SQL DEVELOPER

TASK - 3

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:**
        - `student_id`: Primary Key.
        - `name`: Name of the student.
        - `math_score`: Math test score.
        - `science_score`: Science test score.
        - `english_score`: English test score.
        - `total_score`: The sum of all scores for each student (optional if calculated dynamically).
2. **Insert sample data with scores for Math, Science, and English for multiple students.**

Step 2: Tasks to Perform

Task 1: Identify Top Students by Total Scores

- Use a subquery to calculate the total score (`math_score + science_score + english_score`) 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).

Task 2: Calculate Averages Based on Specific Conditions

Main Flow Services and Technologies Pvt. Ltd.
Contact Us. +91 9389641586, +91 97736 99074
Email-Add. contact.mainflow@gmail.com
www.mainflow.in

- **Use subqueries to filter and group data for average calculations:**
    - Example 1: Calculate the average score of students who scored above 70 in Math.
    - 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).

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.
- **Example:**
    - Use `MAX(math_score)` in a subquery to find the highest score.
    - Use `WHERE math_score < (SELECT MAX(math_score) FROM Students)` to exclude the top score and then use `MAX` again to find the second highest score.

Deliverables

1. **SQL Queries**
    - Include clear, structured SQL queries with comments explaining each step.
2. **Screenshots of Results**
    - Run the queries in your database environment and capture screenshots of the outputs.
3. **Explanation of Subquery Usage**
    - Provide detailed explanations of where and how subqueries were used in each query.
4. **Summary of Findings**
    - Highlight key insights from the results, such as the top-performing students, average scores, or trends in Math performance.

How to Execute

1. **Setup the Database**
    - Create the `Students` table with appropriate data types for scores.
    - Insert sample data for testing.

Main Flow Services and Technologies Pvt. Ltd.
Contact Us. +91 9389641586, +91 97736 99074
Email-Add. contact.mainflow@gmail.com
www.mainflow.in

2. **Run Queries Sequentially**
    - Execute each query one by one and review the results.
3. **Validate Results**
    - Ensure the queries handle edge cases, such as ties in scores or missing data.
4. **Document Findings**
    - Save query outputs, include explanations, and summarize findings in a report.

Example Queries

Query 1: Top Students by Total Scores

```
SELECT name,
    (math_score + science_score + english_score) AS total_score
FROM Students
ORDER BY total_score DESC
LIMIT 5;
```

Query 2: Average Score of Students Who Scored Above 70 in Math

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

Query 3: Find the Second-Highest Math Score

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

General Guidelines

1. Use aliases for subqueries to improve readability and organization.
2. Handle edge cases such as students with identical scores or missing score entries.
3. Use aggregate functions (`SUM`, `AVG`, `MAX`) for summarizing data effectively.
4. Optimize queries for larger datasets by ensuring proper indexing on relevant fields.

Main Flow Services and Technologies Pvt. Ltd.
Contact Us. +91 9389641586, +91 97736 99074
Email-Add. contact.mainflow@gmail.com
www.mainflow.in

Expected Outcomes

1. A ranked list of top-performing students based on total scores.
2. Insights into average performance based on specific score conditions.
3. Identification of the second-highest Math score.
4. Visual proof of SQL queries and outputs documented in screenshots.

Deadline Compliance

- **Restriction**: **Submit the project within 7 days** from the start date.
- **Reason**: Meeting deadlines is crucial in the real-world software development environment. This restriction helps students practice **time management** and **task prioritization**. In professional settings, tight deadlines are often the norm, and learning to meet them without compromising quality is an essential skill.
- **Learning Outcome**: Students will learn to manage their time effectively, complete projects under pressure, and **deliver results on time**, which are all important skills in the workplace.