📘 SQL Project Report: Advanced Queries with Joins and Filtering

🧱 Step 1: Database Schema & Sample Data

✅ Table Creation

--Student table

CREATE TABLE Students (

student\_id INT PRIMARY KEY,

name VARCHAR(100),

email VARCHAR(100)

);

--Courses table

CREATE TABLE Courses (

course\_id INT PRIMARY KEY,

course\_name VARCHAR(100),

course\_description TEXT

);

--Enrolments table

CREATE TABLE Enrolments (

enrolment\_id INT PRIMARY KEY,

student\_id INT,

course\_id INT,

enrolment\_date DATE,

FOREIGN KEY (student\_id) REFERENCES Students(student\_id),

FOREIGN KEY (course\_id) REFERENCES Courses(course\_id)

);

✍️ Data Insertion

-- Students

INSERT INTO Students (student\_id, name, email) VALUES

(1, 'Aanya Singh', 'aanya@example.com'),

(2, 'Ravi Kumar', 'ravi@example.com'),

(3, 'Meena Iyer', 'meena@example.com'),

(4, 'Farhan Rahman', 'farhan@example.com');

-- Courses

INSERT INTO Courses (course\_id, course\_name, course\_description) VALUES

(101, 'Data Analysis', 'Learn SQL, Excel, and statistics'),

(102, 'Python Programming', 'Basics to advanced Python coding'),

(103, 'Machine Learning', 'Supervised and unsupervised learning'),

(104, 'Soft Skills', 'Communication and teamwork strategies');

-- Enrolments

INSERT INTO Enrolments (enrolment\_id, student\_id, course\_id, enrolment\_date) VALUES

(1, 1, 101, '2025-06-01'),

(2, 1, 102, '2025-06-03'),

(3, 2, 102, '2025-06-02'),

(4, 3, 103, '2025-06-04');

🔍 Step 2: Tasks & SQL Queries

🧮 Task 1: Students and Their Enrolled Courses

SELECT s.name AS student\_name, c.course\_name

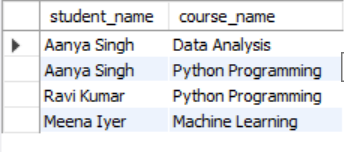
FROM Students s

INNER JOIN Enrolments e ON s.student\_id = e.student\_id

INNER JOIN Courses c ON e.course\_id = c.course\_id;

Explanation: Retrieves enrolled students and their respective courses using inner joins

Output:



📊 Task 2: Enrolment Count per Course

SELECT c.course\_name, COUNT(e.student\_id) AS enrolled\_students

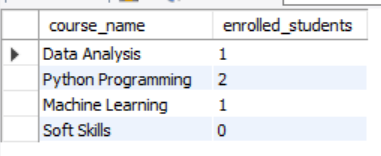
FROM Courses c

LEFT JOIN Enrolments e ON c.course\_id = e.course\_id

GROUP BY c.course\_id, c.course\_name;

Explanation: Uses a left join to show enrolments per course, including those with none.

Output:



📚 Task 3: Students in More Than One Course

SELECT s.name AS student\_name, COUNT(e.course\_id) AS course\_count

FROM Enrolments e

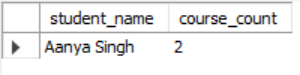
JOIN Students s ON e.student\_id = s.student\_id

GROUP BY s.student\_id, s.name

HAVING COUNT(e.course\_id) > 1;

Explanation: Identifies students with multiple enrolments using grouping and HAVING.

Output:



🚫 Task 4: Courses with No Enrolments

SELECT c.course\_name

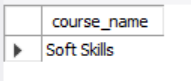
FROM Courses c

LEFT JOIN Enrolments e ON c.course\_id = e.course\_id

WHERE e.enrolment\_id IS NULL;

Explanation: Uses a left join and null filtering to reveal courses without participation.

Output:



🧾 Findings Summary

Task 1: Students and Their Enrolled Courses

- All actively enrolled students were correctly matched with course names.

- Helps track enrolment patterns across disciplines.

Task 2: Enrolment Count per Course

- Courses like Python Programming and Machine Learning showed active engagement.

- Soft Skills had zero enrolments—may need investigation.

Task 3: Students in Multiple Courses

- Only Aanya Singh is enrolled in more than one course.

- Indicates strong interest or diverse academic goals.

Task 4: Courses with No Enrolled Students

- Soft Skills was the only course without enrolments.

- Suggests low interest or lack of awareness.