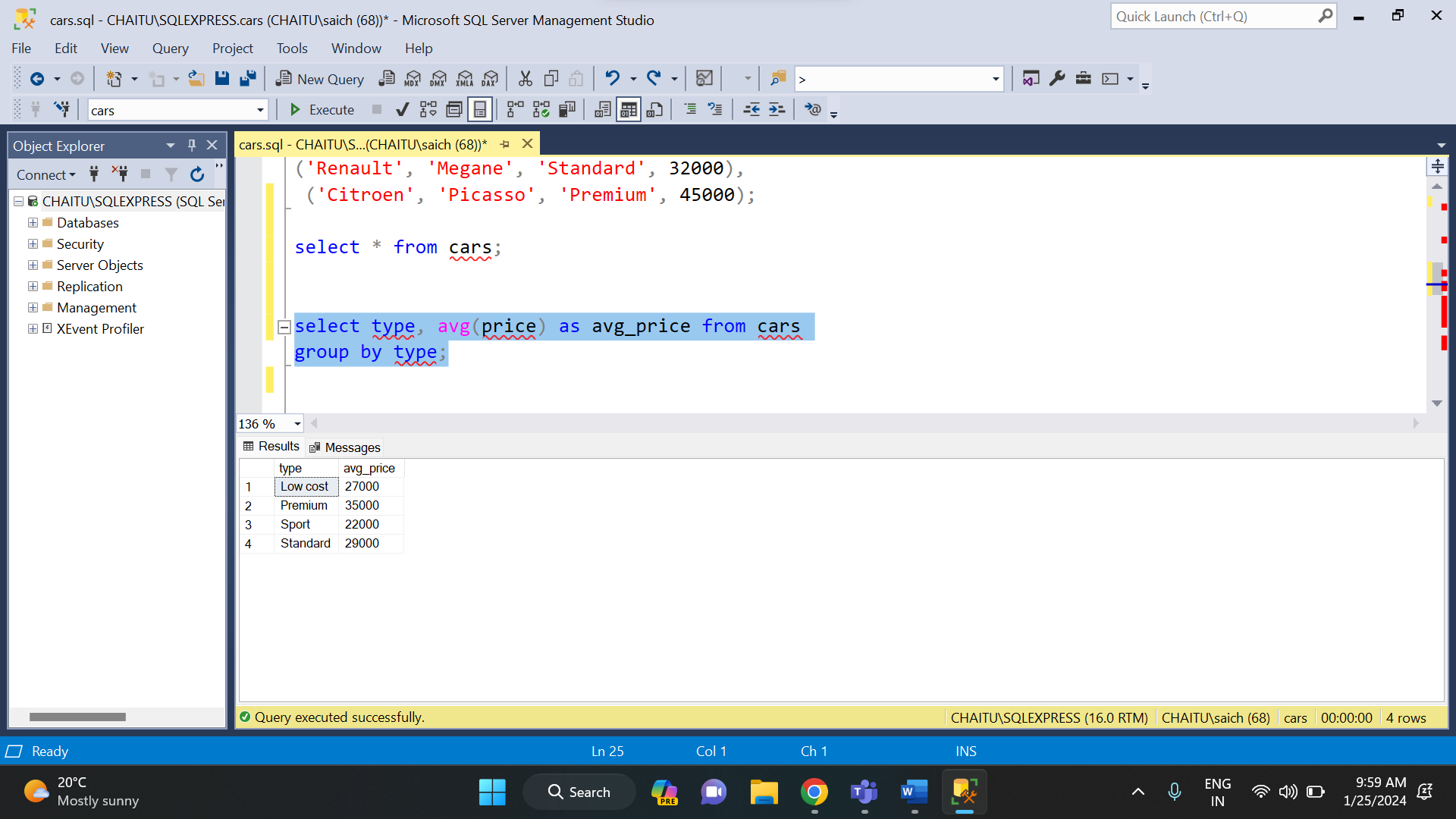
**Data Engineering Assignment – 6**

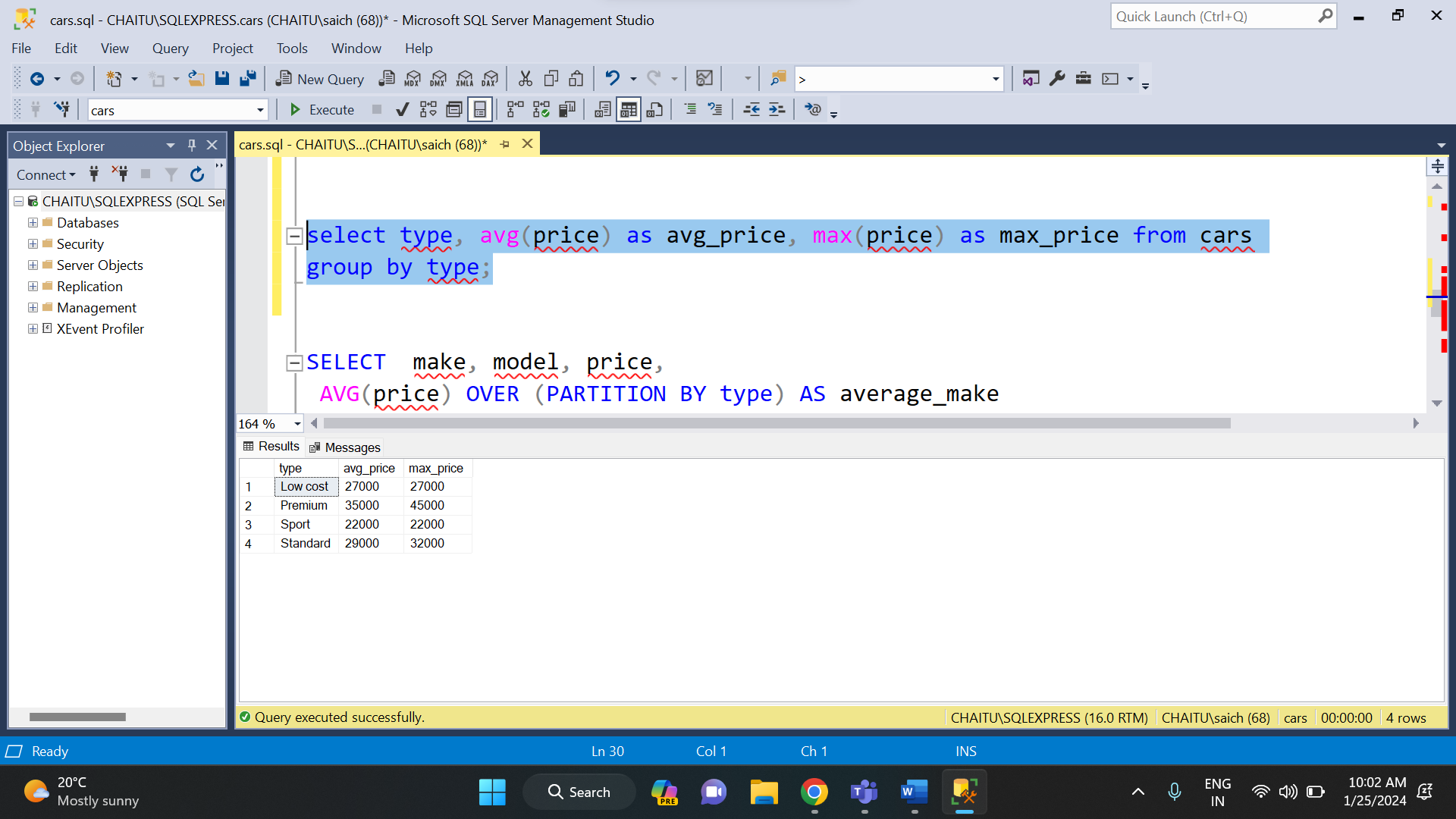
**Total Aggregations using SQL queries:**

Total aggregations are using the aggregate functions on the available database. Below are some of the examples that represent the total aggregations on SQL queries.

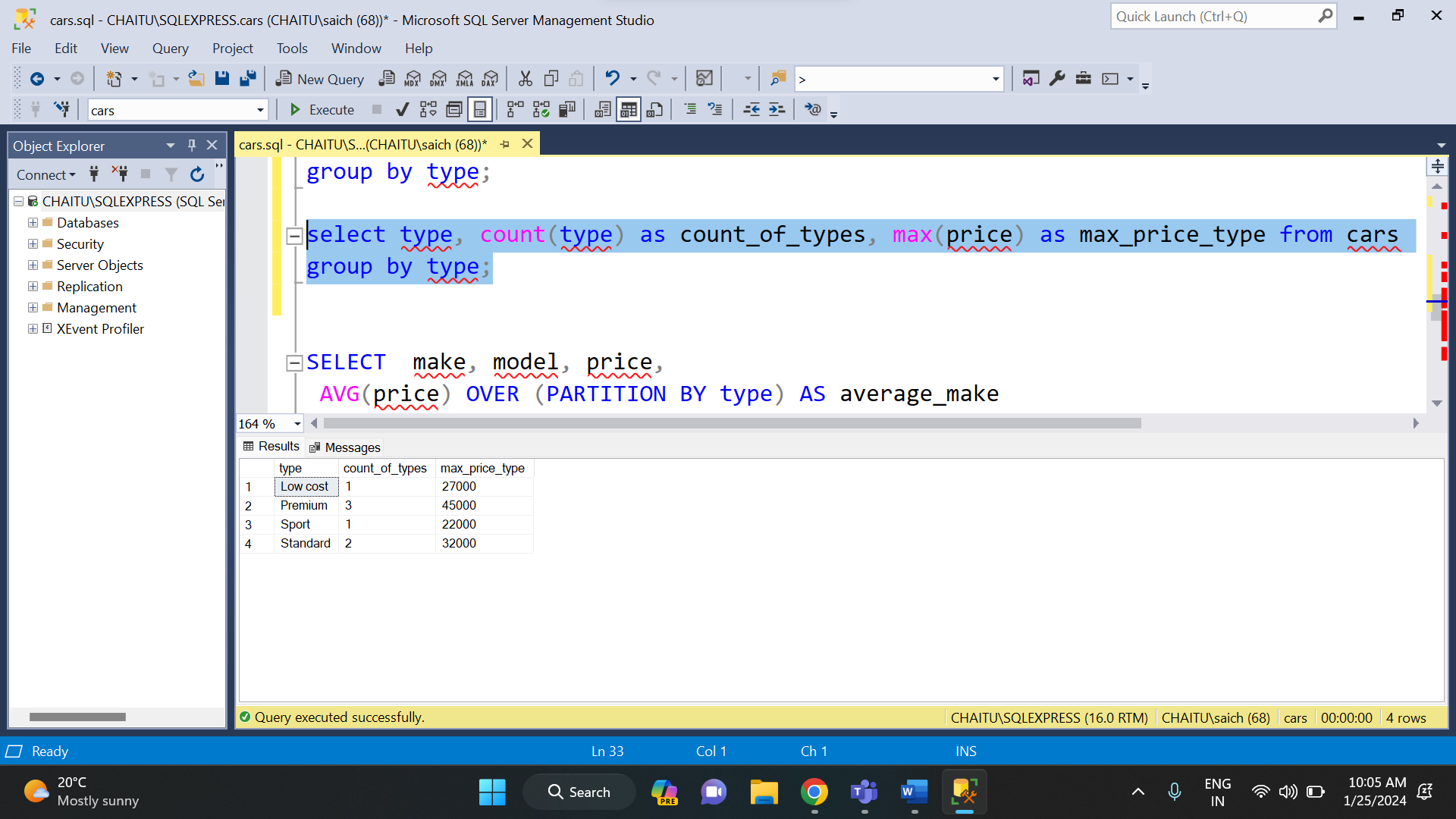
Example: 1



Example: 2

****

Example: 3



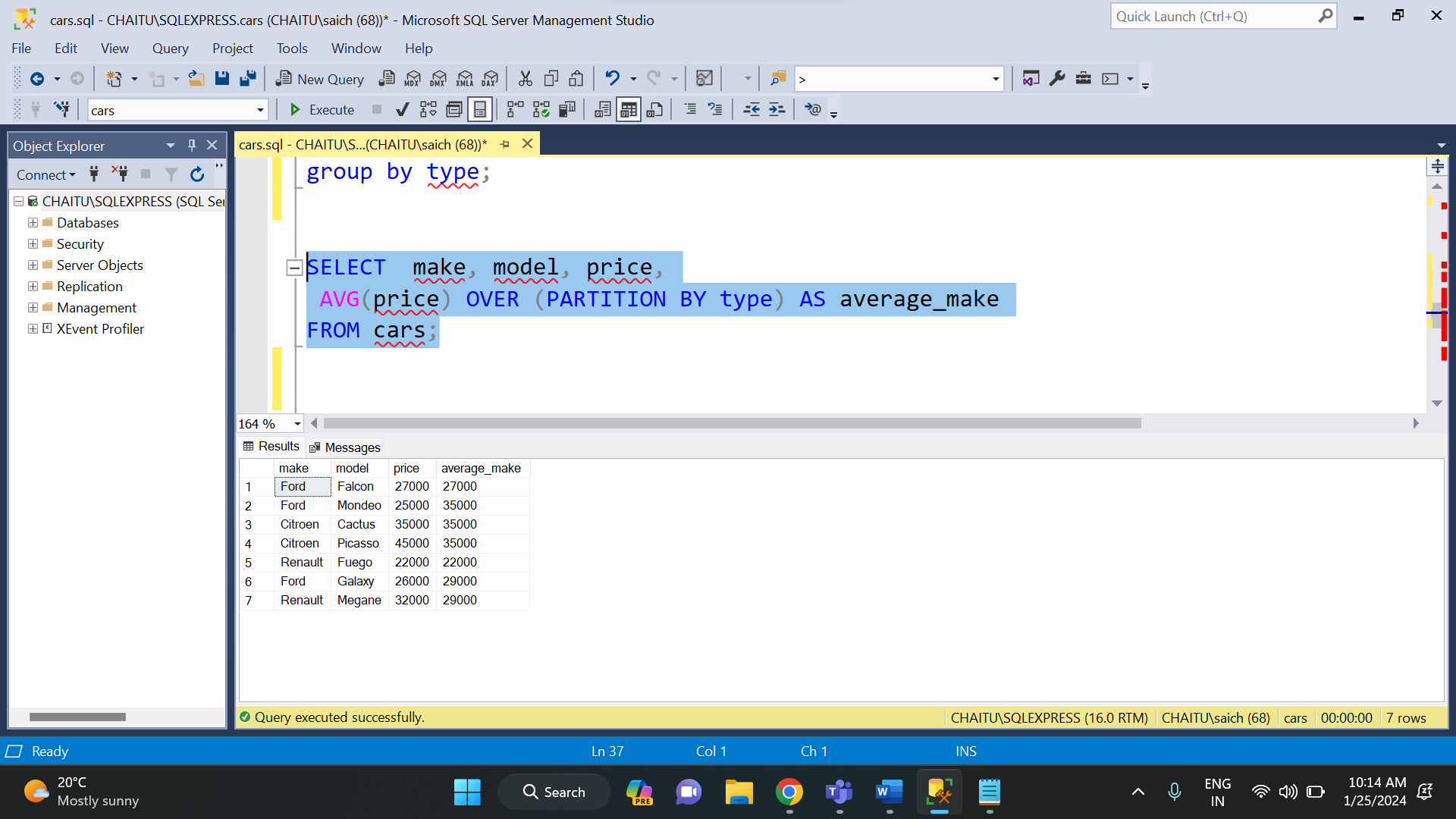
**OVER AND PARTITION BY CLAUSE:**

The PARTITION BY clause is used to divide the result set into partitions to which the window function is applied. It allows you to perform calculations separately for each partition. The OVER () clause defines the window or set of rows to which the window function is applied. If you don't specify PARTITION BY, the window is the entire result set.

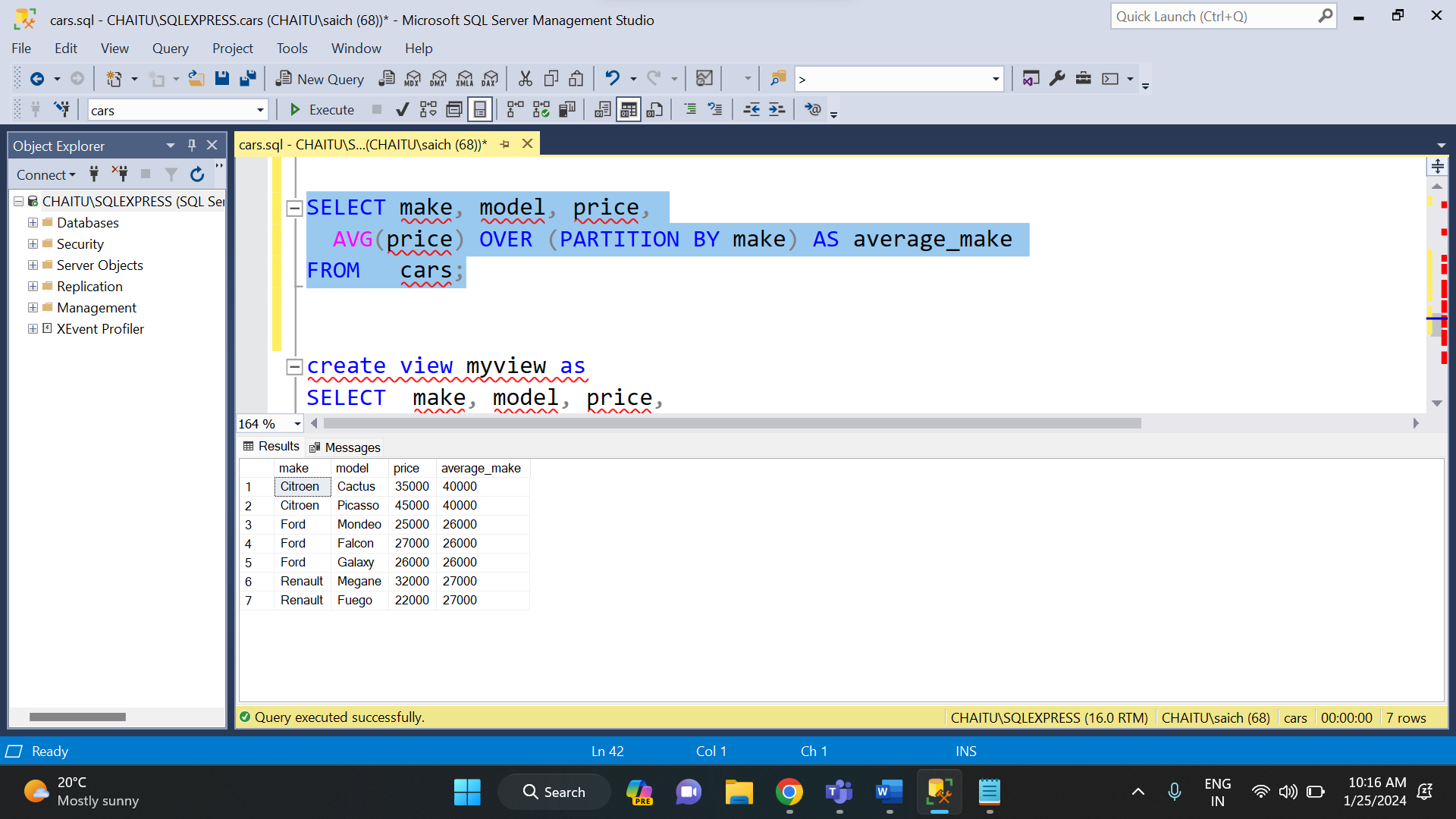
The PARTITION BY clause and the OVER () clause are both used in SQL, specifically in the context of window functions. These clauses are typically used together to perform calculations on a specific "window" or subset of rows within a result set.

And below are some of the examples that demonstrate the over () and partition by clause.

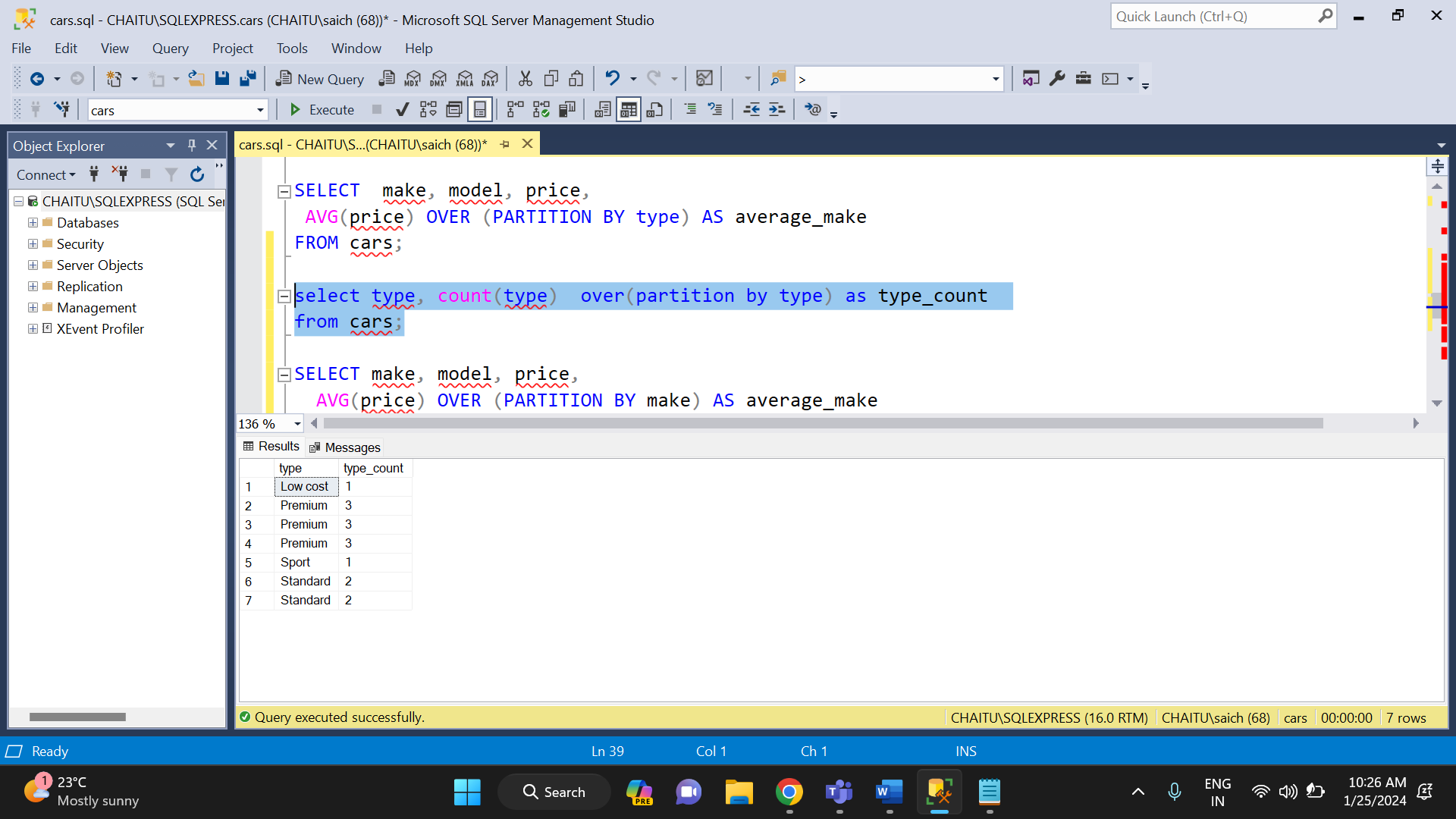
Example: 1



Example: 2



Example: 3

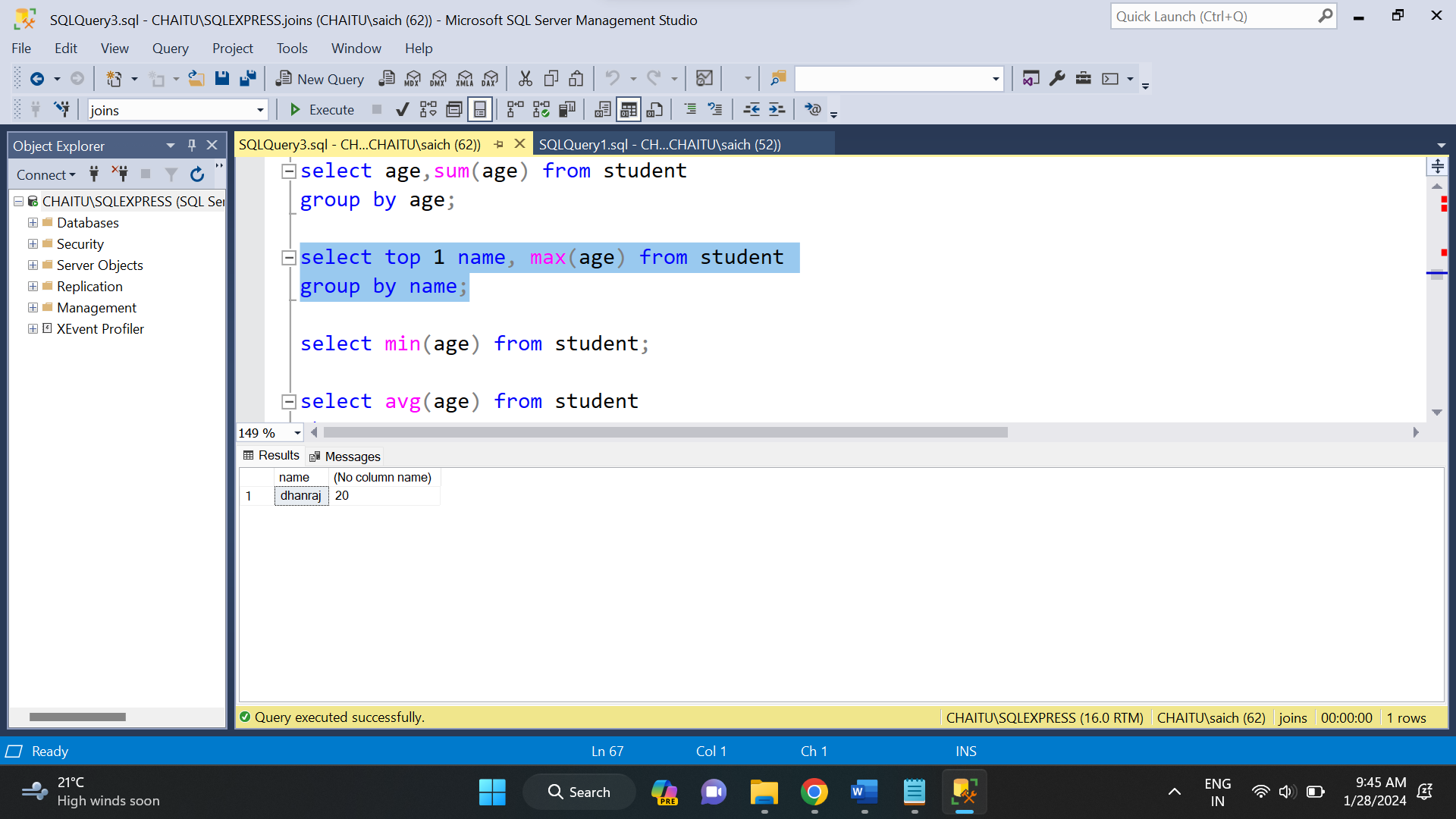


**RULES AND RESTRICTION TO GROUP AND FILTER DATA:**

GROUP BY enables you to use aggregate functions on groups of data returned from a query.

FILTER is a modifier used on an aggregate function to limit the values used in an aggregation. All the columns in the select statement that aren’t aggregated should be specified in a GROUP BY clause in the query.

Example of group by clause:



**ORDER OF EXECUTION OF SQL QUERIES:**

SQL order of execution refers to the sequence in which different clauses and operations within a SQL query are processed by the database management system.

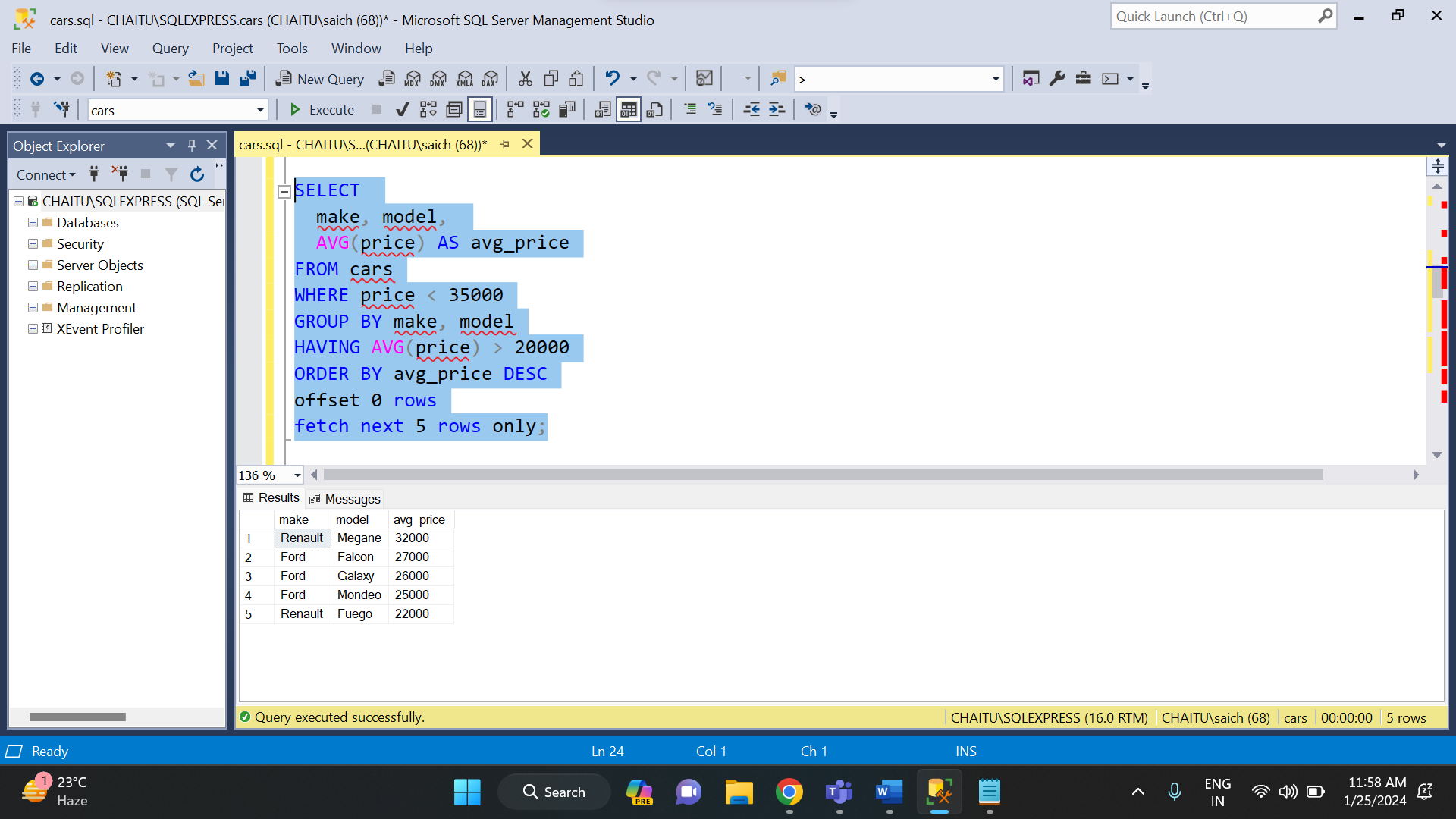
Each SQL query consists of various components such as SELECT, FROM, WHERE, GROUP BY, HAVING, and ORDER BY clauses, along with functions and operators. Understanding the order in which these components are executed is vital for producing accurate and efficient query results.

It is very important to understand the order of execution because it helps in:

* Getting accurate results
* Query optimization
* Efficient resource utilization
* Reduced query complexity

Let us see this order of execution of SQL query with some examples.

Example: 1



So here the order of execution will be as follows:

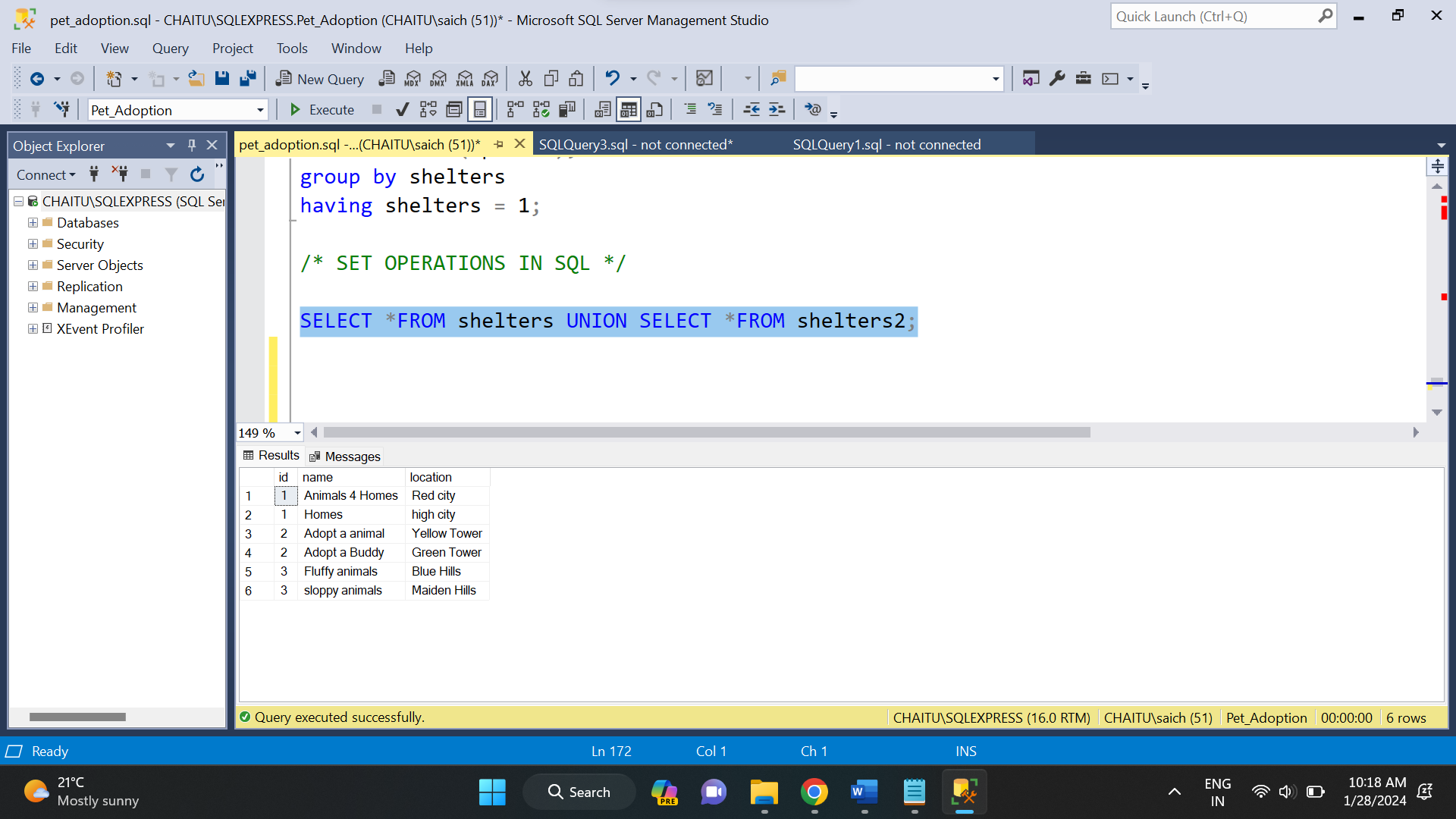
* FROM
* WHERE
* GROUP BY
* HAVING
* SELECT
* ORDER BY
* LIMIT/OFFSET

**DIFFERENCES BETWEEN UNION, EXCEPT AND INTERSECT:**

The main differences between union, except and interest are:

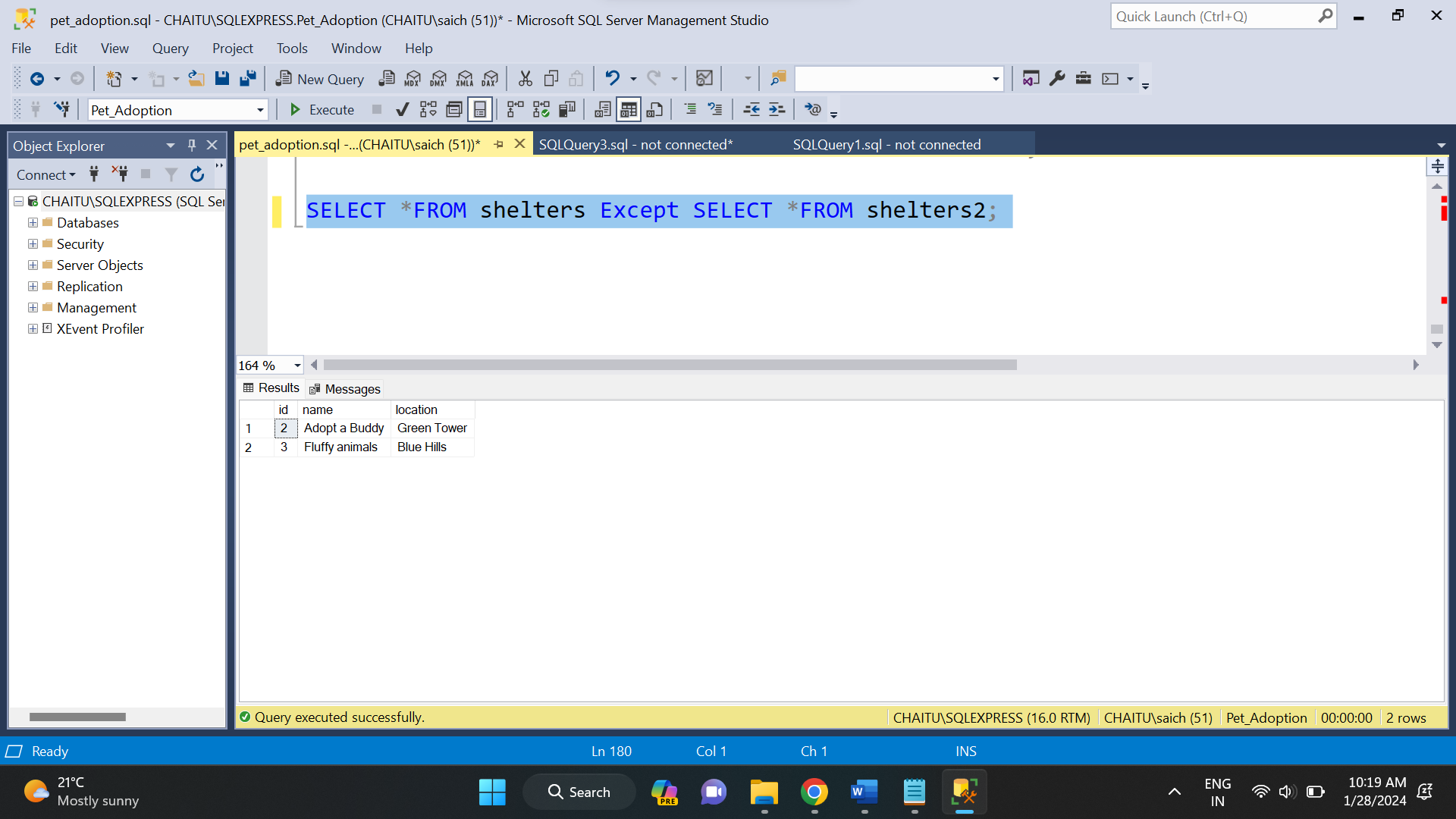
**UNION Operator:**

The UNION operator combines the result sets of two or more SELECT statements into a single result set. It removes duplicate rows from the combined result set by default. The columns in the SELECT statements must have compatible data types, and the number of columns in each SELECT statement must be the same. The order of rows in the final result set may not be the same as in the individual SELECT statements unless you use the ORDER BY clause.



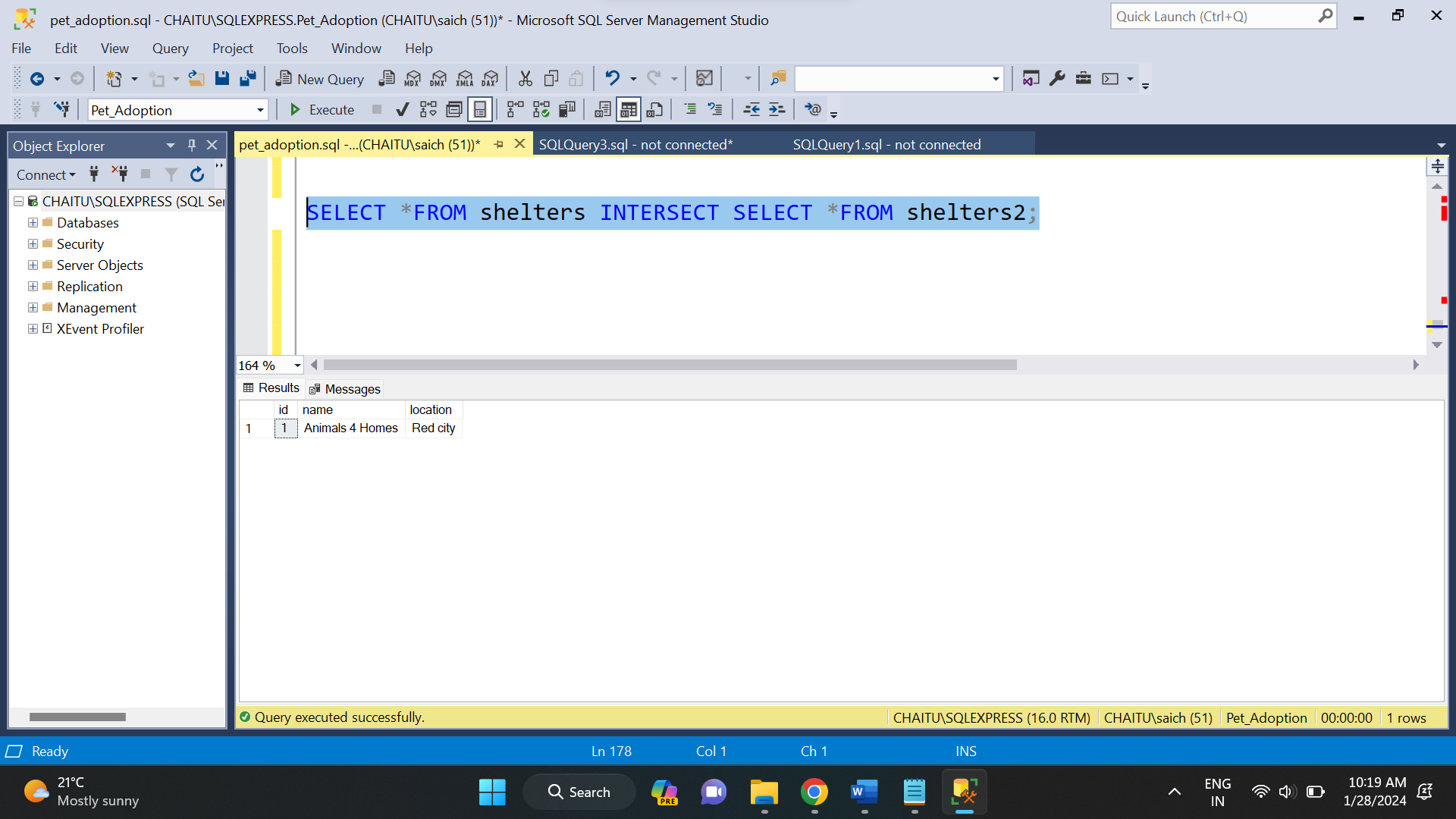
**EXCEPT Operator:**

The EXCEPT operator retrieves the rows present in the first result set but not in the second result set. It returns distinct rows from the first result set that do not have corresponding rows in the second result set. The columns in both SELECT statements must have compatible data types, and the number of columns in both statements must be the same.



**INTERSECT Operator:**

The INTERSECT operator is used to retrieve the rows that are common to both result sets. It returns distinct rows appearing in the first and second result sets. The columns in both SELECT statements must have compatible data types, and the number of columns in both statements must be the same.



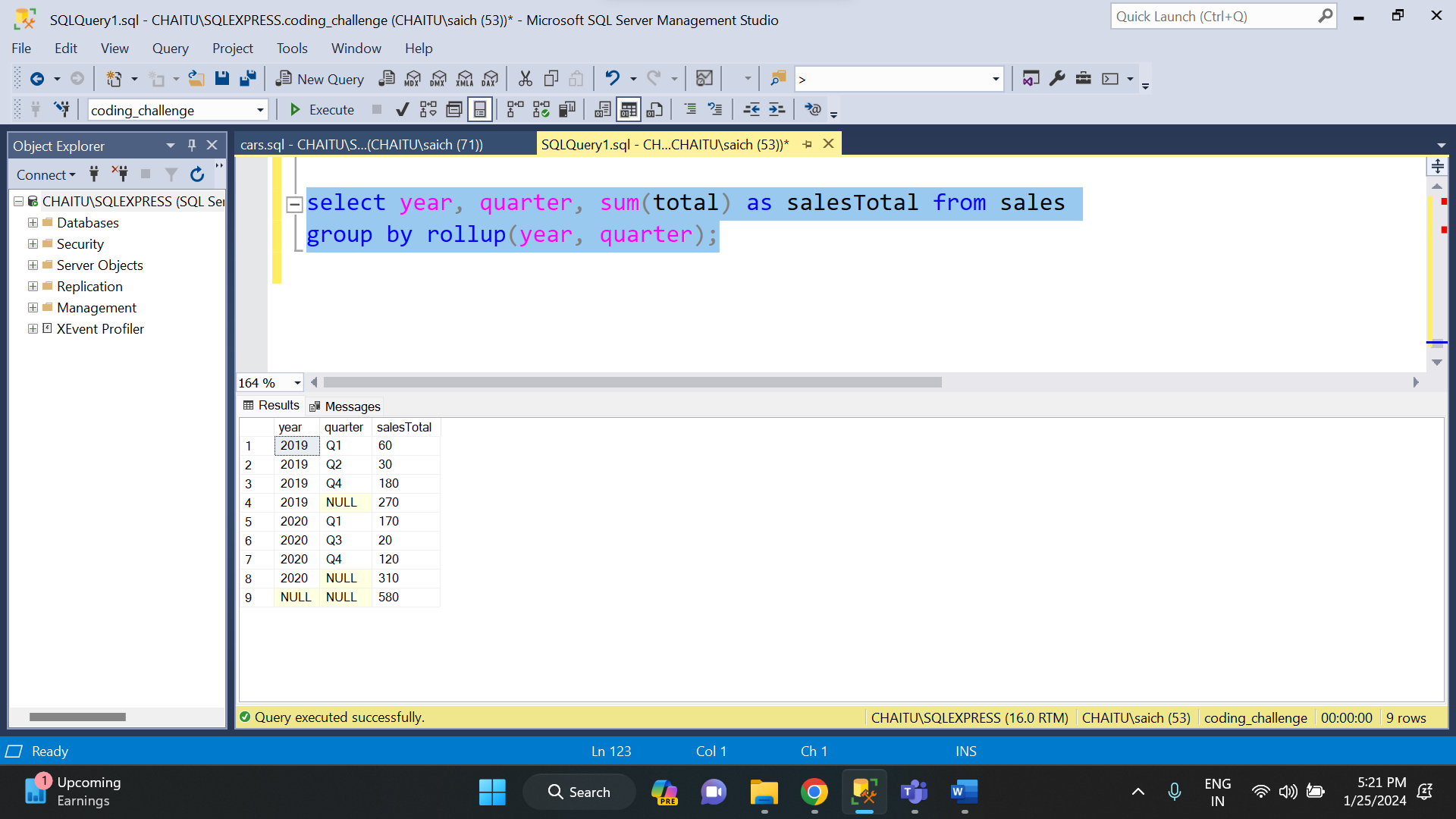
* **CALCULATING SUBTOTALS:**

A subtotal is a number that is formed by summing or adding similar sets of data. It only includes the summation of similar data but it does not include the final total. These subtotals are mainly used in sales and finance fields.

And in order to find the subtotal, we have to use rollup extension of group by clause. This rollup statement allows us to generate hierarchical subtotal rows according to the columns specified and it also adds a grand total row to the final result set.

In order to implement this subtotal, I have created a sales table and have inserted some records into it. Now based on the sales data, I will find the subtotal.

Here the execution on subtotal.

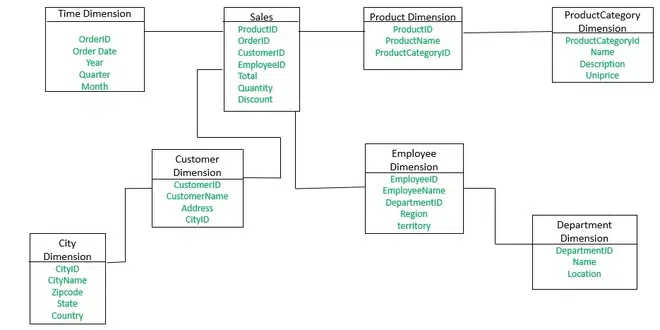


**SNOWFLAKING SCHEMA:**

A snowflake schema is a type of data modeling technique used in data warehousing to represent data in a structured way that is optimized for querying large amounts of data efficiently. In a snowflake schema, the dimension tables are normalized into multiple related tables, creating a hierarchical or “snowflake” structure.

In a snowflake schema, the fact table is still located at the center of the schema, surrounded by the dimension tables. However, each dimension table is further broken down into multiple related tables, creating a hierarchical structure that resembles a snowflake.

Below one is an example of star schema.



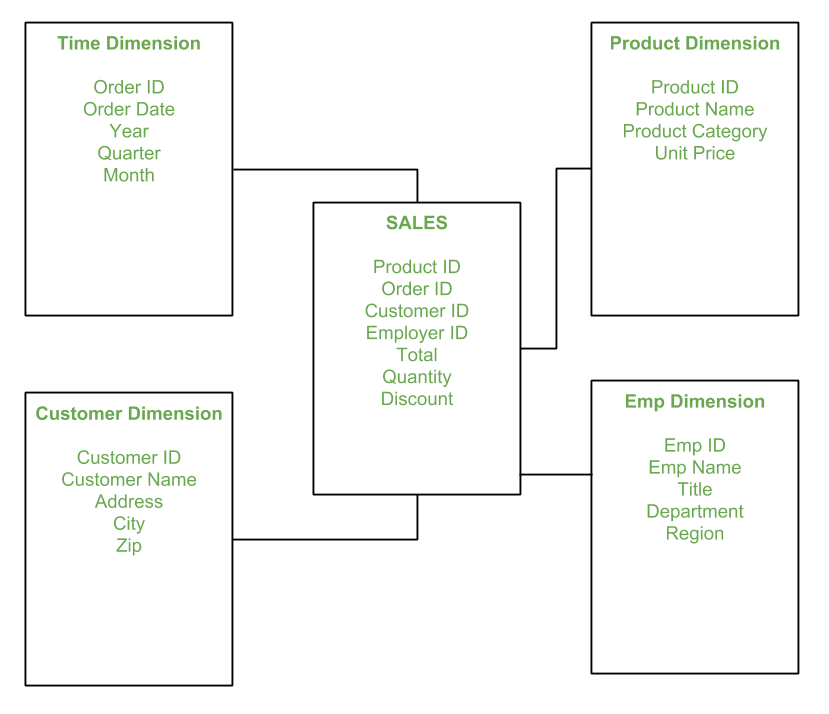
**STAR SCHEMA:**

A star schema is a type of data modeling technique used in data warehousing to represent data in a structured and intuitive way. In a star schema, data is organized into a central fact table that contains the measures of interest, surrounded by dimension tables that describe the attributes of the measures.

The fact table in a star schema contains the measures or metrics that are of interest to the user or organization. For example, in a sales data warehouse, the fact table might contain sales revenue, units sold, and profit margins. Each record in the fact table represents a specific event or transaction, such as a sale or order.

The dimension tables in a star schema contain the descriptive attributes of the measures in the fact table. These attributes are used to slice and dice the data in the fact table, allowing users to analyze the data from different perspectives. For example, in a sales data warehouse, the dimension tables might include product, customer, time, and location.

Below is an example of star schema.



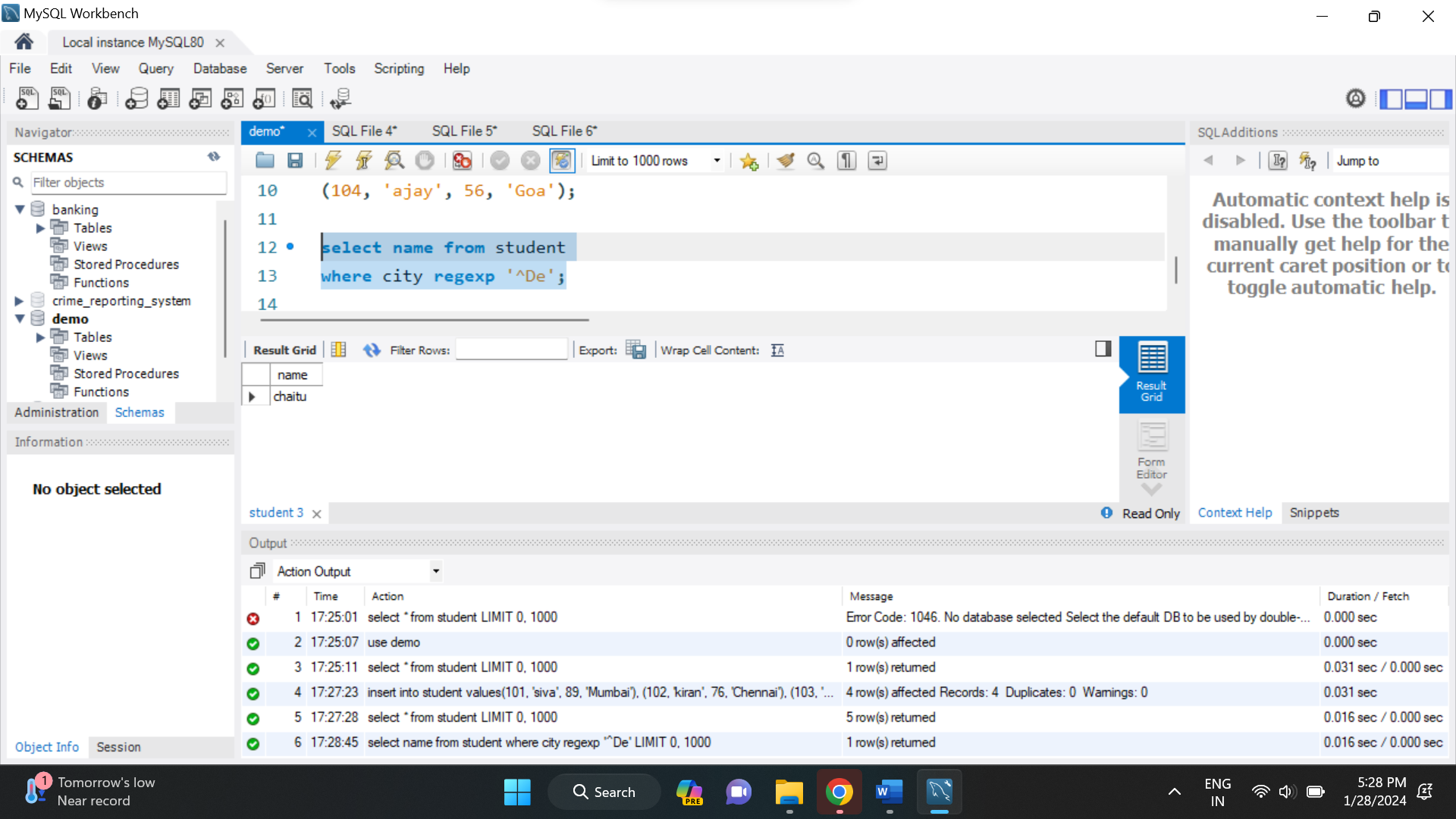
**REGULAR FUNCTIONS IN SQL:**

MySQL supports another type of pattern matching operation based on the regular expressions and the REGEXP operator.

It provides a powerful and flexible pattern match that can help us implement power search utilities for our database systems. REGEXP is the operator used when performing regular expression pattern matches. RLIKE is the synonym.

It also supports a number of metacharacters which allow more flexibility and control when performing pattern matching. The backslash is used as an escape character. It’s only considered in the pattern match if double backslashes have used. These are not case sensitive.

Examples for regular functions.



**VIEWS IN SQL:**

In SQL, a view is a virtual table based on the result-set of an SQL statement. It is a stored SQL query that you can treat like a table. Views are useful for simplifying complex queries, abstracting data access, and providing a layer of security by restricting access to certain columns or rows.

