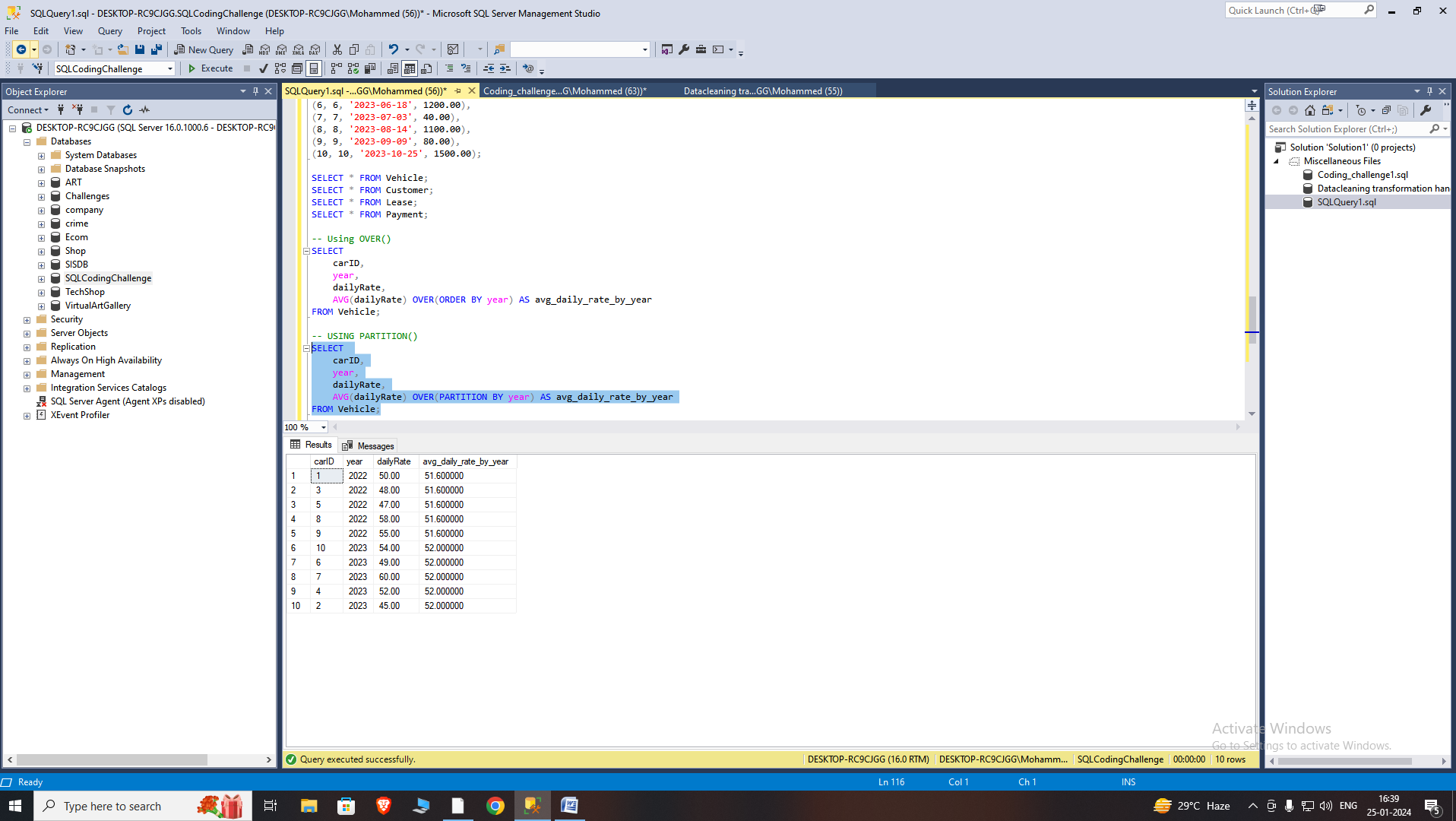
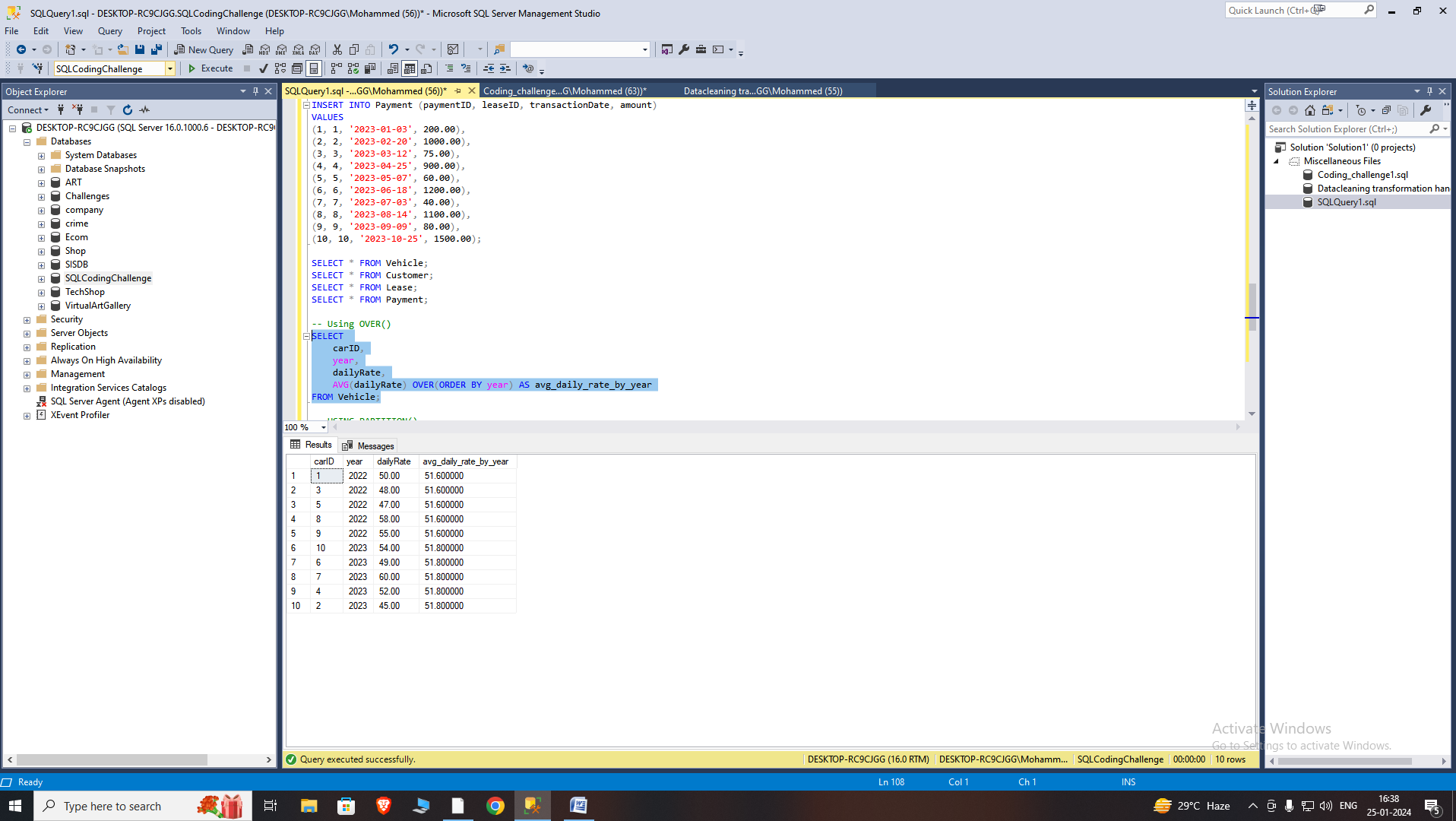
OVER() and PARTITION()

The OVER keyword in SQL is used in conjunction with window functions to perform calculations across a specified range of rows related to the current row within the result set. It's an essential part of the syntax when working with window functions.

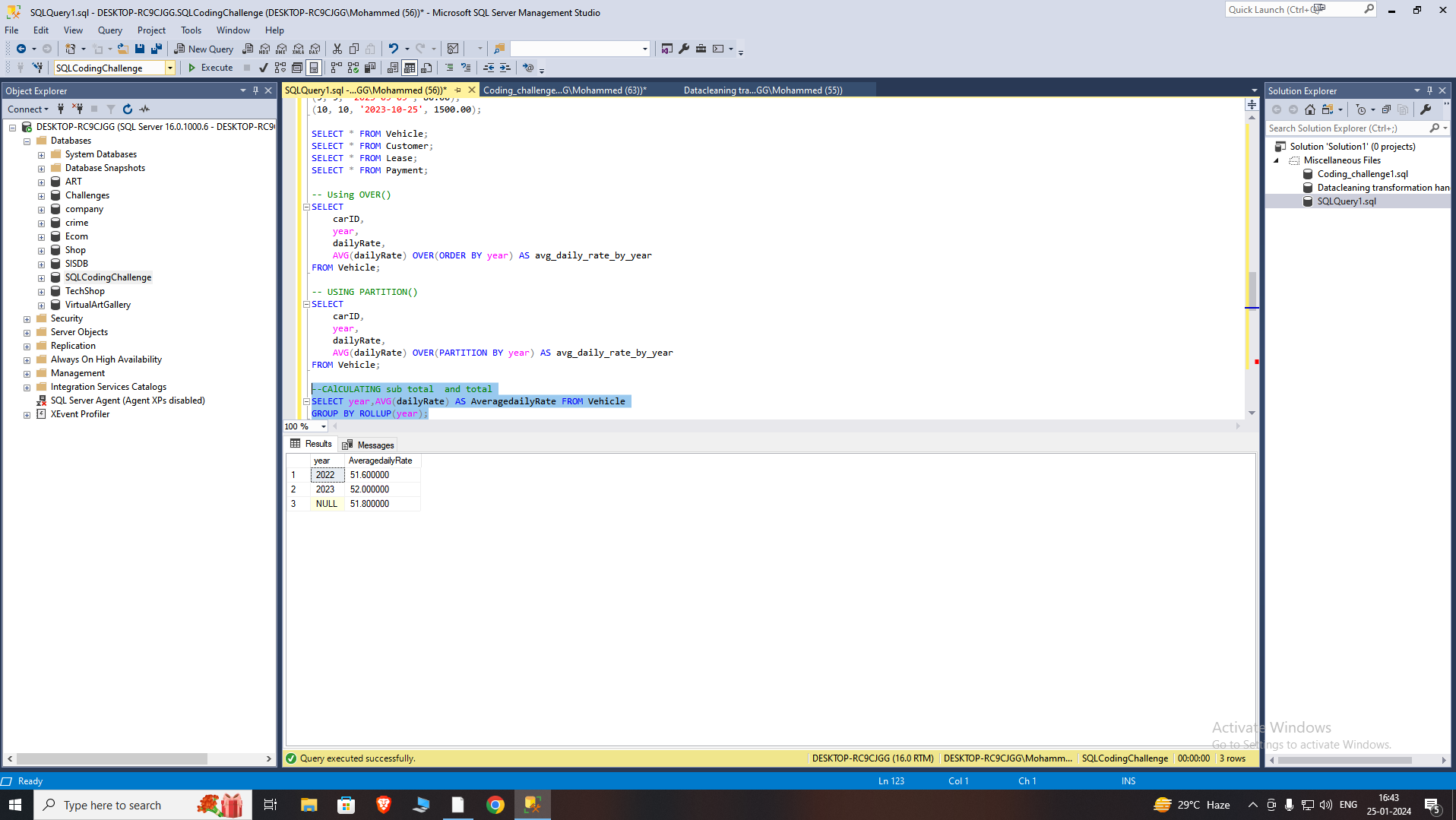


The PARTITION BY clause is used in conjunction with window functions to divide the result set into partitions to which the window function is applied. It allows you to perform calculations on a specific subset of rows within the result set, based on the values in one or more columns.



Creating Subtotals and Totals

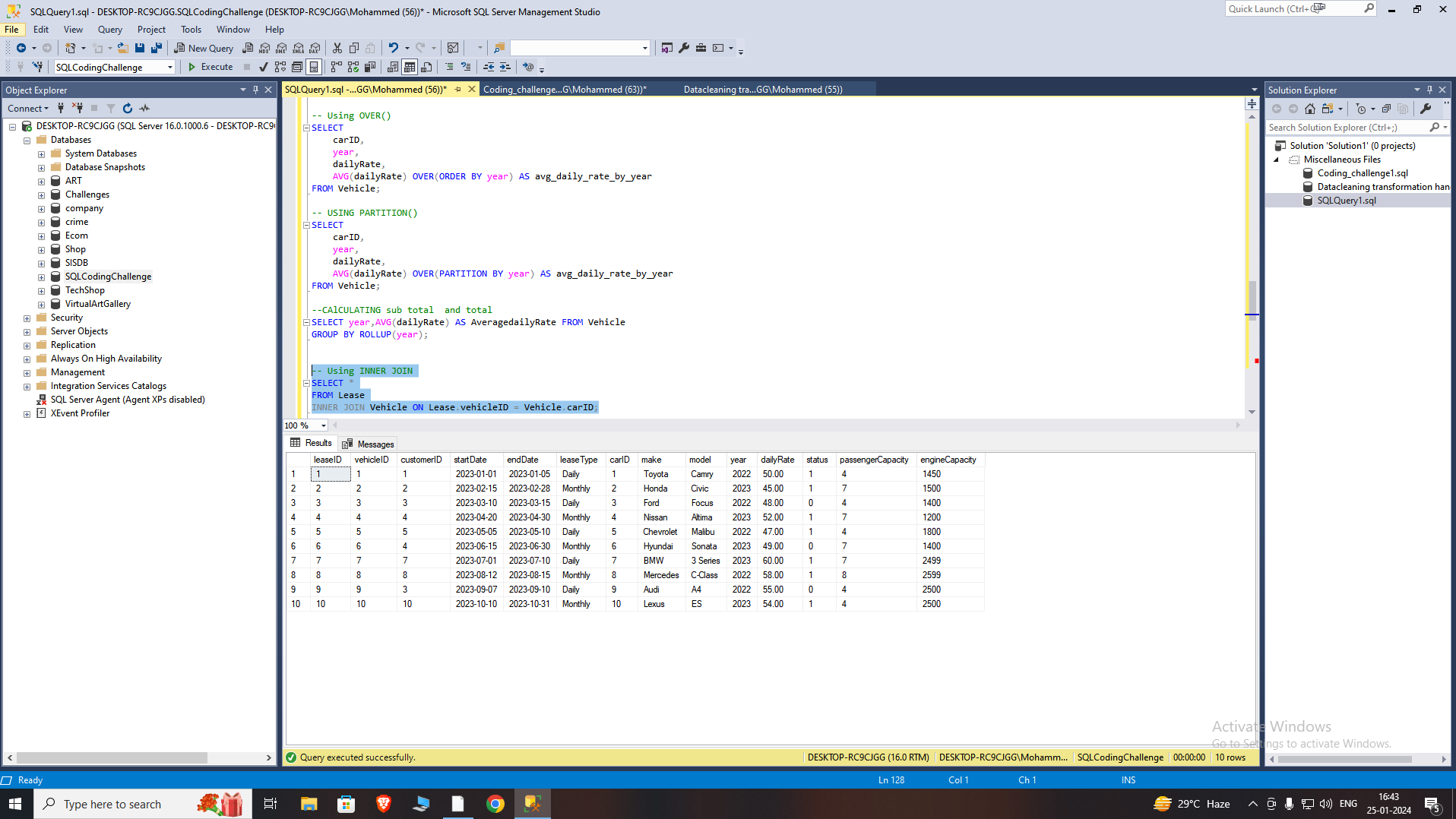
Subtotals and totals are created in SQL using ROLLUP().The ROLLUP operator enhances the capabilities of the GROUP BY clause by enabling the computation of subtotals and grand totals for a set of columns. It produces a result set that incorporates rows at various levels of aggregation.



JOINS

INNER JOIN:

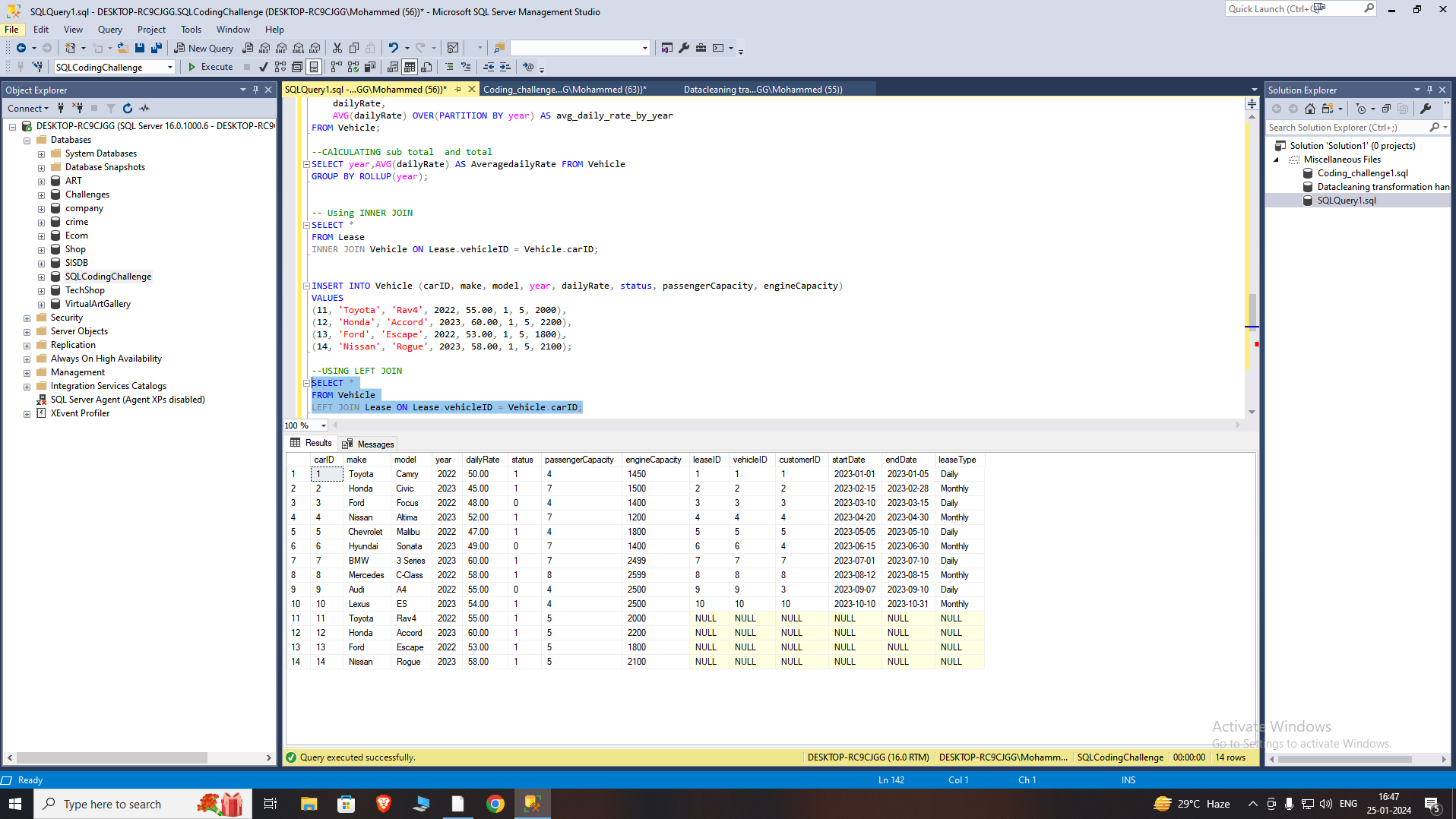
An INNER JOIN returns only the rows where there is a match in both tables based on the specified condition



This query retrieves all columns from the Lease table and Vehicle table where the vehicleID in Lease matches the carID in Vehicle.

LEFT JOIN:

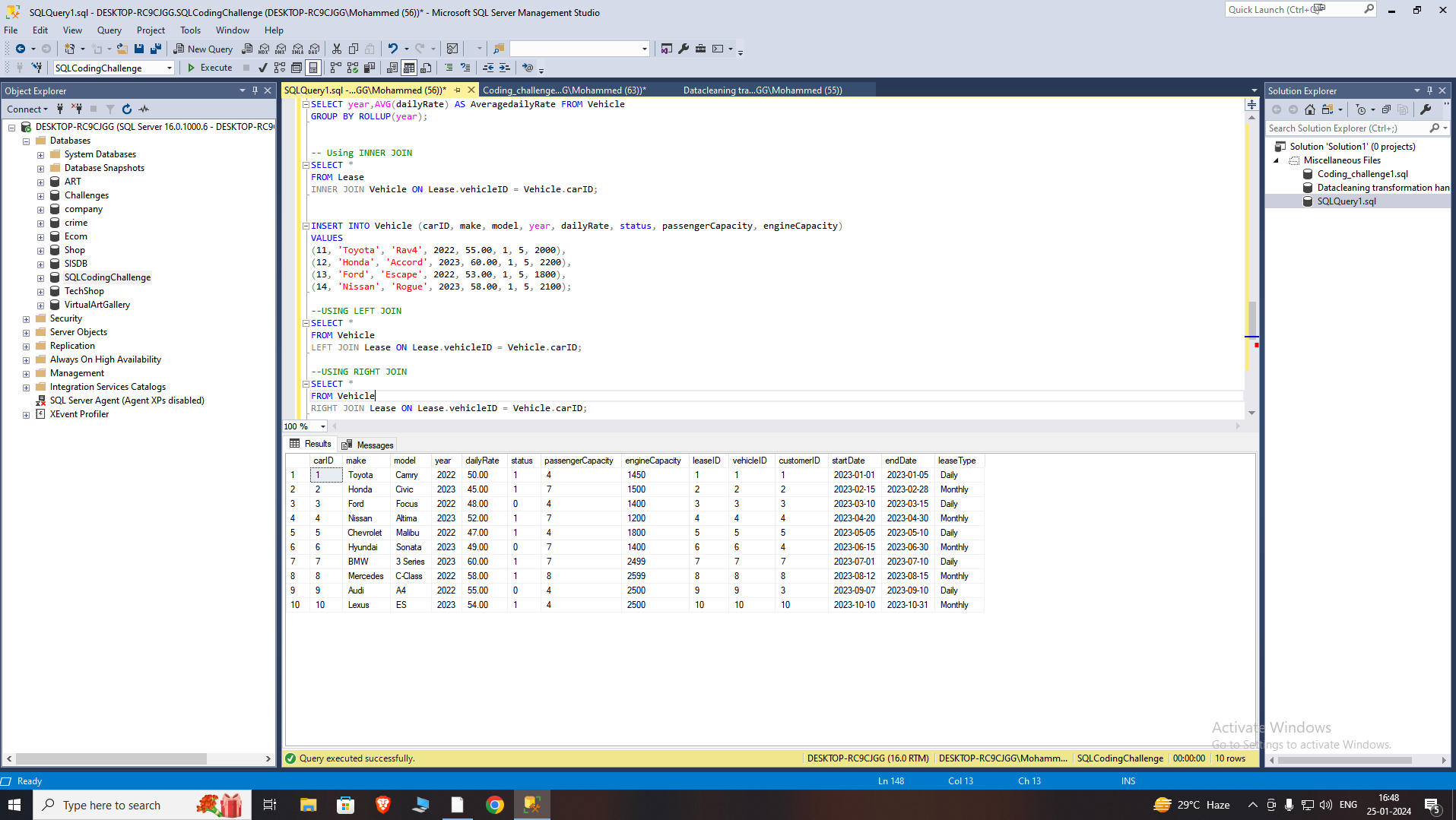
Returns all rows from the left table and the matched rows from the right table. If there is no match, NULL values are returned for columns from the right table.



THE LEFT JOIN here returns all rows from the left table Vehicle, and the matched rows from the right table Lease. If there is no match, NULL values are returned for columns from the right table.

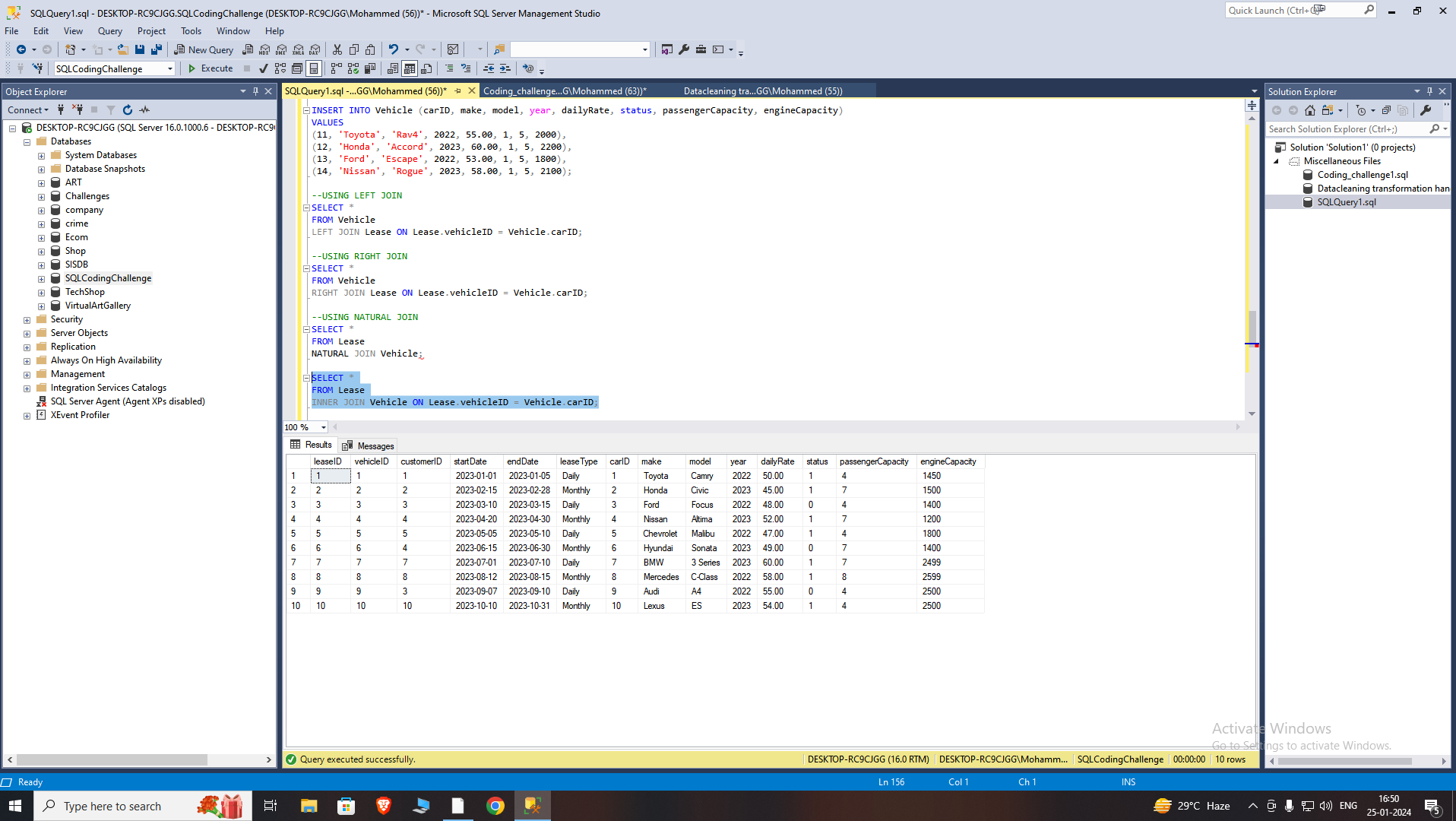
RIGHT JOIN:

A RIGHT JOIN returns all rows from the right table (Vehicle), and the matched rows from the left table (Lease). If there is no match, NULL values are returned for columns from the left table.



NATURAL JOIN:

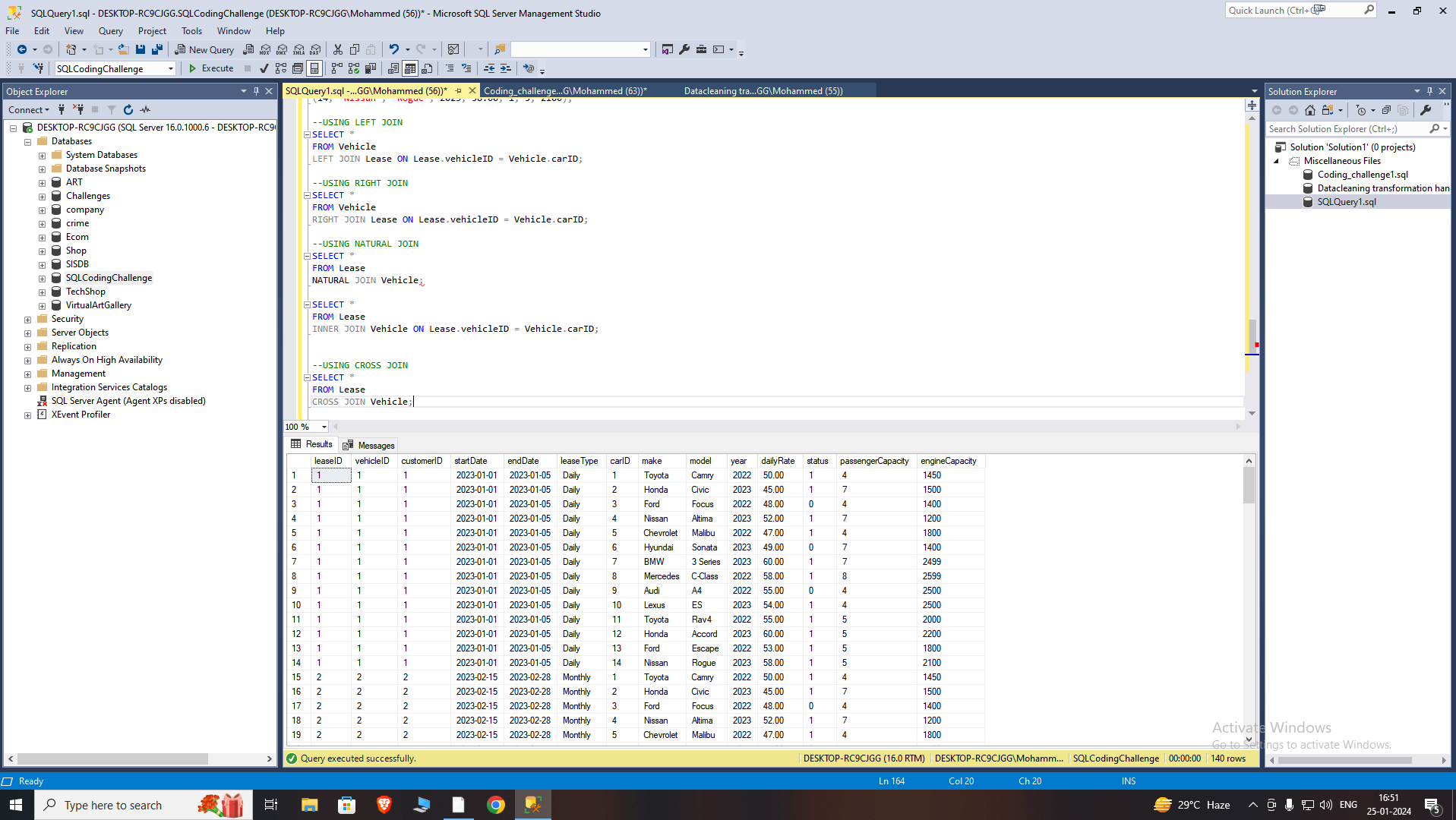
Automatically matches columns with the same name in both tables. It's a shorthand for specifying columns to join on.



SQL Server does not have Natural join so we can use INNER JOIN with explicit column names to get the same output

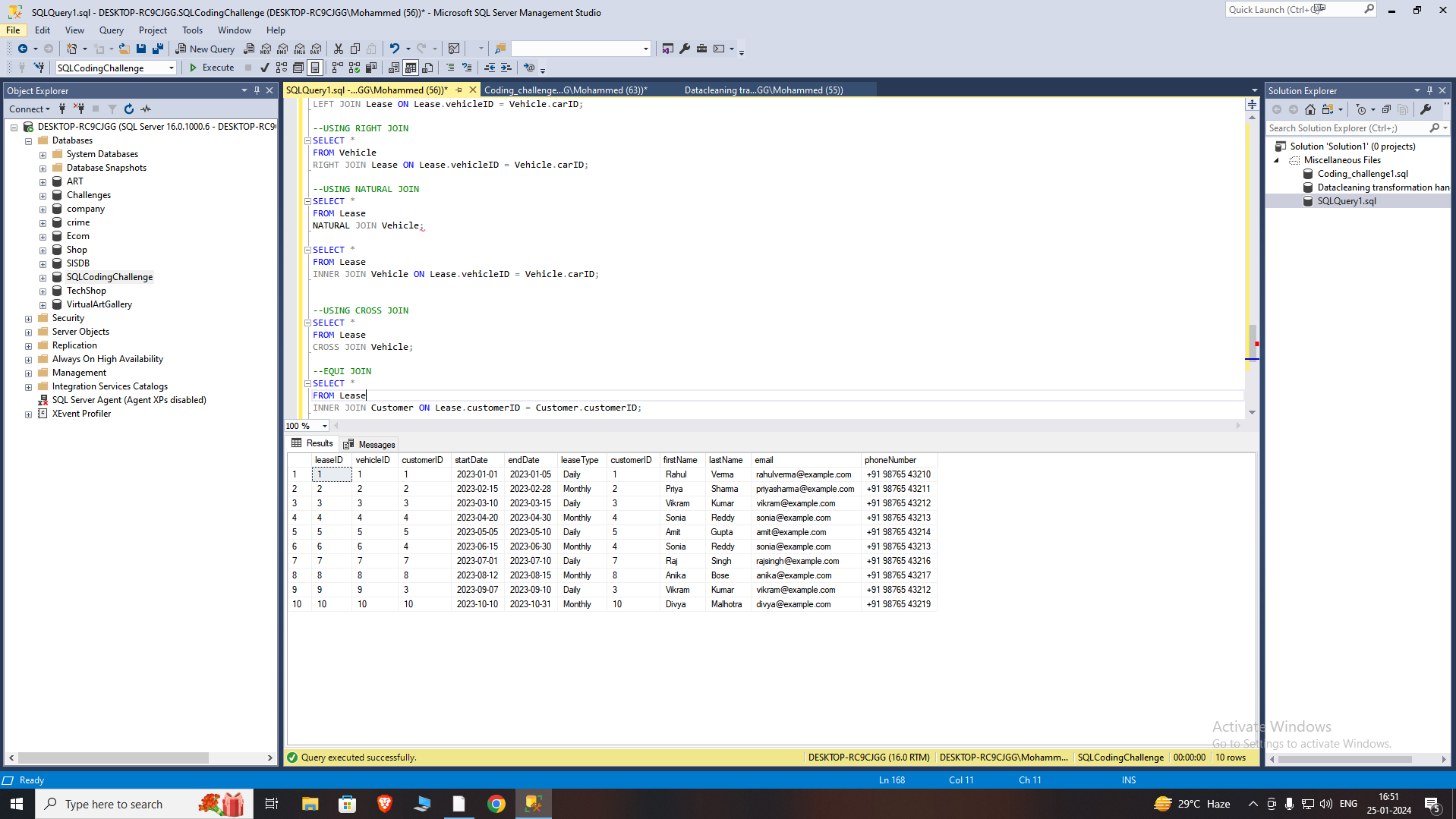
CROSS JOIN:

Returns the Cartesian product of two tables, combining every row from the first table with every row from the second table.



EQUI JOIN:

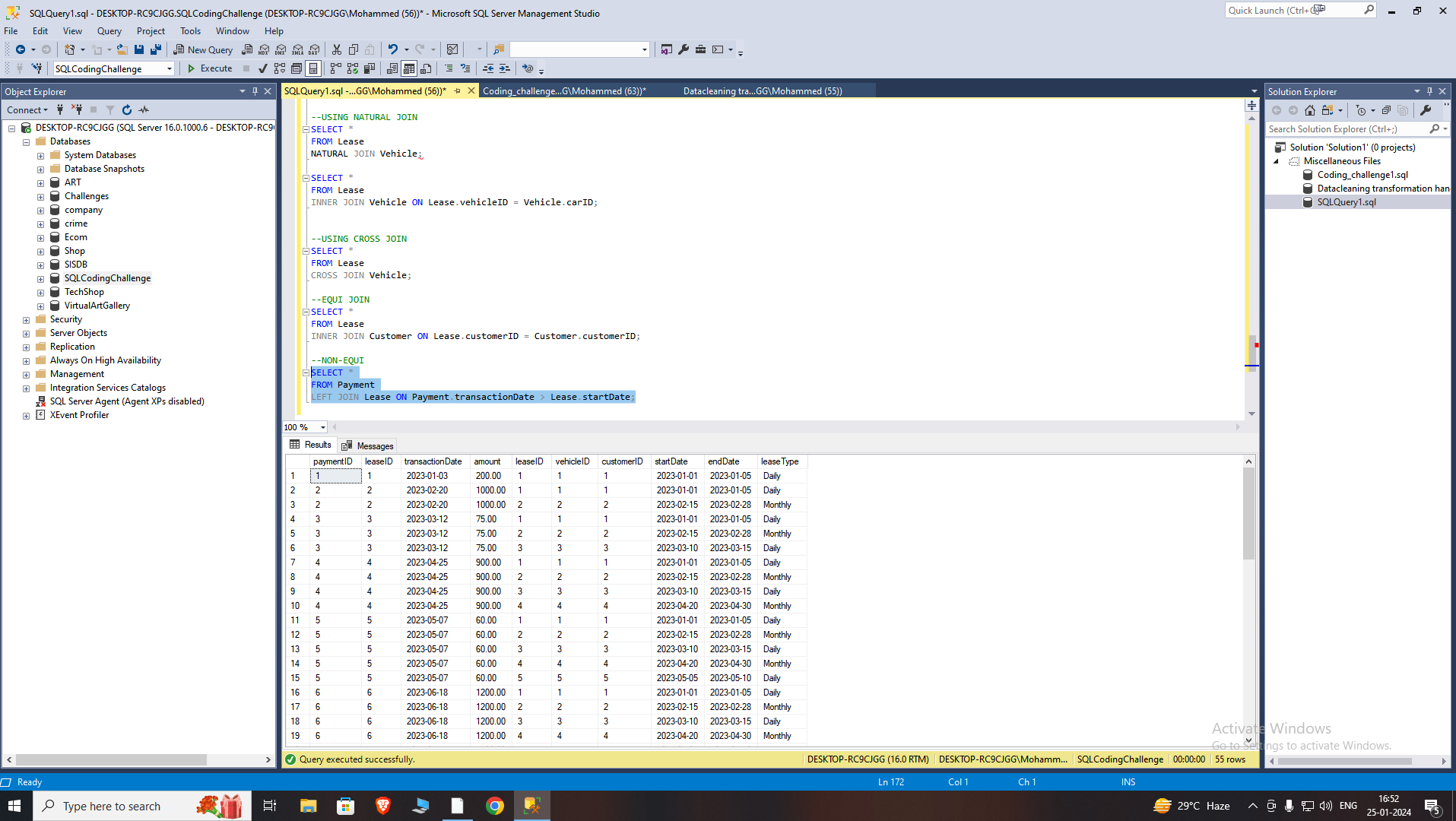
A type of join that uses the equality operator (=) to match rows based on a specified condition.



In this example, the INNER JOIN is based on the equality condition Lease.customerID = Customer.customerID, which means we are matching rows where the customer ID in the Lease table is equal to the customer ID in the Customer table.

NON EQUI JOIN:

A type of join that uses operators other than equality (e.g., <, >, <=, >=, <>) to match rows.



Here the LEFT JOIN is based on a non-equi condition Payment.transactionDate > Lease.startDate, which means we are matching rows where the transactionDate in the Payment table is greater than the startDate in the Lease table.

Creating Database tables and Inserting data