TASK 3

NAME : AKHIL SAI SAMMETA

Task 3: Subqueries and Aggregations

Objective

Use subqueries to extract insights from a dataset and perform data aggregations to summarize and analyze the data.

Project Steps

Step 1: Database Setup

1. **Table: Students**
   - **Fields:**
     - **StudentID: Primary Key.**
     - **Name: Name of the student.**
     - **Gender: Gender of the student (M for Male, F for Female).**
     - **Age: Age of the student.**
     - **Grade: Grade of the student.**
     - **MathScore: Math test score.**
     - **ScienceScore: Science test score.**
     - **EnglishScore: English test score.**
2. **Insert sample data**
   - **Populate the table with scores for Math, Science, and English for multiple students.**

Example Database Setup Commands:

```
CREATE DATABASE StudentManagement01;

USE StudentManagement01;


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

);
```

```sql
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);
```

Tasks to Perform

Task 1: Identify Top Students by Total Scores

- **Use a subquery to calculate the total score (MathScore + ScienceScore + EnglishScore) for each student.**
- **Use an ORDER BY clause to rank students by their total scores in descending order.**
- **Limit the results to show only the top students (e.g., top 5).**

SQL Query:

```sql
-- Calculate total scores for each student and rank them in descending order.
SELECT StudentID, Name, TotalScore
FROM (
   SELECT StudentID, Name, (MathScore + ScienceScore + EnglishScore) AS TotalScore
   FROM Students
) AS Subquery
ORDER BY TotalScore DESC
LIMIT 5;
```

Explanation:

- The subquery calculates the total score for each student by summing their individual subject scores.
- The outer query retrieves these totals, orders them in descending order, and limits the output to the top 5.

StudentID	Name	TotalScore
4	Diana Prince	279
9	Ian Malcolm	273
3	Charlie Brown	265
1	Alice Johnson	263
6	Fiona Apple	260

Task 2: Calculate Averages Based on Specific Conditions

Example 1: Calculate the average score of students who scored above 70 in Math.

SQL Query:

```
-- Calculate the average Math score for students scoring above 70.

SELECT AVG(MathScore) AS AverageMathScore

FROM Students

WHERE MathScore > 70;
```

Explanation:

- The query filters students with Math scores above 70 and calculates their average using the AVG function.

AverageMathScore
85.375

Example 2: Calculate the average total score of students grouped by a specific condition, such as a score range (e.g., students scoring 200–250 in total).

SQL Query:

```
-- Calculate the average total score for students scoring within a specific range.

SELECT AVG(TotalScore) AS AverageTotalScore

FROM (

   SELECT (MathScore + ScienceScore + EnglishScore) AS TotalScore

   FROM Students

) AS Subquery

WHERE TotalScore BETWEEN 200 AND 250;
```

Explanation:

- The subquery calculates total scores for all students.

- The outer query filters these totals to include only those within the specified range and calculates their average.

AverageTotalScore
232.5

Task 3: Find Second-Highest Math Scores

- **Use a subquery to determine the highest Math score and exclude it in a second query to find the next highest value.**

SQL Query:

```
-- Find the second-highest Math score.
SELECT MAX(MathScore) AS SecondHighestMathScore
FROM Students
WHERE MathScore < (
   SELECT MAX(MathScore)
   FROM Students
);
```

Explanation:

- The inner subquery finds the maximum Math score.
- The outer query excludes this maximum and calculates the next highest score using MAX**.**

SecondHighestMathScore
92

Summary of Findings

1. **Top-Performing Students:**
    - **The top 5 students based on their total scores were identified.**
2. **Average Scores:**

- **Students scoring above 70 in Math had an average Math score calculated.**
- **Average total scores were grouped by specific ranges.**

3. **Second-Highest Math Score:**
    - **The second-highest Math score was determined efficiently using subqueries.**