**Exercise 1: Ranking and Window Functions**

Goal: Use ROW\_NUMBER(), RANK(), DENSE\_RANK(), OVER(), and PARTITION BY.

Scenario:

Find the top 3 most expensive products in each category using different ranking functions.

Steps:

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

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

3. Use PARTITION BY Category and ORDER BY Price DESC

**CODE:**

use sampledb;

CREATE TABLE Products (

ProductID INT,

ProductName VARCHAR(100),

Category VARCHAR(100),

Price DECIMAL(10, 2)

);

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

(1, 'iPhone 14', 'Smartphones', 999.99),

(2, 'Galaxy S22', 'Smartphones', 899.99),

(3, 'Pixel 7', 'Smartphones', 899.99),

(4, 'OnePlus 10', 'Smartphones', 749.99),

(5, 'MacBook Pro', 'Laptops', 1999.99),

(6, 'Dell XPS 15', 'Laptops', 1899.99),

(7, 'HP Spectre', 'Laptops', 1899.99),

(8, 'Lenovo Yoga', 'Laptops', 1499.99),

(9, 'Sony WH-1000XM5', 'Headphones', 399.99),

(10, 'Bose QC45', 'Headphones', 379.99),

(11, 'AirPods Max', 'Headphones', 549.99),

(12, 'Jabra Elite 85h', 'Headphones', 249.99),

(13, 'iPad Pro', 'Tablets', 1099.99),

(14, 'Galaxy Tab S8', 'Tablets', 999.99),

(15, 'Surface Pro 9', 'Tablets', 1099.99),

(16, 'Lenovo Tab P11', 'Tablets', 499.99);

SELECT \*

FROM (

SELECT

ProductID,

ProductName,

Category,

Price,

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

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

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

FROM Products

) ranked

WHERE RowNum <= 3 OR RankNum <= 3 OR DenseRankNum <= 3

ORDER BY Category, Price DESC;

**OUTPUT:**

