com | 234-567-8901 | Course D         | Jane                | Jones              | jane.jones@example.com     |
-> |         NULL | NULL           |         3 | Bob              | Johnson         | 1999-03-03  | Male   | bob.johnson@example.com | 345-678-9012 | NULL            | NULL                | NULL               | NULL                       |
-> |         NULL | NULL           |       101 | Jane             | Smith           | 2000-01-01  | Male   | jane_Smith@example.com | 9876543210   | NULL             | NULL                | NULL               | NULL                       |
-> |          402 | 2023-01-02     |       102 | Ishitha          | Iyer            | 2001-02-02  | Female | Ishitha@gmail.com      | 9123456789   | Physics          | Marie               | Curie              | marie.curie@example.com    |
-> |          403 | 2023-01-03     |       103 | Raman            | Bhalla          | 2002-03-03  | Male   | Bhalla@gmail.com       | 9282726252   | Chemistry        | Isaac               | Newton             | isaac.newton@example.com   |
-> |          404 | 2023-01-04     |       104 | Rahi             | Khan            | 2003-04-04  | Female | Rahi@gmail.com         | 9325648971   | Biology          | Charles             | Darwin             | charles.darwin@example.com |
-> |          405 | 2023-01-05     |       105 | Vidyuth          | Sahay           | 2004-05-05  | Male   | Vidyuth@gmail.com      | 9563214789   | Computer Science | Alan                | Turing             | alan.turing@example.com    |
-> +--------------+----------------+-----------+------------------+-----------------+-------------+--------+------------------------+--------------+------------------+---------------------+--------------------+----------------------------+
-> 10 rows in set (0.00 sec)
```

2. Assume a university where students can enroll in various courses. Here are some fiction details:

Student Information:

Student with ID 1: John, email: john@email.com

Student with ID 2: Jane, email: jane@email.com

Student with ID 3: Bob, email: bob@email.com

Enrollment Information:

Enrollment with ID 101: John (StudentID: 1) enrolls in Math (CourseID: MATH101).

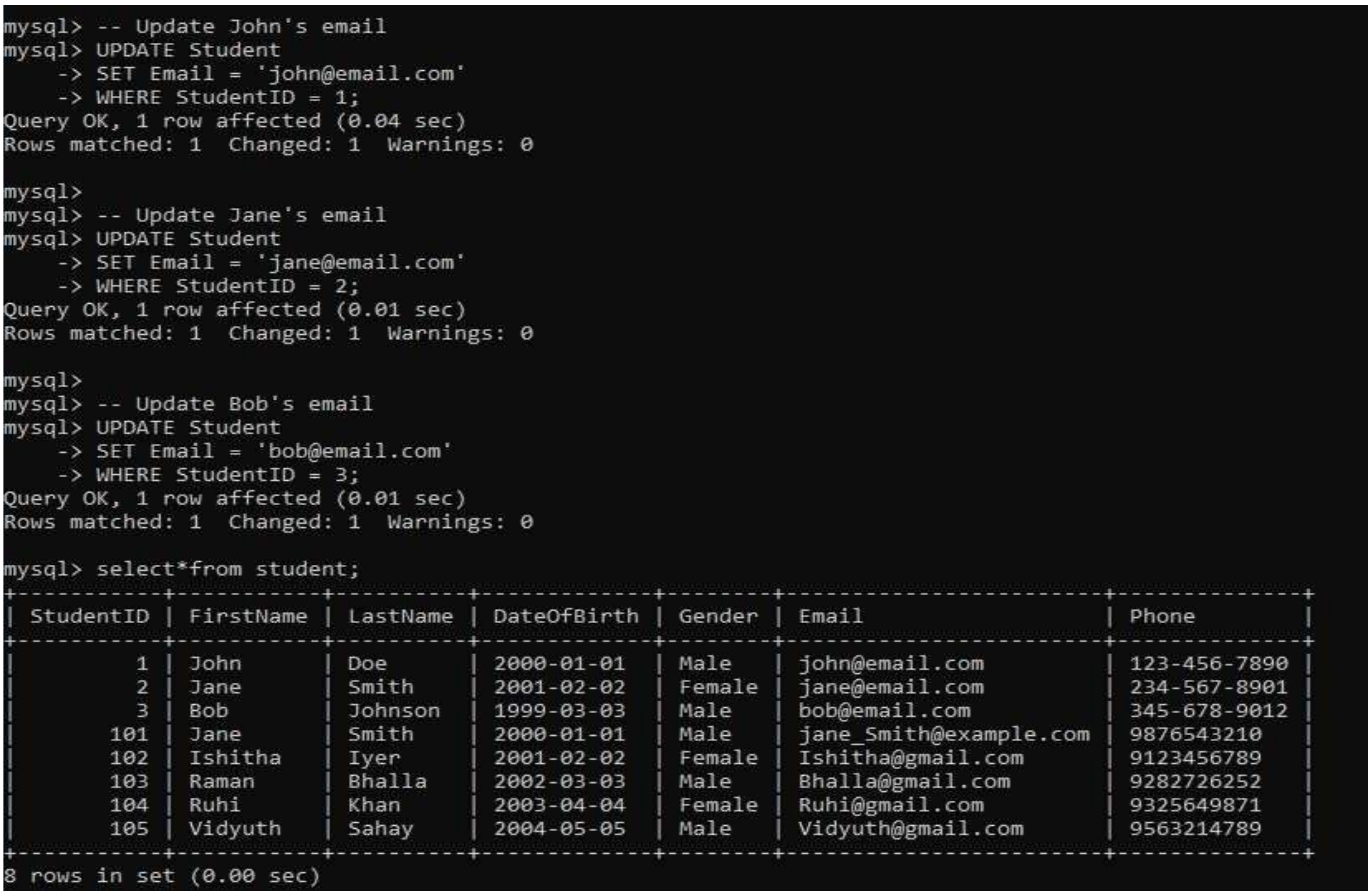
Enrollment with ID 102: John (StudentID: 1) enrolls in History (CourseID: HIST201).

Enrollment with ID 103: Jane (StudentID: 2) enrolls in Physics (CourseID: PHYS301).

Enrollment with ID 104: Bob (StudentID: 3) enrolls in Chemistry (CourseID: CHEM401).

Enrollment with ID 105: Alice (StudentID: 4) enrolls in English (CourseID: ENG501).

Now, write a LEFT JOIN query to retrieve the data.

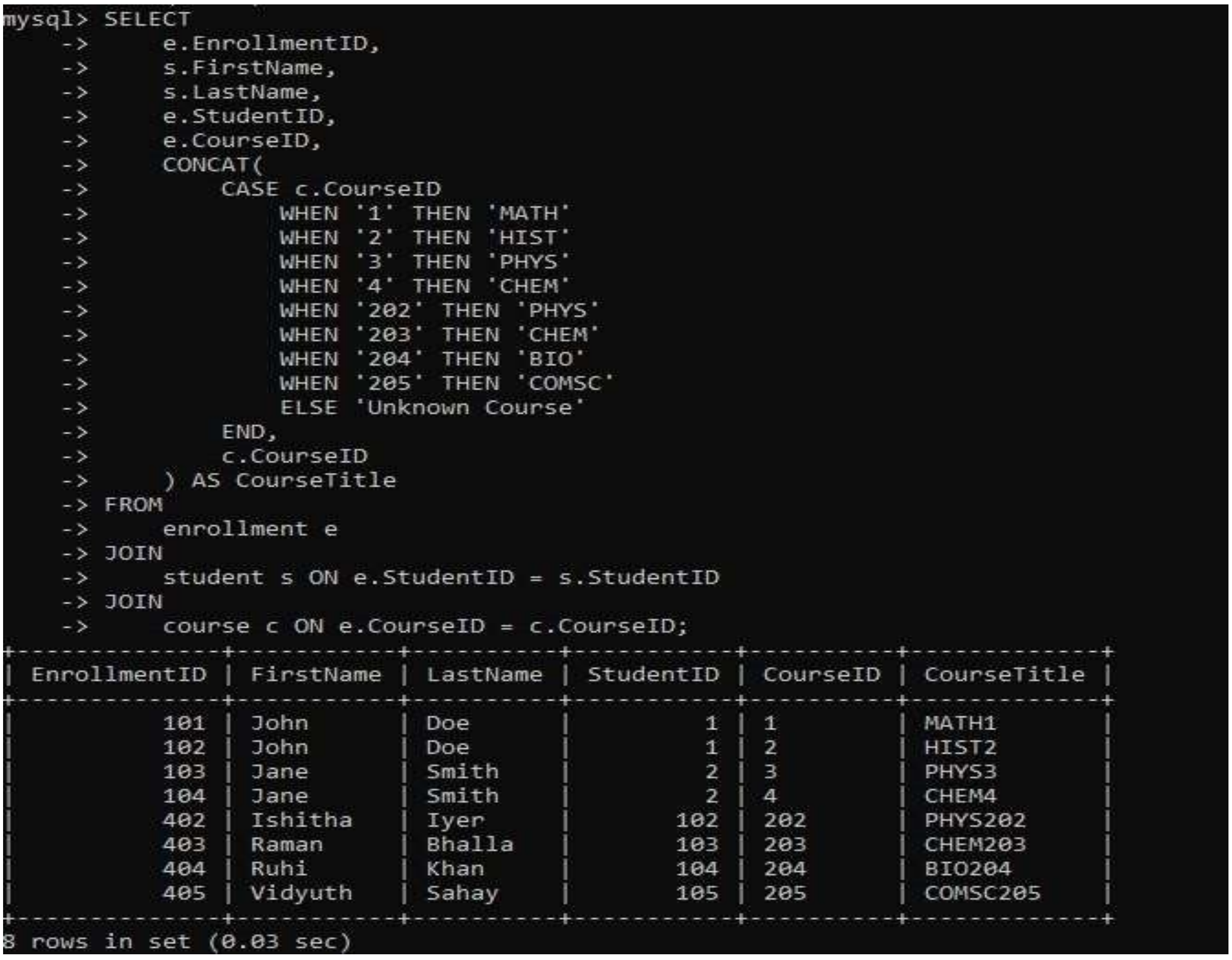


```
mysql> -- Update John's email
mysql> UPDATE Student
    -> SET Email = 'john@email.com'
    -> WHERE StudentID = 1;
Query OK, 1 row affected (0.04 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql>
mysql> -- Update Jane's email
mysql> UPDATE Student
    -> SET Email = 'jane@email.com'
    -> WHERE StudentID = 2;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql>
mysql> -- Update Bob's email
mysql> UPDATE Student
    -> SET Email = 'bob@email.com'
    -> WHERE StudentID = 3;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select*from student;
+-----------+-----------+----------+-------------+--------+-------------------------+--------------+
| StudentID | FirstName | LastName | DateOfBirth | Gender | Email                   | Phone        |
+-----------+-----------+----------+-------------+--------+-------------------------+--------------+
|         1 | John      | Doe      | 2000-01-01  | Male   | john@email.com          | 123-456-7890 |
|         2 | Jane      | Smith    | 2001-02-02  | Female | jane@email.com          | 234-567-8901 |
|         3 | Bob       | Johnson  | 1999-03-03  | Male   | bob@email.com           | 345-678-9012 |
|       101 | Jane      | Smith    | 2000-01-01  | Male   | jane_Smith@example.com  | 9876543210   |
|       102 | Ishitha   | Iyer     | 2001-02-02  | Female | Ishitha@gmail.com       | 9123456789   |
|       103 | Raman     | Bhalla   | 2002-03-03  | Male   | Bhalla@gmail.com        | 9282726252   |
|       104 | Ruhi      | Khan     | 2003-04-04  | Female | Ruhi@gmail.com          | 9325649871   |
|       105 | Vidyuth   | Sahay    | 2004-05-05  | Male   | Vidyuth@gmail.com       | 9563214789   |
+-----------+-----------+----------+-------------+--------+-------------------------+--------------+
8 rows in set (0.00 sec)
```



```
mysql> SELECT
    ->     e.EnrollmentID,
    ->     s.FirstName,
    ->     s.LastName,
    ->     e.StudentID,
    ->     e.CourseID,
    ->     CONCAT(
    ->         CASE c.CourseID
    ->             WHEN '1' THEN 'MATH'
    ->             WHEN '2' THEN 'HIST'
    ->             WHEN '3' THEN 'PHYS'
    ->             WHEN '4' THEN 'CHEM'
    ->             WHEN '202' THEN 'PHYS'
    ->             WHEN '203' THEN 'CHEM'
    ->             WHEN '204' THEN 'BIO'
    ->             WHEN '205' THEN 'COMSC'
    ->             ELSE 'Unknown Course'
    ->         END,
    ->         c.CourseID
    ->     ) AS CourseTitle
    -> FROM
    ->     enrollment e
    -> JOIN
    ->     student s ON e.StudentID = s.StudentID
    -> JOIN
    ->     course c ON e.CourseID = c.CourseID;
+--------------+-----------+----------+-----------+----------+-------------+
| EnrollmentID | FirstName | LastName | StudentID | CourseID | CourseTitle |
+--------------+-----------+----------+-----------+----------+-------------+
|          101 | John      | Doe      |         1 | 1        | MATH1       |
|          102 | John      | Doe      |         1 | 2        | HIST2       |
|          103 | Jane      | Smith    |         2 | 3        | PHYS3       |
|          104 | Jane      | Smith    |         2 | 4        | CHEM4       |
|          402 | Ishitha   | Iyer     |       102 | 202      | PHYS202     |
|          403 | Raman     | Bhalla   |       103 | 203      | CHEM203     |
|          404 | Ruhi      | Khan     |       104 | 204      | BIO204      |
|          405 | Vidyuth   | Sahay    |       105 | 205      | COMSC205    |
+--------------+-----------+----------+-----------+----------+-------------+
8 rows in set (0.03 sec)
```

ChatGPT Exercise

Using ChatGPT generates SQL queries of the below problem .

Scenario 1:You have two tables, employees and departments. Retrieve a list of employees along with their department names using an inner join.

Scenario 2:In an employee database, join the employees table with itself to display each employee along with their manager, including employees without managers, using a left join.

We have an "Employee" table with the following columns: EmployeeID,EmployeeName,ManagerID(Foreign Key) and "Manager" table with following columns:ManagerID,ManagerName.You want to retrieve each employee along with your manager.Generate a chatGPT prompt for the scenario.

Scenario 1: You have two tables, employees and departments. Retrieve a list of employees along with their department names using an inner join.

Scenario 2: In an employee database, join the employees table with itself to display each employee along with their manager, including employees without managers, using a left join.

We have an "Employee" table with the following columns: EmployeeID, EmployeeName, ManagerID (Foreign Key) and "Manager" table with following columns: ManagerID, ManagerName. You want to retrieve each employee along with their manager.

Sample Output:

1. Generate an SQL query for scenario 1.
2. Generate an SQL query for scenario 2.

Generated SQL Queries:

For scenario 1:

```
SELECT employees.EmployeeName, departments.DepartmentName

FROM employees

INNER JOIN departments ON employees.DepartmentID = departments.DepartmentID;
```

For scenario 2:

```
SELECT e.EmployeeName, m.ManagerName AS Manager

FROM Employee e

LEFT JOIN Manager m ON e.ManagerID = m.ManagerID;
```