**Aayushi Dutta  
Superset ID – 6363122  
Exercise 1 Advanced SQL**

CREATE TABLE Products (

ProductID INT PRIMARY KEY,

ProductName VARCHAR(50),

Category VARCHAR(50),

Price DECIMAL(10,2)

);

-- Inserting Data

INSERT INTO Products VALUES

(1, 'Phone', 'Electronics', 70000),

(2, 'Laptop', 'Electronics', 65000),

(3, 'Headphones', 'Electronics', 3000),

(4, 'Fridge', 'Appliances', 40000),

(5, 'Microwave', 'Appliances', 10000),

(6, 'Washing Machine', 'Appliances', 40000),

(7, 'Notebook', 'Stationery', 50),

(8, 'Pen', 'Stationery', 30),

(9, 'File', 'Stationery', 80),

(10, 'Printer', 'Electronics', 20000);

--Use ROW\_NUMBER() to assign a unique rank within each category

SELECT

ProductID,

ProductName,

Category,

Price,

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

FROM Products;

--Use RANK() and DENSE\_RANK() to compare how ties are handled.

SELECT

ProductID,

ProductName,

Category,

Price,

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

FROM Products;

SELECT

ProductID,

ProductName,

Category,

Price,

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

FROM Products;

--Use PARTITION BY Category and ORDER BY Price DESC.

WITH RankedProducts AS (

SELECT \*,

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

FROM Products

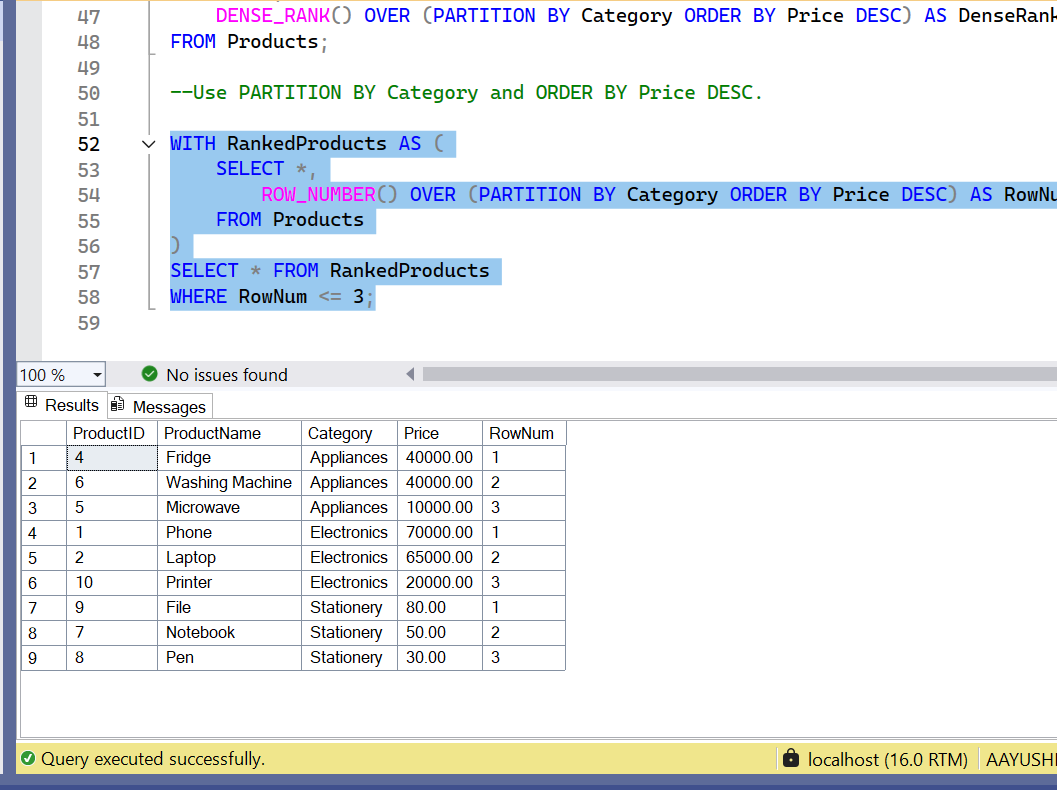
)

SELECT \* FROM RankedProducts

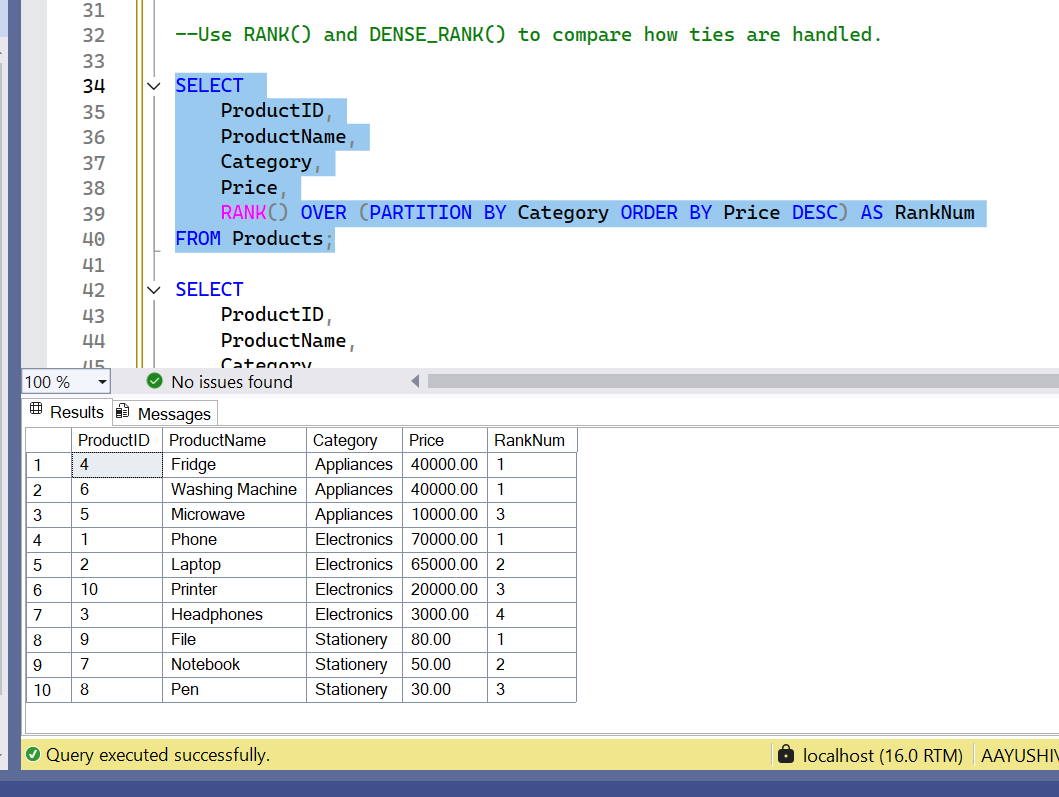
WHERE RowNum <= 3;

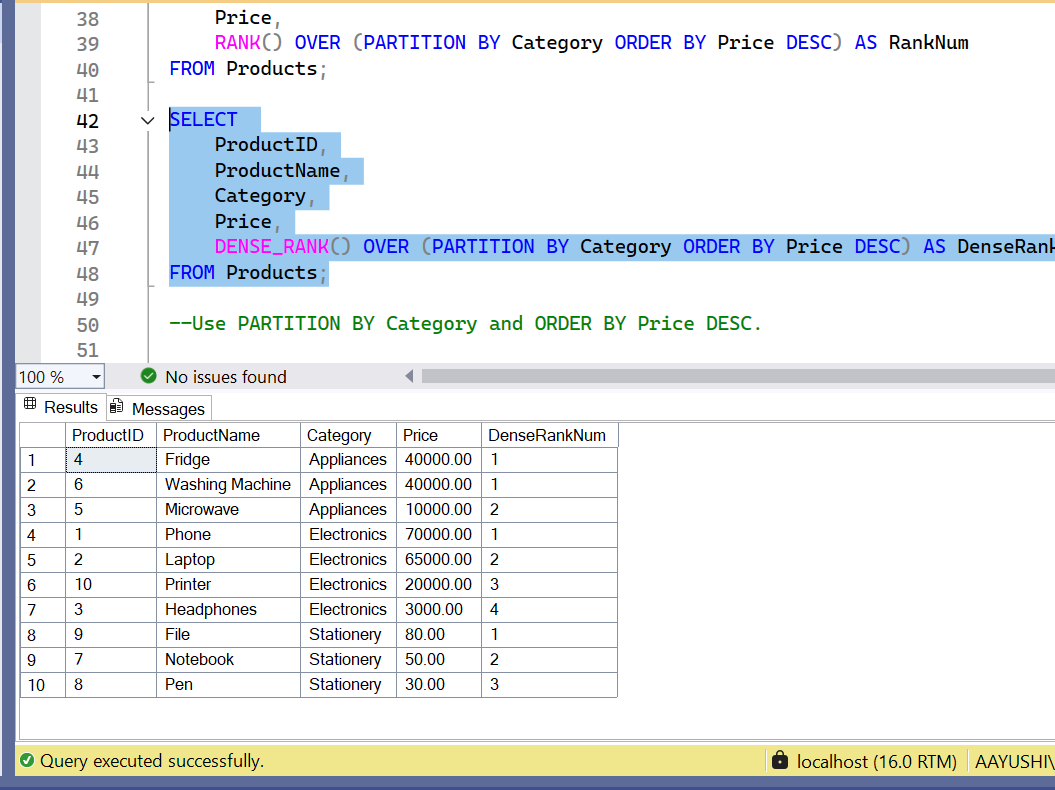
OUTPUT:

1. Use ROW\_NUMBER() to assign a unique rank within each category



1. Use RANK() and DENSE\_RANK() to compare how ties are handled





3. Use PARTITION BY Category and ORDER BY Price DESC

