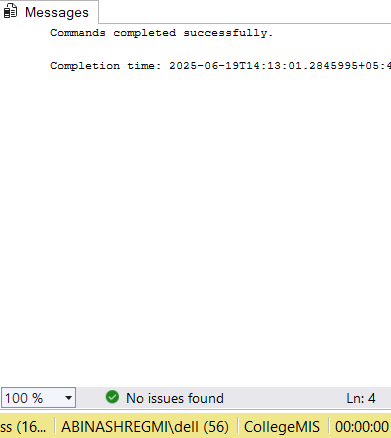
Lab 2: Use of Multiple Tables

**Create Database :**

**Query:** CREATE DATABASE CollegeMIS;

USE CollegeMIS;



**Creating Table:**

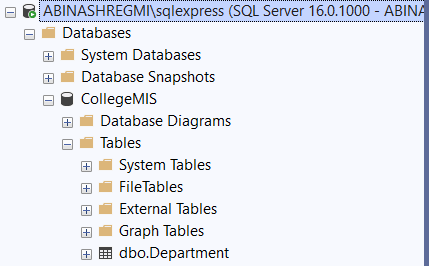
**Create Department table:**

**Query:** CREATE TABLE Department (

DepartmentID INT PRIMARY KEY,

DeptName VARCHAR(50)

);



**Create Student table:**

**Query : CREATE TABLE Student (**

**StudentID INT PRIMARY KEY,**

**Name VARCHAR(50),**

**Gender CHAR(1),**

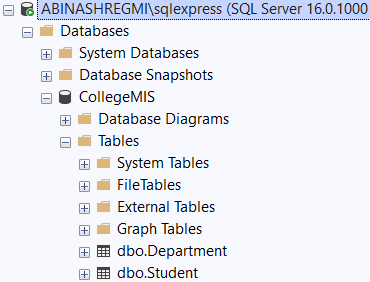
**DOB DATE,**

**DepartmentID INT,**

**Email VARCHAR(100),**

**FOREIGN KEY (DepartmentID) REFERENCES Department(DepartmentID)**

**);**

****

**Create Course table:**

**Query :** CREATE TABLE Course (

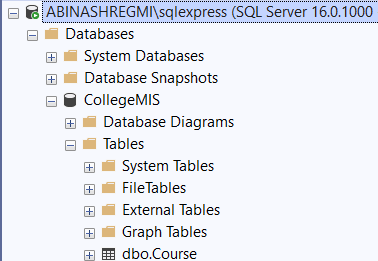
CourseID INT PRIMARY KEY,

CourseName VARCHAR(100),

DepartmentID INT,

FOREIGN KEY (DepartmentID) REFERENCES Department(DepartmentID)

);



**Create Enrollment table:**

**Query : CREATE TABLE Enrollment (**

**EnrollmentID INT PRIMARY KEY,**

**StudentID INT,**

**CourseID INT,**

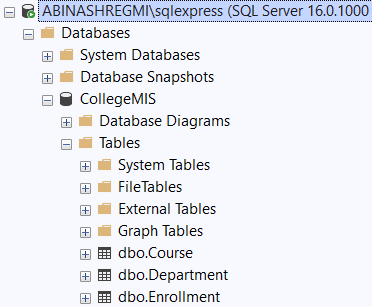
**Semester VARCHAR(10),**

**Marks INT,**

**FOREIGN KEY (StudentID) REFERENCES Student(StudentID),**

**FOREIGN KEY (CourseID) REFERENCES Course(CourseID)**

**);**

****

**Insert sample data into all four tables (Student, Department, Course, Enrollment).**

**Departments:**

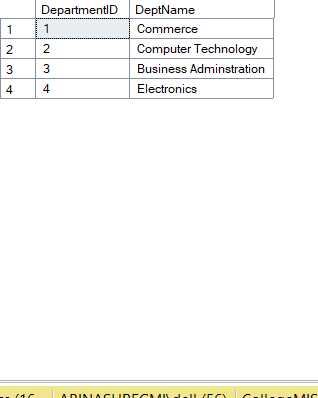
**Query : INSERT INTO Department(DepartmentID,DeptName) VALUES**

**(1,'Commerce'),**

**(2, 'Computer Technology'),**

**(3, 'Business Adminstration'),**

**(4, 'Electronics');**

****

**Students :**

**Query :**  **INSERT INTO Student (StudentID, Name, Gender, DOB, DepartmentID, Email)**

**VALUES**

**(201, 'Aayush Bhattarai', 'M', '2003-05-01','1', 'notaayush@gmail.com'),**

**(202, 'Albina Tamang', 'F', '2005-04-03','2' ,'albina@yahoo.com'),**

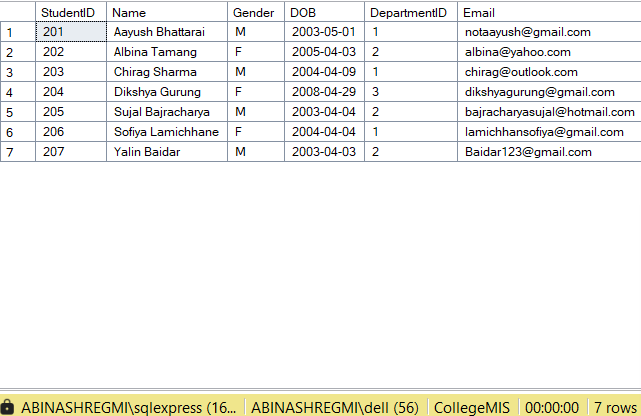
**(203, 'Chirag Sharma', 'M', '2004-04-09', '1','chirag@outlook.com'),**

**(204, 'Dikshya Gurung', 'F', '2008-04-29','3', 'dikshyagurung@gmail.com'),**

**(205, 'Sujal Bajracharya', 'M', '2003-04-04','2', 'bajracharyasujal@hotmail.com'),**

**(206, 'Sofiya Lamichhane','F', '2004-04-04','1', 'lamichhansofiya@gmail.com'),**

**(207, 'Yalin Baidar', 'M', '2003-04-03', '2' ,'Baidar123@gmail.com');**

****

**Courses :**

**Query : INSERT INTO Course (CourseID, CourseName, DepartmentID) VALUES**

**(201, 'Data Structures', 1),**

**(202, 'Operating Systems', 1),**

**(203, 'Database Systems', 2),**

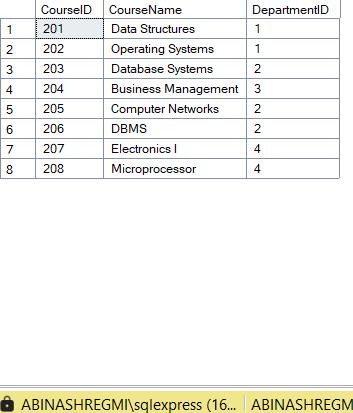
**(204, 'Business Management', 3),**

**(205, 'Computer Networks', 2),**

**(206, 'DBMS', 2),**

**(207, 'Electronics I', 4),**

**(208, 'Microprocessor', 4);**

****

**Enrollments:**

**Query : INSERT INTO Enrollment (EnrollmentID, StudentID, CourseID, Semester,**

**Marks) VALUES**

**(301, 204, 201, 'Semester3', 88),**

**(302, 201, 202, 'Semester3', 91),**

**(303, 202, 203, 'Semester3', 85),**

**(304, 203, 201, 'Semester3', 92),**

**(305, 203, 204, 'Semester3', 78),**

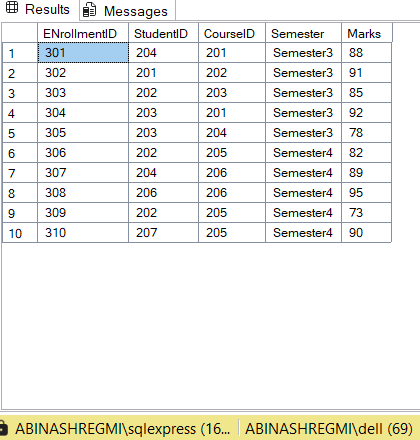
**(306, 202, 205, 'Semester4', 82),**

**(307, 204, 206, 'Semester4', 89),**

**(308, 206, 206, 'Semester4', 95),**

**(309, 202, 205, 'Semester4', 73),**

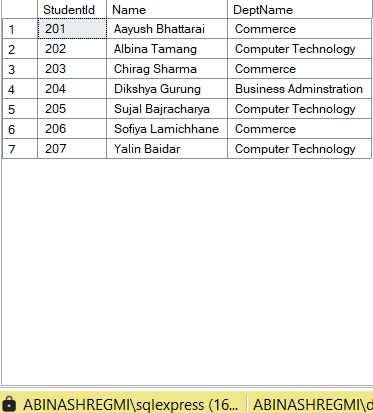
**(310, 207, 205, 'Semester4', 90);**

****

**Display all students with their department name using JOIN.**

**Query : SELECT s.StudentID, s.Name, d.DeptName FROM Student s**

**JOIN Department d ON s.DepartmentID = d.DepartmentID;**

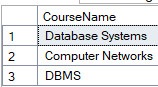
****

**Show all courses offered by the "Computer Technology" department.**

**Query : SELECT c.CourseName FROM Course c**

**JOIN Department d ON c.DepartmentID = d.DepartmentID**

**WHERE d.DeptName = 'Computer Technology;**

****

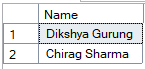
**List all students enrolled in "Data Structures" course.**

**Query : SELECT s.Name FROM Student s**

**JOIN Enrollment e ON s.StudentID = e.StudentID**

**JOIN Course c ON e.CourseID = c.CourseID**

**WHERE c.CourseName = 'Data Structures';**

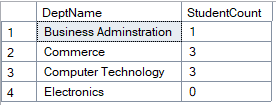
****

**Count the number of students in each department.**

**Query :** **SELECT d.DeptName, COUNT(s.StudentID) AS StudentCount FROM Department d**

**LEFT JOIN Student s ON d.DepartmentID = s.DepartmentID**

**GROUP BY d.DeptName;**

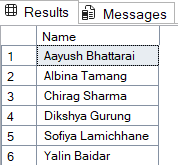
****

**List students who scored more than 80 in any course.**

**Query :** **SELECT DISTINCT s.Name FROM Student s**

**JOIN Enrollment e ON s.StudentID = e.StudentID**

**WHERE e.Marks > 80;**

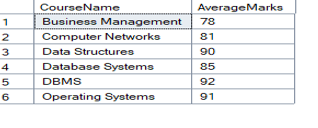
****

**Find the average marks for each course using GROUP BY.**

**Query :** **SELECT c.CourseName, AVG(e.Marks) AS AverageMarks FROM Course c**

**JOIN Enrollment e ON c.CourseID = e.CourseID**

**GROUP BY c.CourseName;**

****

**Display students who have not enrolled in any course.**

**Query :** **SELECT s.Name FROM Student s**

**LEFT JOIN Enrollment e ON s.StudentID = e.StudentID**

**WHERE e.EnrollmentID IS NULL;**

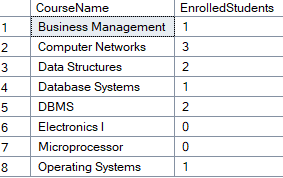
****

**List all courses with the number of students enrolled in them.**

**Query :** **SELECT c.CourseName, COUNT(e.StudentID) AS EnrolledStudents FROM Course c**

**LEFT JOIN Enrollment e ON c.CourseID = e.CourseID**

**GROUP BY c.CourseName;**

****

**Find the student with the highest marks in "DBMS" course.**

**Query :** **SELECT TOP 1 s.Name, e.Marks FROM Student s**

**JOIN Enrollment e ON s.StudentID = e.StudentID**

**JOIN Course c ON e.CourseID = c.CourseID**

**WHERE c.CourseName = 'DBMS'**

**ORDER BY e.Marks DESC;**

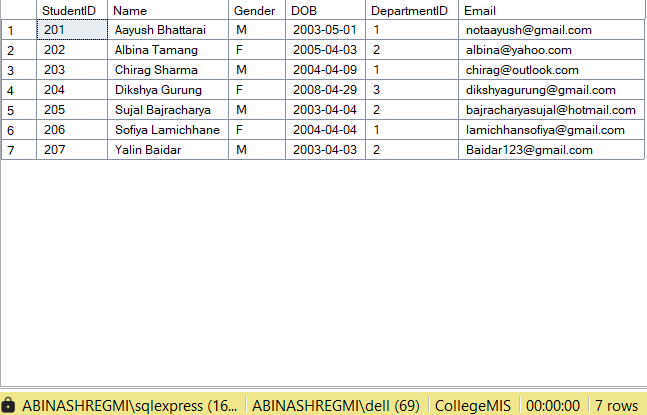
****

**Update a student’s department based on StudentID.**

**Query :** **UPDATE Student**

**SET DepartmentID = 3**

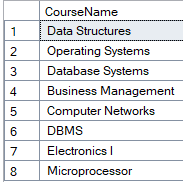
**WHERE StudentID = 109;**

**After :**

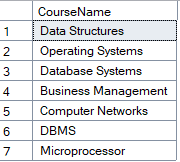
**Delete a course from the course list and ensure cascade deletion in enrollment.**

**Query :** **DELETE FROM Course**

**WHERE CourseName = 'Electronics I';**

**Before:**

**After:**

****

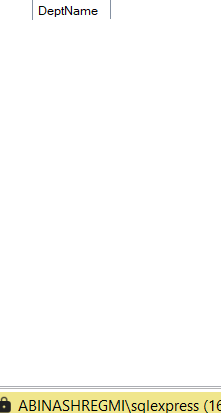
**Find the departments that offer more than 3 courses.**

**Query :** **SELECT d.DeptName FROM Department d**

**JOIN Course c ON d.DepartmentID = c.DepartmentID**

**GROUP BY d.DeptName**

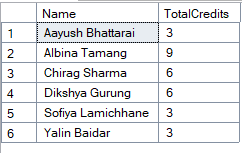
**HAVING COUNT(c.CourseID) > 3;**

****

**Show the names of students along with total credits they are enrolled in** **(assuming each course = 3 credits).**

**Query :** **SELECT s.Name, COUNT(e.CourseID) \* 3 AS TotalCredits FROM Student s**

**JOIN Enrollment e ON s.StudentID = e.StudentID**

**GROUP BY s.Name;**

**Create a view TopPerformers that shows students with average marks > 85.**

**Query : GO**

**CREATE VIEW TopPerformers AS**

**SELECT s.StudentID, s.Name, AVG(e.Marks) AS AvgMarks**

**FROM Student s**

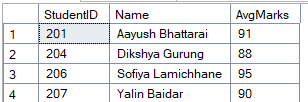
**JOIN Enrollment e ON s.StudentID = e.StudentID**

**GROUP BY s.StudentID, s.Name**

**HAVING AVG(e.Marks) > 85;**

**GO**

**Data in the view TopPerformers :**

****

**Create a view StudentDetails to show StudentID, Name, Department, and Email.**

**Query : GO**

**CREATE VIEW StudentDetails AS**

**SELECT s.StudentID, s.Name, d.DeptName AS Department, s.Email FROM Student s**

**JOIN Department d ON s.DepartmentID = d.DepartmentID;**

**GO**

**Data in the view StudentDetails :**

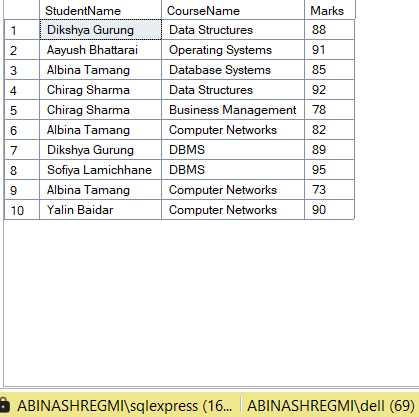
****

**Write a query to show student name, course name, and marks for all enrollments.**

**Query : SELECT s.Name AS StudentName, c.CourseName, e.Marks FROM Enrollment e**

**JOIN Student s ON e.StudentID = s.StudentID**

**JOIN Course c ON e.CourseID = c.CourseID;**

****

**Find students who have enrolled in courses from more than one department.**

**Query : SELECT s.Name FROM Student s**

**JOIN Enrollment e ON s.StudentID = e.StudentID**

**JOIN Course c ON e.CourseID = c.CourseID**

**GROUP BY s.StudentID, s.Name**

**HAVING COUNT(DISTINCT c.DepartmentID) > 1;**

****

**Display the list of departments that currently have no students.**

**Query : SELECT d.DeptName FROM Department d**

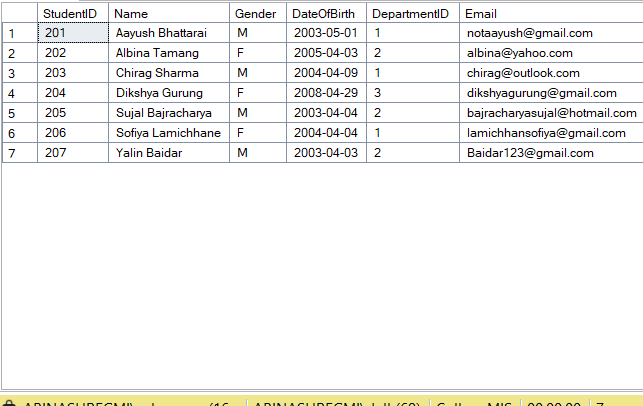
**LEFT JOIN Student s ON d.DepartmentID = s.DepartmentID**

**WHERE s.StudentID IS NULL;**

****

**Rename the column DOB in Student to DateOfBirth.**

**Query : EXEC sp\_rename 'Student.DOB', 'DateOfBirth', 'COLUMN';**

****

**Find students who are both in the "CT" department and enrolled in "Computer Networks"**

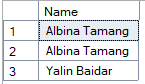
**Query :** **SELECT s.Name FROM Student s**

**JOIN Department d ON s.DepartmentID = d.DepartmentID**

**JOIN Enrollment e ON s.StudentID = e.StudentID**

**JOIN Course c ON e.CourseID = c.CourseID**

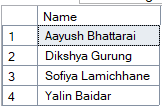
**WHERE d.DeptName = Computer Technology' AND c.CourseName = 'Computer Networks';**

****

**Find students whose email ends with '@gmail.com'**

**Query :** **SELECT Name FROM Student**

**WHERE Email LIKE '%@gmail.com';**

****

**Display students who scored more than the average mark of all students.**

**Query :** **SELECT s.Name, e.Marks FROM Student s**

**JOIN Enrollment e ON s.StudentID = e.StudentID**

**WHERE e.Marks > (SELECT AVG(Marks) FROM Enrollment);**

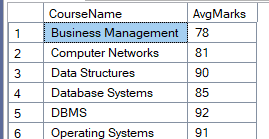
****

**Show the average marks of students in each course.**

**Query :** **SELECT c.CourseName, AVG(e.Marks) AS AvgMarks FROM Course c**

**JOIN Enrollment e ON c.CourseID = e.CourseID**

**GROUP BY c.CourseName;**

****

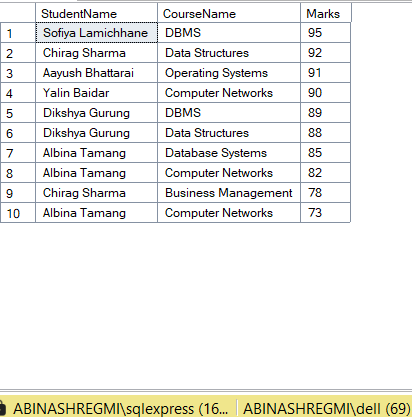
**Show all students with course name and sort by Marks descending.**

**Query : SELECT s.Name AS StudentName, c.CourseName, e.Marks FROM Student s**

**JOIN Enrollment e ON s.StudentID = e.StudentID**

**JOIN Course c ON e.CourseID = c.CourseID**

**ORDER BY e.Marks DESC;**

****