**Queries Used In Carrying Out the Week2 Advanced SQL HandsOn:-**

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

*Query:-*

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

(5, 'Smartwatch', 'Electronics', 800.00),

(6, 'Charger', 'Accessories', 50.00),

(7, 'Mouse', 'Accessories', 150.00),

(8, 'Backpack', 'Accessories', 80.00),

(9, 'Notebook', 'Stationery', 10.00),

(10, 'Pen', 'Stationery', 2.00),

(11, 'Marker', 'Stationery', 5.00),

(12, 'Printer', 'Electronics', 1000.00),

(13, 'Desk Lamp', 'Home Decor', 300.00),

(14, 'Cushion', 'Home Decor', 200.00),

(15, 'Rug', 'Home Decor', 500.00);

SELECT

Category,

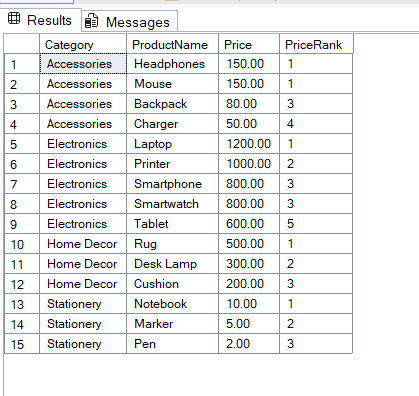
ProductName,

Price,

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

FROM Products;

*Output:-*



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

*Query:-*

SELECT

Category,

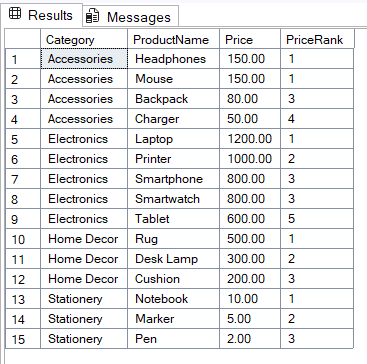
ProductName,

Price,

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

FROM Products;

***Output:-***



***Query:-***

SELECT

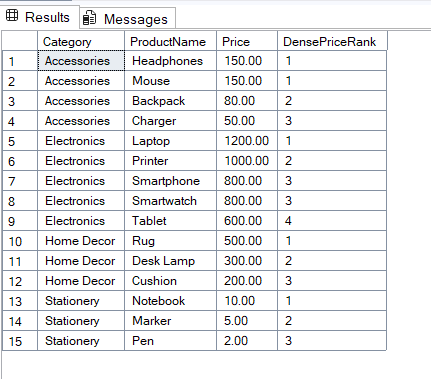
Category,

ProductName,

Price,

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

FROM Products;

******

*3. Use PARTITION BY Category and ORDER BY Price DESC.*

*Query:-*

SELECT \* FROM (

SELECT

Category,

ProductName,

Price,

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

FROM Products

) AS Ranked

WHERE DensePriceRank <= 3;

