Student Report System – SQL Mini Project

# 📘 Project Overview

This SQL mini project simulates a simple Student Report System using SQL Server. It includes table creation, data insertion, and several advanced queries using JOINs, GROUP BY, CASE, subqueries, and window functions.

# 🛠️ Step 1: Create Tables

CREATE TABLE Students (  
 StudentID INT PRIMARY KEY,  
 Name VARCHAR(100),  
 Gender CHAR(1),  
 Class VARCHAR(10),  
 DOB DATE  
);  
  
CREATE TABLE Subjects (  
 SubjectID INT PRIMARY KEY,  
 Name VARCHAR(100)  
);  
  
CREATE TABLE Marks (  
 MarkID INT PRIMARY KEY,  
 StudentID INT FOREIGN KEY REFERENCES Students(StudentID),  
 SubjectID INT FOREIGN KEY REFERENCES Subjects(SubjectID),  
 Marks INT,  
 ExamType VARCHAR(50)  
);

# 📝 Step 2: Insert Sample Data

INSERT INTO Students VALUES  
(1, 'Arjun', 'M', '10A', '2008-04-12'),  
(2, 'Diya', 'F', '10A', '2008-07-19'),  
(3, 'Ravi', 'M', '10B', '2007-11-03');  
  
INSERT INTO Subjects VALUES  
(101, 'Maths'), (102, 'Science'), (103, 'English');  
  
INSERT INTO Marks VALUES  
(1, 1, 101, 88, 'Final'),  
(2, 1, 102, 72, 'Final'),  
(3, 2, 101, 95, 'Final'),  
(4, 2, 103, 90, 'Final'),  
(5, 3, 101, 65, 'Final'),  
(6, 3, 102, 58, 'Final');

# 📊 Step 3: Advanced SQL Queries

## 1. Student Total & Average Marks

SELECT s.StudentID, s.Name, s.Class,  
 COUNT(m.Marks) AS SubjectCount,  
 SUM(m.Marks) AS TotalMarks,  
 AVG(m.Marks) AS AverageMarks  
FROM Students s  
JOIN Marks m ON s.StudentID = m.StudentID  
GROUP BY s.StudentID, s.Name, s.Class;

## 2. Rank Students in Class by Total Marks

SELECT s.Name, s.Class, SUM(m.Marks) AS TotalMarks,  
 RANK() OVER (PARTITION BY s.Class ORDER BY SUM(m.Marks) DESC) AS ClassRank  
FROM Students s  
JOIN Marks m ON s.StudentID = m.StudentID  
GROUP BY s.Name, s.Class;

## 3. Grade Report Using CASE

SELECT s.Name, s.Class, AVG(m.Marks) AS AverageMarks,  
 CASE   
 WHEN AVG(m.Marks) >= 90 THEN 'A+'  
 WHEN AVG(m.Marks) >= 75 THEN 'A'  
 WHEN AVG(m.Marks) >= 60 THEN 'B'  
 WHEN AVG(m.Marks) >= 45 THEN 'C'  
 ELSE 'D'  
 END AS Grade  
FROM Students s  
JOIN Marks m ON s.StudentID = m.StudentID  
GROUP BY s.Name, s.Class;

## 4. Subject-wise Toppers

SELECT \*  
FROM (  
 SELECT s.Name, sub.Name AS Subject, m.Marks,  
 RANK() OVER (PARTITION BY m.SubjectID ORDER BY m.Marks DESC) AS RankInSubject  
 FROM Marks m  
 JOIN Students s ON m.StudentID = s.StudentID  
 JOIN Subjects sub ON m.SubjectID = sub.SubjectID  
) AS ranked  
WHERE RankInSubject = 1;

## 5. Students Above Class Average (Subquery)

SELECT s.Name, s.Class, SUM(m.Marks) AS TotalMarks  
FROM Students s  
JOIN Marks m ON s.StudentID = m.StudentID  
GROUP BY s.Name, s.Class  
HAVING SUM(m.Marks) > (  
 SELECT AVG(Total)  
 FROM (  
 SELECT SUM(m2.Marks) AS Total  
 FROM Students s2  
 JOIN Marks m2 ON s2.StudentID = m2.StudentID  
 GROUP BY s2.StudentID  
 ) AS ClassAvg  
);