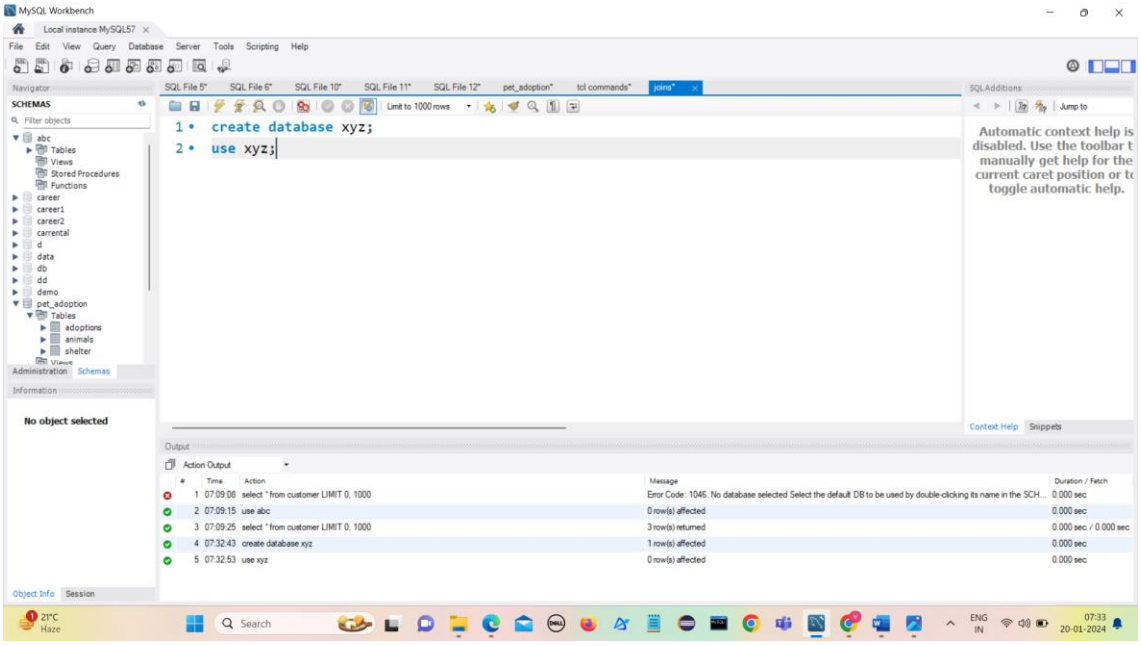
Assignment-4

Joins

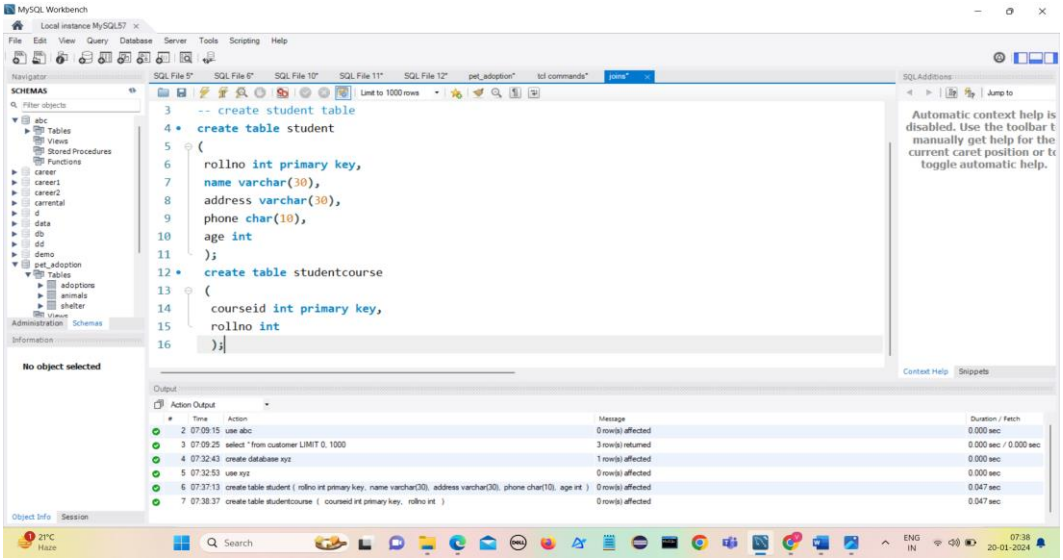
Join statement is used to combine data or rows from two or more tables based on a common field between them.

1) INNER JOIN

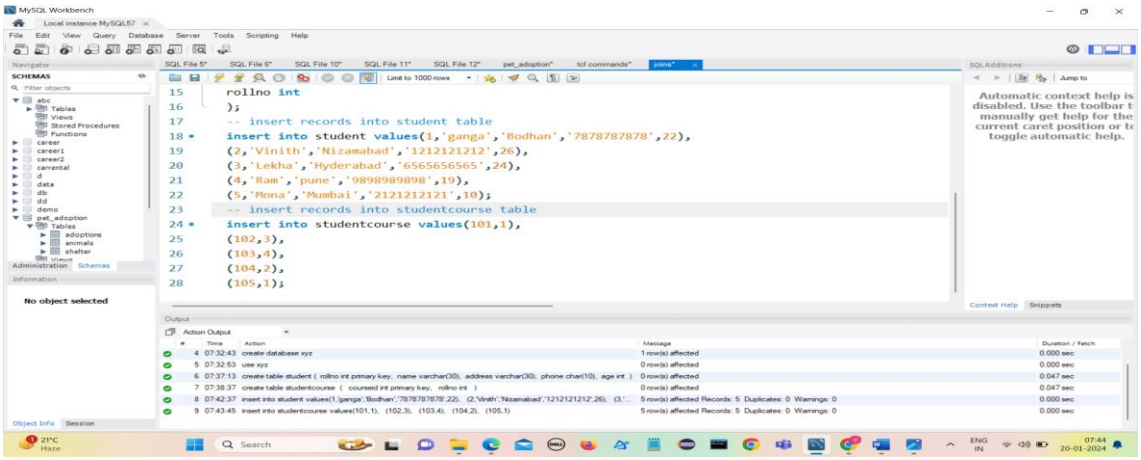
Create a table xyz and use the database to make it as the current working database.



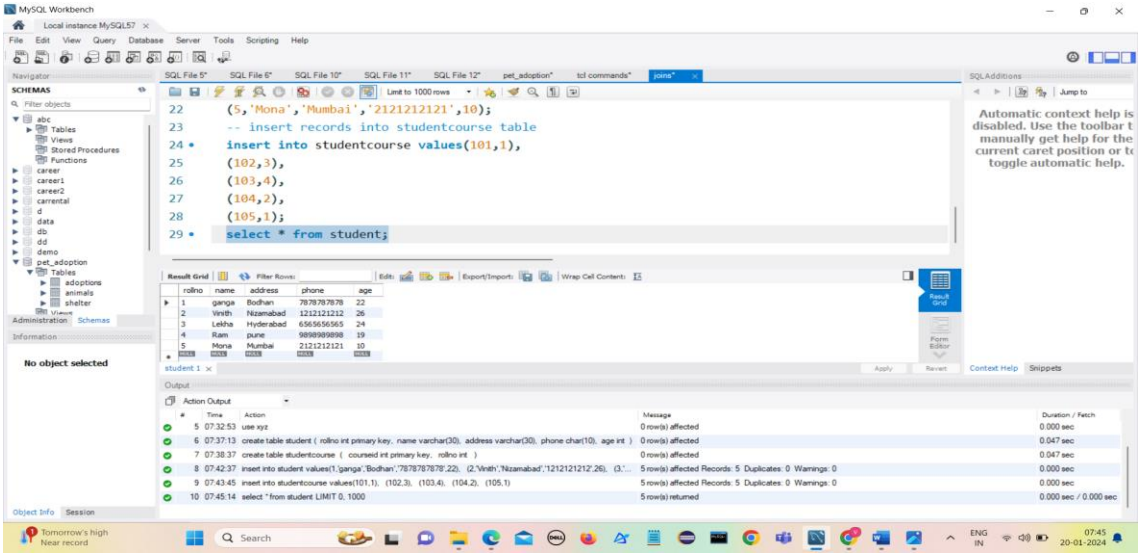
Create two tables Student and Studentcourse using create table statement.

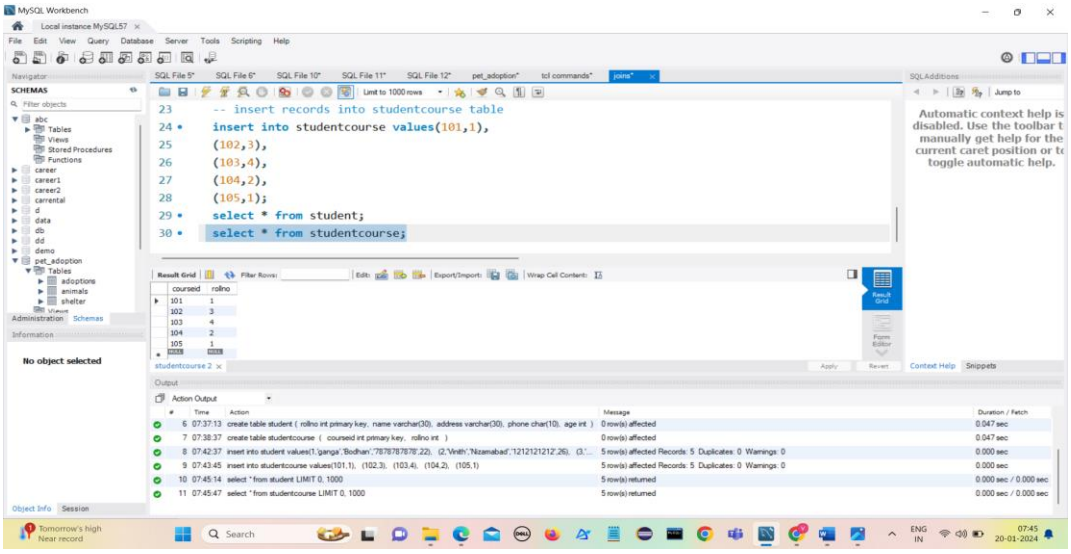


Insert records into Student and Studentcourse table using insert statement.



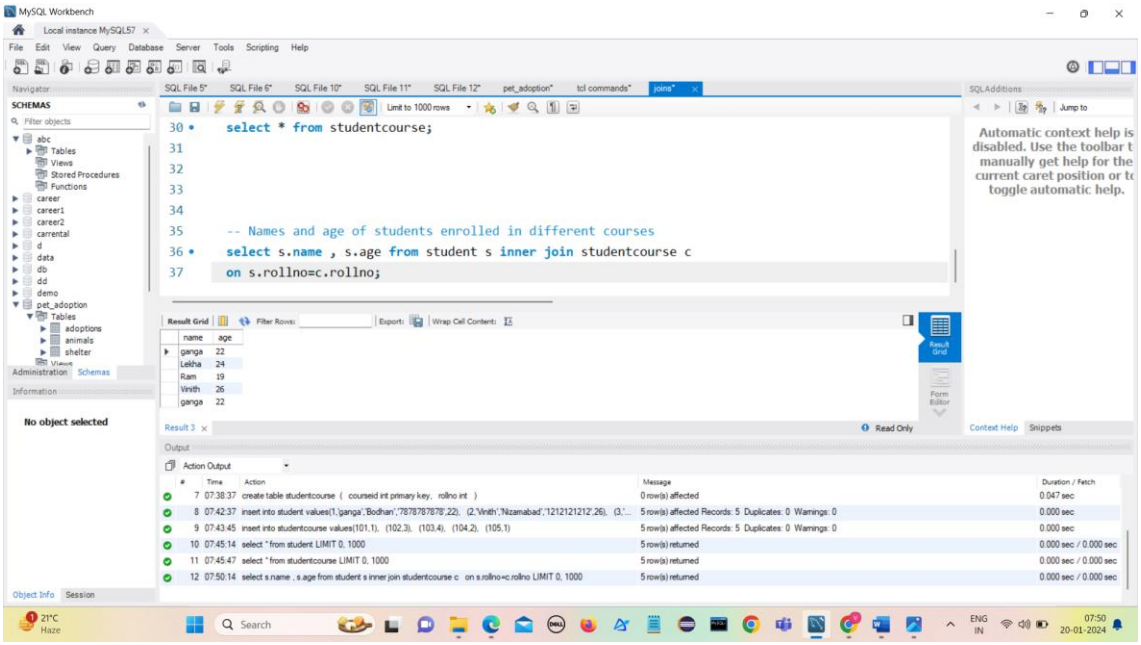
Retrieve records from student and studentcourse table





Inner join

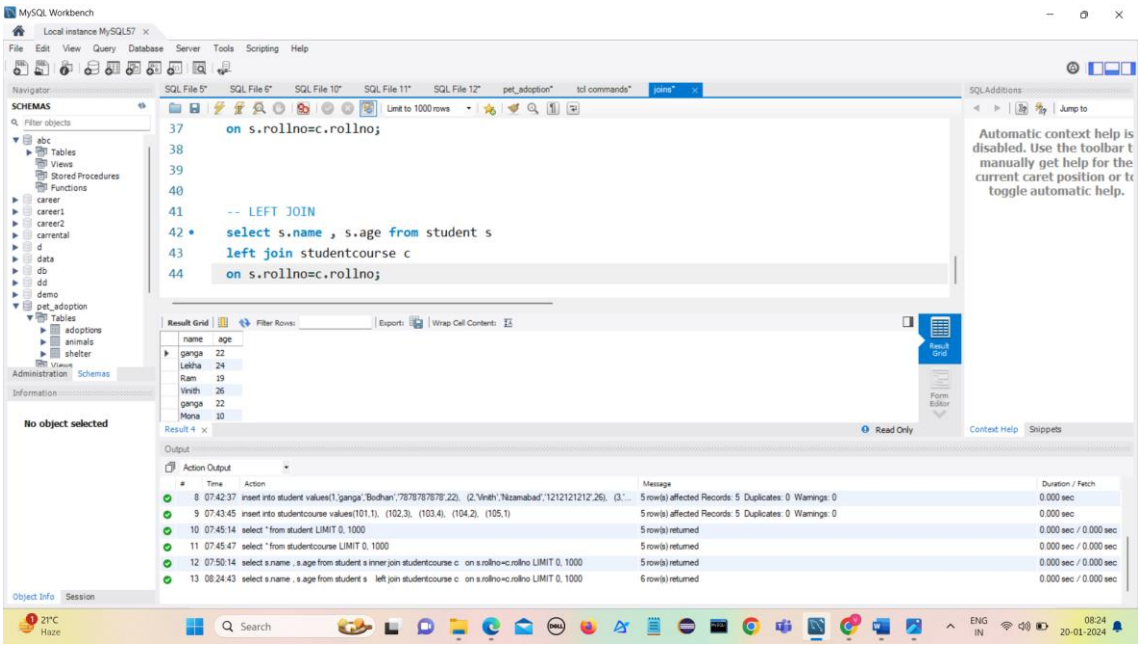
The INNER JOIN keyword selects all rows from both the tables as long as the condition is satisfied. This keyword will create the result-set by combining all rows from both the tables where the condition satisfies i.e value of the common field will be the same.



This query gives the names and age of students enrolled in different courses.

2) **LEFT JOIN**

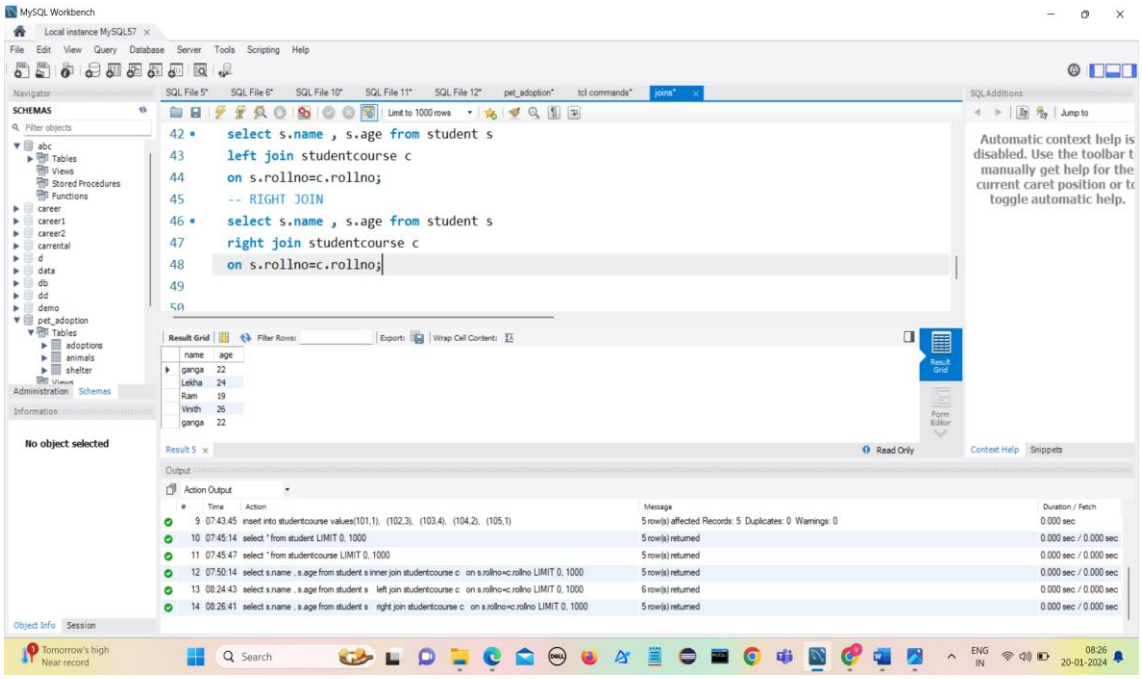
This join returns all the rows of the table on the left side of the join and matches rows for the table on the right side of the join. For the rows for which there is no matching row on the right side, the result-set will contain *null*.



3) RIGHT JOIN

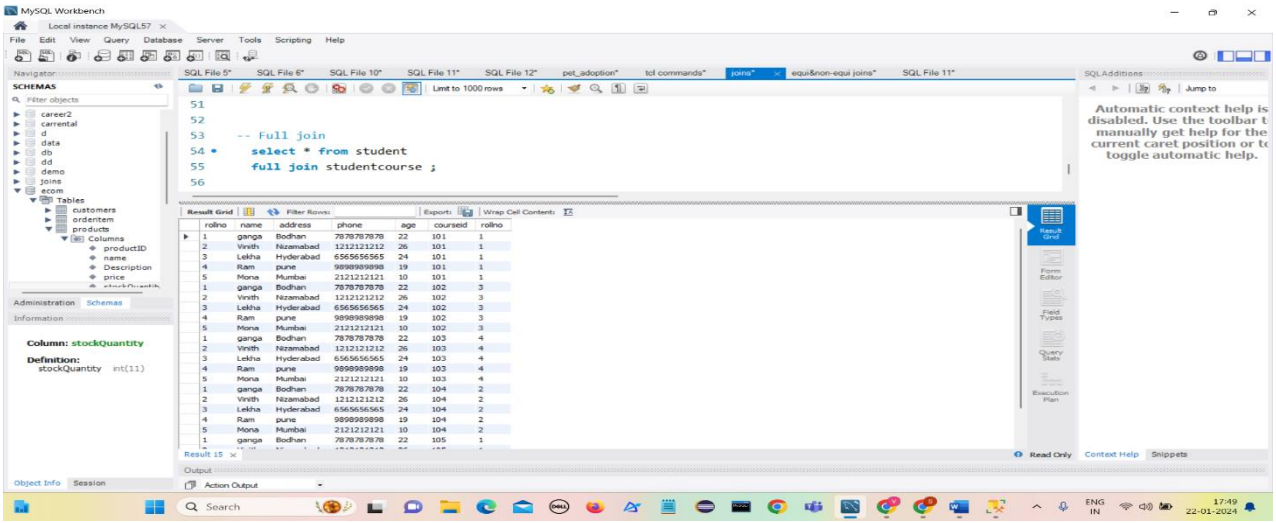
RIGHT JOIN is similar to LEFT JOIN. This join returns all the rows of the table on the right side of the join and matching rows for the table on the left side of the join.

For the rows for which there is no matching row on the left side, the result-set will contain *null*



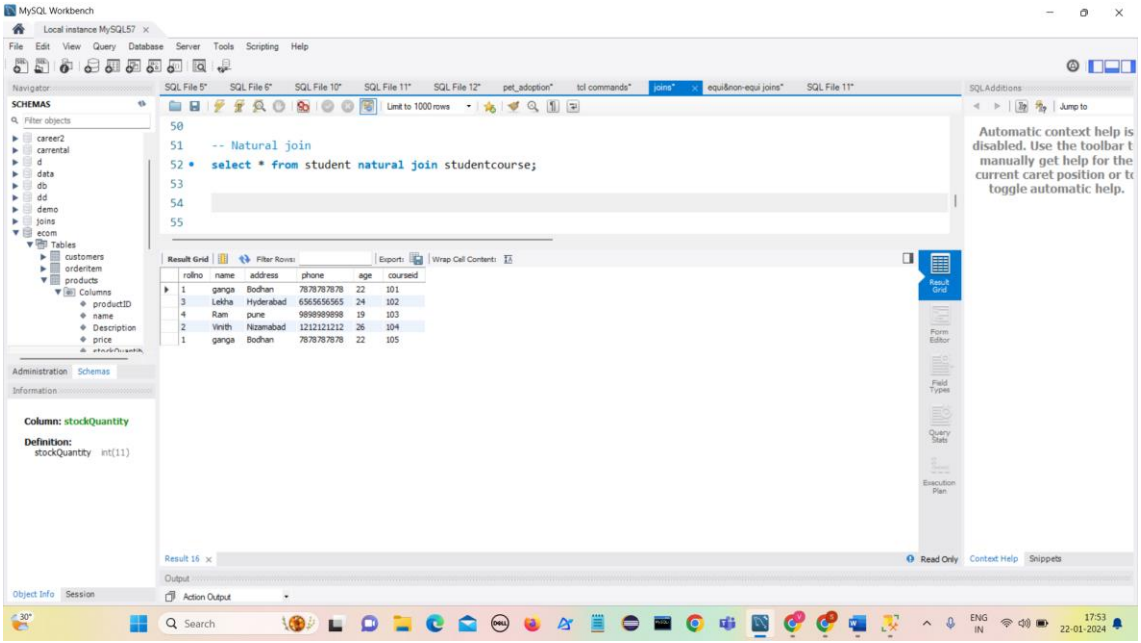
4) FULL JOIN

FULL JOIN creates the result-set by combining results of both LEFT JOIN and RIGHT JOIN. The result-set will contain all the rows from both tables. For the rows for which there is no matching, the result-set will contain *NULL* values.



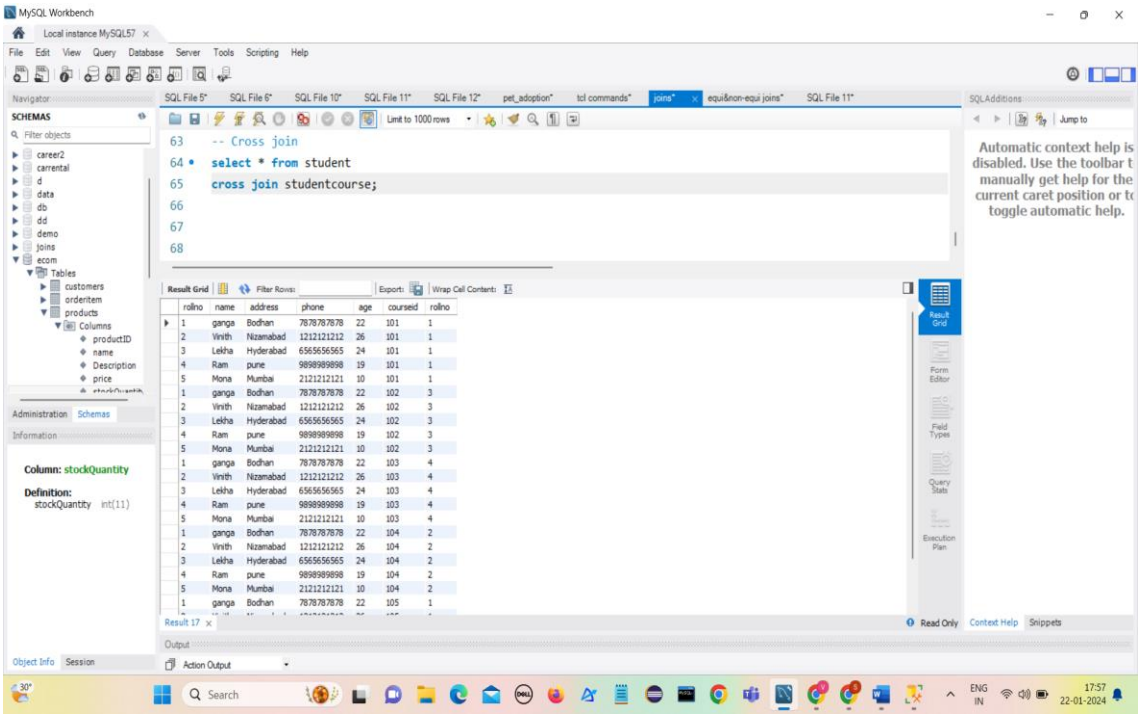
5) NATURAL JOIN

A natural join returns all rows by matching values in common columns having same name and data type of columns and that column should be present in both tables.



6) CROSS JOIN

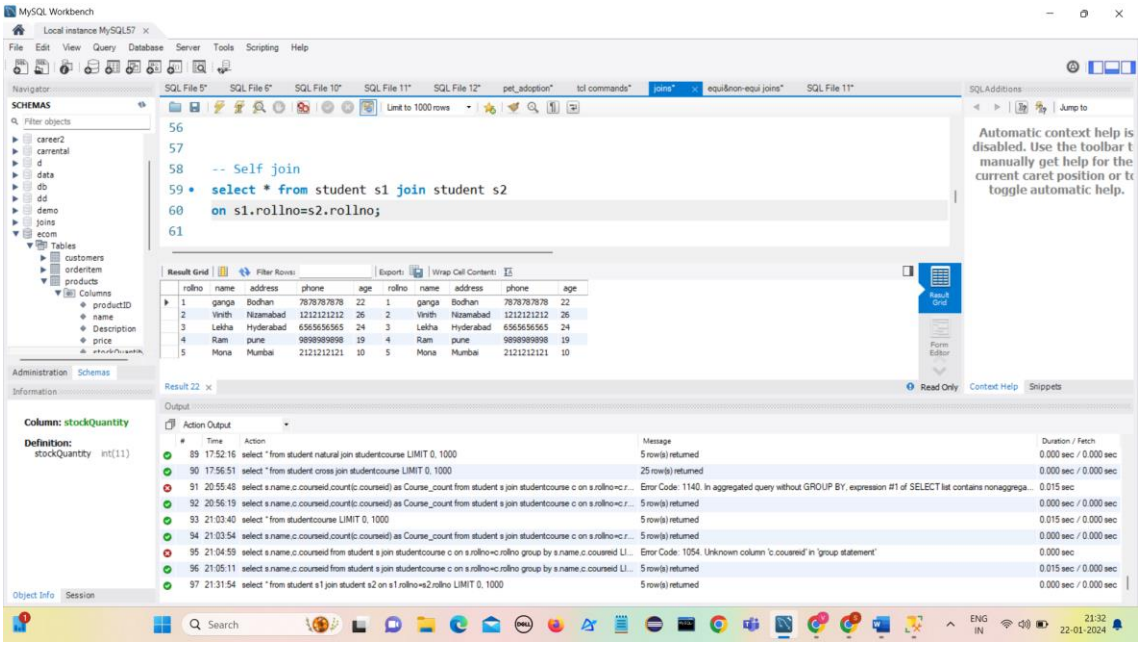
A *cross join* is a type of join that returns the Cartesian product of rows from the tables in the join. It combines each row from the first table with each row from the second table.



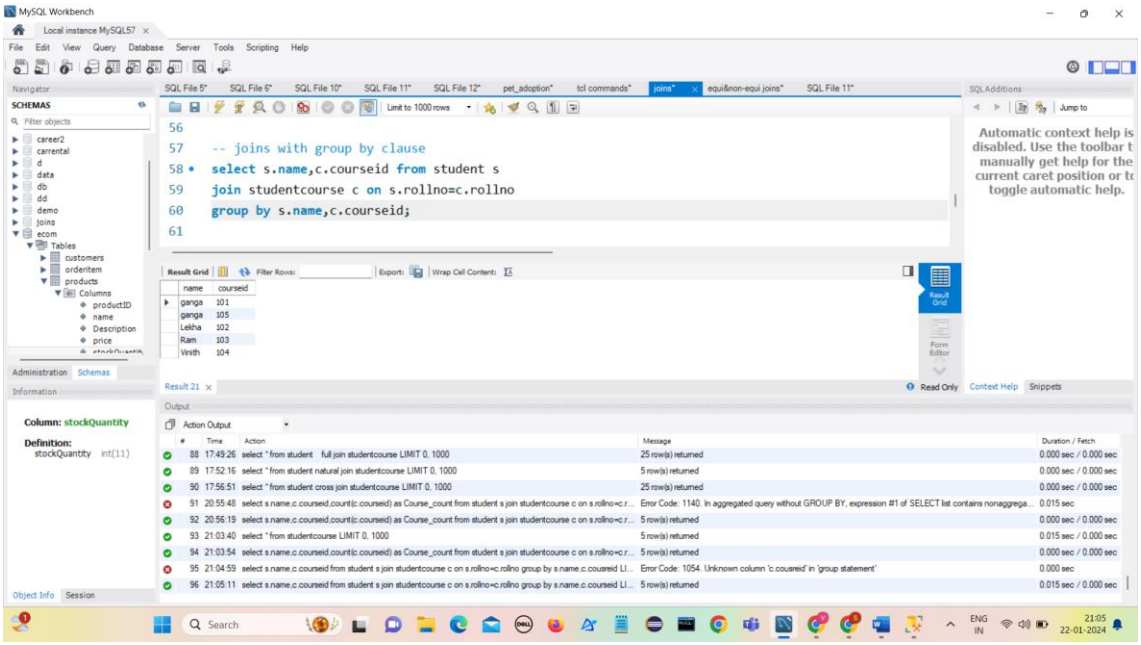
7) SELF JOIN

A self join is a regular join that is used to join a table with itself.

It basically allows us to combine the rows from the same table based on some specific conditions.



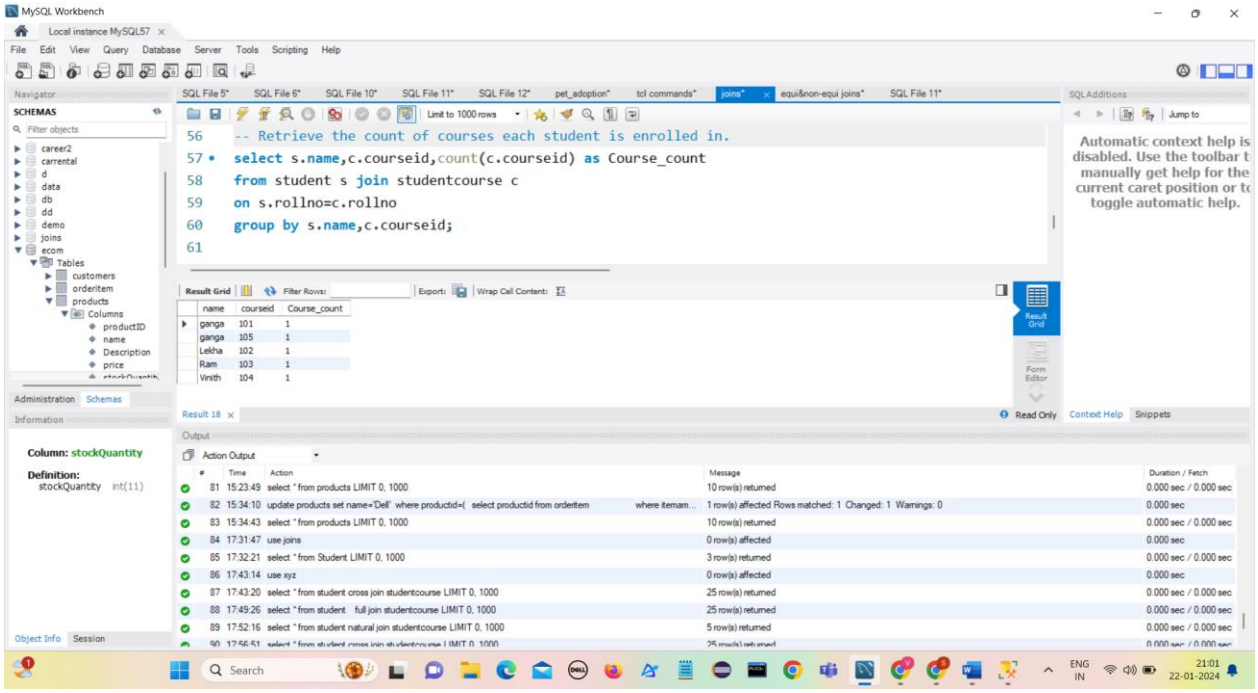
8) Joins with group by clause



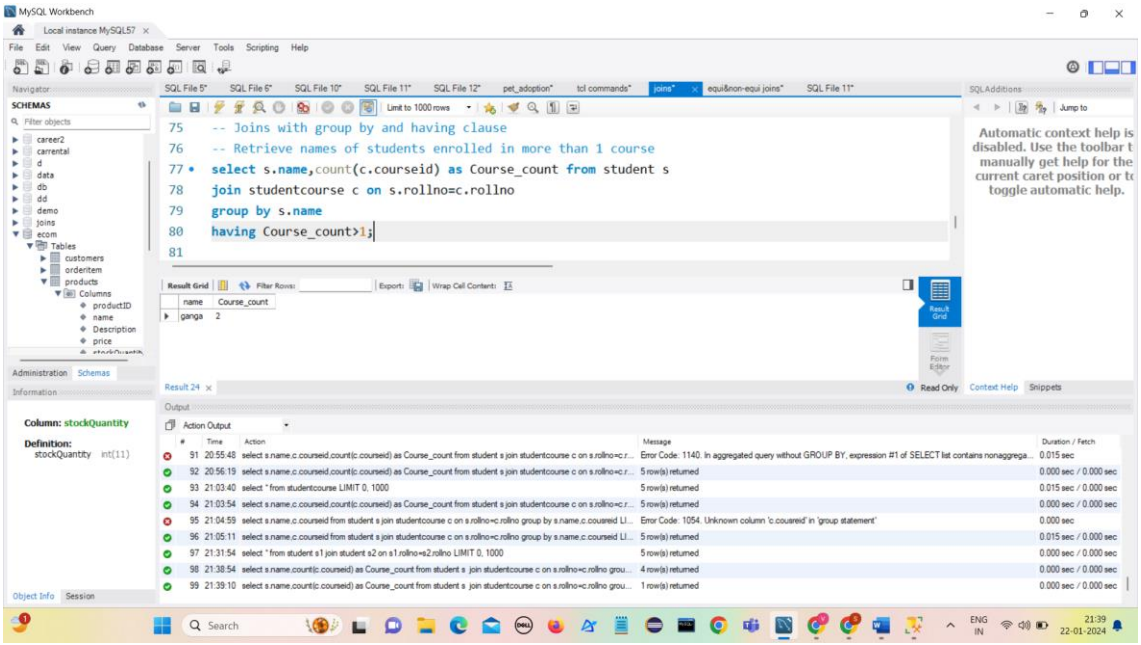
9) Join with aggregate functions

Aggregate functions like SUM, AVG , COUNT, MIN,MAX can be used with joins.

Retrieve the count of courses each student is enrolled in



10) Joins with group by and having clause



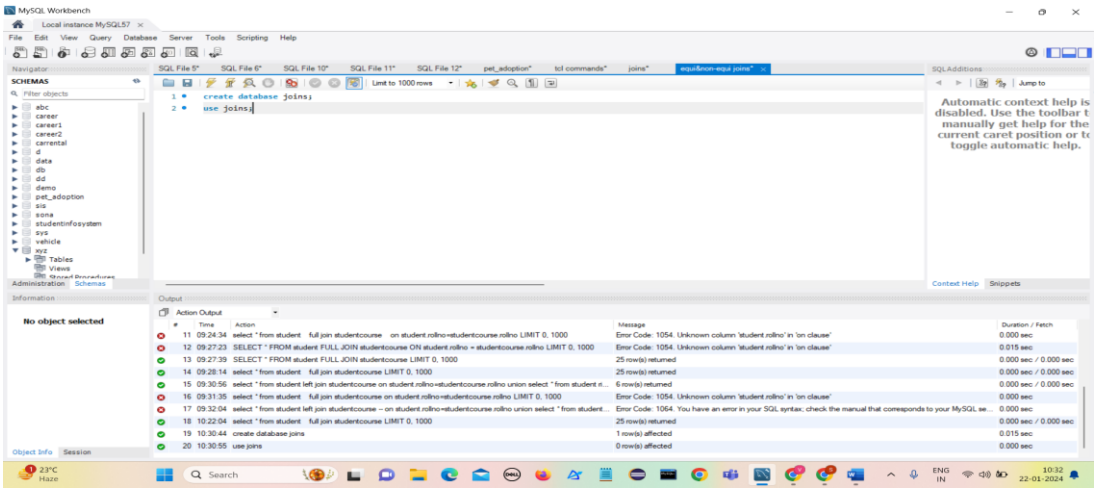
Equi joins & Non-Equi Joins

Create database:

create database joins;

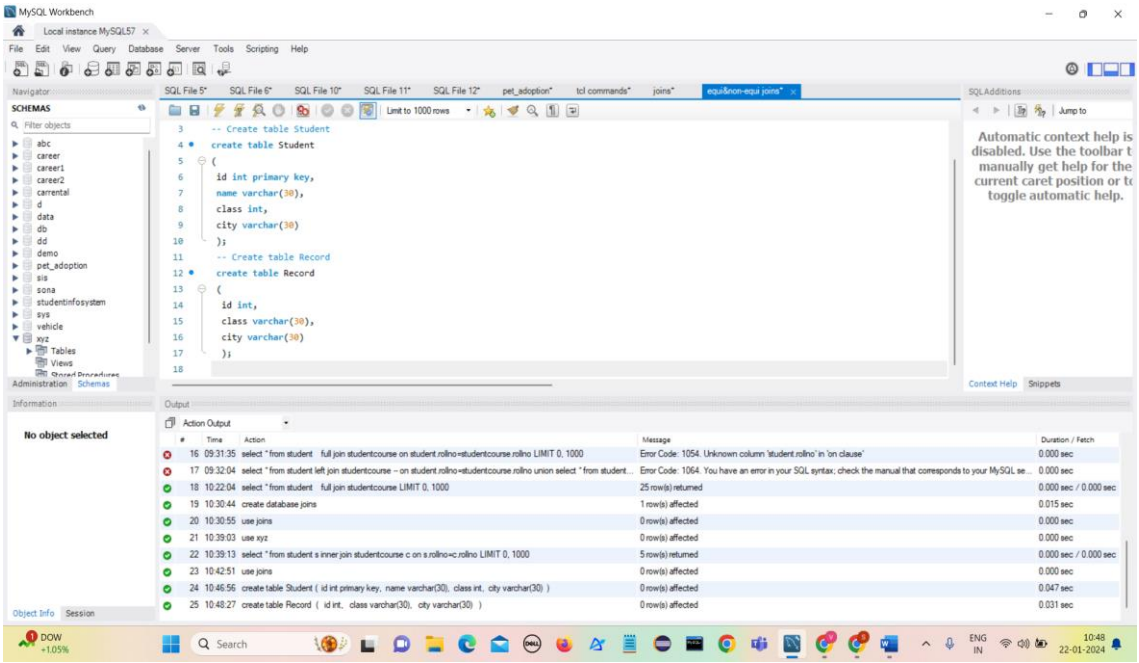
Use database:

use joins; make joins as the current working database

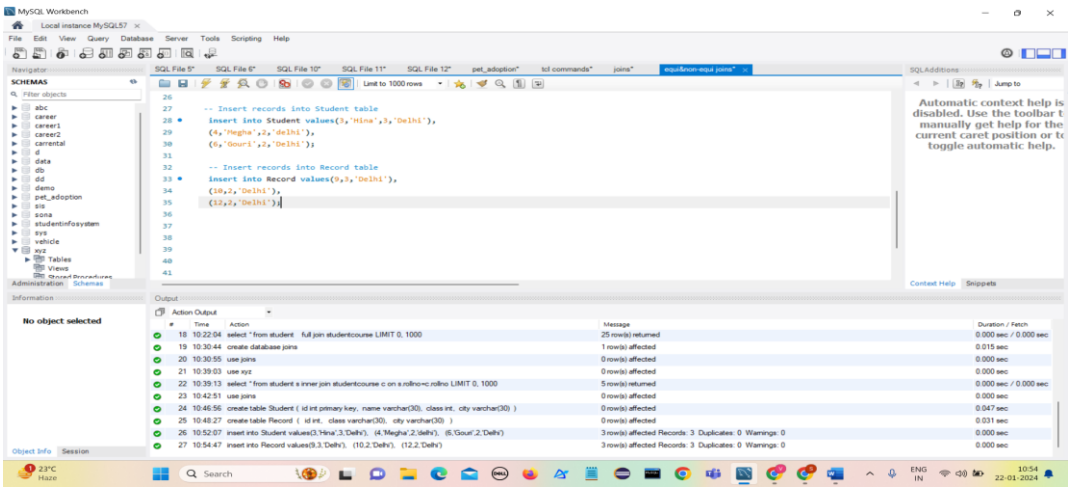


Create table Student

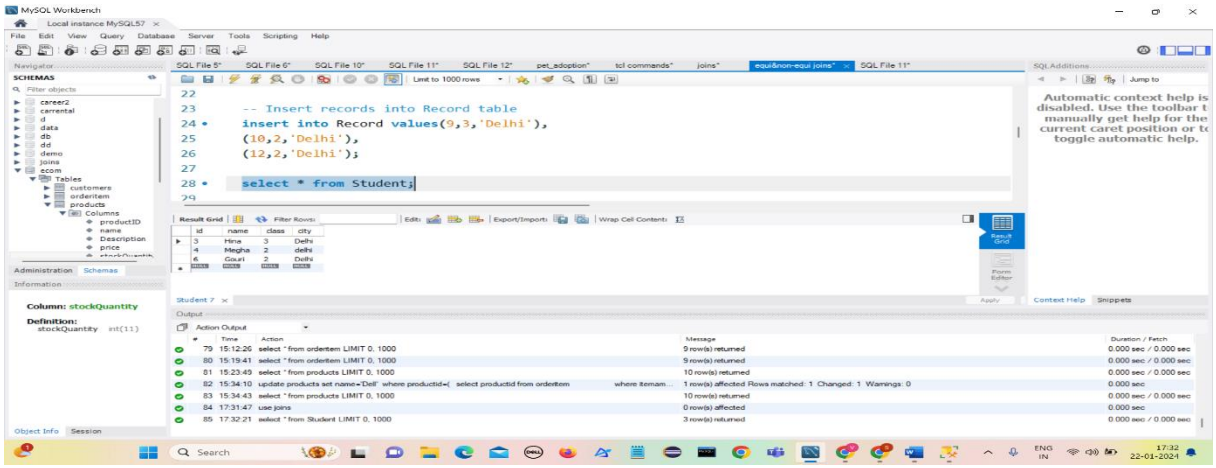
Create table Record



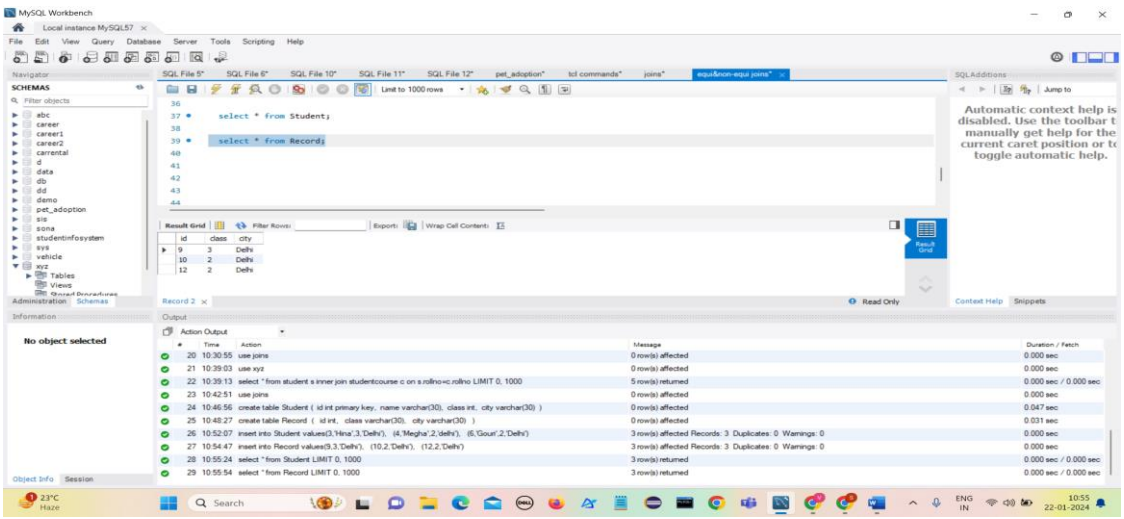
Insert records into Student and Record Table



Retrieve records from Student table using select statement

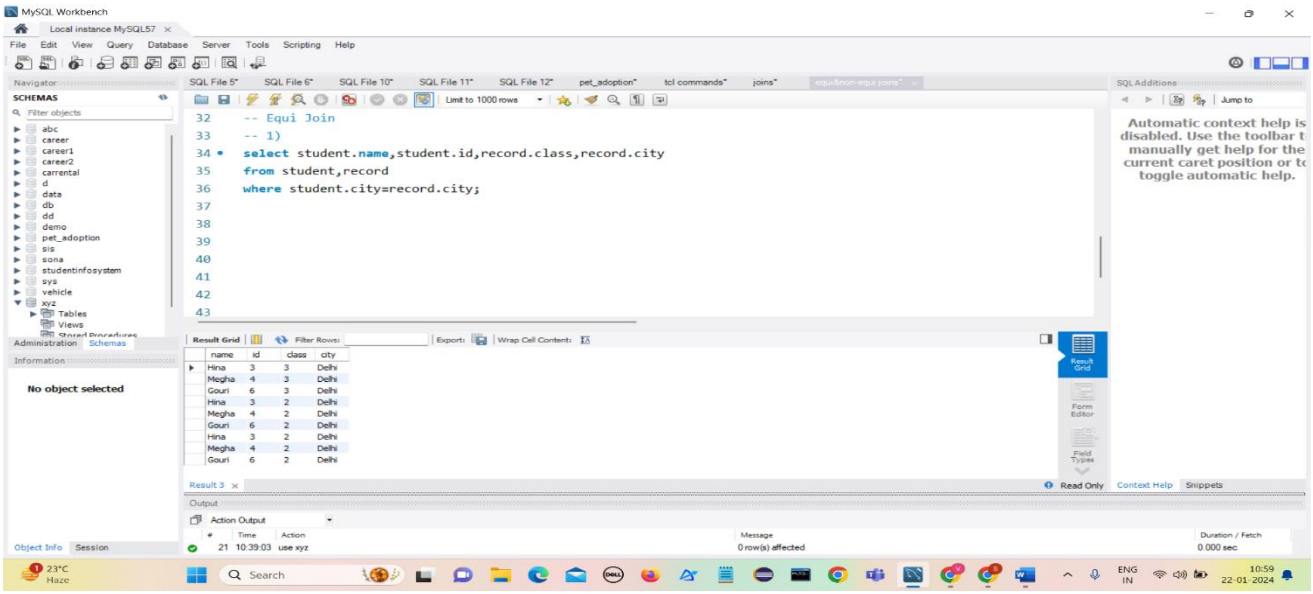


Retrieve records from Record table using select statement

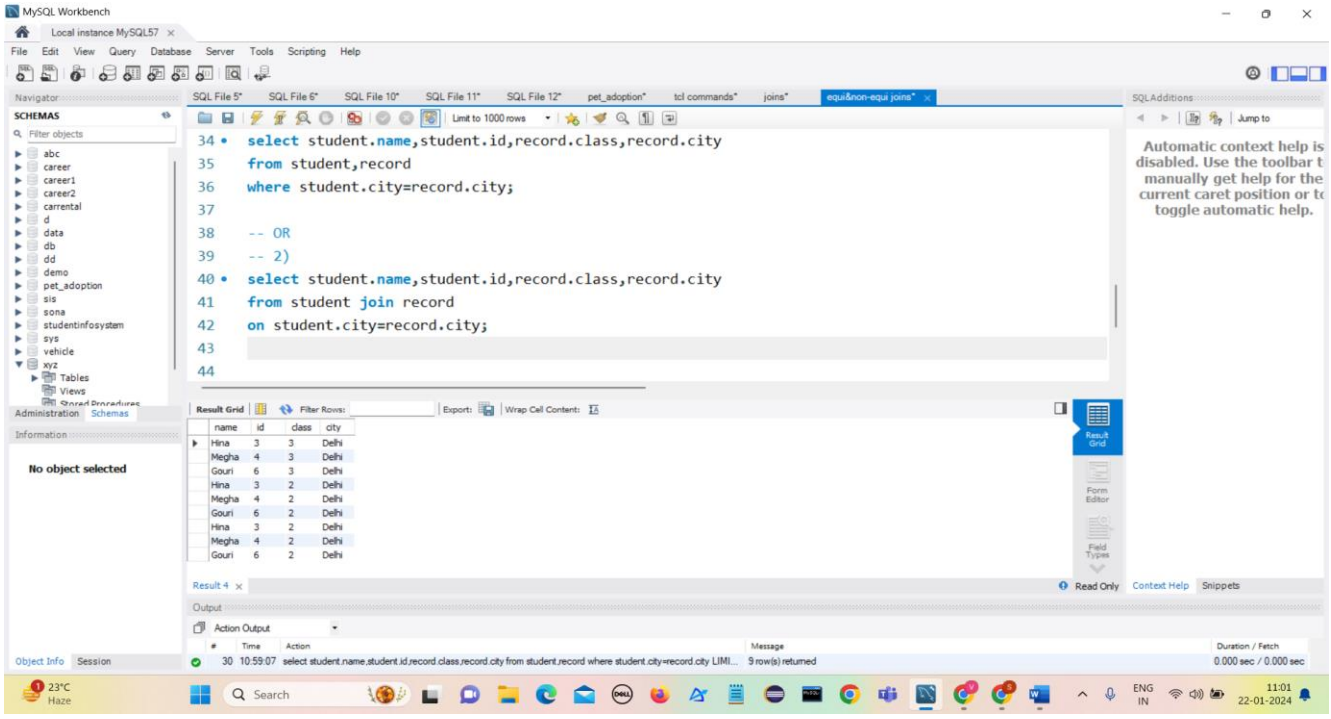


11) Equi Join:

EQUI JOIN creates a JOIN for equality or matching column(s) values of the relative tables. EQUI JOIN also create JOIN by using JOIN with ON and then providing the names of the columns with their relative tables to check equality using equal sign (=).

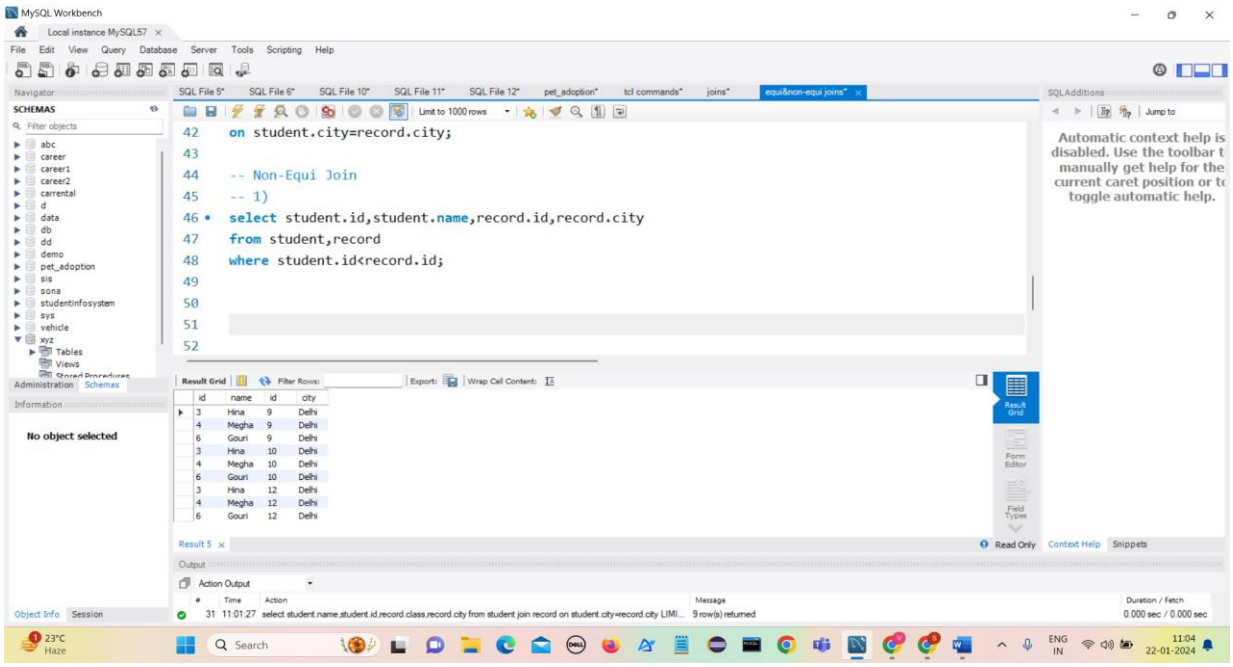


OR

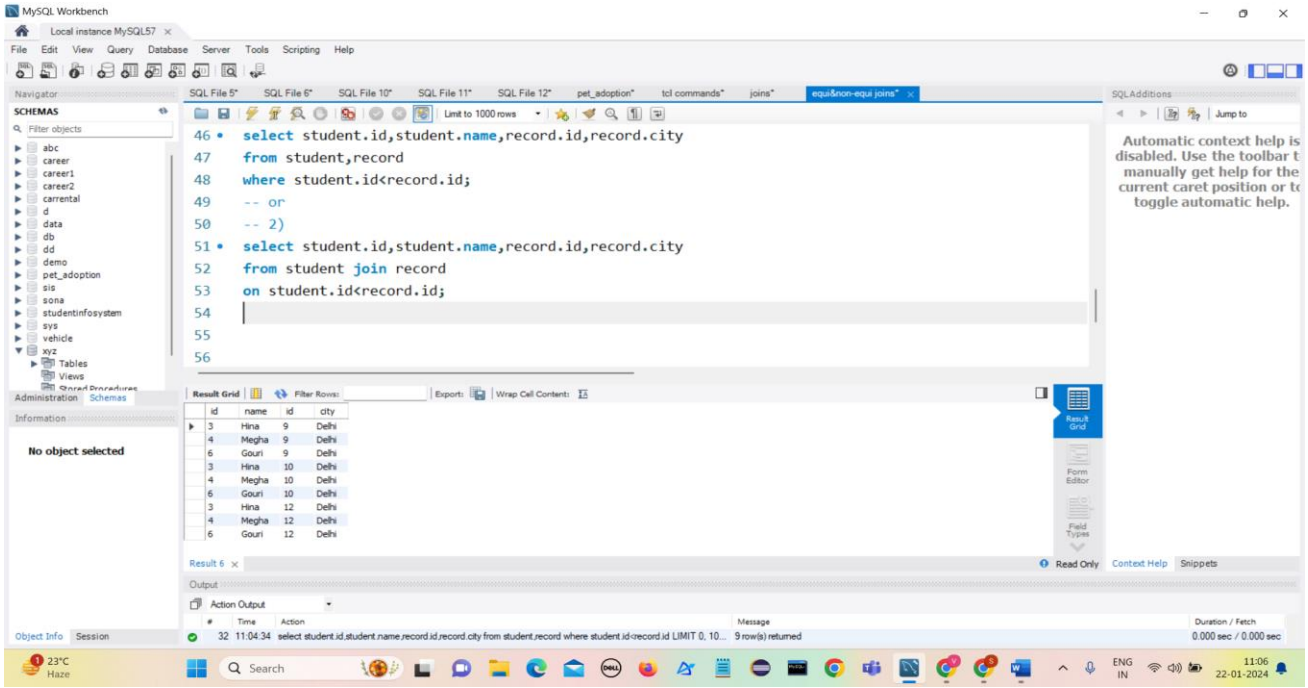


12) Non- Equi join

NON EQUI JOIN performs a JOIN using comparison operator other than equal(=) sign like >, <, >=, <= with conditions.

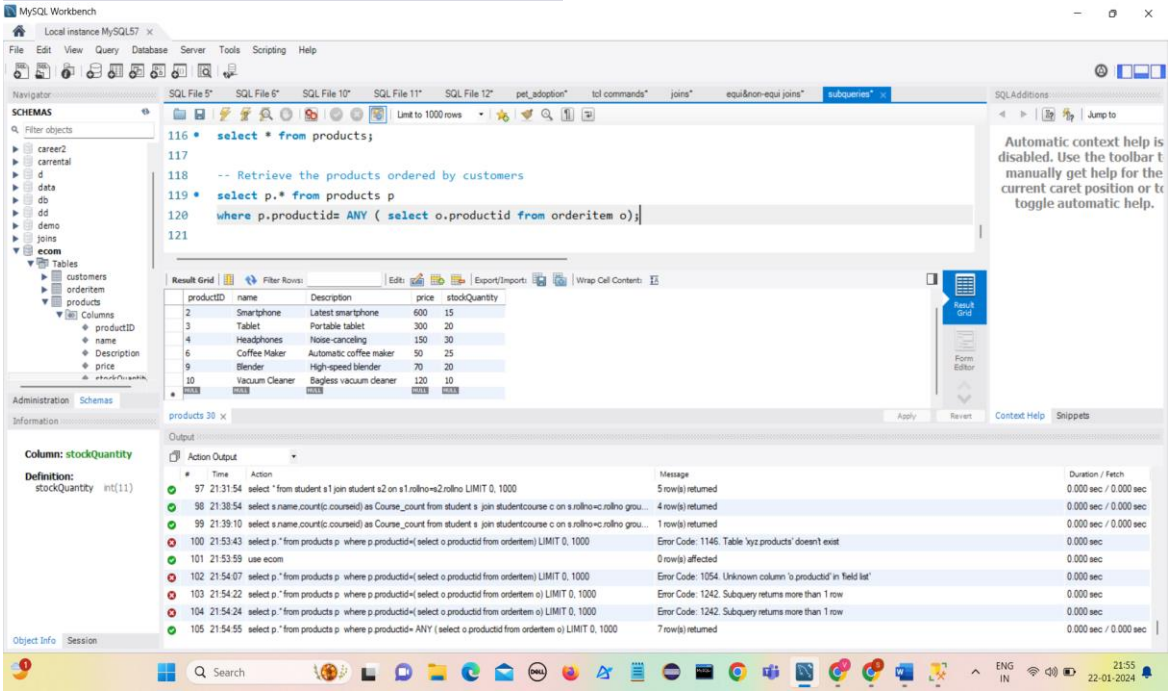


Or

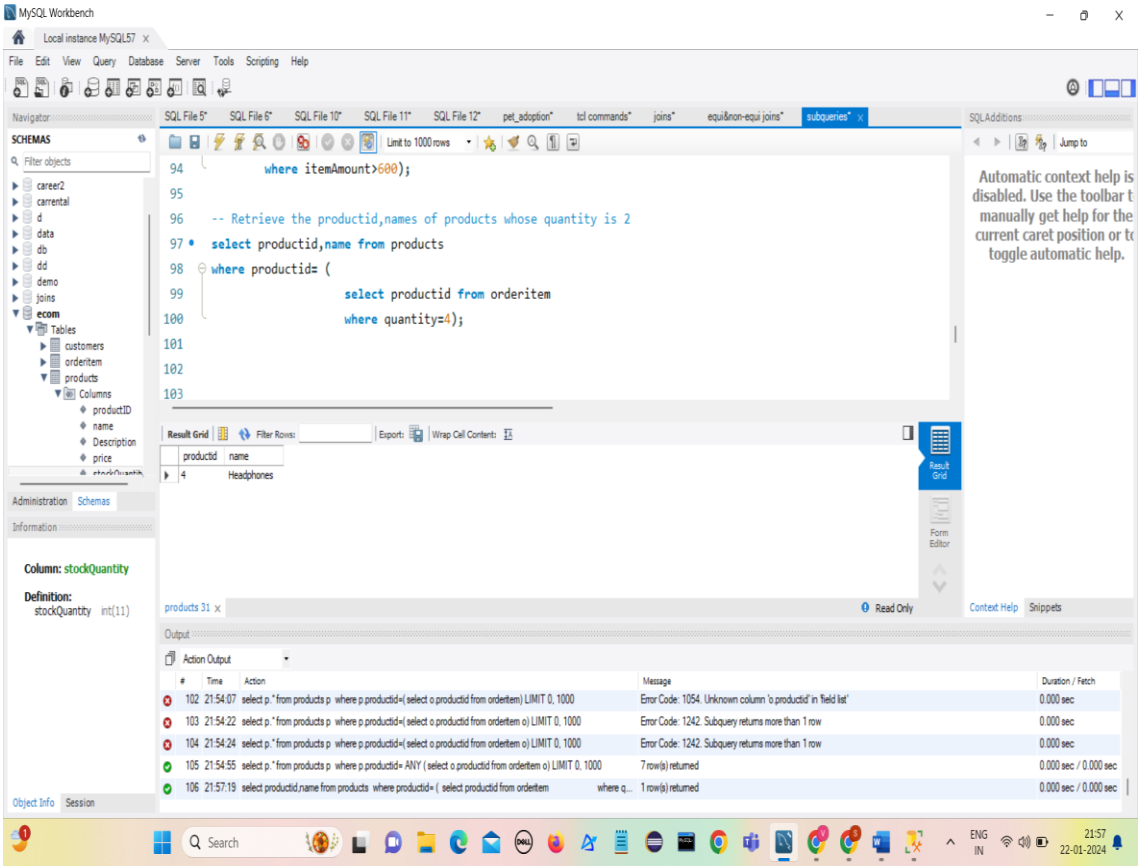


13) Querying data using Subqueries

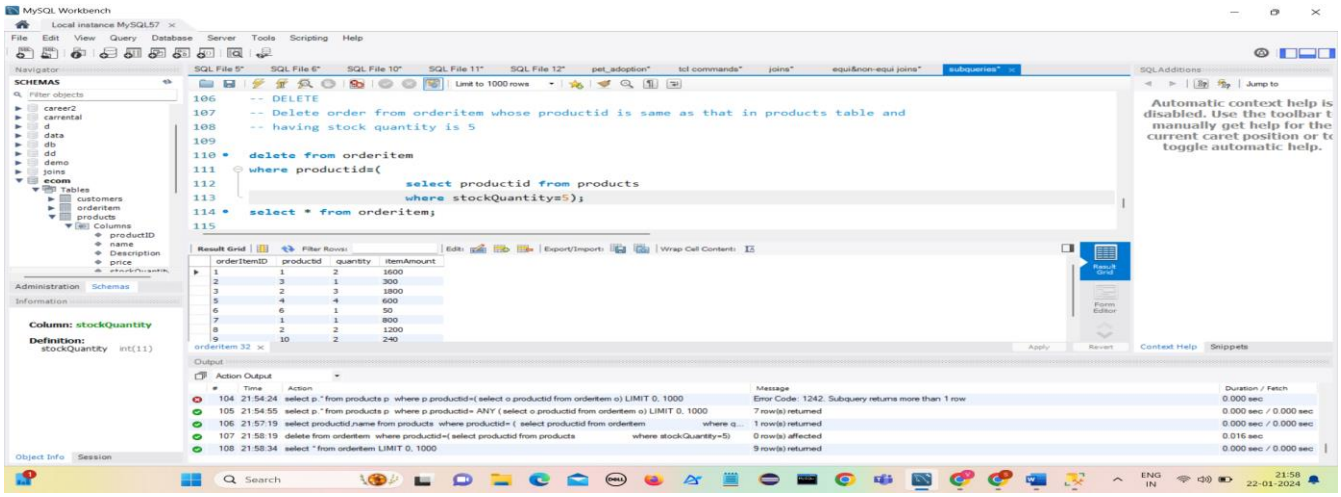
Retrieve the products ordered by customers



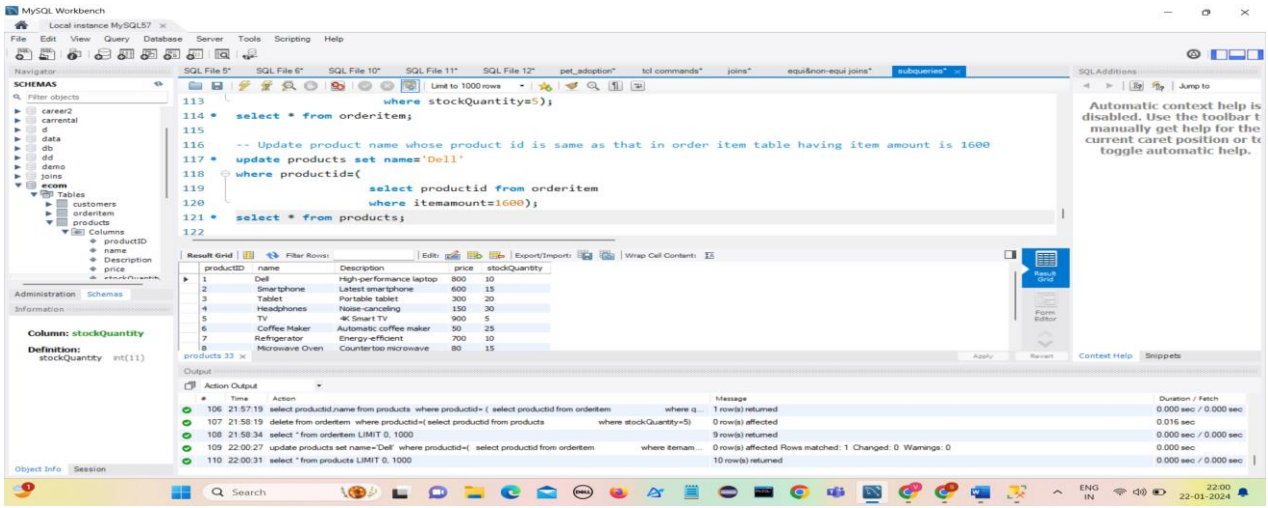
Retrieve the productid,names of products whose quantity is 2



Delete order from orderitem whose productid is same as that in products table and having stock quantity is 5

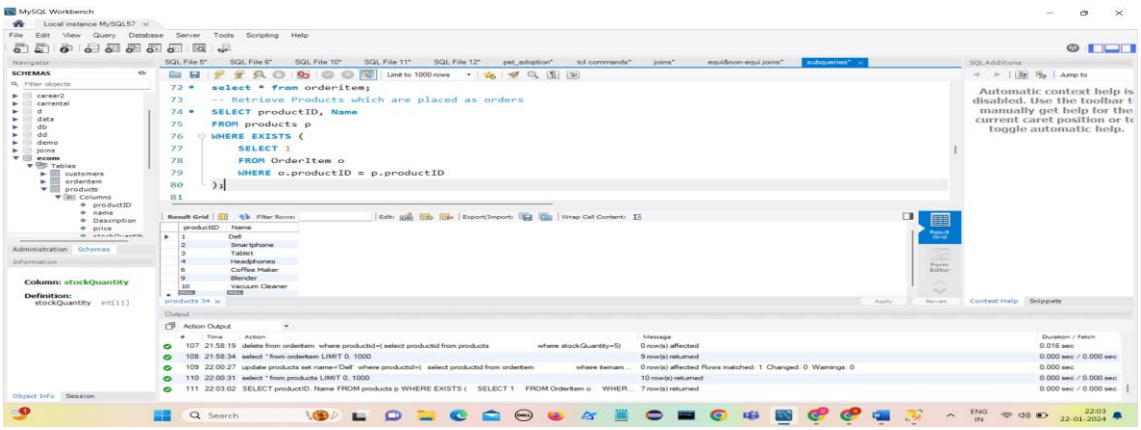


Update product name whose product id is same as that in order item table having item amount is 1600.



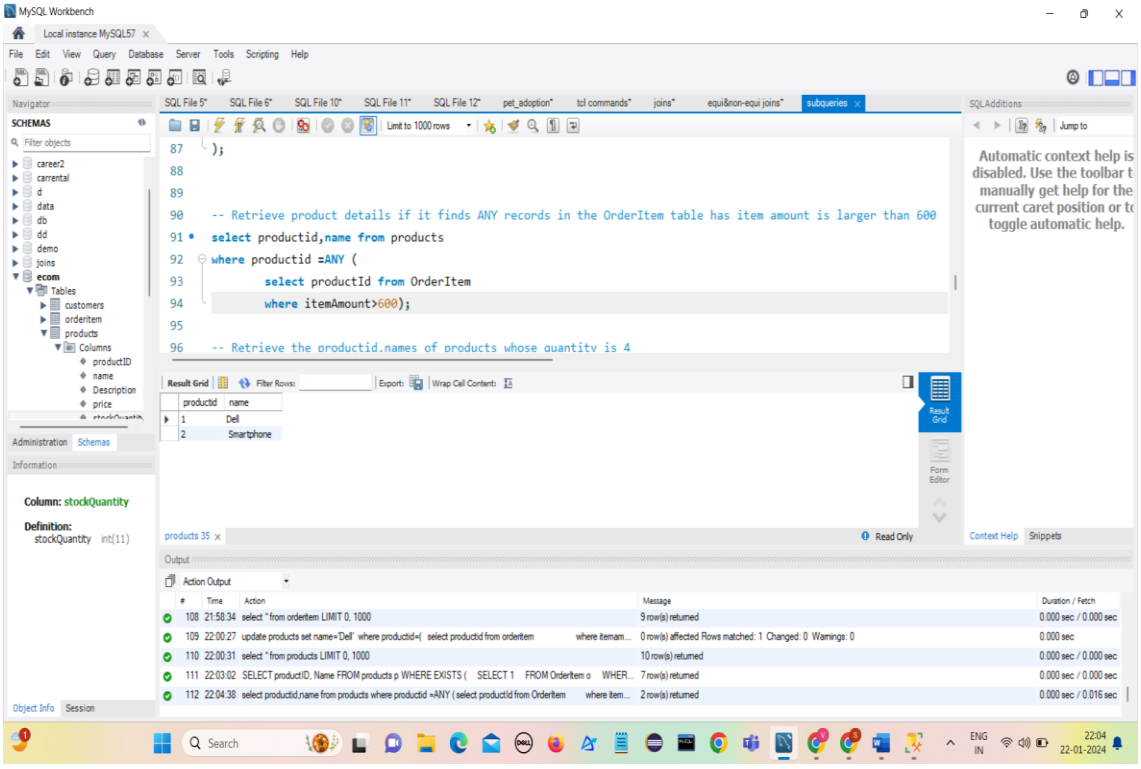
14) EXISTS operator

Retrieve Products which are placed as orders



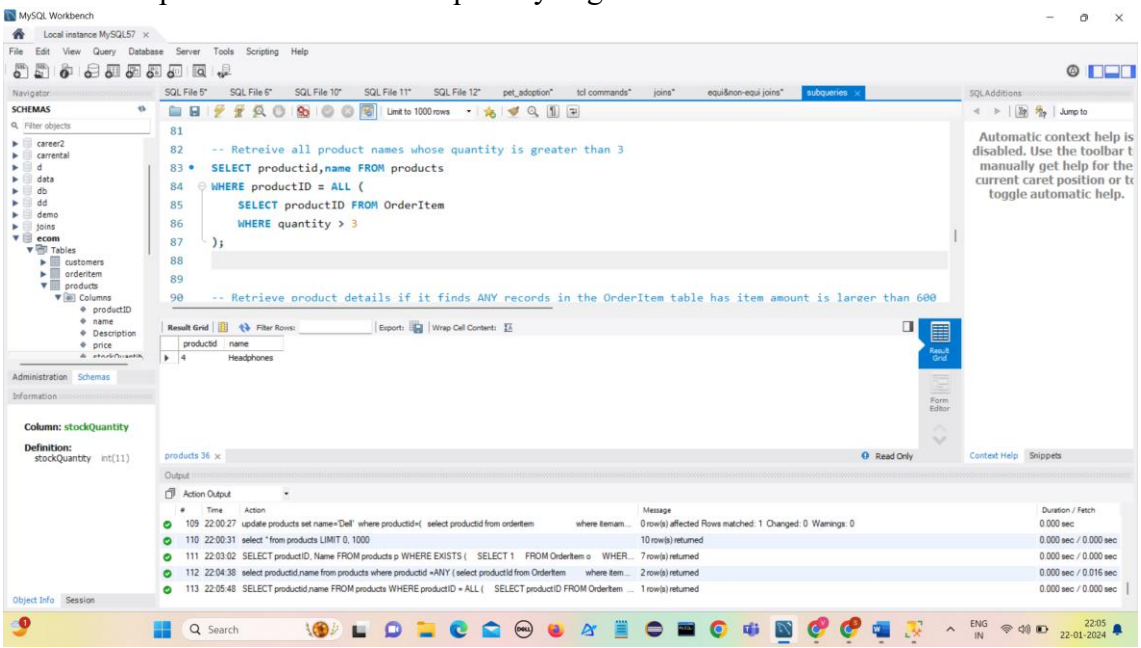
15) ANY operator

Retrieve product details if it finds ANY records in the OrderItem table has item amount is larger than 600.



16) ALL operator

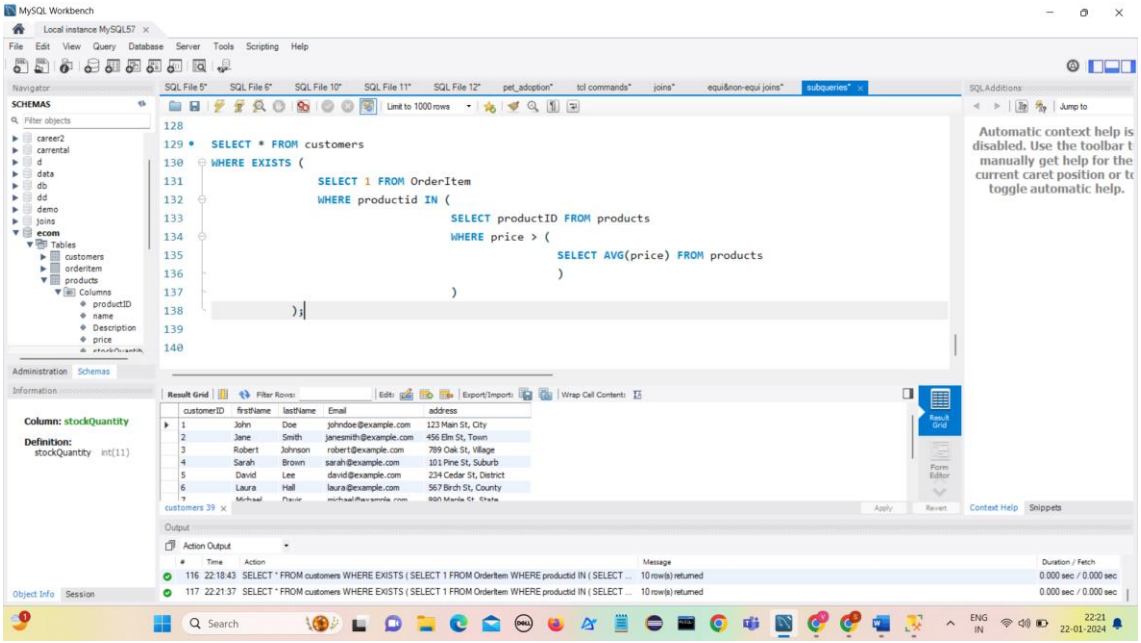
Retreive all product names whose quantity is greater than 3



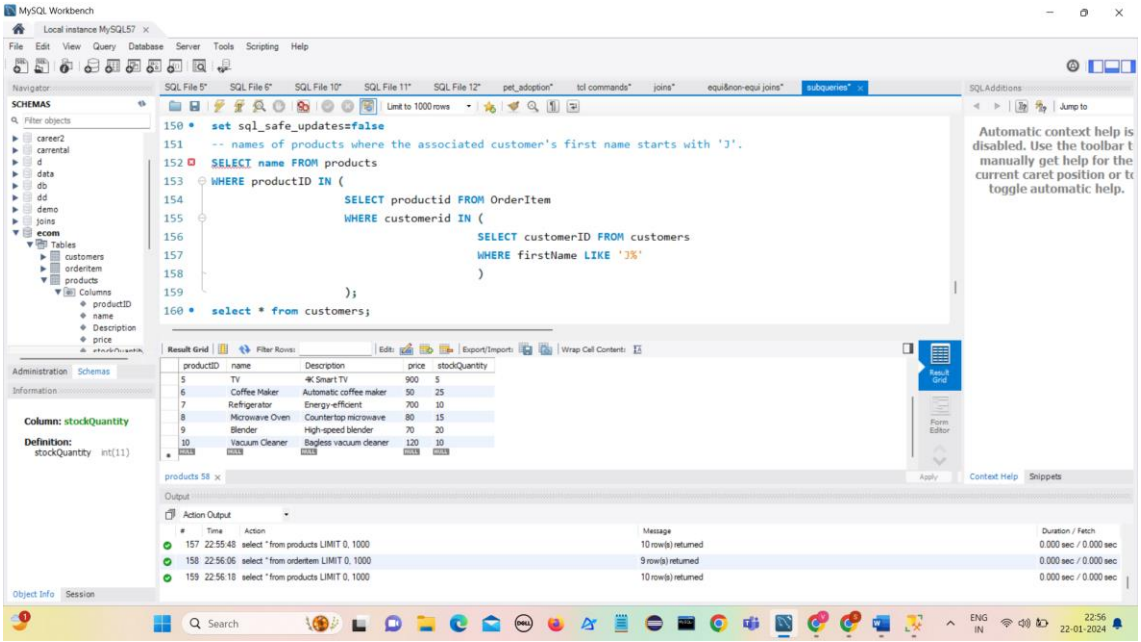
17) NESTED SUB QUERIES

A nested subquery is a subquery that is used inside another query. The inner query runs first, and the result of the inner query is used as the input for the outer query.

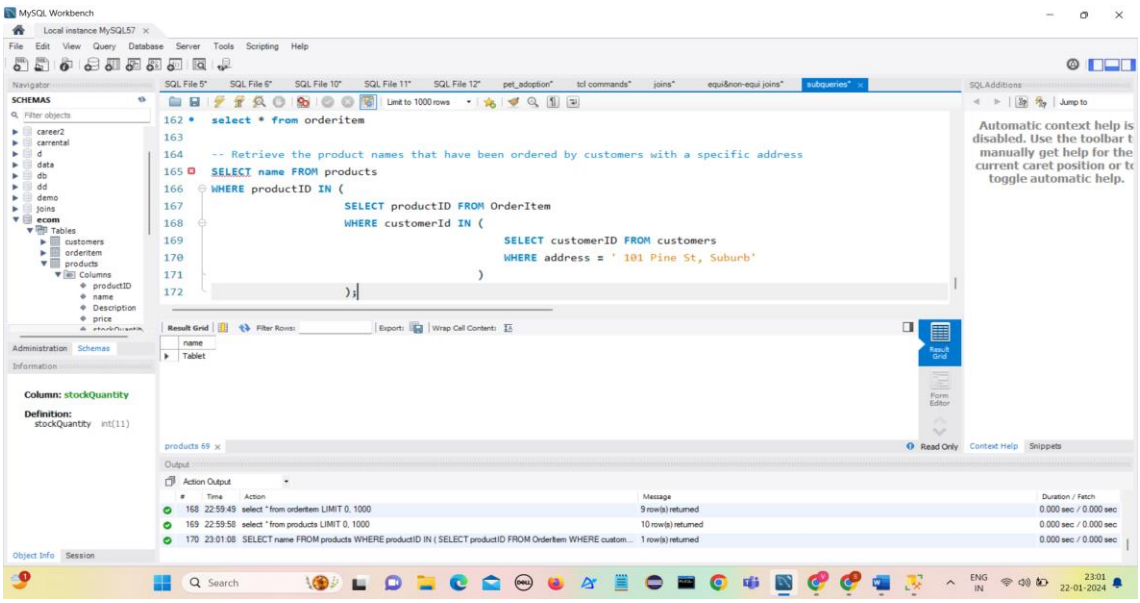
1) Retrieve customers who have placed orders for products with a price greater than the average price of all products



2) Names of products where the associated customer's first name starts with 'J'



3) Retrieve the product names that have been ordered by customers with a specific address

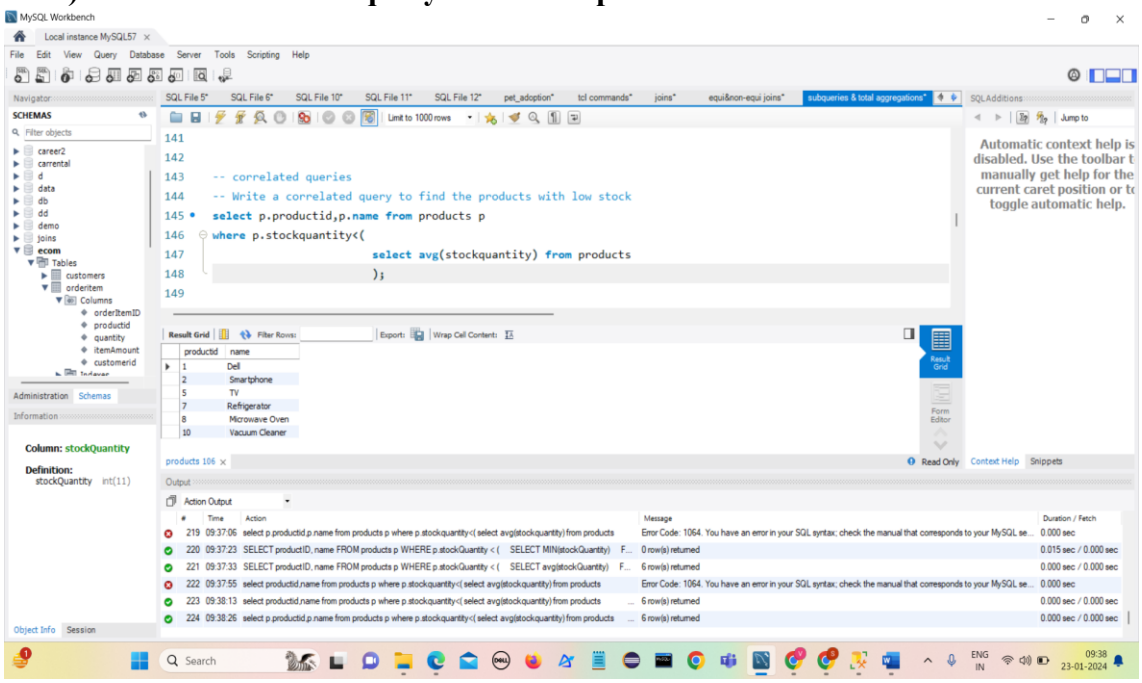


18) Correlated sub queries

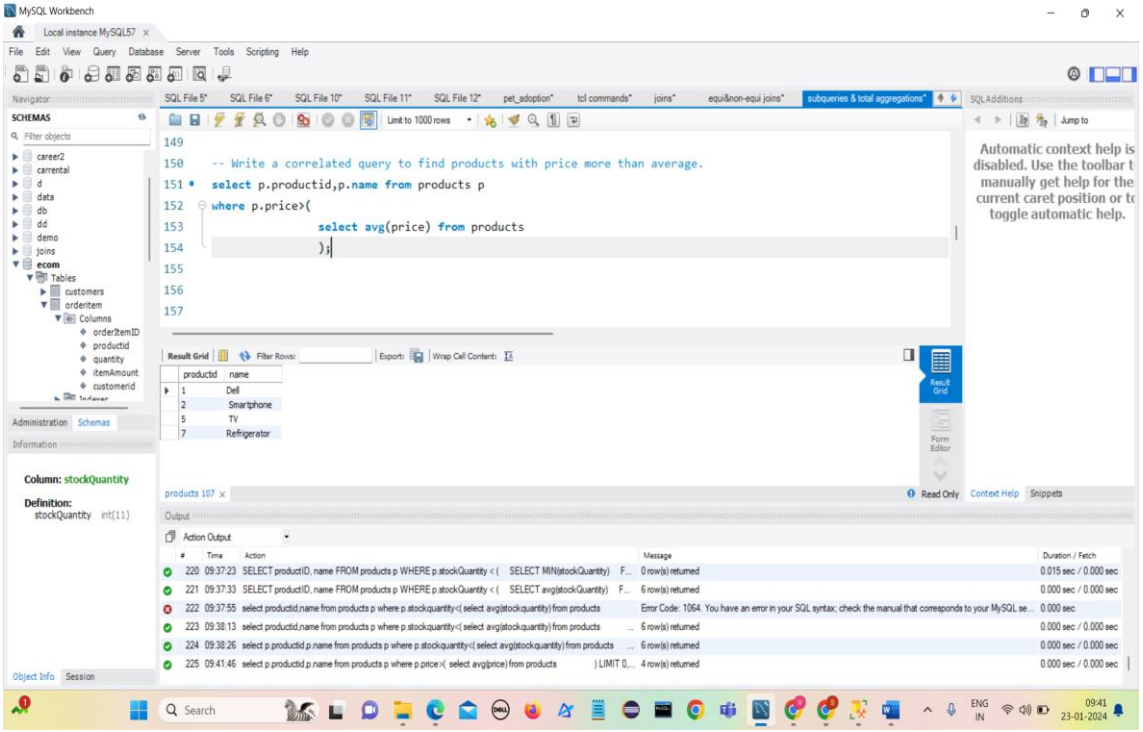
SQL Correlated Subqueries are used to select data from a table referenced in the outer query.

The subquery is known as a correlated because the subquery is related to the outer query.

1) Write a correlated query to find the products with low stock.



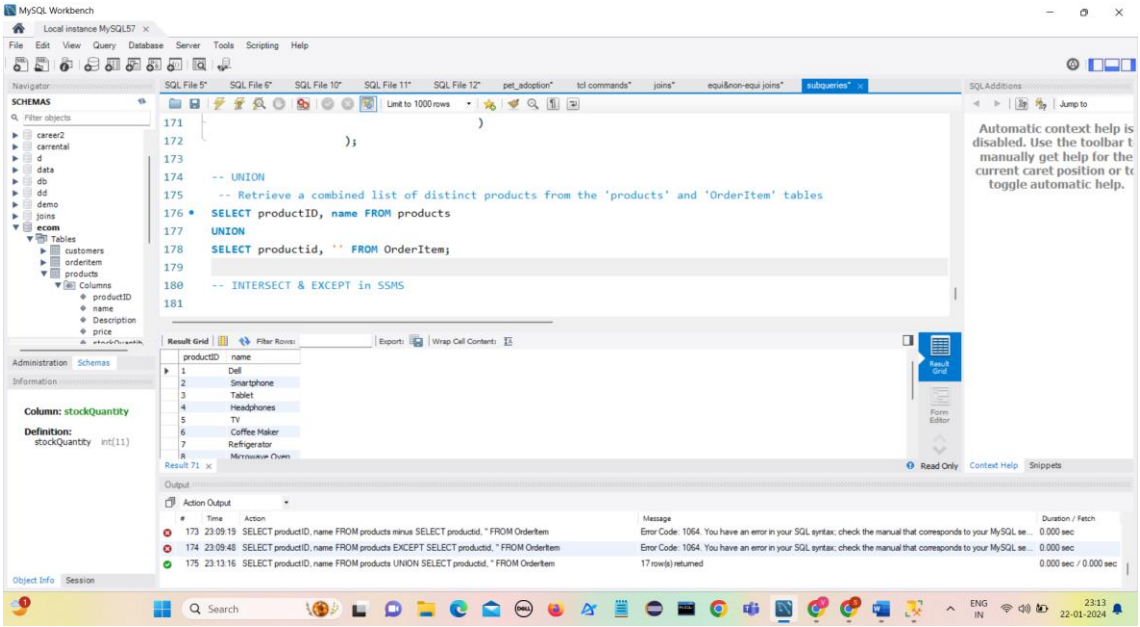
2) **Write a correlated query to find products with price more than average.**



19) UNION, INTERSECT, EXCEPT

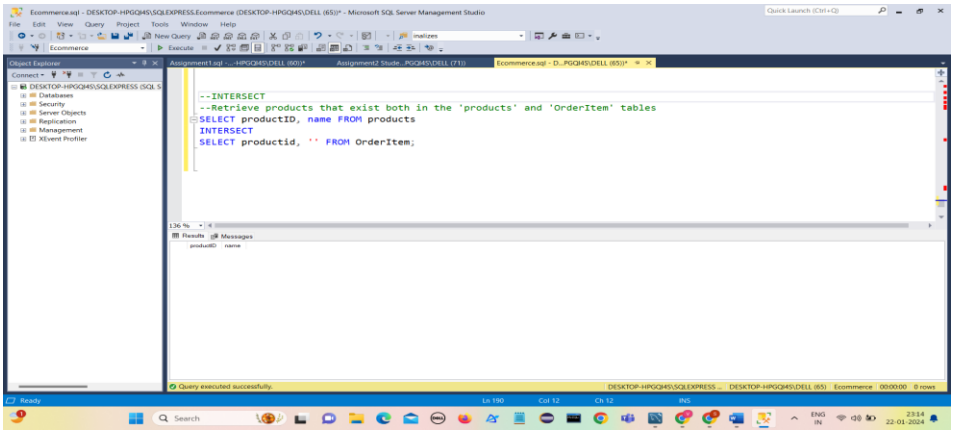
1) UNION

Retrieve a combined list of distinct products from the 'products' and 'OrderItem' tables



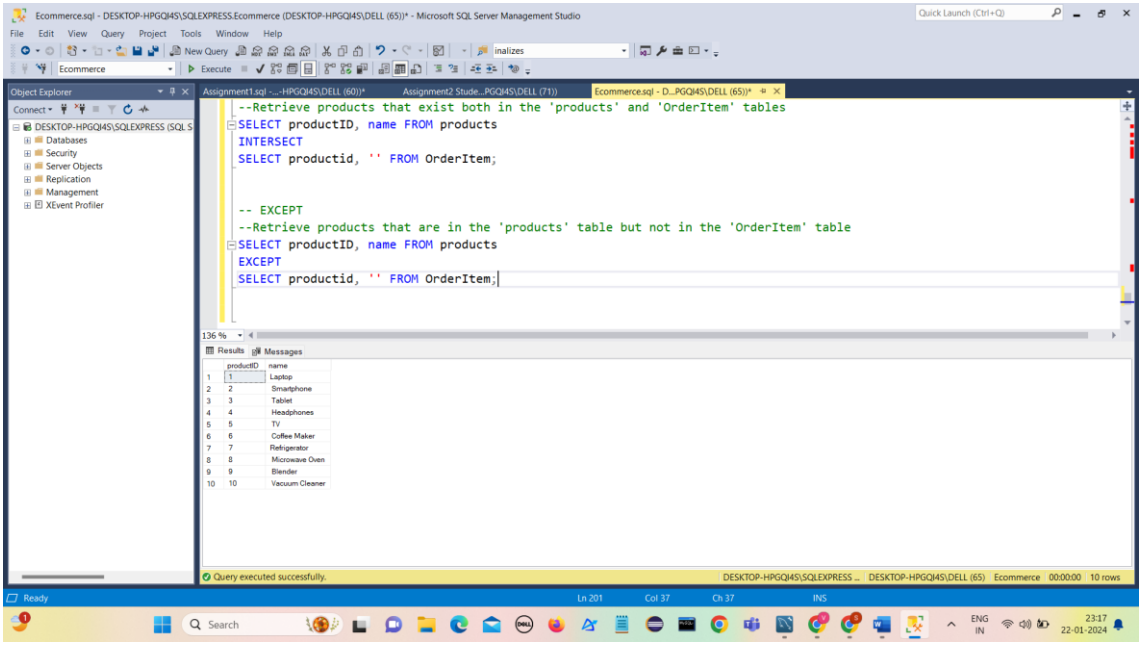
2) INTERSECT

Retrieve products that exist both in the 'products' and 'OrderItem' tables



3) EXCEPT

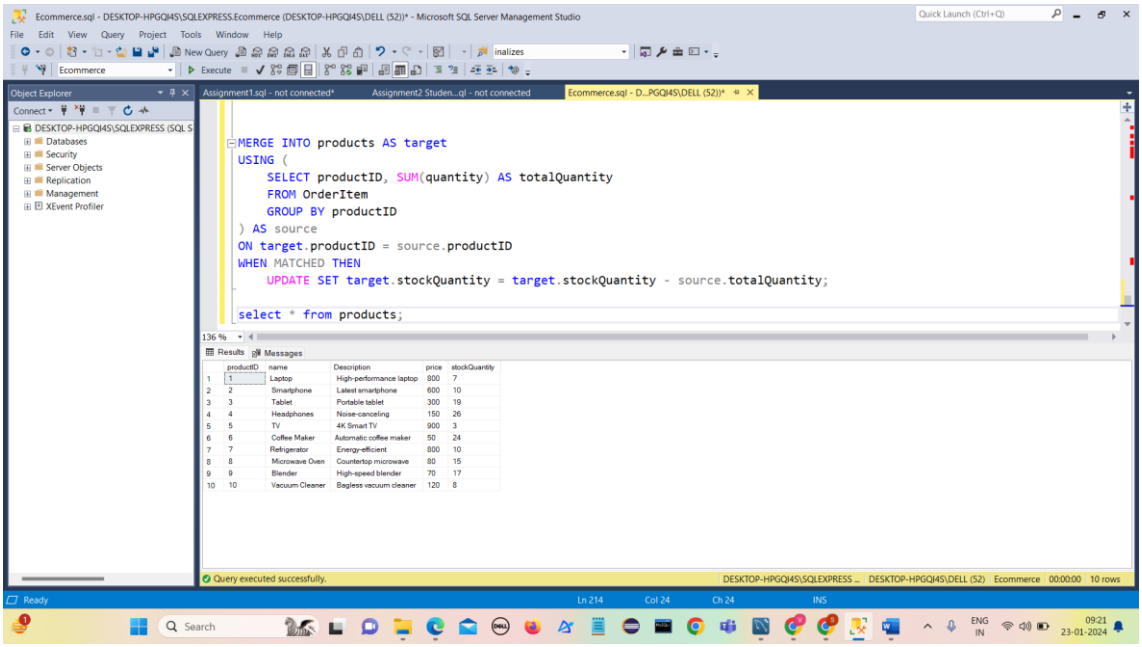
Retrieve products that are in the 'products' table but not in the 'OrderItem' table



4) MERGE

The **MERGE** statement in SQL is typically used for performing operations such as **INSERT**, **UPDATE**, or **DELETE** based on a condition.

Merge to update stock quantity in the products table



Vuthur Sriganga

Hexaware Data Engineering Batch-1