COGNIZANT DN - 4.0 DEEP SKILLING

HANDS ON WEEK-2

Advanced SQL

**1. SQL Exercise - Advanced concepts**

**Exercise 1- Ranking and Window Functions**

**CODE:**

1. **Query with ROW\_NUMBER()**

WITH RankedProducts AS (

SELECT \*,

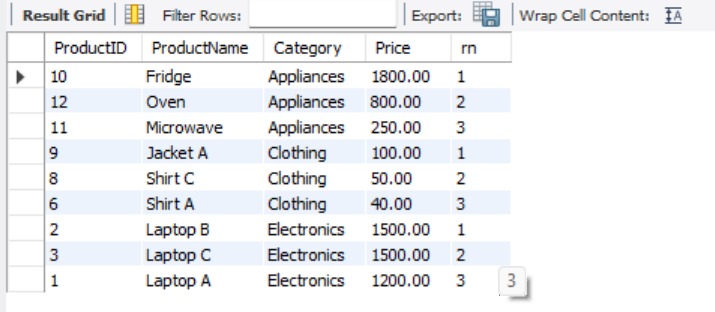
ROW\_NUMBER() OVER (PARTITION BY Category ORDER BY Price DESC) AS rn

FROM Products

)

SELECT \* FROM RankedProducts

WHERE rn <= 3;



1. **Query with RANK()**

WITH RankedProducts AS (

SELECT \*,

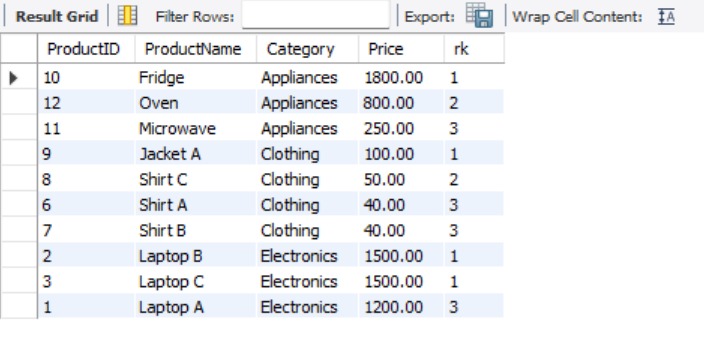
RANK() OVER (PARTITION BY Category ORDER BY Price DESC) AS rk

FROM Products

)

SELECT \* FROM RankedProducts

WHERE rk <= 3;



### Query with DENSE\_RANK and PARTITION BY

### WITH RankedProducts AS (

### SELECT \*,

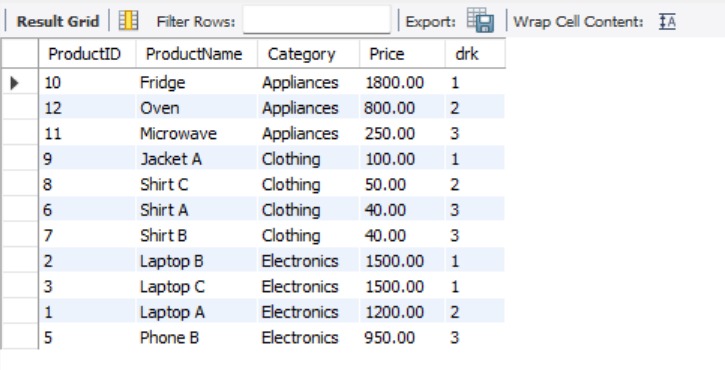
### DENSE\_RANK() OVER (PARTITION BY Category ORDER BY Price DESC) AS drk

### FROM Products

### )

### SELECT \* FROM RankedProducts

### WHERE drk <= 3;



**4. SQL Exercise - Stored procedure**

**Exercise 1- Create a Stored Procedure**

**CODE:**

1. **. Define the stored procedure with a parameter for DepartmentID.**

DELIMITER //

CREATE PROCEDURE sp\_GetEmployeesByDepartment(IN dept\_id INT)

BEGIN

SELECT

e.EmployeeID,

e.FirstName,

e.LastName,

e.Salary,

e.JoinDate,

d.DepartmentName

FROM Employees e

JOIN Departments d ON e.DepartmentID = d.DepartmentID

WHERE e.DepartmentID = dept\_id;

END //

DELIMITER ;

**2. Write the SQL query to select employee details based on the DepartmentID**

SELECT

e.EmployeeID,

e.FirstName,

e.LastName,

e.Salary,

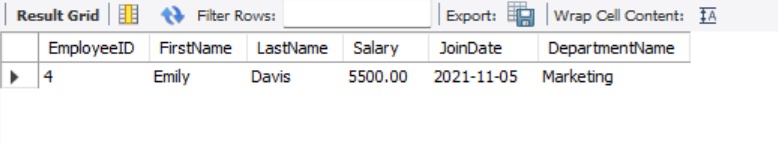
e.JoinDate,

d.DepartmentName

FROM Employees e

JOIN Departments d ON e.DepartmentID = d.DepartmentID

WHERE e.DepartmentID = 4;



**3.Create a stored procedure named `sp\_InsertEmployee`**

DELIMITER //

CREATE PROCEDURE sp\_InsertEmployee (

IN FirstName VARCHAR(50),

IN LastName VARCHAR(50),

IN DepartmentID INT,

IN Salary DECIMAL(10,2),

IN JoinDate DATE

)

BEGIN

INSERT INTO Employees (FirstName, LastName, DepartmentID, Salary, JoinDate)

VALUES (FirstName, LastName, DepartmentID, Salary, JoinDate);

END //

DELIMITER ;

### Exercise 5: Return Data from a Stored Procedure

### CODE:

### Define the stored procedure with a parameter for DepartmentID

### DELIMITER //

### CREATE PROCEDURE sp\_GetEmployeesByDepartment(IN dept\_id INT)

### BEGIN

### SELECT

### e.EmployeeID,

### e.FirstName,

### e.LastName,

### e.Salary,

### e.JoinDate,

### d.DepartmentName

### FROM Employees e

### JOIN Departments d ON e.DepartmentID = d.DepartmentID

### WHERE e.DepartmentID = dept\_id;

### END //

### DELIMITER ;

### Write the SQL query to count the number of employees in the specified department.

### DELIMITER //

### CREATE PROCEDURE sp\_CountEmployeesByDepartment(IN dept\_id INT)

### BEGIN

### SELECT

### COUNT(\*) AS TotalEmployees

### FROM Employees

### WHERE DepartmentID = dept\_id;

### END //

### DELIMITER ;

