SQL DEVELOPER

NAME : AKHIL SAI SAMMETA **MAIL ID : Gmail.com**

TASK - 1

Student Management System Project for SQL Developers

Objective

The project focuses on providing students with practical experience in SQL database creation,

data manipulation, and analysis using student performance data.

1. Database Setup

```
CREATE DATABASE StudentManagement;
USE StudentManagement;

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

Purpose:

- The first query creates a new database named StudentManagement to store all related data.
- The second query creates a table named Students with fields for student details, including an auto-incrementing primary key (StudentID), name, gender, age, grade, and scores in Math, Science, and English.

Observations:

- The table structure allows for easy data entry and retrieval, ensuring that all necessary information about students is captured.

---

2. Insert Sample Data

```
INSERT INTO Students (Name, Gender, Age, Grade, MathScore, ScienceScore, EnglishScore) VALUES
('Alice Johnson', 'F', 20, 'A', 85, 90, 88),
('Bob Smith', 'M', 21, 'B', 78, 82, 80),
('Charlie Brown', 'M', 19, 'C', 92, 88, 85),
('Diana Prince', 'F', 22, 'A', 95, 94, 90),
('Ethan Hunt', 'M', 20, 'B', 70, 75, 72),
('Fiona Apple', 'F', 21, 'A', 88, 85, 87),
('George Clooney', 'M', 23, 'C', 60, 65, 70),
('Hannah Montana', 'F', 19, 'B', 80, 78, 82),
('Ian Malcolm', 'M', 22, 'A', 90, 92, 91),
('Julia Roberts', 'F', 20, 'C', 75, 80, 78);
```

Purpose:

- This query populates the Students table with 10 sample records, providing a diverse set of names, genders, ages, grades, and scores.

Observations:

- The sample data includes a mix of genders and grades, allowing for comprehensive analysis of performance across different demographics.

---

3. TASKS TO PERFORM

1. Display All Students

```
SELECT * FROM Students;
```

Purpose:

- This query retrieves all records from the Students table, providing a complete overview of the data.

Observations:

- The output allows for a quick review of all student details, which can help identify any data entry errors or inconsistencies.

StudentID	Name	Gender	Age	Grade	MathScore	ScienceSco	EnglishScore
1	Alice Johns	F	20	A	85	90	88
2	Bob Smith	M	21	B	78	82	80
3	Charlie Brc	M	19	C	92	88	85
4	Diana Prin	F	22	A	95	94	90
5	Ethan Hunt	M	20	B	70	75	72
6	Fiona Appl	F	21	A	88	85	87
7	George Clc	M	23	C	60	65	70
8	Hannah M	F	19	B	80	78	82
9	Ian Malcol	M	22	A	90	92	91
10	Julia Rober	F	20	C	75	80	78

---

2. Calculate Average Scores for Each Subject

```
SELECT
   AVG(MathScore) AS AverageMathScore,
   AVG(ScienceScore) AS AverageScienceScore,
   AVG(EnglishScore) AS AverageEnglishScore
FROM Students;
```

Purpose:

- This query calculates the average scores for Math, Science, and English across all students.
-

Observations:

- The average scores provide insights into overall subject performance, helping educators identify areas where students may need additional support.

Gender	AverageMathScore	AverageScienceScore	AverageEnglishScore
F	84.6	85.4	85
M	78	80.4	79.6

---

3. Find the Student(s) with the Highest Total Score

```
SELECT Name, (MathScore + ScienceScore + EnglishScore) AS TotalScore
FROM Students
ORDER BY TotalScore DESC
LIMIT 1;
```

Purpose:

- This query identifies the student with the highest total score across all subjects.

Observations:

- The result highlights the top performer, which can be useful for recognizing achievements and motivating other students.

Name	TotalScore
Diana Prince	279

---

4. Count the Number of Students in Each Grade

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

Purpose:

- This query counts the number of students in each grade category.

Observations:

- The distribution of students across grades can help educators understand the overall performance levels and identify trends in student achievement.

Grade	NumberOfStudents
A	4
B	3
C	3

5. Find the Average Score for Male and Female Students

```
SELECT Gender,
    AVG(MathScore) AS AverageMathScore,
    AVG(ScienceScore) AS AverageScienceScore,
    AVG(EnglishScore) AS AverageEnglishScore
FROM Students
GROUP BY Gender;
```

Purpose:

- This query calculates the average scores for male and female students, allowing for a comparison of performance by gender.

Observations:

- The results can reveal any significant differences in performance between genders

Gender	AverageMathScore	AverageScienceScore	AverageEnglishScore
F	84.6	85.4	85
M	78	80.4	79.6

6. Identify Students Whose Math Score is Above 80

```
SELECT * FROM Students
WHERE MathScore > 80;
```

Purpose:

- This query retrieves all records from the Students table where the Math score is greater than 80. This helps in identifying high achievers in Math.

Observations:

- The output will list students who excel in Math, which can be useful for recognizing and encouraging students who perform well in this subject. It may also help in identifying students who could be considered for advanced Math programs or competitions.

StudentID	Name	Gender	Age	Grade	MathScore	ScienceScore	EnglishScore
1	Alice Johnson	F	20	A	85	90	88
3	Charlie Brown	M	19	C	92	88	85
4	Diana Prince	F	22	A	95	94	90
6	Fiona Apple	F	21	A	88	85	87
9	Ian Malcolm	M	22	A	90	92	91

7. Update the Grade of a Student with a Specific Student ID

```
UPDATE Students
SET Grade = 'A'
WHERE StudentID = 5;
```

Purpose:

- This query updates the Grade field of a specific student identified by their StudentID. In this example, it sets the grade to 'A' for the student with StudentID 5.

Observations:

- This operation is useful for correcting or reflecting changes in a student's academic performance. It allows for dynamic updates to the database, ensuring that the information remains accurate and up-to-date. After executing this query, it would be prudent to verify the change by running a SELECT query to confirm that the grade has been updated correctly.

StudentID	Name	Gender	Age	Grade	MathScore	ScienceScore	EnglishScore
1	Alice Johns	F	20	A	85	90	88
2	Bob Smith	M	21	B	78	82	80
3	Charlie Brc	M	19	C	92	88	85
4	Diana Prin	F	22	A	95	94	90
5	Ethan Hun	M	20	A	70	75	72
6	Fiona Appl	F	21	A	88	85	87
7	George Clc	M	23	C	60	65	70
8	Hannah M	F	19	B	80	78	82
9	Ian Malcol	M	22	A	90	92	91
10	Julia Rober	F	20	C	75	80	78