Name-Sangram Mandal

Superset Id-6363848

Week-2(Handson-Exercise)

1. Advance SQL Queries

Exercise 1 :Ranking and Window Functions :

Code:

In Script.sql

    DROP TABLE IF EXISTS Products;

    CREATE TABLE Products (

        ProductID INTEGER PRIMARY KEY,

        ProductName TEXT,

        Category TEXT,

        Price REAL

    );

    INSERT INTO Products (ProductName, Category, Price) VALUES

    ('Laptop A', 'Electronics', 1000),

    ('Laptop B', 'Electronics', 950),

    ('Laptop C', 'Electronics', 950),

    ('Phone A', 'Electronics', 700),

    ('Blender', 'Home', 120),

    ('Toaster', 'Home', 80),

    ('Microwave', 'Home', 120),

    ('Vacuum', 'Home', 150),

    ('Shirt', 'Clothing', 40),

    ('Jeans', 'Clothing', 50),

    ('Jacket', 'Clothing', 100),

    ('Shoes', 'Clothing', 100);

    SELECT

        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;

    SELECT \*

    FROM (

        SELECT

            ProductName,

            Category,

            Price,

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

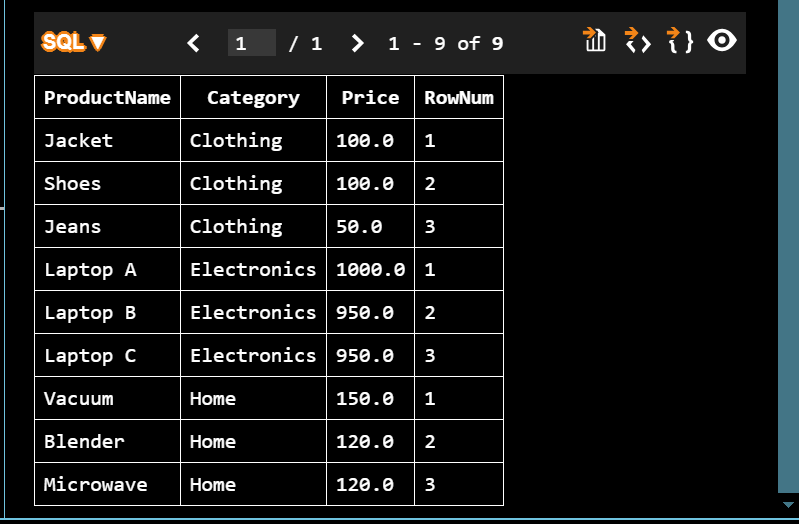
        FROM Products

    ) AS Ranked

    WHERE RowNum <= 3;

Output :





Exercise 1: Create a Stored Procedure

Code :

USE master;

GO

DROP DATABASE IF EXISTS xdb;

GO

CREATE DATABASE xdb;

GO

USE xdb;

GO

CREATE TABLE x1 (

x1id INT PRIMARY KEY,

x1name VARCHAR(100)

);

GO

CREATE TABLE x2 ( -- Employees

x2id INT PRIMARY KEY IDENTITY(1,1),

x2fname VARCHAR(50),

x2lname VARCHAR(50),

x1id INT FOREIGN KEY REFERENCES x1(x1id),

x2sal DECIMAL(10,2),

x2join DATE

);

GO

INSERT INTO x1 (x1id, x1name) VALUES

(1, 'HR'),

(2, 'Finance'),

(3, 'IT'),

(4, 'Marketing');

GO

INSERT INTO x2 (x2fname, x2lname, x1id, x2sal, x2join) VALUES

('John', 'Doe', 1, 5000.00, '2020-01-15'),

('Jane', 'Smith', 2, 6000.00, '2019-03-22'),

('Michael', 'Johnson', 3, 7000.00, '2018-07-30'),

('Emily', 'Davis', 4, 5500.00, '2021-11-05');

GO

GO

CREATE PROCEDURE sp\_x1

@x1id INT

AS

BEGIN

SELECT x2id, x2fname, x2lname, x2sal, x2join

FROM x2

WHERE x1id = @x1id;

END;

GO

GO

CREATE PROCEDURE sp\_x2

@x2fname VARCHAR(50),

@x2lname VARCHAR(50),

@x1id INT,

@x2sal DECIMAL(10,2),

@x2join DATE

AS

BEGIN

INSERT INTO x2 (x2fname, x2lname, x1id, x2sal, x2join)

VALUES (@x2fname, @x2lname, @x1id, @x2sal, @x2join);

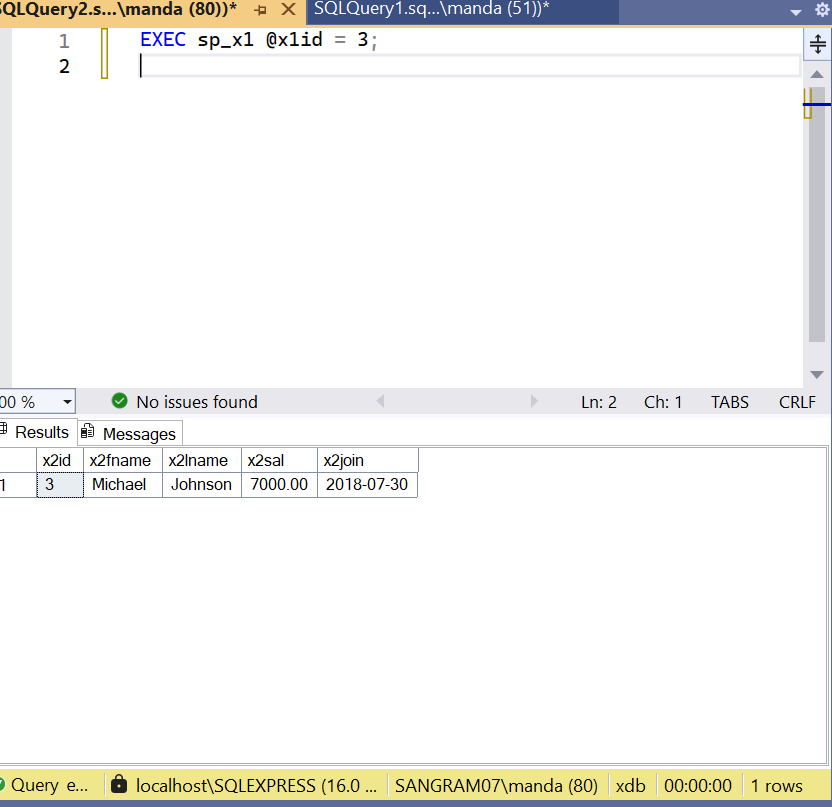
END;

GO

Output :

1. Get employees in IT department:

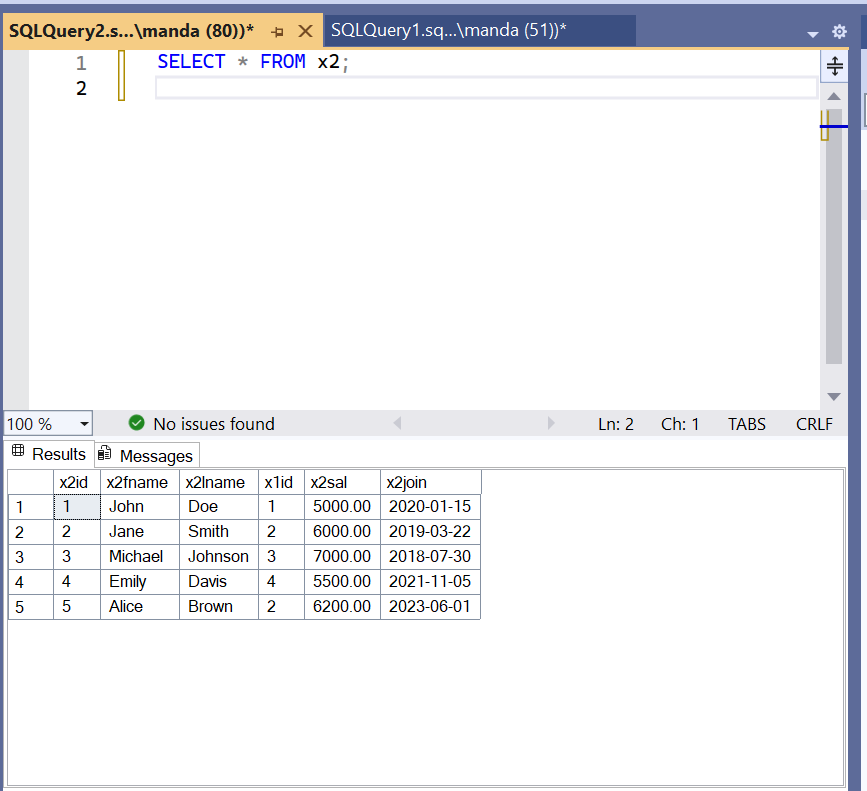
EXEC sp\_x1 @x1id = 3;



1. Insert a new employee:



### Check inserted data:



Exercise 5: Return Data from a Stored Procedure

**Code:**

CREATE PROCEDURE sp\_CountEmployeesByDepartment

@DepartmentID INT

AS

BEGIN

SELECT COUNT(\*) AS TotalEmployees

FROM Employees

WHERE DepartmentID = @DepartmentID;

A screenshot of a computer

AI-generated content may be incorrect.

**Output:**

EXEC sp\_CountEmployeesByDepartment @DepartmentID = 1;

A screenshot of a computer

AI-generated content may be incorrect.

EXEC sp\_CountEmployeesByDepartment @DepartmentID = 5;

**A screenshot of a computer

AI-generated content may be incorrect.**