# *Title-Window Functions (Task04)*

# *Presented by-Shoaib Asim*

# *Submission Date-03-08-2025*

# *Work Platform-My SQL Workbench*

**Objective**: Utilize SQL window functions to rank students and perform cumulative analysis.

**1.Dataset Setup**

**- CREATE TABLE Students:**

CREATE TABLE Students (

StudentID INT PRIMARY KEY,

Name VARCHAR(50),

MathScore INT,

TotalScore INT

);

**-Populate with sample data for the analysis:**

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

VALUES

(1, 'Aisha', 85, 260),

(2, 'Rahul', 92, 280),

(3, 'Mei', 75, 240),

(4, 'Omar', 88, 270),

(5, 'Lina', 95, 290),

(6, 'Kenji', 80, 250),

(7, 'Sara', 78, 245),

(8, 'Zain', 90, 275);

**Task 1**: **Rank Students Based on Total Scores:**

**SQL Query**

SELECT

StudentID,

Name,

TotalScore,

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

FROM Students;

**OUTPUT:**

A screenshot of a computer

AI-generated content may be incorrect.

**Explanation**

* RANK() OVER (ORDER BY TotalScore DESC): Assigns ranks in descending order of TotalScore.
* Ties share the same rank, and the next rank is skipped.

**Task 2**: **Calculate Running Totals for Math Scores:**

**SQL Query**

SELECT

StudentID,

Name,

MathScore,

SUM(MathScore) OVER (ORDER BY StudentID) AS RunningTotalMath

FROM Students;

**OUTPUT:**

A screenshot of a computer

AI-generated content may be incorrect.

**Explanation**

* SUM(MathScore) OVER (ORDER BY StudentID): Calculates the cumulative sum by StudentID order.
* Helps track how the total Math score grows row by row.

**Summary of Findings**

| Insight | Description |
| --- | --- |
| Top Performers | Ranked by TotalScore, useful for awards or recognition. |
| Ties Handled Smartly | RANK handles duplicate scores cleanly without misleading placement. |
| Progress Trend in Math Performance | Cumulative math scores help identify improvement or consistency over student IDs. |

**SQL QUERY FILE:**

