Lecture 4 Activity: Advanced SQL & Query Optimization

(Medium Learner)

Case Study: Academic Records Management System

A university maintains records of its students, courses, instructors, departments, and exam results. The aim is to help the university administration generate useful reports using SQL, optimize performance, and make informed academic decisions.

Database Schema:

Students(StudentID, Name, DepartmentID, Year)

Departments(DepartmentID, DepartmentName)

Courses(CourseID, CourseName, DepartmentID)

Results(ResultID, StudentID, CourseID, Marks)

Use the schema above to solve the following tasks. Assume sample data exists in all tables. If needed, create small sample datasets for testing.

1. Find the names of students who scored **above average marks** across all results.

```
SELECT s.Name
FROM Students s
JOIN Results r ON s.StudentID = r.StudentID
GROUP BY s.StudentID, s.Name
HAVING AVG(r.Marks) > (
    SELECT AVG(Marks) FROM Results
);
```

2. List students who took courses **outside their department**.

```
SELECT DISTINCT s.Name
FROM Students s
JOIN Results r ON s.StudentID = r.StudentID
JOIN Courses c ON r.CourseID = c.CourseID
WHERE s.DepartmentID <> c.DepartmentID;
```

3. Get the names of students who took the course **"Data Structures"**.

```
SELECT DISTINCT s.Name
FROM Students s
JOIN Results r ON s.StudentID = r.StudentID
JOIN Courses c ON r.CourseID = c.CourseID
WHERE c.CourseName = 'Data Structures';
```

4. List students who scored **more than any student in the Electrical department**.

```
SELECT DISTINCT s.Name
FROM Students s
JOIN Results r ON s.StudentID = r.StudentID
WHERE r.Marks > (
    SELECT MAX(r2.Marks)
    FROM Results r2
    JOIN Students s2 ON r2.StudentID = s2.StudentID
    WHERE s2.DepartmentID = (
        SELECT DepartmentID FROM Departments WHERE DepartmentName = 'Electrical'
    )
);
```

5. Get students who are either in the **CSE department** or have taken a course in the CSE department (using `UNION`).

```
SELECT Name
FROM Students
WHERE DepartmentID = (
    SELECT DepartmentID FROM Departments WHERE DepartmentName = 'CSE'
)
UNION
SELECT DISTINCT s.Name
FROM Students s
JOIN Results r ON s.StudentID = r.StudentID
JOIN Courses c ON r.CourseID = c.CourseID
WHERE c.DepartmentID = (
    SELECT DepartmentID FROM Departments WHERE DepartmentName = 'CSE'
);
```

6. Find **CSE students who have NOT taken** the course **"Operating Systems"**.

```
SELECT Name
FROM Students
WHERE DepartmentID = (
    SELECT DepartmentID FROM Departments WHERE DepartmentName = 'CSE'
)
AND StudentID NOT IN (
    SELECT r.StudentID
    FROM Results r
    JOIN Courses c ON r.CourseID = c.CourseID
    WHERE c.CourseName = 'Operating Systems'
);
```

7. Show student names and their **average marks above 75**.

```
SELECT s.Name, AVG(r.Marks) AS AvgMarks

FROM Students s

JOIN Results r ON s.StudentID = r.StudentID

GROUP BY s.StudentID, s.Name

HAVING AVG(r.Marks) > 75;
```

8. Suggest optimization for fetching student names and marks greater than 90.

```
-- Create index on Marks to speed up search

CREATE INDEX idx_marks ON Results(Marks);


-- Optimized query

SELECT s.Name, r.Marks

FROM Students s

JOIN Results r ON s.StudentID = r.StudentID

WHERE r.Marks > 90;
```