

```

CREATE TABLE students (
    student_id INT PRIMARY KEY,
    name VARCHAR(50),
    department VARCHAR(50),
    year INT
);

CREATE TABLE subjects (
    subject_id INT PRIMARY KEY,
    subject_name VARCHAR(50)
);

CREATE TABLE marks (
    mark_id INT PRIMARY KEY,
    student_id INT,
    subject_id INT,
    marks INT,
    attendance_percentage DECIMAL(5,2),
    FOREIGN KEY (student_id) REFERENCES students (student_id),
    FOREIGN KEY (subject_id) REFERENCES subjects (subject_id)
);

INSERT INTO students
VALUES (1, 'Aarav', 'Computer Science', 1),
      (2, 'Meera', 'Computer Science', 1),
      (3, 'Rohan', 'Electronics', 2),
      (4, 'Ananya', 'Mechanical', 2),
      (5, 'Kabir', 'Electronics', 1);

INSERT INTO subjects
VALUES (101, 'Mathematics'),
      (102, 'Programming'),
      (103, 'Physics');

INSERT INTO marks VALUES
(1, 1, 101, 85, 92.5),
(2, 1, 102, 90, 95.0),
(3, 2, 101, 72, 78.0),
(4, 2, 102, 65, 70.0),
(5, 3, 103, 55, 68.0),
(6, 3, 101, 60, 72.5),
(7, 4, 103, 48, 65.0),
(8, 5, 102, 88, 90.0);

SELECT * FROM students;
SELECT * FROM subjects;
SELECT * FROM marks;

--SECTION A: Data Understanding

--Q1. Display all records from the students table
SELECT * FROM students;

```

```
--Q2. List all unique departments available in the database.
SELECT DISTINCT department FROM students;
```

```
--Q3. Show all subjects offered.
SELECT * FROM subjects;
```

```
--Question 4. Count the total number of students in the database.
SELECT count(student_id) as total_students FROM students;
```

```
--Question 5. Count how many students are enrolled in each department.
select department, count(student_id) as student_count
group by department;
```

--SECTION B: Basic Performance Analysis

```
--Question 6. Display each student's marks along with their subject names.
select s.name as student_name, sub.subject_name, m.marks
from marks m
join students s on m.student_id = s.student_id
join subjects sub on m.subject_id = sub.subject_id;
```

```
--Question 7 - Find the average marks scored in each subject.
select sub.subject_name, avg(m.marks) as avg_marks
join subjects sub on m.subject_id=sub.subject_id
group by sub.subject_name;
```

```
--Question 8 - Identify the highest and lowest marks in each subject.
select sub.subject_name, max(m.marks) as highest_marks, min(m.marks) as lowest_marks
join subjects sub on m.subject_id=sub.subject_id
group by sub.subject_name;
```

```
--Question 9 - List students who scored more than 80 marks in any subject.
select s.name as student_name , m.marks from marks m
join students s on s.student_id=m.student_id
where m.marks > 80;
```

```
--Question 10 - Count how many students appear in each subject.
SELECT sub.subject_name, COUNT(m.student_id) as student_count
JOIN subjects sub ON m.subject_id = sub.subject_id
GROUP BY sub.subject_name;
```

--SECTION C - Attendance Analysis

```
--Question 11 Show students whose attendance is less than 75%.
select s.name , m.attendance_percentage from marks m
JOIN students s ON m.student_id = s.student_id
where m.attendance_percentage < 75;
```

```
--Question 12 - Calculate the average attendance percentage for each subject.
select sub.subject_name , avg(m.attendance_percentage) as avg_attendance
join subjects sub on m.subject_id=sub.subject_id
group by sub.subject_name;
```

```
--Question 13 - Find students who have both attended and scored marks in a subject.
select s.name , m.attendance_percentage , m.marks
join students s on s.student_id=m.student_id
join marks m on m.student_id=s.student_id;
```

```

where m.attendance_percentage < 75 and m.marks > 50;

--Question 14 - Determine the percentage of students with attendance less than 75%
SELECT (COUNT(DISTINCT student_id) * 100.0 / (SELECT COUNT(DISTINCT student_id) FROM marks)) AS percentage
FROM marks
WHERE attendance_percentage < 75;

--Question 15 Rank students based on attendance percentage
SELECT
    s.name AS student_name,
    m.attendance_percentage,
    RANK() OVER (ORDER BY m.attendance_percentage) AS rank
FROM marks m
JOIN students s ON m.student_id = s.student_id;

--SECTION D: Combined Academic & Attendance Information
--Question 16 Calculate each student's overall average marks
SELECT
    s.name AS student_name,
    AVG(m.marks) AS average_marks
FROM marks m
JOIN students s ON m.student_id = s.student_id
GROUP BY s.name;

--Question 17 - Rank students based on their average marks
SELECT
    student_name,
    average_marks,
    RANK() OVER (ORDER BY average_marks DESC) AS rank
FROM (
    SELECT
        s.name AS student_name,
        AVG(m.marks) AS average_marks
    FROM marks m
    JOIN students s ON m.student_id = s.student_id
    GROUP BY s.name
) AS avg_table;

```

Student Performance & Attendance Analysis (SQL Project)

Project Overview

This project focuses on analyzing student performance and attendance data using SQL. The objective is to extract meaningful insights that help understand academic trends, attendance impact, and overall student performance.

Objectives

- Analyze student attendance patterns
- Evaluate academic performance using marks and grades
- Identify relationships between attendance and performance
- Practice real-world SQL queries on structured data

Dataset Description

Tables used:

- Students
- Attendance
- Marks

Connected using Student ID.

Tools Used

- SQL
- Google Colab

SQL Concepts

- JOIN
- GROUP BY
- Aggregate functions
- Filtering

Sample Queries

```
SELECT student_id,  
       (classes_attended * 100.0 / total_classes) AS attendance_percentage  
FROM attendance;
```

Key Insight

Higher attendance generally leads to better academic performance.

Conclusion

This project helped me apply SQL concepts to real-world data analysis.

