Task 1

DATABASE CREATION AND DATA INSERTION

```sql
CREATE DATABASE STUDENTMANAGEMENT;
USE STUDENTMANAGEMENT;
CREATE TABLE STUDENTS(
StudentID INT AUTO_INCREMENT PRIMARY KEY,
Name VARCHAR(50) NOT NULL,
Gender VARCHAR(1) CHECK (Gender IN ('M', 'F')),
Age INT CHECK (Age >= 5 AND Age <= 25),
Grade VARCHAR(10) NOT NULL,
MathScore INT CHECK (MathScore BETWEEN 0 AND 100),
ScienceScore INT CHECK (ScienceScore BETWEEN 0 AND 100),
EnglishScore INT CHECK (EnglishScore BETWEEN 0 AND 100)
);
INSERT INTO Students (Name, Gender, Age, Grade, MathScore, ScienceScore, EnglishScore)
VALUES
('GEETA', 'F', 16, 'A', 95, 90, 88),
('SHYAM', 'M', 17, 'B', 80, 85, 82),
('RAM', 'M', 15, 'A', 92, 87, 91),
('ANUSHKA', 'F', 18, 'C', 70, 75, 78),
('GYAN', 'M', 16, 'B', 85, 89, 84),
('RIYA', 'F', 17, 'A', 96, 94, 93),
('MUKESH', 'M', 15, 'C', 68, 70, 65),
('SITA', 'F', 16, 'B', 83, 81, 85),
('ISAAN', 'M', 17, 'A', 98, 97, 96),
('JACKY', 'M', 18, 'C', 72, 74, 71);
```

PURPOSE OF DATABASE CREATION- This query creates a database named **StudentManagement** and a table **Students** with the necessary fields to store student information, including scores in three subjects.

PURPOSE OF DATA INSERTION: This query inserts 10 sample records representing students with diverse grades, genders, and scores.

Query No-1

```sql
-- 1. Display all students and their details to get an overview of the data--
SELECT* FROM STUDENTS;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

StudentID	Name	Gender	Age	Grade	MathScore	ScienceScore	EnglishScore
1	GEETA	F	16	A	95	90	88
2	SHYAM	M	17	B	80	85	82
3	RAM	M	15	A	92	87	91
4	ANUSHKA	F	18	C	70	75	78
5	GYAN	M	16	B	85	89	84
6	RIYA	F	17	A	96	94	93
7	MUKESH	M	15	C	68	70	65
8	SITA	F	16	B	83	81	85
9	ISAAN	M	17	A	98	97	96
10	JACKY	M	18	C	72	74	71
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Purpose:

- **It Retrieves all student records for an overview of the dataset.**

Query No-2

```
27      -- 2. Calculate the average scores for each subject to understand subject-wise performance
28      SELECT
29          AVG(MathScore) AS Avg_MathScore,
30          AVG(ScienceScore) AS Avg_ScienceScore,
31          AVG(EnglishScore) AS Avg_EnglishScore
32      FROM Students;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Avg_MathScore	Avg_ScienceScore	Avg_EnglishScore
83.9000	84.2000	83.3000

Purpose: It Provides insights into subject-wise performance and can identify which subject students excel in or struggle with.

Query No-3

Limit to 1000 rows

```
33      -- 3. Find the student(s) with the highest total score across all subjects to identify the top performer.
34      SELECT Name, Gender, Age, Grade,
35             MathScore, ScienceScore, EnglishScore,
36             (MathScore + ScienceScore + EnglishScore) AS TotalScore
37      FROM Students
38      ORDER BY TotalScore DESC
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

Name	Gender	Age	Grade	MathScore	ScienceScore	EnglishScore	TotalScore
ISAAN	M	17	A	98	97	96	291

Purpose: It Identifies the student with the highest total score and helps in recognizing academic excellence.

Query No-4

Limit to 1000 rows

```
40      -- 4. Count the number of students in each grade to observe grade distributions
41      SELECT GRADE,COUNT(*) AS STUDENTCOUNT
42      FROM STUDENTS
43      GROUP BY GRADE;
44      SELECT GENDER,
45             AVG(MathScore) AS Avg_MathScore,
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

GRADE	STUDENTCOUNT
A	4
B	3
C	3

Purpose : It Helps analyze the distribution of students across different grade and Identifies whether most students belong to a specific grade.

Query No-5

```
44      -- 5. Find the average score for male and female students to compare performance by gender.
45 •    SELECT GENDER,
46             AVG(MathScore) AS Avg_MathScore,
47             AVG(ScienceScore) AS Avg_ScienceScore,
48             AVG(EnglishScore) AS Avg_EnglishScore
49             FROM STUDENTS
50             GROUP BY GENDER;
```

esult Grid | Filter Rows: | Export: | Wrap Cell Content:

GRADE	STUDENTCOUNT
A	4
B	3
C	3

Purpose : It Compares male and female students' academic performance and Can highlight potential gender disparities in subject performance.

Query No-6

```
51      -- 6. Identify students whose Math score is above 80 to highlight high achievers in Math.
52 •    SELECT * FROM Students
53      WHERE MathScore > 80;
54      -- 7. Update the grade of a student with a specific Student ID to reflect changes or corrections.
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

StudentID	Name	Gender	Age	Grade	MathScore	ScienceScore	EnglishScore
1	GEETA	F	16	A	95	90	88
3	RAM	M	15	A	92	87	91
5	GYAN	M	16	B	85	89	84
6	RIYA	F	17	A	96	94	93
8	SITA	F	16	B	83	81	85
9	ISAAN	M	17	A	98	97	96
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Purpose : It Lists students who scored above 80 in Math and helps in identifying strong performers in Mathematics.

Query No-7

```
-- 7. Update the grade of a student with a specific Student ID to reflect changes or corrections.
UPDATE Students
SET Grade = 'B'
WHERE StudentID = 5;
```

Context Help | Snippets

Action Output

	Time	Action	Message	Duration / Fetch
10	14:09:40	SELECT AVG(MathScore) AS Avg_MathScore, AVG(ScienceScore) AS Avg_ScienceScore, AVG(E...	1 row(s) returned	0.000 sec / 0.000 se
11	14:11:26	SELECT Name, Gender, Age, Grade, MathScore, ScienceScore, EnglishScore, (MathScore + Scien...	1 row(s) returned	0.000 sec / 0.000 se
12	14:12:41	SELECT GRADE,COUNT(*) AS STUDENTCOUNT FROM STUDENTS GROUP BY GRADE LIMIT 0, 1000	3 row(s) returned	0.031 sec / 0.000 se
13	14:16:52	SELECT * FROM Students WHERE MathScore > 80 LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 se
14	14:17:21	SELECT * FROM Students WHERE MathScore > 80 LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 se
15	14:19:10	UPDATE Students SET Grade = 'B' WHERE StudentID = 5	0 row(s) affected Rows matched: 1 Changed: 0 Warnings: 0	0.031 sec

Purpose: Updates a student's grade based on StudentID and Useful for correcting data entry mistakes or reflecting academic improvements.