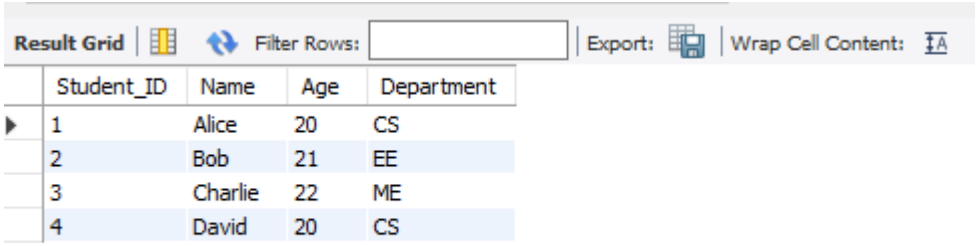
Database Management System Laboratory (MCA 291)

Assignment – 5

1. Create Students table and insert value:

```
CREATE TABLE Students (
    Student_ID INT PRIMARY KEY,
    Name VARCHAR(50),
    Age INT,
    Department VARCHAR(10)
);
INSERT INTO Students (Student_ID, Name, Age, Department) VALUES
(1, 'Alice', 20, 'CS'),
(2, 'Bob', 21, 'EE'),
(3, 'Charlie', 22, 'ME'),
(4, 'David', 20, 'CS');
```

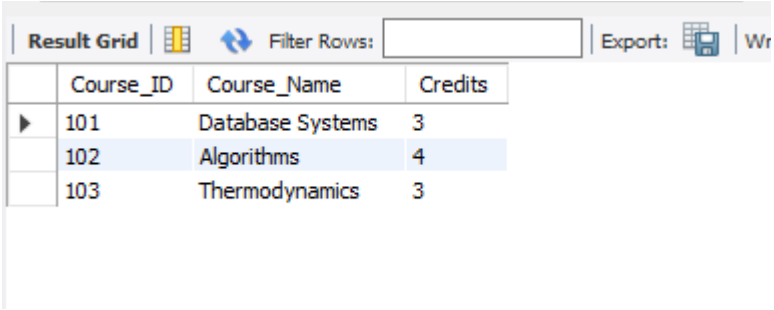


Student_ID	Name	Age	Department
1	Alice	20	CS
2	Bob	21	EE
3	Charlie	22	ME
4	David	20	CS

2. Create and insert into Courses table:

```
CREATE TABLE Courses (
    Course_ID INT PRIMARY KEY,
```

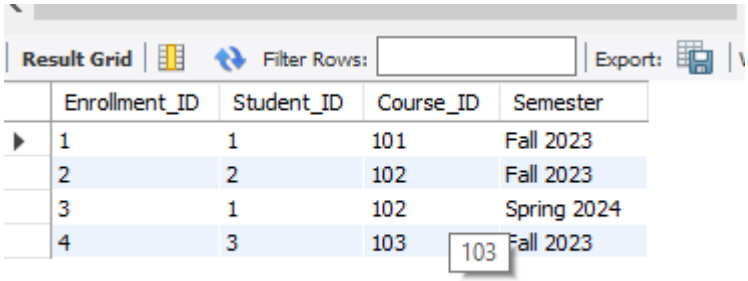
```
    Course_Name VARCHAR(50),
    Credits INT
);
INSERT INTO Courses (Course_ID, Course_Name, Credits) VALUES
(101, 'Database Systems', 3),
(102, 'Algorithms', 4),
(103, 'Thermodynamics', 3);
```



Course_ID	Course_Name	Credits
101	Database Systems	3
102	Algorithms	4
103	Thermodynamics	3

3. Create and insert into Enrollments table:

```
CREATE TABLE Enrollments (
    Enrollment_ID INT PRIMARY KEY,
    Student_ID INT,
    Course_ID INT,
    Semester VARCHAR(20)
);
INSERT INTO Enrollments (Enrollment_ID, Student_ID, Course_ID, Semester)
VALUES
(1, 1, 101, 'Fall 2023'),
(2, 2, 102, 'Fall 2023'),
(3, 1, 102, 'Spring 2024'),
(4, 3, 103, 'Fall 2023');
```

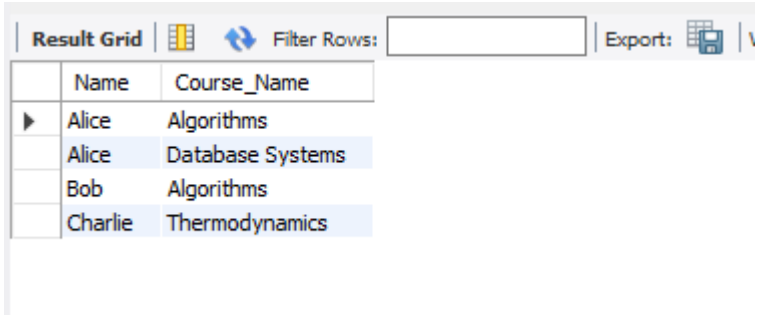


Enrollment_ID	Student_ID	Course_ID	Semester
1	1	101	Fall 2023
2	2	102	Fall 2023
3	1	102	Spring 2024
4	3	103	Fall 2023

- **INNER JOIN:**

- **Write a SQL query to display student names along with the courses they are enrolled in.**

  ⇨ SELECT Students.Name, Courses.Course_Name FROM Students
  INNER JOIN Enrollments ON Students.Student_ID = Enrollments.Student_ID
  INNER JOIN Courses ON Enrollments.Course_ID = Courses.Course_ID;



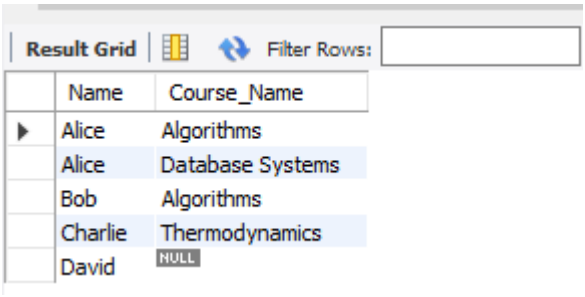
Name	Course_Name
Alice	Algorithms
Alice	Database Systems
Bob	Algorithms
Charlie	Thermodynamics

- **LEFT JOIN:**

- **Write a SQL query to display all students and the courses they are enrolled in. If a student is not enrolled in any course, display NULL for the course name.**

  ⇨ SELECT Students.Name, Courses.Course_Name FROM Students

LEFT JOIN Enrollments ON Students.Student_ID = Enrollments.Student_ID
LEFT JOIN Courses ON Enrollments.Course_ID = Courses.Course_ID;



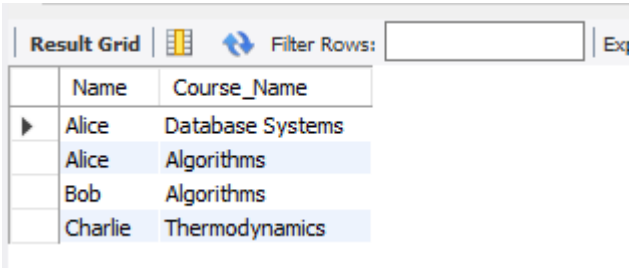
Result Grid | Filter Rows:

Name	Course_Name
Alice	Algorithms
Alice	Database Systems
Bob	Algorithms
Charlie	Thermodynamics
David	NULL

- **RIGHT JOIN:**

- **Write a SQL query to display all courses along with student names who have enrolled in them. If no student is enrolled in a course, display NULL for student names.**

⇨ SELECT Students.Name, Courses.Course_Name FROM Students
RIGHT JOIN Enrollments ON Students.Student_ID = Enrollments.Student_ID
RIGHT JOIN Courses ON Enrollments.Course_ID = Courses.Course_ID;



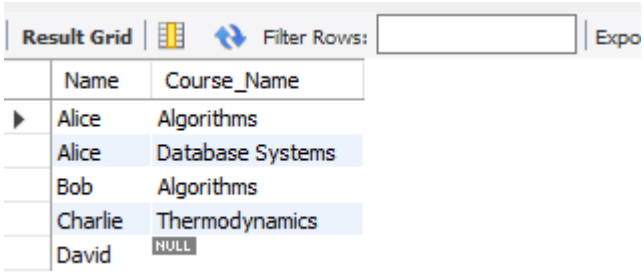
Result Grid | Filter Rows:

Name	Course_Name
Alice	Database Systems
Alice	Algorithms
Bob	Algorithms
Charlie	Thermodynamics

- **FULL JOIN:**

- **Write a SQL query to display all students and all courses, ensuring that students without courses and courses without students are also included.**

⇨ SELECT Students.Name, Courses.Course_Name FROM Students
FULL JOIN Enrollments ON Students.Student_ID = Enrollments.Student_ID
FULL JOIN Courses ON Enrollments.Course_ID = Courses.Course_ID;



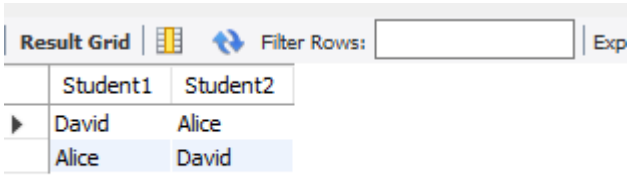
Result Grid | Filter Rows: | Expo

Name	Course_Name
Alice	Algorithms
Alice	Database Systems
Bob	Algorithms
Charlie	Thermodynamics
David	NULL

- **SELF JOIN:**

- **Write a SQL query to find students from the same department by joining the students table with itself.**

  ⇨ SELECT A.Name AS Student1, B.Name AS Student2 FROM Students A JOIN Students B ON A.Department = B.Department
  WHERE A.Student_ID != B.Student_ID;



Result Grid | Filter Rows: | Exp

Student1	Student2
David	Alice
Alice	David