Name : Akhil Sai Sammeta

SQL in Action: Ranking & Running Totals

Overview

This document details an SQL exercise focused on ranking students based on their total scores and calculating running totals for Math scores. The exercise was implemented on **Mianflow**.

Steps

Step 1: Create the Students Table

The following SQL query is used to create a Students table with relevant fields:

```
CREATE TABLE Students (

    StudentID INT PRIMARY KEY,

    Name VARCHAR(100),

    MathScore INT,

    TotalScore INT

);
```

Step 2: Insert Sample Data

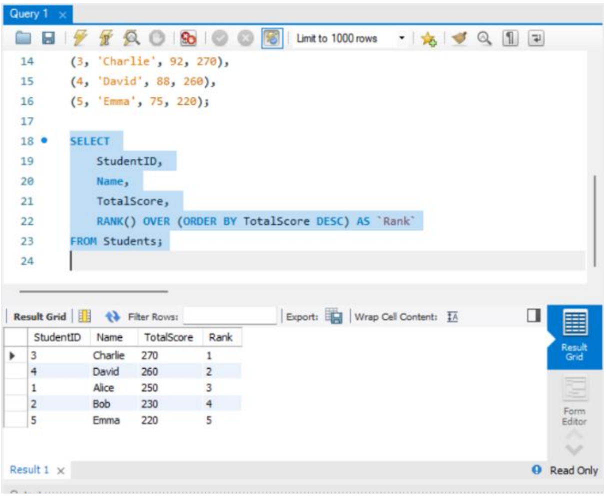
Sample data is inserted into the table for analysis:

```
INSERT INTO Students (StudentID, Name, MathScore, TotalScore) VALUES

(1, 'Alice', 85, 250),

(2, 'Bob', 78, 230),

(3, 'Charlie', 92, 270),

(4, 'David', 88, 260),

(5, 'Emma', 75, 220);
```

Task 1: Rank Students Based on Total Scores

To rank students based on their TotalScore, the following query is executed:

```
SELECT

    StudentID,

    Name,

    TotalScore,

    RANK() OVER (ORDER BY TotalScore DESC) AS Rank

FROM Students;
```



```
(3, 'Charlie', 92, 270),
(4, 'David', 88, 260),
(5, 'Emma', 75, 220);

SELECT
    StudentID,
    Name,
    TotalScore,
    RANK() OVER (ORDER BY TotalScore DESC) AS `Rank`
FROM Students;
```

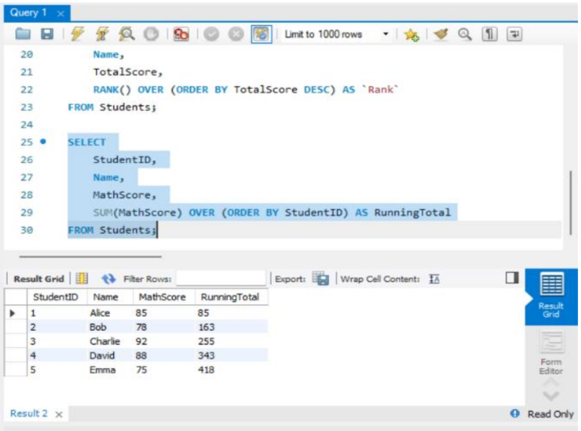
StudentID	Name	TotalScore	Rank
3	Charlie	270	1
4	David	260	2
1	Alice	250	3
2	Bob	230	4
5	Emma	220	5

This query assigns ranks in descending order of TotalScore.

Task 2: Calculate Running Totals for Math Scores

To compute the cumulative total of MathScore ordered by StudentID, the following query is used:

```
SELECT
    StudentID,
    Name,
    MathScore,
    SUM(MathScore) OVER (ORDER BY StudentID) AS RunningTotal
FROM Students;
```



```
        Name,
        TotalScore,
        RANK() OVER (ORDER BY TotalScore DESC) AS `Rank`
    FROM Students;

    SELECT
        StudentID,
        Name,
        MathScore,
        SUM(MathScore) OVER (ORDER BY StudentID) AS RunningTotal
    FROM Students;
```

StudentID	Name	MathScore	RunningTotal
1	Alice	85	85
2	Bob	78	163
3	Charlie	92	255
4	David	88	343
5	Emma	75	418

Conclusion

The use of SQL window functions such as RANK() and SUM() OVER() proves to be effective in analyzing student performance data. These queries help in ranking students efficiently and computing cumulative totals, which are essential for reporting and analytics.