**Advanced SQL Exercises for Online Retail Store**

**Exercise 1: Ranking and Window Function**

* **CODE**:

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

DROP TABLE Products;

CREATE TABLE Products (

ProductID INT,

ProductName VARCHAR(100),

Category VARCHAR(100),

Price DECIMAL(10, 2)

);

-- 2. Insert Sample Data

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

(1, 'Pen', 'Stationery', 10.00),

(2, 'Pencil', 'Stationery', 5.00),

(3, 'Notebook', 'Stationery', 20.00),

(4, 'Milk', 'Grocery', 30.00),

(5, 'Bread', 'Grocery', 25.00),

(6, 'Butter', 'Grocery', 30.00),

(7, 'Marker', 'Stationery', 20.00),

(8, 'Cheese', 'Grocery', 20.00),

(9, 'Eraser', 'Stationery', 5.00);

Step 1: ROW\_NUMBER() - Unique rank per category

SELECT

ProductID,

ProductName,

Category,

Price,

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

FROM Products;

Step 2: RANK() and DENSE\_RANK() - How ties are handled

SELECT

ProductID,

ProductName,

Category,

Price,

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

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

FROM Products;

Step 3: Top 3 most expensive products per category using ROW\_NUMBER()

WITH RankedProducts AS (

SELECT

ProductID,

ProductName,

Category,

Price,

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

FROM Products

)

SELECT \* FROM RankedProducts

WHERE RowNum <= 3;

SELECT \* FROM Products;

OUTPUT:





