SQL EXERCISE – ADVANCED CONCEPTS

Exercise 1: Ranking and Window Functions

Code :

-- Create a sample database

CREATE DATABASE ShopRankingDB;

GO

USE ShopRankingDB;

GO

-- Drop the table if it already exists

IF OBJECT\_ID('ProductCatalog', 'U') IS NOT NULL

DROP TABLE ProductCatalog;

-- Create a sample table

CREATE TABLE ProductCatalog (

ProductID INT PRIMARY KEY,

ProductName VARCHAR(100),

Category VARCHAR(50),

Price DECIMAL(10, 2)

);

GO

-- Insert sample product data

INSERT INTO ProductCatalog (ProductID, ProductName, Category, Price) VALUES

(1, 'Laptop Pro', 'Electronics', 1200.00),

(2, 'Smartphone X', 'Electronics', 999.99),

(3, 'Bluetooth Speaker', 'Electronics', 150.00),

(4, 'Tablet Z', 'Electronics', 450.00),

(5, 'Smartwatch Y', 'Electronics', 450.00),

(6, 'Office Chair', 'Furniture', 300.00),

(7, 'Wooden Desk', 'Furniture', 500.00),

(8, 'Bookshelf', 'Furniture', 200.00),

(9, 'Recliner Sofa', 'Furniture', 1000.00),

(10, 'Dining Table', 'Furniture', 800.00),

(11, 'Running Shoes', 'Apparel', 120.00),

(12, 'Leather Jacket', 'Apparel', 250.00),

(13, 'Casual Shirt', 'Apparel', 80.00),

(14, 'Denim Jeans', 'Apparel', 150.00),

(15, 'Sneakers', 'Apparel', 150.00);

GO

-- Query 1: Using ROW\_NUMBER() to assign unique rank within each category

WITH RankedProducts AS (

SELECT

ProductID,

ProductName,

Category,

Price,

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

FROM ProductCatalog

)

SELECT

ProductID,

ProductName,

Category,

Price,

RowNum

FROM RankedProducts

WHERE RowNum <= 3

ORDER BY Category, RowNum;

GO

-- Query 2: Using RANK() to get top 3 most expensive products per category (allows gaps in ranking if ties occur)

SELECT \*

FROM (

SELECT \*, RANK() OVER (PARTITION BY Category ORDER BY Price DESC) AS RankNum

FROM ProductCatalog

) AS Ranked

WHERE RankNum <= 3

ORDER BY Category, RankNum;

GO

-- Query 3: Using DENSE\_RANK() to get top 3 most expensive products per category (no gaps in ranking for ties)

SELECT \*

FROM (

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

FROM ProductCatalog

) AS Ranked

WHERE DenseRankNum <= 3

ORDER BY Category, DenseRankNum;

GO

-- Query 4: Top 3 most expensive dishes per category using DENSE\_RANK()

SELECT \*

FROM (

SELECT \*, DENSE\_RANK() OVER (PARTITION BY Category ORDER BY Price DESC) AS DenseRank

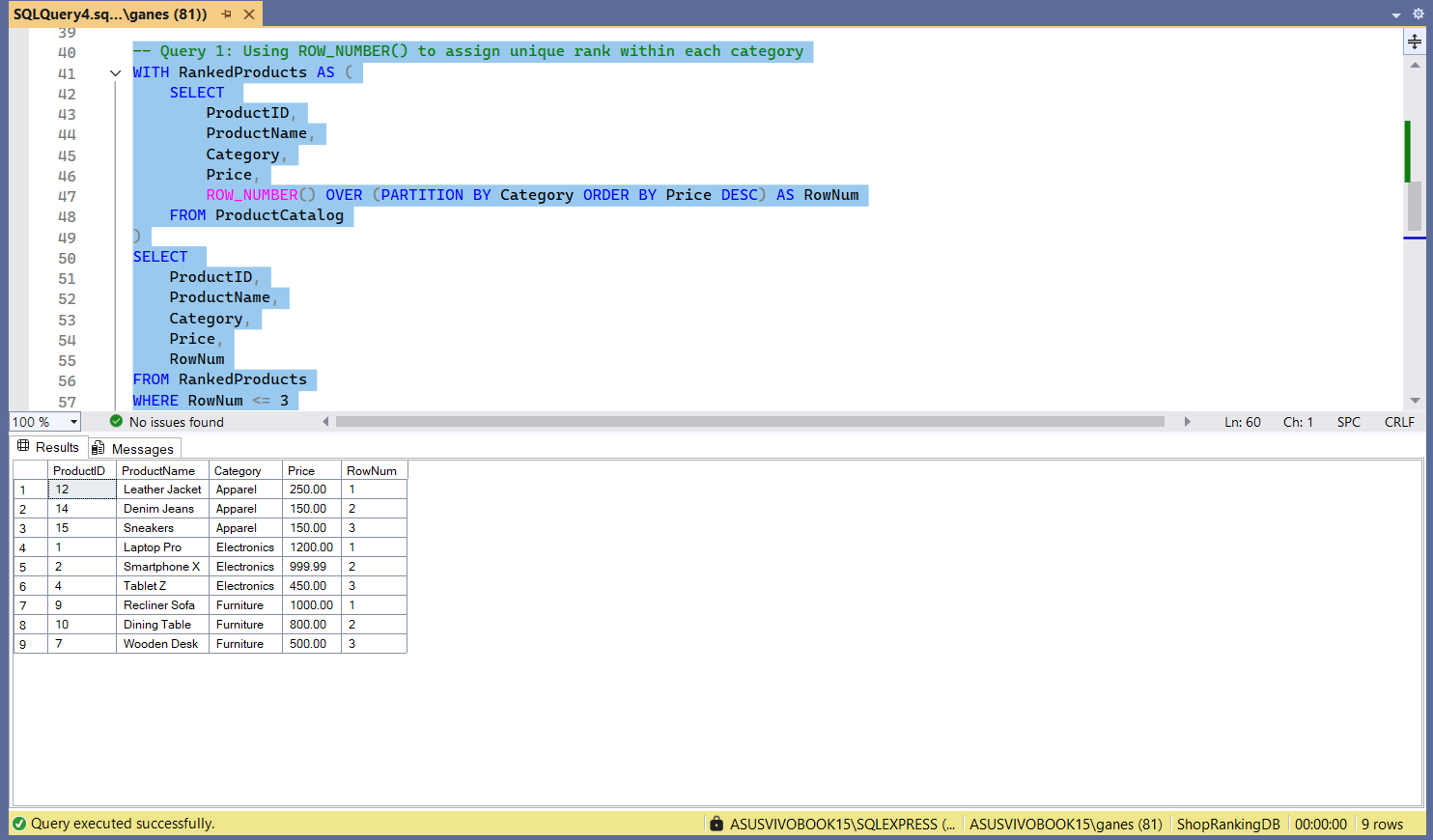
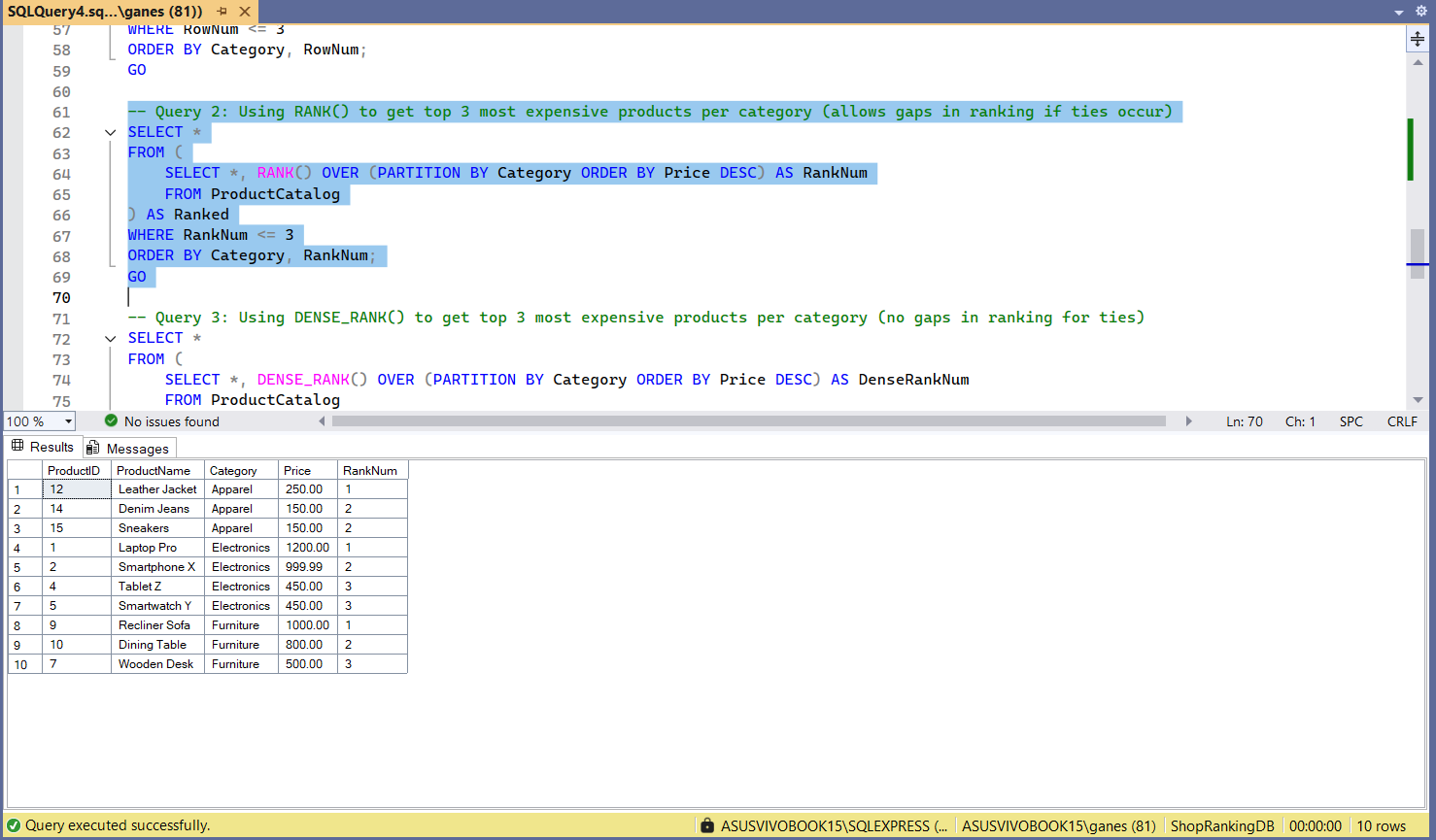
FROM ProductCatalog

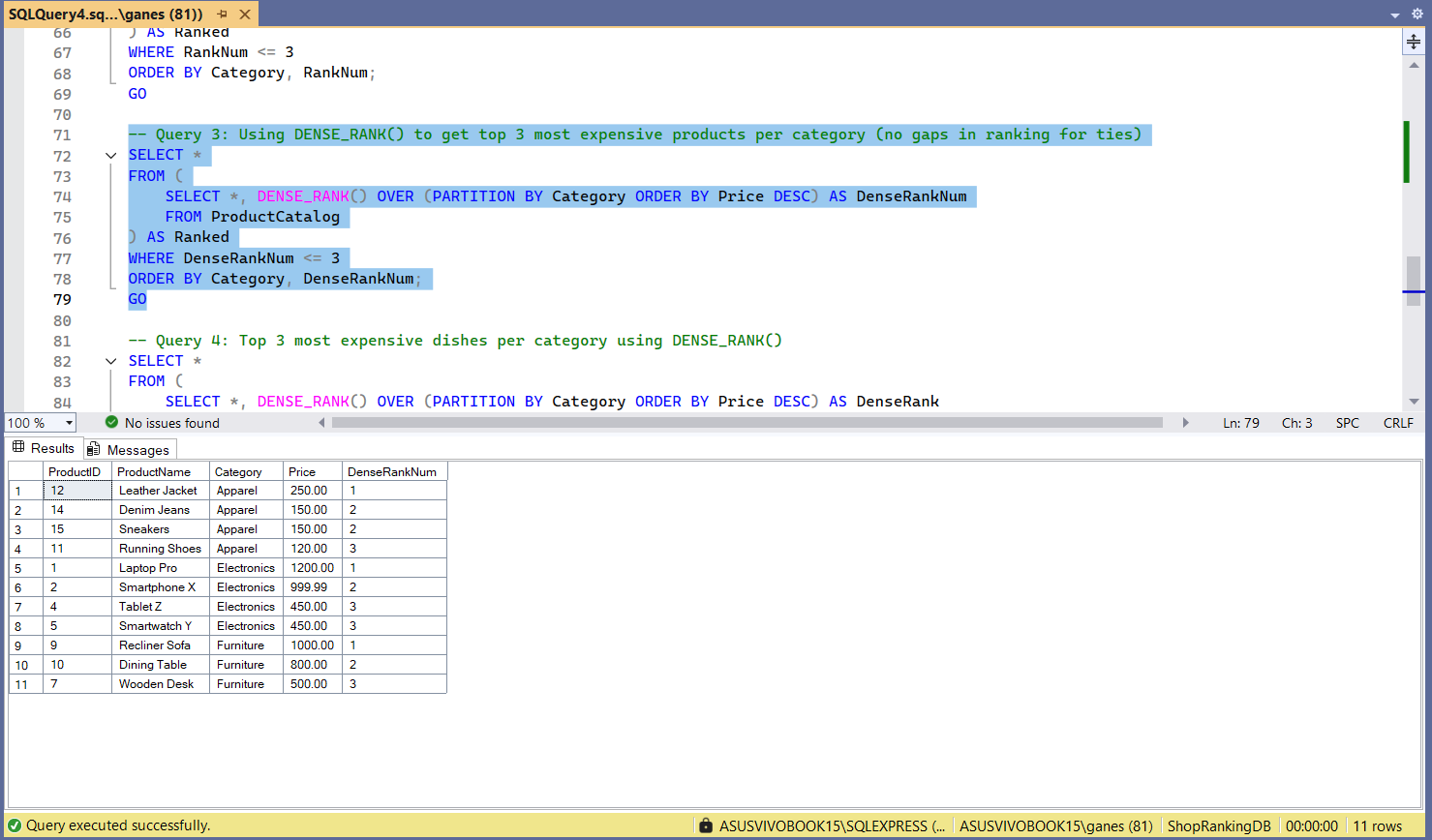
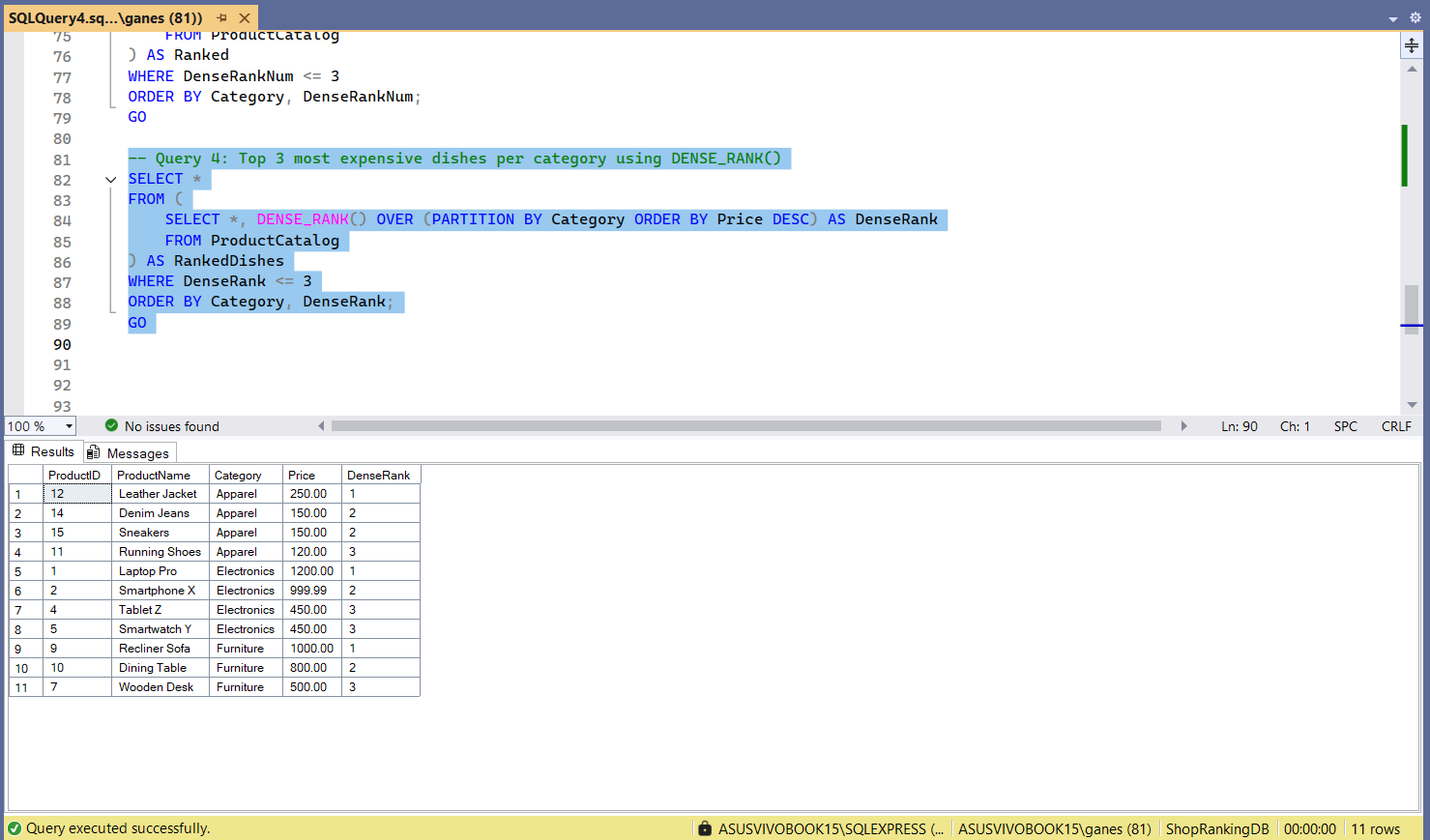
) AS RankedDishes

WHERE DenseRank <= 3

ORDER BY Category, DenseRank;

GO

**OUTPUT :**

****