# WEEK-2 {ADVANCED SQL}\_{ADVANCED\_CONCEPT

1. EXERCISE-1{ Ranking and Window Functions}

CODE-1

-- Create the Products table

CREATE TABLE Products (

ProductID INT PRIMARY KEY,

ProductName VARCHAR(100),

Category VARCHAR(50),

Price DECIMAL(10, 2)

);

-- Insert sample data

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

VALUES

(1, 'iPhone 15', 'Electronics', 1200.00),

(2, 'Samsung S23', 'Electronics', 1100.00),

(3, 'MacBook Air', 'Electronics', 1500.00),

(4, 'Dell XPS 13', 'Electronics', 1400.00),

(5, 'HP Envy', 'Electronics', 1100.00),

(6, 'T-shirt', 'Clothing', 20.00),

(7, 'Jeans', 'Clothing', 40.00),

(8, 'Jacket', 'Clothing', 60.00),

(9, 'Shoes', 'Clothing', 60.00),

(10, 'Blazer', 'Clothing', 80.00);

CODE-2{Use ROW\_NUMBER() to assign a unique rank within each category.}

SELECT \*

FROM (

SELECT

ProductID,

ProductName,

Category,

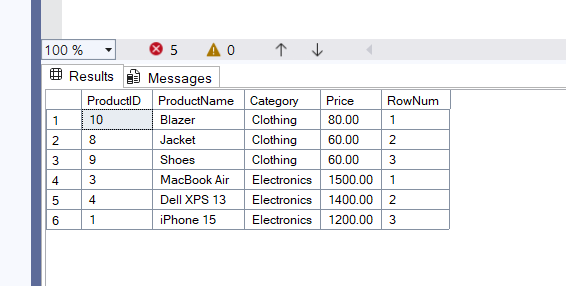
Price,

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

FROM Products

) AS Ranked

WHERE RowNum <= 3;

Output:-

Code-3{Use RANK() and DENSE\_RANK() to compare how ties are handled.}

SELECT \*

FROM (

SELECT

ProductID,

ProductName,

Category,

Price,

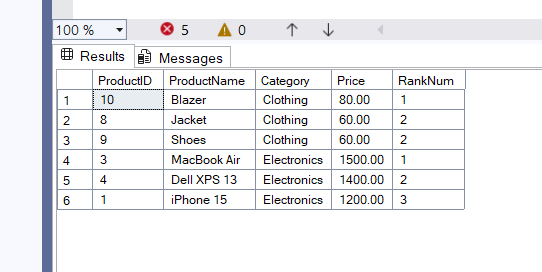
RANK() OVER (PARTITION BY Category ORDER BY Price DESC) AS RankNum

FROM Products

) AS Ranked

WHERE RankNum <= 3;

OUTPUT-



SELECT \*

FROM (

SELECT

ProductID,

ProductName,

Category,

Price,

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

FROM Products

) AS Ranked

WHERE DenseRankNum <= 3;

OUTPUT-

