■ Student Result Processing System - MySQL Report

1. Schema Design

```
-- Students
CREATE TABLE Students (
    student_id INT AUTO_INCREMENT PRIMARY KEY,
    student_name VARCHAR(100) NOT NULL,
    roll_no VARCHAR(20) UNIQUE NOT NULL,
    department VARCHAR(50)
);

-- Courses
CREATE TABLE Courses (
    course_id INT AUTO_INCREMENT PRIMARY KEY,
    course_name VARCHAR(100) NOT NULL,
    credits INT NOT NULL
);

-- Semesters
CREATE TABLE Semesters (
    semester_id INT AUTO_INCREMENT PRIMARY KEY,
    semester_name VARCHAR(50) NOT NULL
);

-- Grades
CREATE TABLE Grades (
    grade_id INT AUTO_INCREMENT PRIMARY KEY,
    student_id INT,
    course_id INT,
    semester_id INT,
    grade CHAR(2) CHECK (grade IN ('O','A+','A','B+','B','C','F')),
    gpa_points DECIMAL(4,2),
    FOREIGN KEY (student_id) REFERENCES Students(student_id),
    FOREIGN KEY (course_id) REFERENCES Courses(course_id),
    FOREIGN KEY (semester_id) REFERENCES Semesters(semester_id)
);
```

2. Sample Data Inserts

```
-- Students
INSERT INTO Students (student_name, roll_no, department) VALUES
('Alice', 'CSE001', 'CSE'),
('Bob', 'CSE002', 'CSE'),
('Charlie', 'ECE001', 'ECE');

-- Courses
INSERT INTO Courses (course_name, credits) VALUES
('DBMS', 4), ('Operating Systems', 3), ('Maths', 4);

-- Semesters
INSERT INTO Semesters (semester_name) VALUES ('Semester 1'), ('Semester 2');

-- Grades
INSERT INTO Grades (student_id, course_id, semester_id, grade, gpa_points) VALUES
(1, 1, 1, 'O', 10), (1, 2, 1, 'A+', 9), (1, 3, 1, 'A', 8),
(2, 1, 1, 'B+', 7), (2, 2, 1, 'B', 6), (2, 3, 1, 'C', 5),
(3, 1, 1, 'O', 10), (3, 2, 1, 'O', 10), (3, 3, 1, 'A+', 9);
```

3. GPA & Pass/Fail Queries

```
-- GPA per student per semester
SELECT s.student_name, sem.semester_name,
       SUM(g.gpa_points * c.credits) / SUM(c.credits) AS GPA
FROM Grades g
```

```
JOIN Students s ON g.student_id = s.student_id
JOIN Courses c ON g.course_id = c.course_id
JOIN Semesters sem ON g.semester_id = sem.semester_id
GROUP BY s.student_id, sem.semester_id;

-- Pass/Fail per semester
SELECT sem.semester_name,
       COUNT(CASE WHEN g.grade != 'F' THEN 1 END) AS PassCount,
       COUNT(CASE WHEN g.grade = 'F' THEN 1 END) AS FailCount
FROM Grades g
JOIN Semesters sem ON g.semester_id = sem.semester_id
GROUP BY sem.semester_id;
```

4. Rank List (Window Functions)

```
SELECT s.student_name, sem.semester_name,
       SUM(g.gpa_points * c.credits) / SUM(c.credits) AS GPA,
       RANK() OVER (PARTITION BY sem.semester_id ORDER BY SUM(g.gpa_points * c.credits) / SUM(c.credi
FROM Grades g
JOIN Students s ON g.student_id = s.student_id
JOIN Courses c ON g.course_id = c.course_id
JOIN Semesters sem ON g.semester_id = sem.semester_id
GROUP BY s.student_id, sem.semester_id;
```

5. Triggers for GPA Calculation

```
DELIMITER $$

CREATE TRIGGER set_gpa_points
BEFORE INSERT ON Grades
FOR EACH ROW
BEGIN
    SET NEW.gpa_points = CASE NEW.grade
        WHEN 'O'  THEN 10
        WHEN 'A+' THEN 9
        WHEN 'A'  THEN 8
        WHEN 'B+' THEN 7
        WHEN 'B'  THEN 6
        WHEN 'C'  THEN 5
        ELSE 0
    END;
END$$

DELIMITER ;
```

6. Export Semester-wise Summary

```
-- Export Results to CSV
SELECT s.student_name, sem.semester_name,
       SUM(g.gpa_points * c.credits) / SUM(c.credits) AS GPA
FROM Grades g
JOIN Students s ON g.student_id = s.student_id
JOIN Courses c ON g.course_id = c.course_id
JOIN Semesters sem ON g.semester_id = sem.semester_id
GROUP BY s.student_id, sem.semester_id
INTO OUTFILE '/tmp/semester_results.csv'
FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n';
```