SQL Exercise: Advanced Ranking and Window Functions

* Objective:

Use ROW\_NUMBER(), RANK(), and DENSE\_RANK() with OVER(PARTITION BY ...) to retrieve the top 3 priced products per category and explore how ranking functions behave with ties.

# Step 1: Create and Use Database

CREATE DATABASE RetailStore; GO

USE RetailStore;

GO

# Step 2: Create Products Table

CREATE TABLE Products (

ProductID INT PRIMARY KEY,

ProductName VARCHAR(100),

Category VARCHAR(50),

Price DECIMAL(10,2)

);

# Step 3: Insert Sample Data

INSERT INTO Products (ProductID, ProductName, Category, Price)

VALUES

(1, 'Laptop', 'Electronics', 80000),

(2, 'Smartphone', 'Electronics', 60000),

(3, 'TV', 'Electronics', 60000),

(4, 'Tablet', 'Electronics', 30000),

(5, 'Washing Machine', 'Appliances', 25000),

(6, 'Refrigerator', 'Appliances', 35000),

(7, 'Microwave', 'Appliances', 15000),

(8, 'Vacuum Cleaner', 'Appliances', 25000),

(9, 'Jeans', 'Clothing', 2000),

(10, 'Shirt', 'Clothing', 1500),

(11, 'Jacket', 'Clothing', 4000),

(12, 'Shoes', 'Clothing', 4000);

# Step 4: Ranking with ROW\_NUMBER()

SELECT

ProductID,

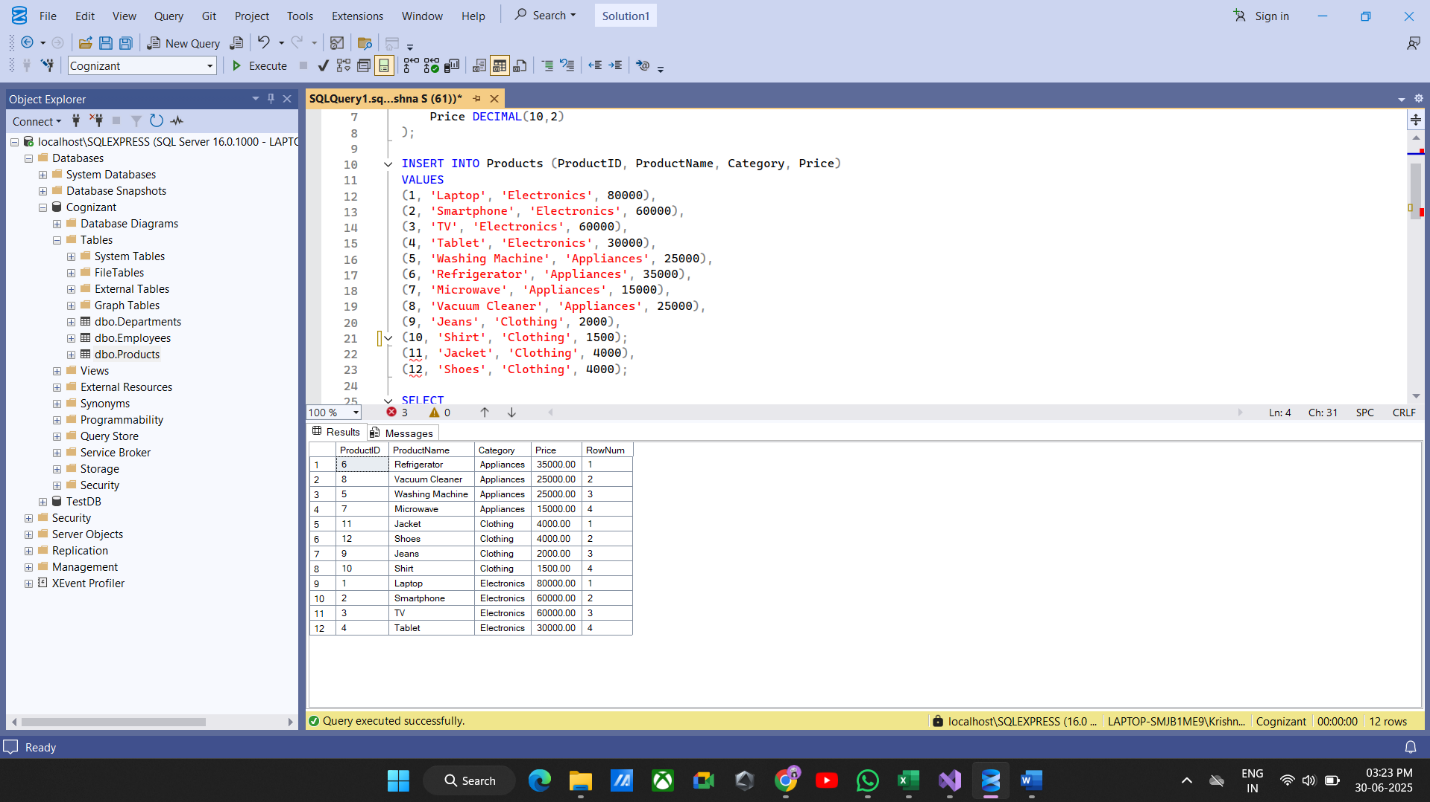
ProductName,

Category,

Price,

ROW\_NUMBER() OVER (PARTITION BY Category ORDER BY Price DESC) AS RowNum

FROM Products;



# Step 5: Ranking with RANK() and DENSE\_RANK()

SELECT

ProductID,

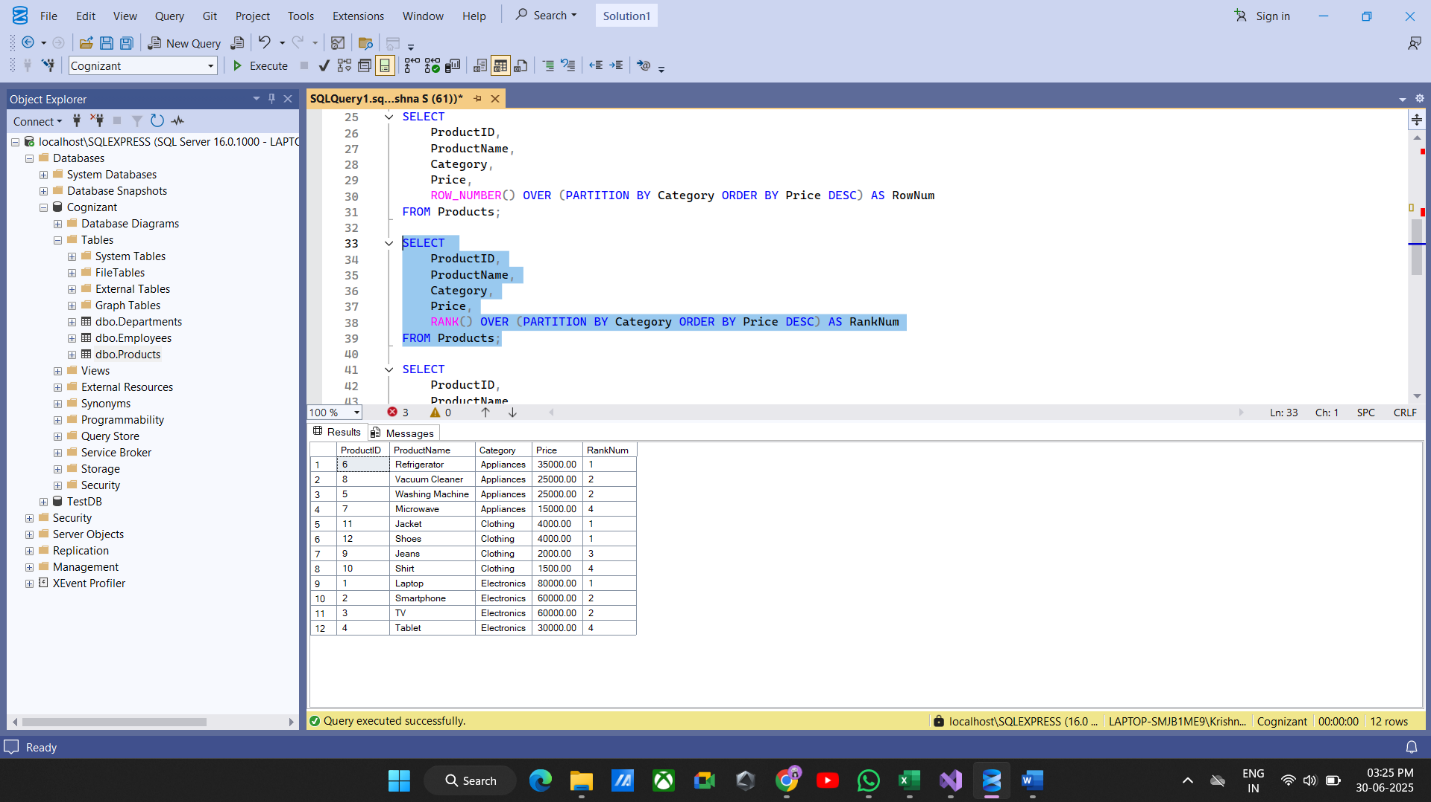
ProductName,

Category,

Price,

RANK() OVER (PARTITION BY Category ORDER BY Price DESC) AS RankNum

FROM Products;



# Step 6: Get Top 3 Products Using ROW\_NUMBER()

SELECT

ProductID,

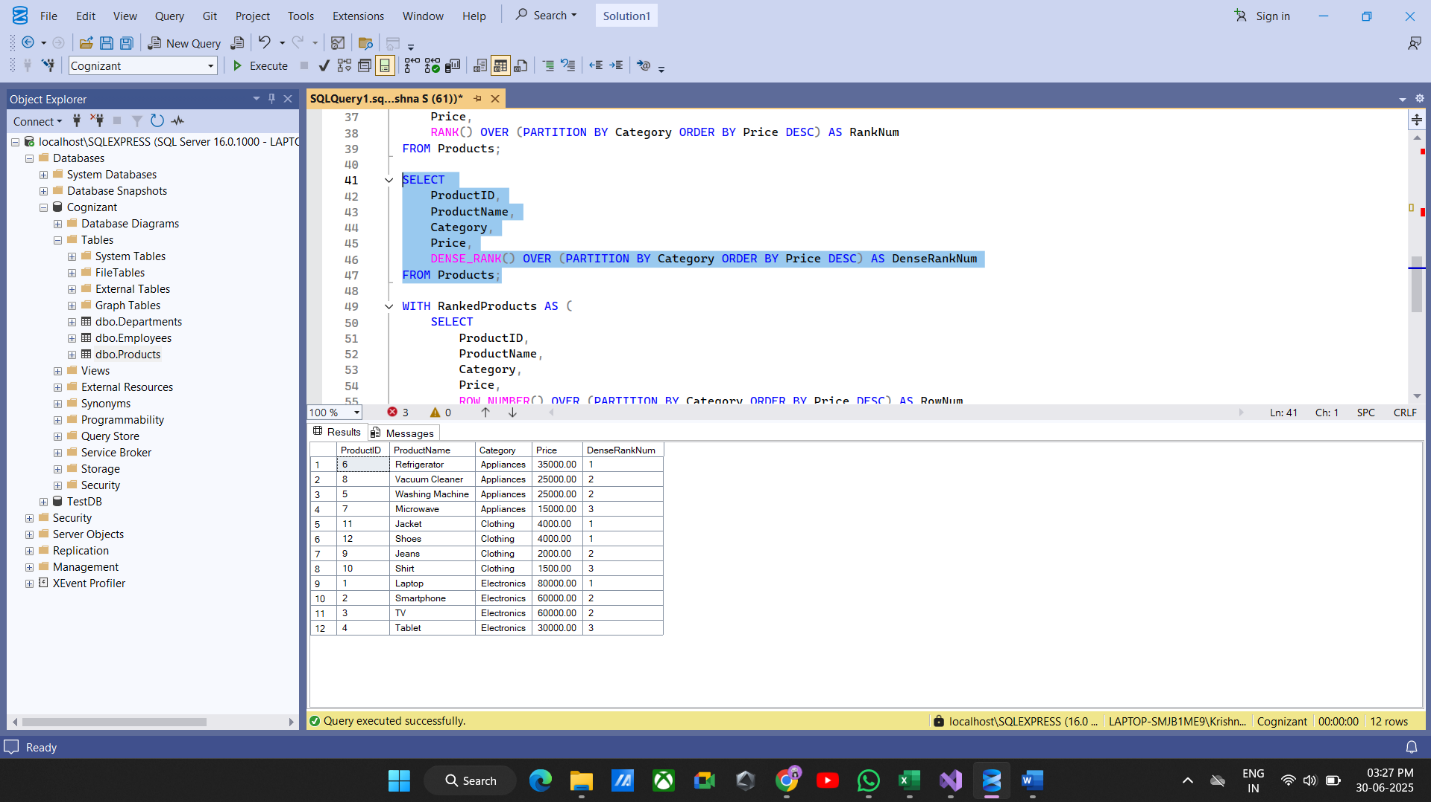
ProductName,

Category,

Price,

DENSE\_RANK() OVER (PARTITION BY Category ORDER BY Price DESC) AS DenseRankNum

FROM Products;



# Step 7: Get Top 3 Products Using RANK() and DENSE\_RANK()

WITH RankedProducts AS (

SELECT

ProductID,

ProductName,

Category,

Price,

ROW\_NUMBER() OVER (PARTITION BY Category ORDER BY Price DESC) AS RowNum

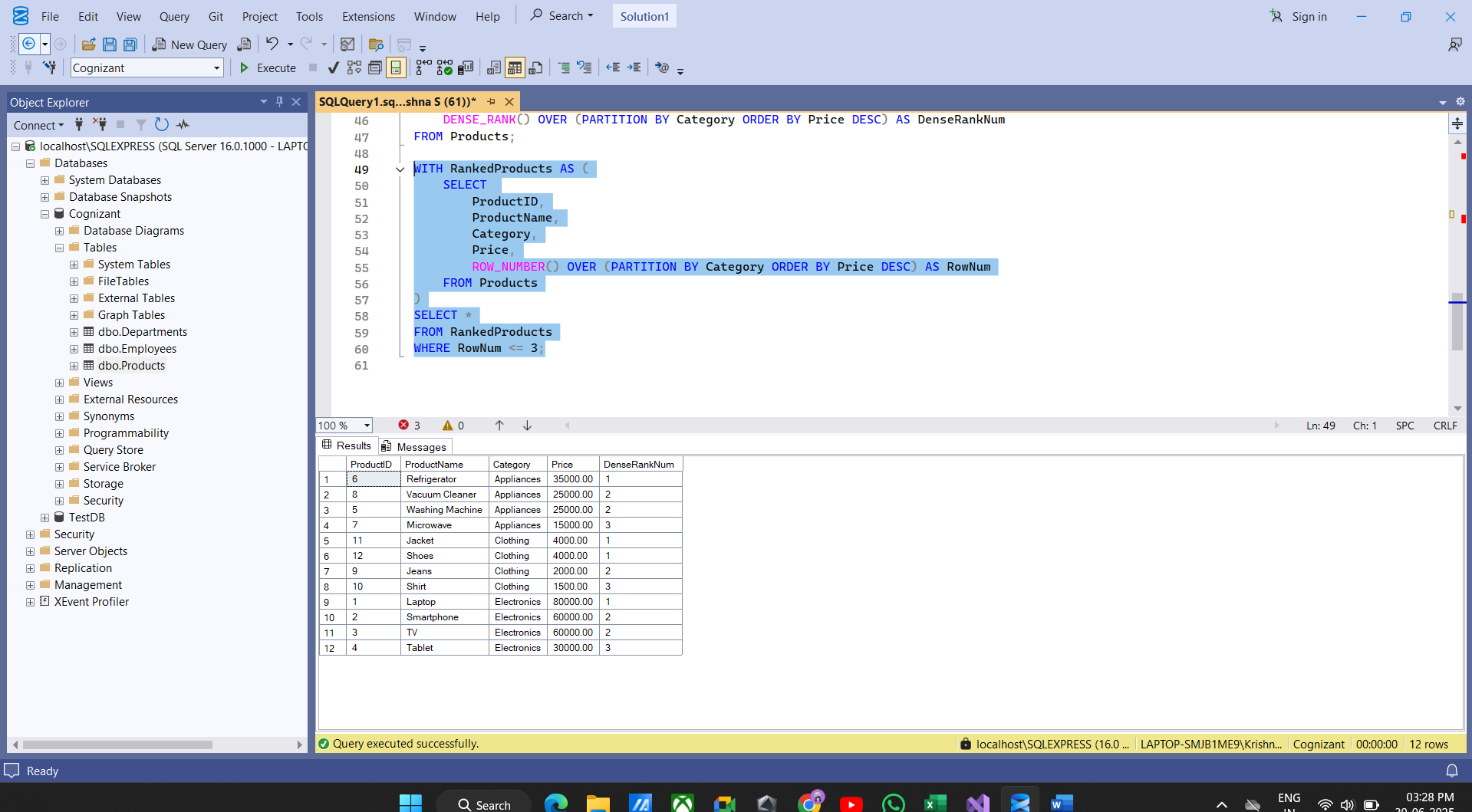
FROM Products

)

SELECT \*

FROM RankedProducts

WHERE RowNum <= 3;



# Summary

* ROW\_NUMBER() always gives unique ranks, no matter if prices are tied.
* RANK() skips rank numbers when there are ties.
* DENSE\_RANK() does not skip rank numbers when there are ties.

The queries return top 3 items per category based on these differences.