

SQL Internship – Task 2

```
-- Create Courses table
DROP TABLE IF EXISTS Courses;
CREATE TABLE Courses (
    course_id INTEGER PRIMARY KEY AUTOINCREMENT,
    course_name VARCHAR(50),
    course_description VARCHAR(100)
);

-- Create Enrolments table
DROP TABLE IF EXISTS Enrolments;
CREATE TABLE Enrolments (
    enrolment_id INTEGER PRIMARY KEY AUTOINCREMENT,
    student_id INT,
    course_id INT,
    enrolment_date DATE,
    FOREIGN KEY(student_id) REFERENCES Students(StudentID),
    FOREIGN KEY(course_id) REFERENCES Courses(course_id)
);

-- Insert courses
INSERT INTO Courses (course_name, course_description) VALUES
('Mathematics', 'Math course covering algebra and geometry'),
('Science', 'General science course'),
('English', 'English language and literature course'),
('Computer Science', 'Basics of programming and algorithms');

-- Insert enrolments
INSERT INTO Enrolments (student_id, course_id, enrolment_date) VALUES
(1, 1, '2025-01-10'),
(1, 2, '2025-01-11'),
(2, 1, '2025-01-12'),
(3, 1, '2025-01-10'),
(3, 2, '2025-01-11'),
(3, 4, '2025-01-15'),
(4, 3, '2025-01-13'),
(5, 2, '2025-01-12'),
(6, 1, '2025-01-14'),
(6, 3, '2025-01-15');

-- Query 1: List students with courses
SELECT s.Name AS StudentName, c.course_name AS CourseName
FROM Students s
INNER JOIN Enrolments e ON s.StudentID = e.student_id
INNER JOIN Courses c ON e.course_id = c.course_id
ORDER BY s.StudentID;

-- Query 2: Count students per course
SELECT c.course_name, COUNT(e.student_id) AS StudentCount
FROM Courses c
LEFT JOIN Enrolments e ON c.course_id = e.course_id
GROUP BY c.course_id, c.course_name;

-- Query 3: Students enrolled in more than one course
SELECT s.Name AS StudentName, COUNT(e.course_id) AS CoursesEnrolled
FROM Students s
INNER JOIN Enrolments e ON s.StudentID = e.student_id
```

```
GROUP BY s.StudentID, s.Name  
HAVING COUNT(e.course_id) > 1;
```

```
-- Query 4: Course with highest enrolments  
SELECT c.course_name, COUNT(e.student_id) AS StudentCount  
FROM Courses c  
INNER JOIN Enrolments e ON c.course_id = e.course_id  
GROUP BY c.course_id, c.course_name  
ORDER BY StudentCount DESC  
LIMIT 1;
```

```
-- Delete enrolment example  
DELETE FROM Enrolments WHERE student_id = 1 AND course_id = 2;
```