Task-1-SQL-Developer

Student Management System Project for SQL Developer domain

Objective

The project focuses on providing students with practical experience in SQL database creation,data manipulation, and analysis using student performance data.

Project Steps

1. Database Setup ● Create a database named StudentManagement. ● Create a table named Students with the following fields: ○ StudentID: Primary Key, INT, an auto-incrementing integer. ○ Name: Stores the student's name (up to 50 characters). ○ Gender: A single character (VARCHAR, 1 - 'M' for Male, 'F' for Female). ○ Age: INT ○ Grade: Academic grade (VARCHAR, 10 - e.g., 'A', 'B', 'C', etc.). ○ MathScore, ScienceScore, EnglishScore: Integers representing scores in respective subjects.
2. Insert Data Populate the Students table with at least 10 sample records, including a variety of names, genders, grades, and scores in Math, Science, and English.
3. Tasks to Perform
    i. Display all students and their details to get an overview of the data.
    ii. Calculate the average scores for each subject to understand subject-wise performance.
    iii. Find the student(s) with the highest total score across all subjects to identify the top performer.
    iv. Count the number of students in each grade to observe grade distributions.
    v. Find the average score for male and female students to compare performance by gender.
    vi. Identify students whose Math score is above 80 to highlight high achievers in Math.

vii. Update the grade of a student with a specific Student ID to reflect changes or corrections.

Insert Data.

```
Show Database
Create Database StudentManagement
use StudentManagement
CREATE TABLE Students (
 StudentID INT AUTO_INCREMENT,
 Name VARCHAR(50),
 Gender VARCHAR(1),
 Age INT,
 Grade VARCHAR(10),
 MathScore INT,
 ScienceScore INT,
 EnglishScore INT,
 PRIMARY KEY (StudentID)
);
INSERT INTO Students (Name, Gender, Age, Grade, MathScore, ScienceScore, EnglishScore)
VALUES
 ('Sagar', 'M', 16, 'A', 90, 85, 95),
 ('Shiwani, 'F', 17, 'A', 95, 90, 92),
 ('Aanand', 'M', 16, 'B', 80, 75, 85),
 ('Sneha', 'F', 17, 'A', 92, 88, 90),
 ('Aryan', 'M', 16, 'C', 70, 65, 75),
 ('Sarika', 'F', 17, 'B', 85, 80, 88),
 ('priyanshu', 'M', 16, 'B', 75, 70, 80),
 ('Sarah', 'F', 17, 'A', 98, 95, 92),
 ('Rahul', 'M', 16, 'C', 65, 60, 70),
 ('Nandini kumari', 'F', 17, 'A', 90, 85, 95);
```

Task 1:

1. Display of all students and their details.

```
-> SELECT * FROM Students;
```

Student ID	Name	Gender	Age	Grade	Mathscore	Sciencescore	English Score
101	Sagar	male	16	A	90	85	95

102	Shiwani	female	17	A	95	90	92
103	Anand	male	16	B	80	75	85
104	Sneha	female	17	A	92	88	90
105	Aryan	male	16	C	70	65	75
106	Sarika	female	17	B	85	80	88
107	Priyanshu	male	16	B	75	70	80
108	Sarah	female	17	A	98	95	92
109	Rahul	male	16	C	65	60	70
110	Nandini	female	17	F	90	85	95

2. Calculate the average scores for each subject.

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

AvgMathScore	AvgScienceScore	AvgEnglishScore
73.9	**70.2**	**75.5**

3. Find the students with the highest total scores.

```
-> SELECT
     StudentID,
      Name,
     MathScore + ScienceScore + EnglishScore AS TotalScore
     FROM Students
     ORDER BY TotalScore DESC
```

```
LIMIT 1;
```

Name	Totalscore
Sarah	285

4.Count the number of students in each grade.

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

Grade	StudentCount
A	4
B	3
C	2
F	1

5. Find the average score for male and female students.

```
-> SELECT
    Gender,
    AVG((MathScore + ScienceScore + EnglishScore) / 3) AS AverageScore
    FROM Students
    GROUP BY Gender;
```

Gender	AvgMathScore	AvgScienceScore	AvgEnglishScore
Male	76	71	81
Female	92	87.6	91.4

6. Identify students whose math score is above 80.

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

Name	MathScore
Sagar	90
Shiwani	95
Sneha	92
Sarika	85
Sarah	98
Nandini	90

7.Update the grade of a student with a specific Student ID.

```
-> UPDATE Students
    SET Grade = 'A'
    WHERE StudentID = 110;
```

StudentID	Name	Gender	Age	Grade	MathScore	ScienceScore	EnglishScore
110	Nandini	Female	17	A	90	85	95