SQL Project Documentation – StudentManagement Database

1. Create Database and Table

SQL Query:

```
CREATE DATABASE StudentManagement;
USE StudentManagement;

CREATE TABLE Students (
  StudentID INT AUTO_INCREMENT PRIMARY KEY,
  Name VARCHAR(50),
  Gender VARCHAR(1),
  Age INT,
  Grade VARCHAR(10),
  MathScore INT,
  ScienceScore INT,
  EnglishScore INT);
```

Purpose: Creates a new database and a table for storing student academic data.

Observation: Establishes the structure to store detailed student information.

2. Insert Sample Records

SQL Query:

```
INSERT INTO Students (Name, Gender, Age, Grade, MathScore, ScienceScore, EnglishScore) VALUES
('Alice Johnson', 'F', 16, 'A', 95, 92, 88),
('Bablu', 'M', 17, 'B', 78, 85, 80),
('Catherine', 'F', 15, 'A', 89, 94, 90),
('David', 'M', 16, 'C', 65, 70, 75),
('Eva Mariam', 'F', 17, 'B', 82, 79, 85),
('Fadalu', 'M', 16, 'A', 90, 88, 91),
('Grace', 'F', 15, 'B', 77, 81, 78),
('Harry', 'M', 17, 'C', 60, 65, 70),
('Isabella', 'F', 16, 'A', 93, 96, 94),
('Jis Thomas', 'M', 15, 'B', 80, 83, 79);
```

Purpose: Populates the Students table with 10 records for analysis.

Observation: Data is varied by age, gender, grade, and subject scores.

3. Display All Students

SQL Query:

```
SELECT * FROM Students;
```

Purpose: Displays all student records.

Observation: Useful for reviewing the inserted data.

StudentID	Name	Gender	Age	Grade	MathScore	ScienceScore	EnglishScore
1	Alice Johnson	F	16	A	95	92	88
2	Bablu	M	17	B	78	85	80
3	Catherine	F	15	A	89	94	90
4	David	M	16	C	65	70	75
5	Eva Mariam	F	17	B	82	79	85
6	Fadalu	M	16	A	90	88	91
7	Grace	F	15	B	77	81	78
8	Harry	M	17	C	60	65	70
9	Isabella	F	16	A	93	96	94
10	Jis Thomas	M	15	B	80	83	79

4. Subject-Wise Average Scores

SQL Query:

```
SELECT
  AVG(MathScore) AS Average_Math_Score,
  AVG(ScienceScore) AS Average_Science_Score,
  AVG(EnglishScore) AS Average_English_Score
FROM Students;
```

Purpose: Calculates average scores for each subject.

Observation: Shows how students are performing by subject.

Average_Math_Score	Average_Science_Score	Average_English_Score
80.9	83.3	83

5. Top Performer

SQL Query:

```
SELECT StudentID, Name, (MathScore + ScienceScore + EnglishScore) AS TotalScore
FROM Students
WHERE (MathScore + ScienceScore + EnglishScore) = (
SELECT MAX(MathScore + ScienceScore + EnglishScore) FROM Students);
```

Purpose: Finds the student(s) with the highest total score.

Observation: Helps identify the top performer.

StudentID	Name	Gender	Age	Grade	MathScore	ScienceScore	EnglishScore	TotalScore
9	Isabella	F	16	A	93	96	94	283

6. Grade Distribution

SQL Query:

```
SELECT Grade, COUNT(*) AS NumberOfStudents
FROM Students
GROUP BY Grade;
```

Purpose: Counts how many students are in each grade.

Observation: Visualizes grade distribution.

Grade	NumberOfStudents
A	4
B	4
C	2

7. Gender-Wise Average Score

SQL Query:

```
SELECT Gender,
  AVG(MathScore) AS Avg_MathScore,
  AVG(ScienceScore) AS Avg_ScienceScore,
  AVG(EnglishScore) AS Avg_EnglishScore
FROM Students
GROUP BY Gender;
```

Purpose: Compares subject-wise scores by gender.

Observation: Analyzes performance trends across genders.

Gender	Avg_MathScore	Avg_ScienceScore	Avg_EnglishScore
F	87.2	88.4	87
M	74.6	78.2	79

8. High Math Achievers

SQL Query:

```
SELECT Name, MathScore
FROM Students
WHERE MathScore > 80;
```

Purpose: Identifies students scoring above 80 in Math.

Observation: Highlights students strong in Mathematics.

Name	MathScore
Alice Johnson	95
Catherine	89
Eva Mariam	82
Fadalu	90
Isabella	93

9. Update Student Grade

SQL Query:

```
UPDATE Students
SET Grade = 'A'
WHERE StudentID = 2;
```

Purpose: Updates a student's grade using their ID.

Observation: Used to reflect academic improvements or corrections.

StudentID	Name	Gender	Age	Grade	MathScore	ScienceScore	EnglishScore
1	Alice Johnson	F	16	A	95	92	88
2	Bablu	M	17	A	78	85	80
3	Catherine	F	15	A	89	94	90
4	David	M	16	C	65	70	75
5	Eva Mariam	F	17	B	82	79	85
6	Fadalu	M	16	A	90	88	91
7	Grace	F	15	B	77	81	78
8	Harry	M	17	C	60	65	70
9	Isabella	F	16	A	93	96	94
10	Jis Thomas	M	15	B	80	83	79