SQL TASK – 3 - StudentManagement Database

1. Create Database and Table

SQL Query:

```
CREATE DATABASE Students_New;
USE Students_New;

CREATE TABLE Students (
 student_id INT AUTO_INCREMENT PRIMARY KEY,
 name VARCHAR(250),
 mathScore INT NOT NULL,
 scienceScore INT NOT NULL,
 englishScore INT NOT NULL,
 total_score INT AS (math_score + science_score + english_score));
```

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,math_score, science_score, english_score) VALUES
('Abhishek', 71, 87, 81),
('Alice', 85, 80, 84),
('Anirudh', 69, 65, 75),
('Basil', 88, 90, 95),
('Binu', 60, 65, 50),
('Cyna’, 75, 85, 87),
('Danniel', 88, 85, 84),
('Diya', 93, 90, 92),
('Fadalu', 95, 94, 89),
('Fathima', 68, 75, 77);
```

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

Observation: Data is varied by subject scores.

3. Display All Students

SQL Query:

```
SELECT * FROM Students;
```

Purpose: Displays all student records.

Observation: Useful for reviewing the inserted data.

student_id	name	math_score	science_score	english_score	total_score
1	Abhishek	71	87	81	239
2	Alice	85	80	84	249
3	Anirudh	69	65	75	209
4	Basil	88	90	95	273
5	Binu	60	65	50	175
6	Cyna	75	85	87	247
7	Danniel	88	85	84	257
8	Diya	93	90	92	275
9	Fadalu	95	94	89	278
10	Fathima	68	75	77	220

4. Top Students by Total Scores

SQL Query:

```
SELECT
  student_id, name, total_score
  FROM Students
  ORDER BY total_score DESC LIMIT 5;
```

Purpose: List the top students based on their total scores.

Observation: Displays the 5 highest total scores, allowing us to recognize the top-performing students.

student_id	name	total_score
9	Fadalu	278
8	Diya	275
4	Basil	273
7	Danniel	257
2	Alice	249

5. Averages Based on Specific Conditions

SQL Query:

i) Average score of students who scored above 70 in Math.

```
SELECT AVG(total_score) AS average_total_score FROM Students WHERE math_score > 70;
```

Purpose: Finds the average total score of the student(s) who have scored above 70 In math.

Observation: Highlights the average total score of students scoring above 70 in math.

average_total_score
259.7143

ii) Average total score of students grouped by a specific condition

```
SELECT CASE
        WHEN total_score BETWEEN 200 AND 250 THEN '200-250'
        WHEN total_score BETWEEN 251 AND 281 THEN '251-281'
    ELSE 'Other'
  END AS score_range,
  AVG(total_score) AS average_total_score FROM Students GROUP BY score_range;
```

Purpose: Finds the average total score within a specified score range.

Observation: Helps identify the distribution of total marks, allowing us to see the average total score for each score range

score_range	average_total_score
200-250	232.8
251-281	270.75
Other	175

6. Second-Highest Math Scores

SQL Query:

i) Highest Math Score

```
SELECT MAX(math_score) AS highest_math_score FROM Students;
```

Purpose: Find the highest math score.

Observation: Helps identify the top math mark.

highest_math_score
95

ii) Second Highest Math Score

```
SELECT MAX(math_score) AS second_highest_math_score
FROM Students WHERE math_score < (SELECT MAX(math_score) FROM Students);
```

Purpose: Find the second highest math score.

Observation: Helps identify the second top math mark.

second_highest_math_score
93