|  |  |
| --- | --- |
| **Cognizant Digital Nurture 4.0: Deep Skilling** | |
| Name: Rajatesh Paul | Superset ID: 6365356 |
| Week: 2 | Advance SQL (Exercise - Advanced concepts) |

**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.

**SQL Code:**

create database Excersise1;

use Excersise1;

CREATE TABLE Products (

ProductID INT PRIMARY KEY,

ProductName VARCHAR(50),

Category VARCHAR(50),

Price DECIMAL(10, 2)

);

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

(1, 'Laptop A', 'Electronics', 1000.00),

(2, 'Laptop B', 'Electronics', 950.00),

(3, 'Laptop C', 'Electronics', 950.00),

(4, 'Tablet A', 'Electronics', 600.00),

(5, 'Phone A', 'Electronics', 500.00),

(6, 'Chair A', 'Furniture', 200.00),

(7, 'Chair B', 'Furniture', 180.00),

(8, 'Desk A', 'Furniture', 300.00),

(9, 'Desk B', 'Furniture', 300.00),

(10, 'Sofa', 'Furniture', 400.00);

**1.** SELECT \* FROM (

SELECT

ProductID,

ProductName,

Category,

Price,

ROW\_NUMBER() OVER (

PARTITION BY Category

ORDER BY Price DESC

) AS RowNum

FROM Products

) AS RankedProducts

WHERE RowNum <= 3;

**2.** WITH Ranked AS (

SELECT

ProductID,

ProductName,

Category,

Price,

RANK() OVER (

PARTITION BY Category

ORDER BY Price DESC

) AS RankNum

FROM Products

)

SELECT \* FROM Ranked

WHERE RankNum <= 3;

**3.** SELECT \* FROM (

SELECT

ProductID,

ProductName,

Category,

Price,

DENSE\_RANK() OVER (

PARTITION BY Category

ORDER BY Price DESC

) AS DenseRank

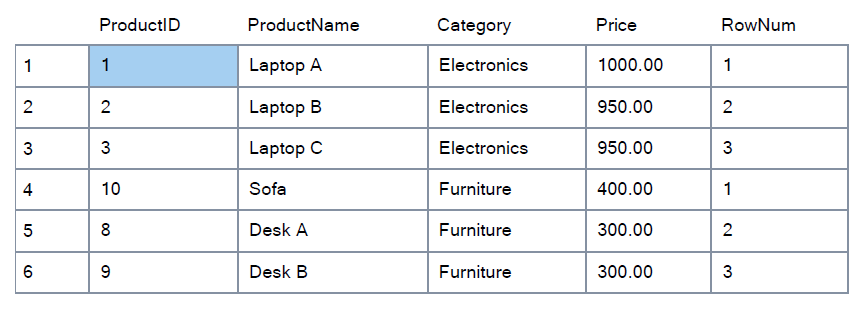
FROM Products

) AS Ranked

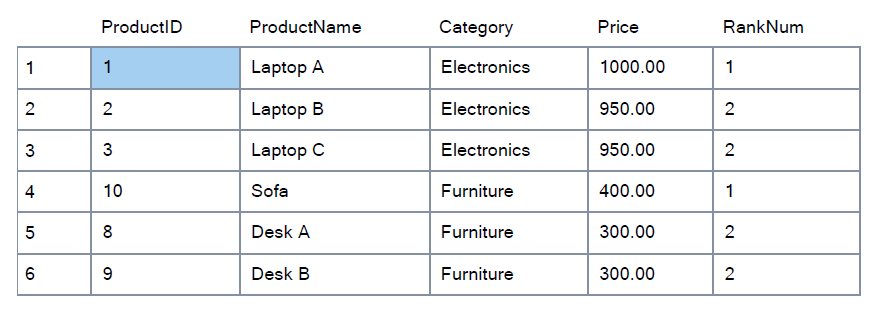
WHERE DenseRank <= 3;

**Output:**

1.ROW\_NUMBER():



2. RANK() and DENSE\_RANK():



3. PARTITION BY:

