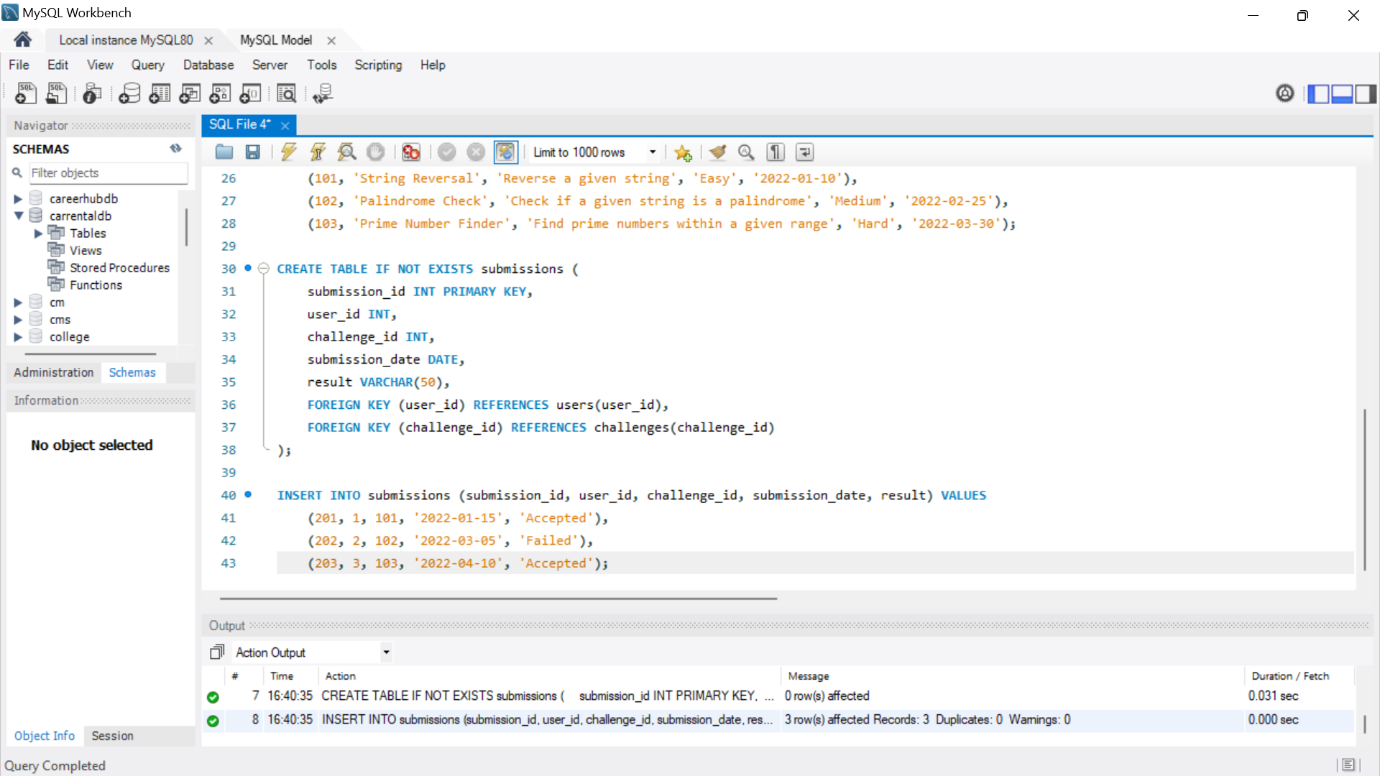
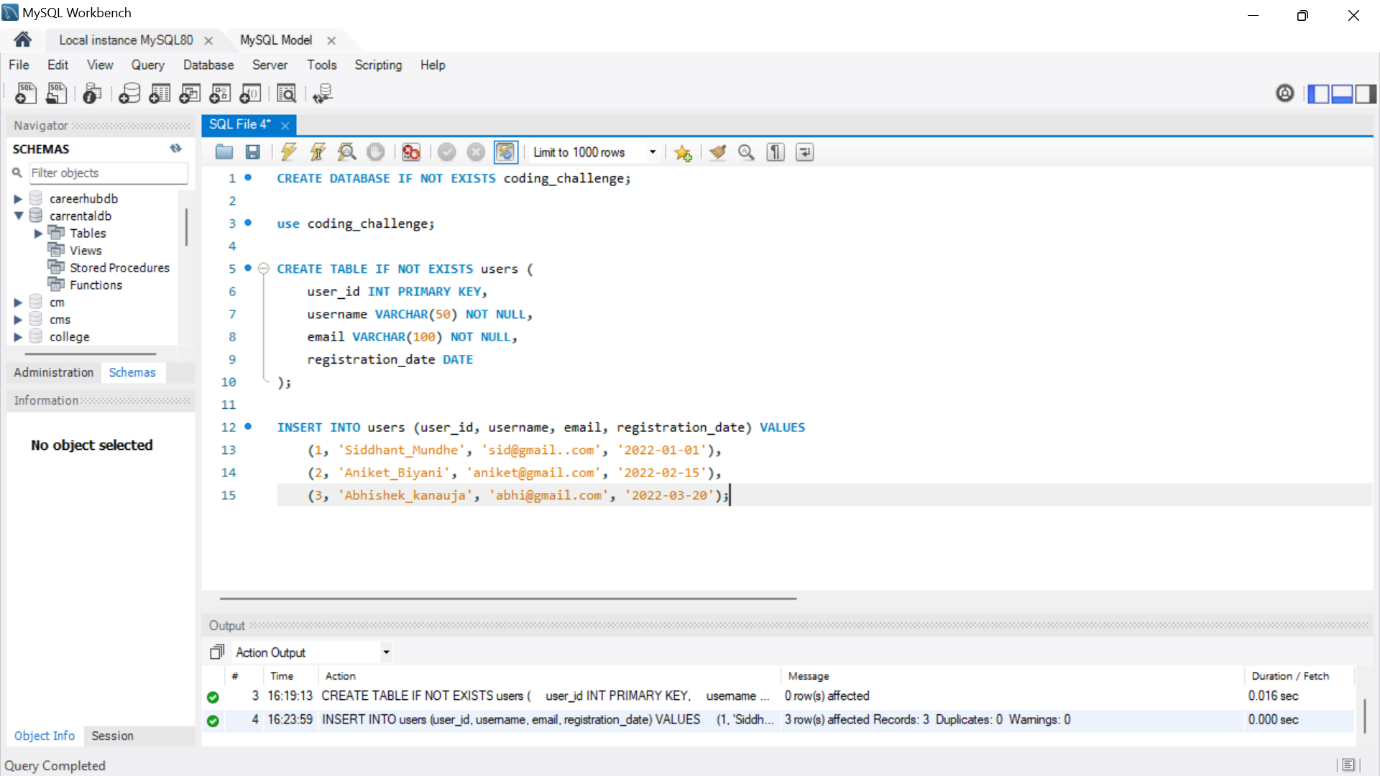
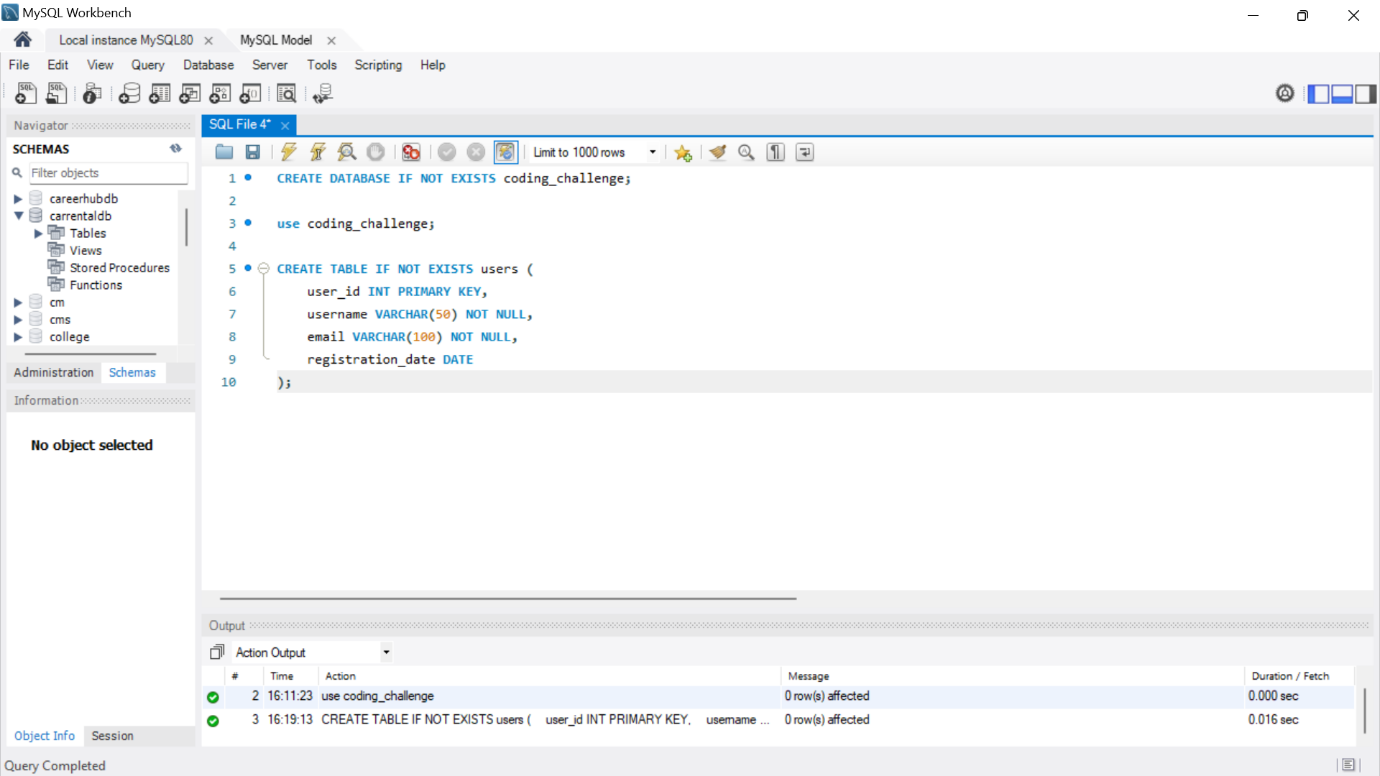
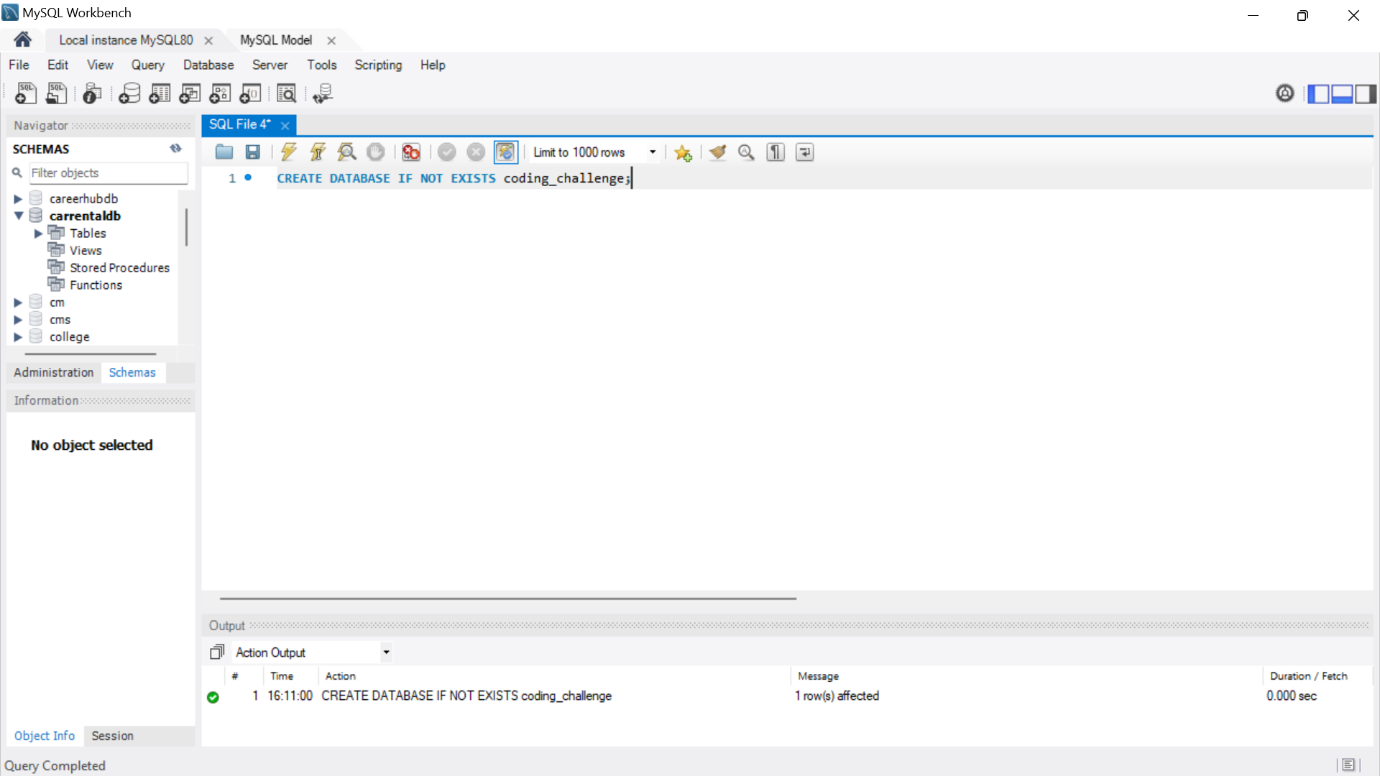
**Siddhant Mundhe**

**Data engineering batch 1**

CODING CHALLENGE

I have created a database named coding challenges with three tables: users, challenges, and submissions.



1. PARTITION BY Clause:

The PARTITION BY clause is used to divide the result set into partitions. It basically groups the rows based on a specified column or columns.

Example:

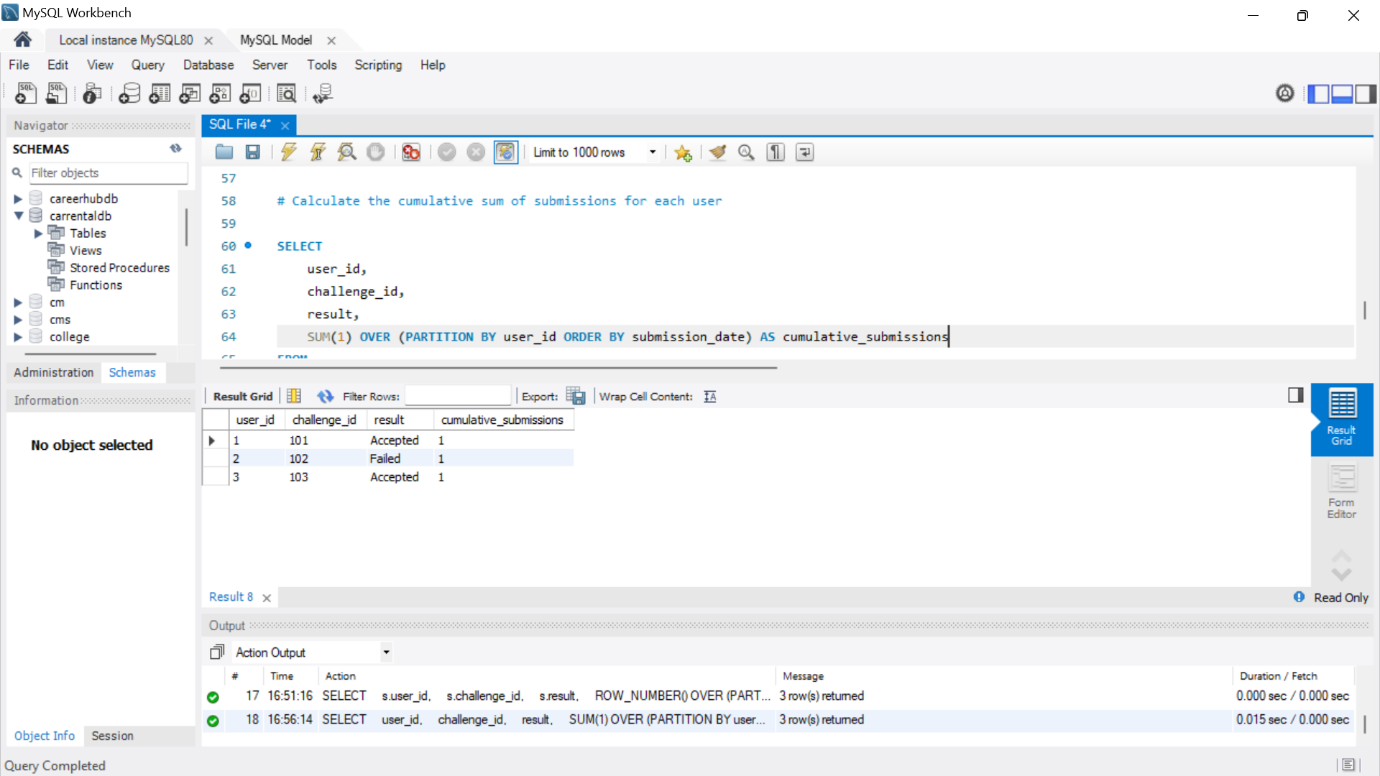
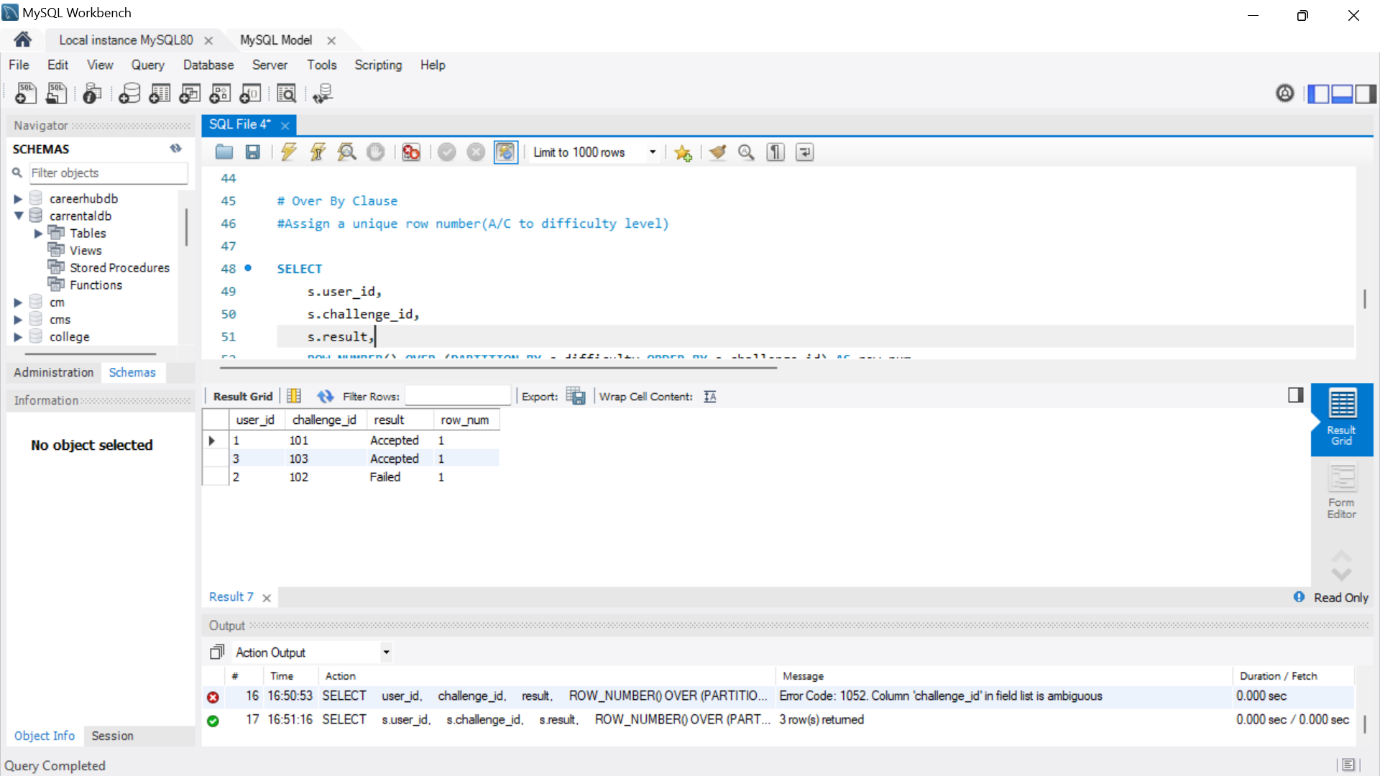
Suppose you have a table named Sales with columns Product, Region, and Revenue. You want to calculate the average revenue for each product within its respective region. You can use the PARTITION BY clause to achieve this.

1. OVER BY Clause:

Over clause is used with functions to compute aggregated values over a group of rows, referred to as a window. This clause gives you control over where the window starts and ends for each row in the result set.

Example:

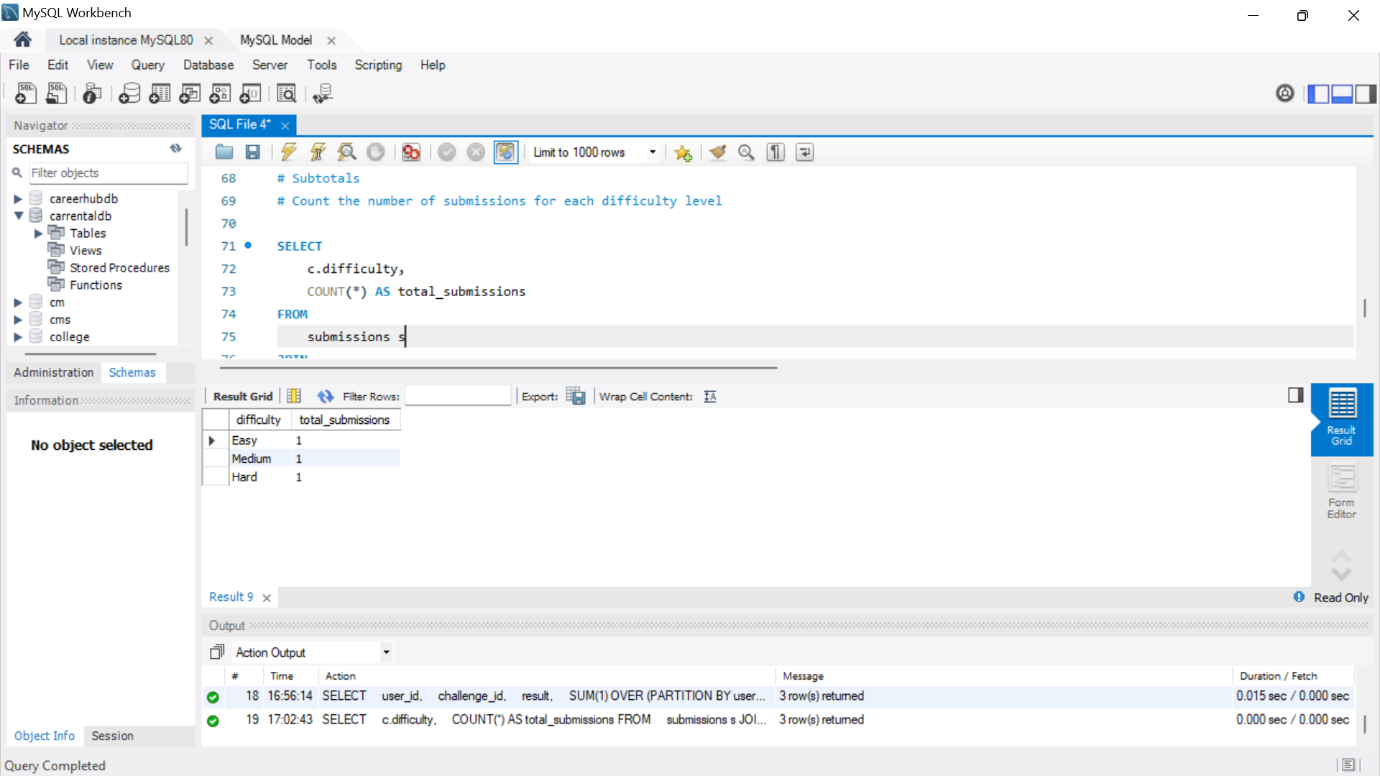
Imagine you have a table of students with columns for their names and scores. You want to find the average score for each student compared to the entire class. The OVER clause helps you do that.



1. Creating Subtotals

A subtotal is a figure that shows the sum of similar sets of data but it does not indicate the final total.

Creating subtotals in SQL involves using the GROUP BY clause along with aggregate functions to calculate summary values for each group.



1. Total Aggregations

Total aggregation in SQL involves calculating summary values for an entire dataset, without grouping the results based on specific criteria. It's like getting the grand total or overall summary for a particular column across all rows.

