Name : Aniket Khandelwal
Superset id : 6365314
Mail id : 2206156@kiit.ac.in
University Roll : 2206156

WEEK - 02 : HandsOn Solutions

#1 - SOL Exercise - Advanced concepts

- **Exercise - 1 : Ranking and Window Functions**

Code :

Query 1 | users | category | product

Limit to 1000 r

```
2
3 •  CREATE DATABASE IF NOT EXISTS cognizant_dn;
4 •  USE cognizant_dn;
5 •  CREATE TABLE Products (
6        ProductID INT PRIMARY KEY,
7        ProductName VARCHAR(100),
8        Category VARCHAR(50),
9        Price DECIMAL(10, 2)
10   );
11 • INSERT INTO Products (ProductID, ProductName, Category, Price) VALUES
12   (1, 'iPhone 15', 'Electronics', 999.99),
13   (2, 'Samsung Galaxy S23', 'Electronics', 899.99),
14   (3, 'Sony Headphones', 'Electronics', 199.99),
15   (4, 'Dell Laptop', 'Electronics', 999.99),
16   (5, 'Banana', 'Groceries', 0.99),
17   (6, 'Apple', 'Groceries', 1.29),
18   (7, 'Mango', 'Groceries', 2.50),
19   (8, 'Pineapple', 'Groceries', 2.50),
20   (9, 'Shampoo', 'Personal Care', 4.50),
21   (10, 'Toothpaste', 'Personal Care', 3.75),
22   (11, 'Face Wash', 'Personal Care', 4.50),
23   (12, 'Moisturizer', 'Personal Care', 6.00);
24
25 • SELECT
26       ProductID,
27       ProductName,
```

Query 1 | users | category | product

Limit to 1000 r

```sql
SELECT
    ProductID,
    ProductName,
    Category,
    Price,
    ROW_NUMBER() OVER (PARTITION BY Category ORDER BY Price DESC) AS RowNum
FROM Products;

SELECT
    ProductID,
    ProductName,
    Category,
    Price,
    RANK() OVER (PARTITION BY Category ORDER BY Price DESC) AS RankPos,
    DENSE_RANK() OVER (PARTITION BY Category ORDER BY Price DESC) AS DenseRankPos
FROM Products;

WITH Ranked AS (
    SELECT
        ProductID,
        ProductName,
        Category,
        Price,
        ROW_NUMBER() OVER (PARTITION BY Category ORDER BY Price DESC) AS RowNum
    FROM Products
)
SELECT *
FROM Ranked
WHERE RowNum <= 3;
```

Output :

1. Using ROW_NUMBER( )

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ProductID	ProductName	Category	Price	RowNum
1	iPhone 15	Electronics	999.99	1
4	Dell Laptop	Electronics	999.99	2
2	Samsung Galaxy S23	Electronics	899.99	3
3	Sony Headphones	Electronics	199.99	4
7	Mango	Groceries	2.50	1
8	Pineapple	Groceries	2.50	2
6	Apple	Groceries	1.29	3
5	Banana	Groceries	0.99	4
12	Moisturizer	Personal Care	6.00	1
9	Shampoo	Personal Care	4.50	2
11	Face Wash	Personal Care	4.50	3
10	Toothpaste	Personal Care	3.75	4

2. Using RANK( ) & DENSE_RANK( )

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ProductID	ProductName	Category	Price	RankPos	DenseRankPos
1	iPhone 15	Electronics	999.99	1	1
4	Dell Laptop	Electronics	999.99	1	1
2	Samsung Galaxy S23	Electronics	899.99	3	2
3	Sony Headphones	Electronics	199.99	4	3
7	Mango	Groceries	2.50	1	1
8	Pineapple	Groceries	2.50	1	1
6	Apple	Groceries	1.29	3	2
5	Banana	Groceries	0.99	4	3
12	Moisturizer	Personal Care	6.00	1	1
9	Shampoo	Personal Care	4.50	2	2
11	Face Wash	Personal Care	4.50	2	2
10	Toothpaste	Personal Care	3.75	4	3

3. Using PARTITION BY category & ORDER BY price DESC

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ProductID	ProductName	Category	Price	RowNum
1	iPhone 15	Electronics	999.99	1
4	Dell Laptop	Electronics	999.99	2
2	Samsung Galaxy S23	Electronics	899.99	3
7	Mango	Groceries	2.50	1
8	Pineapple	Groceries	2.50	2
6	Apple	Groceries	1.29	3
12	Moisturizer	Personal Care	6.00	1
9	Shampoo	Personal Care	4.50	2
11	Face Wash	Personal Care	4.50	3

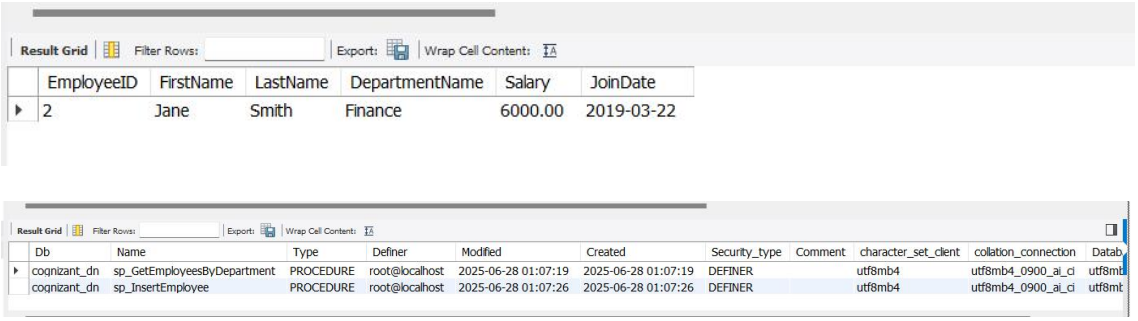
#2. SQL Exercise - Stored procedure

➢ Exercise 1: Create a stored procedure

```
81  DELIMITER //
82
83  CREATE PROCEDURE sp_GetEmployeesByDepartment(IN dept_id INT)
84  BEGIN
85      SELECT
86          E.EmployeeID,
87          E.FirstName,
88          E.LastName,
89          D.DepartmentName,
90          E.Salary,
91          E.JoinDate
92      FROM Employees E
93      JOIN Departments D ON E.DepartmentID = D.DepartmentID
94      WHERE E.DepartmentID = dept_id;
95  END //
96
97  DELIMITER ;
```

```
98   DELIMITER //
99
100  CREATE PROCEDURE sp_InsertEmployee(
101      IN first_name VARCHAR(50),
102      IN last_name VARCHAR(50),
103      IN dept_id INT,
104      IN salary DECIMAL(10,2),
105      IN join_date DATE
106  )
107  BEGIN
108      INSERT INTO Employees (FirstName, LastName, DepartmentID, Salary, JoinDate)
109      VALUES (first_name, last_name, dept_id, salary, join_date);
110  END //
111
112  DELIMITER ;
113  CALL sp_GetEmployeesByDepartment(2);
114  CALL sp_InsertEmployee('Robert', 'Brown', 3, 7500.00, '2023-08-01');
115
116  SHOW PROCEDURE STATUS WHERE Db = 'cognizant_dn';
```

Output :



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

EmployeeID	FirstName	LastName	DepartmentName	Salary	JoinDate
2	Jane	Smith	Finance	6000.00	2019-03-22

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

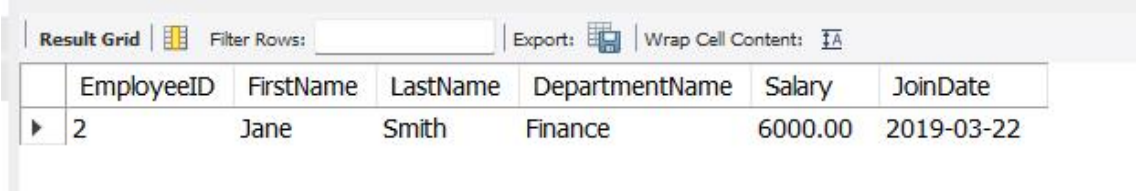
Db	Name	Type	Definer	Modified	Created	Security_type	Comment	character_set_client	collation_connection	Datab
cognizant_dn	sp_GetEmployeesByDepartment	PROCEDURE	root@localhost	2025-06-28 01:07:19	2025-06-28 01:07:19	DEFINER		utf8mb4	utf8mb4_0900_ai_ci	utf8mb
cognizant_dn	sp_InsertEmployee	PROCEDURE	root@localhost	2025-06-28 01:07:26	2025-06-28 01:07:26	DEFINER		utf8mb4	utf8mb4_0900_ai_ci	utf8mb

➢ Exercise 4: Execute a stored procedure

Code :

```
113  CALL sp_GetEmployeesByDepartment(2);
114  CALL sp_InsertEmployee('Robert', 'Brown', 3, 7500.00, '2023-08-01');
115
116  SHOW PROCEDURE STATUS WHERE Db = 'cognizant_dn';
117  CALL sp_GetEmployeesByDepartment(3);
```

Output :



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

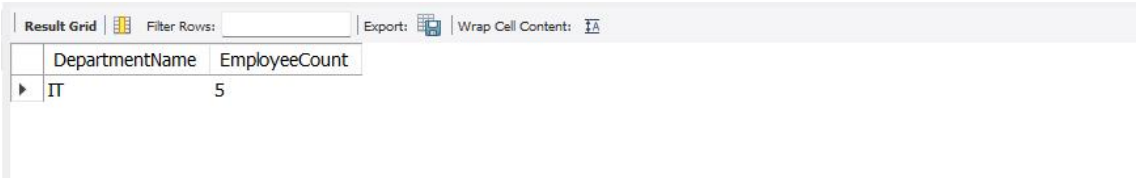
EmployeeID	FirstName	LastName	DepartmentName	Salary	JoinDate
2	Jane	Smith	Finance	6000.00	2019-03-22

➢ Exercise 5: Return data from a stored procedure

Code :

```
118   DELIMITER //
119
120 • CREATE PROCEDURE sp_GetEmployeeCountByDepartment(IN dept_id INT)
121   BEGIN
122       SELECT
123           D.DepartmentName,
124           COUNT(E.EmployeeID) AS EmployeeCount
125       FROM Departments D
126       LEFT JOIN Employees E ON D.DepartmentID = E.DepartmentID
127       WHERE D.DepartmentID = dept_id
128       GROUP BY D.DepartmentName;
129   END //
130
131   DELIMITER ;
132
133 • CALL sp_GetEmployeeCountByDepartment(3);
```

Output :


Result Grid | Filter Rows: | Export: | Wrap Cell Content:

DepartmentName	EmployeeCount
IT	5