**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.

**1. Use ROW\_NUMBER() to assign a unique rank within each category.**

SELECT

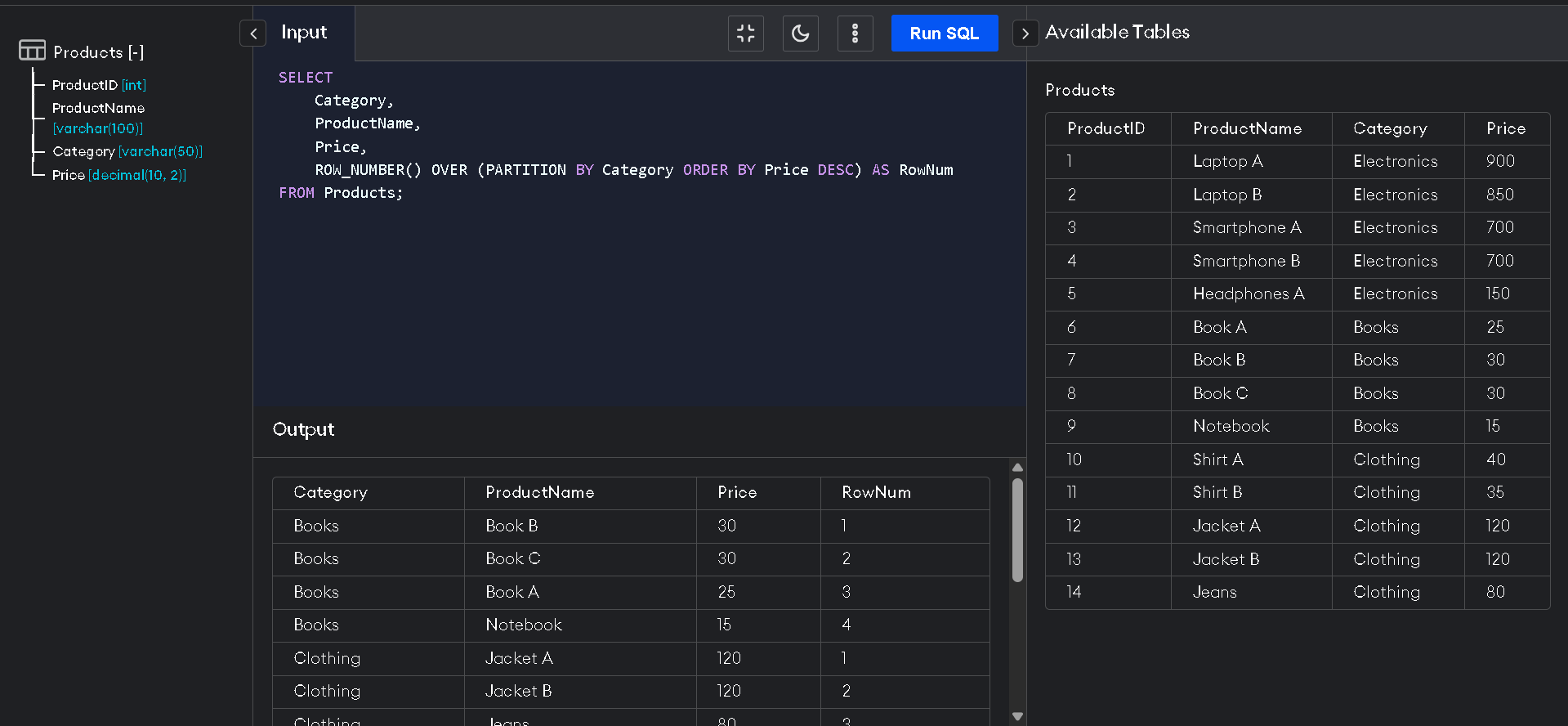
Category,

ProductName,

Price,

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

FROM Products;



**2. Use RANK() and DENSE\_RANK() to compare how ties are handled.**

SELECT

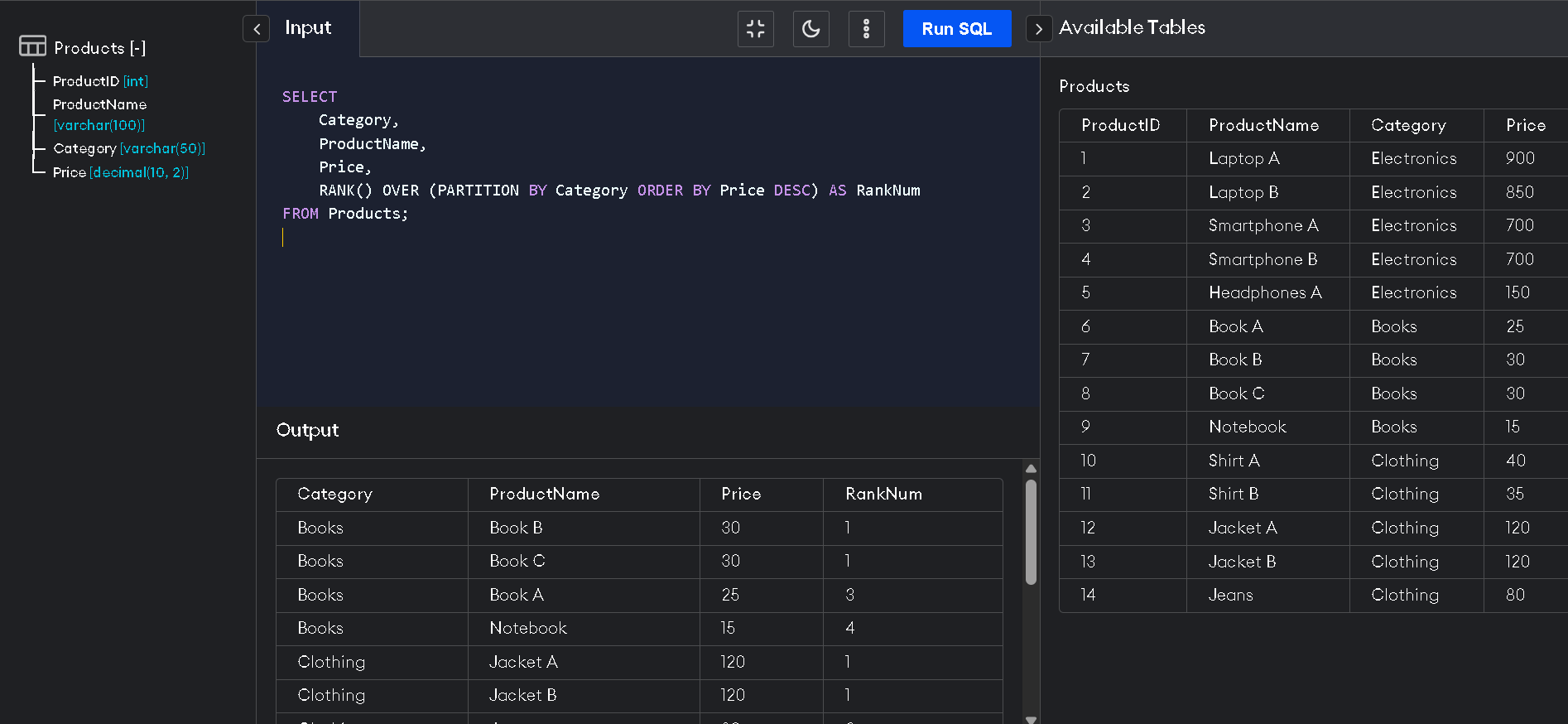
Category,

ProductName,

Price,

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

FROM Products;



**3. Use PARTITION BY Category and ORDER BY Price DESC.**

SELECT

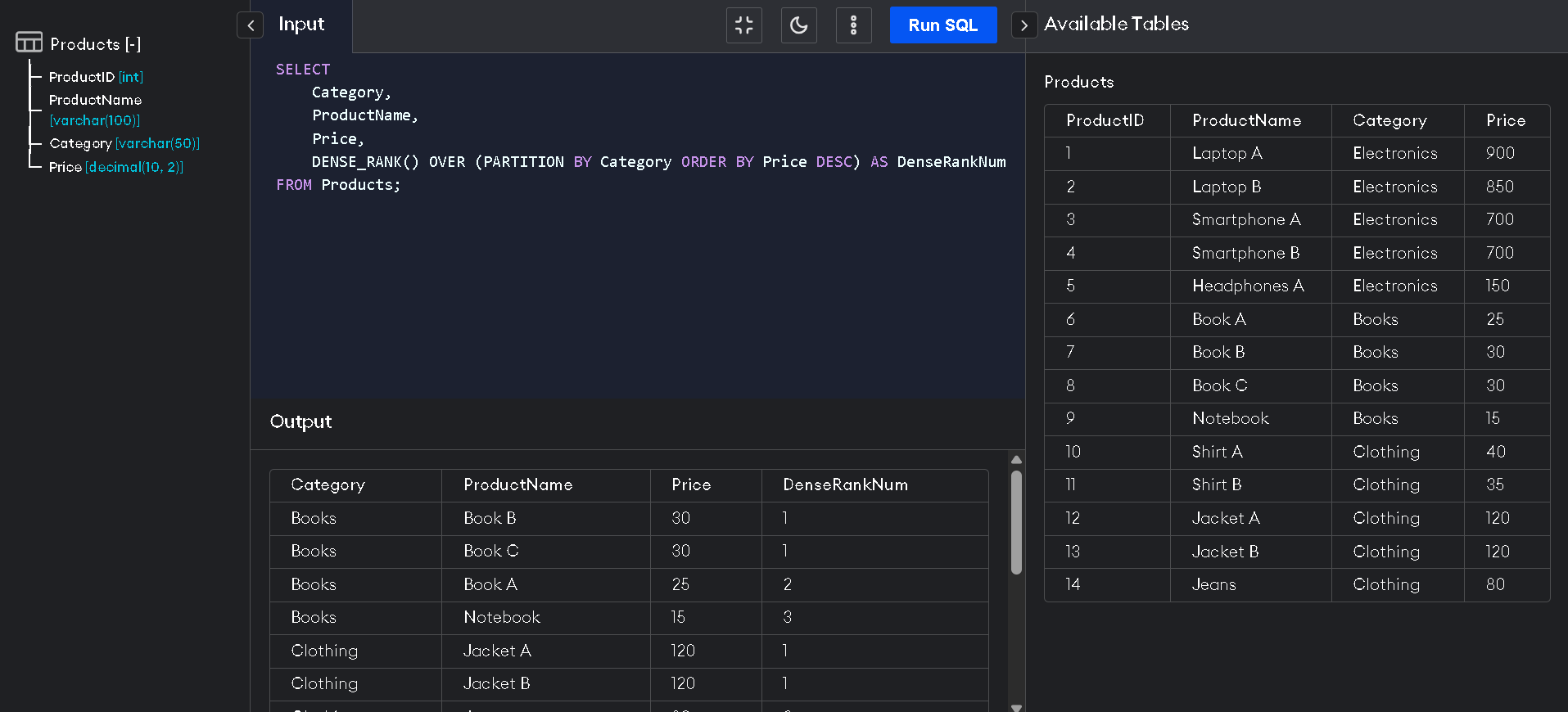
Category,

ProductName,

Price,

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

FROM Products;



**Top 3 Most Expensive Products per Category**

WITH Ranked AS (

SELECT \*,

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

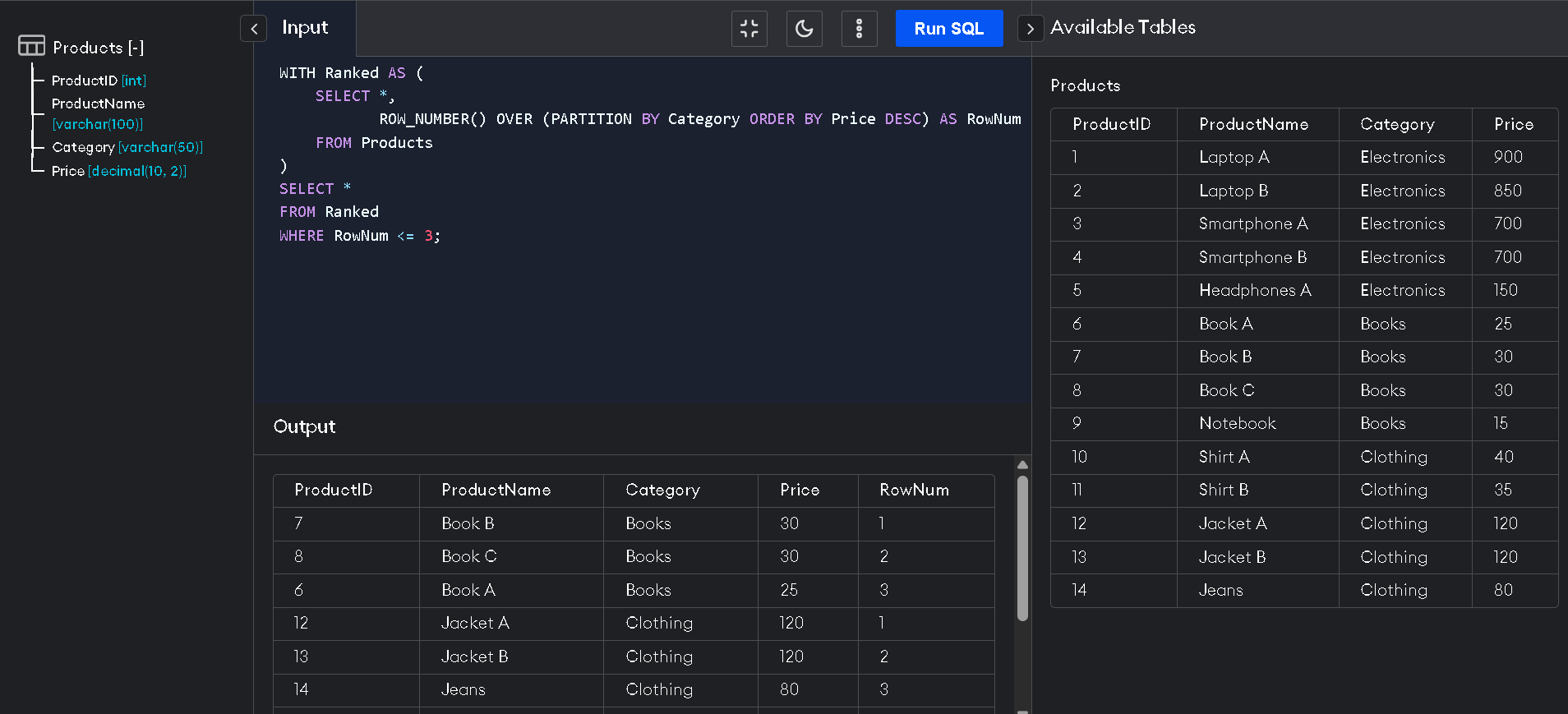
FROM Products

)

SELECT \*

FROM Ranked

WHERE RowNum <= 3;



| **2.Exercise 1: Create a Stored Procedure**  **CREATE A STORED PROCEDURE TO RETRIEVE EMPLOYEES BY DEPARTMENT** |
| --- |

DELIMITER $$

CREATE PROCEDURE GetEmployeesByDepartment(IN dept\_id INT)

BEGIN

SELECT

EmployeeID, FirstName, LastName, DepartmentID, Salary, JoinDate

FROM

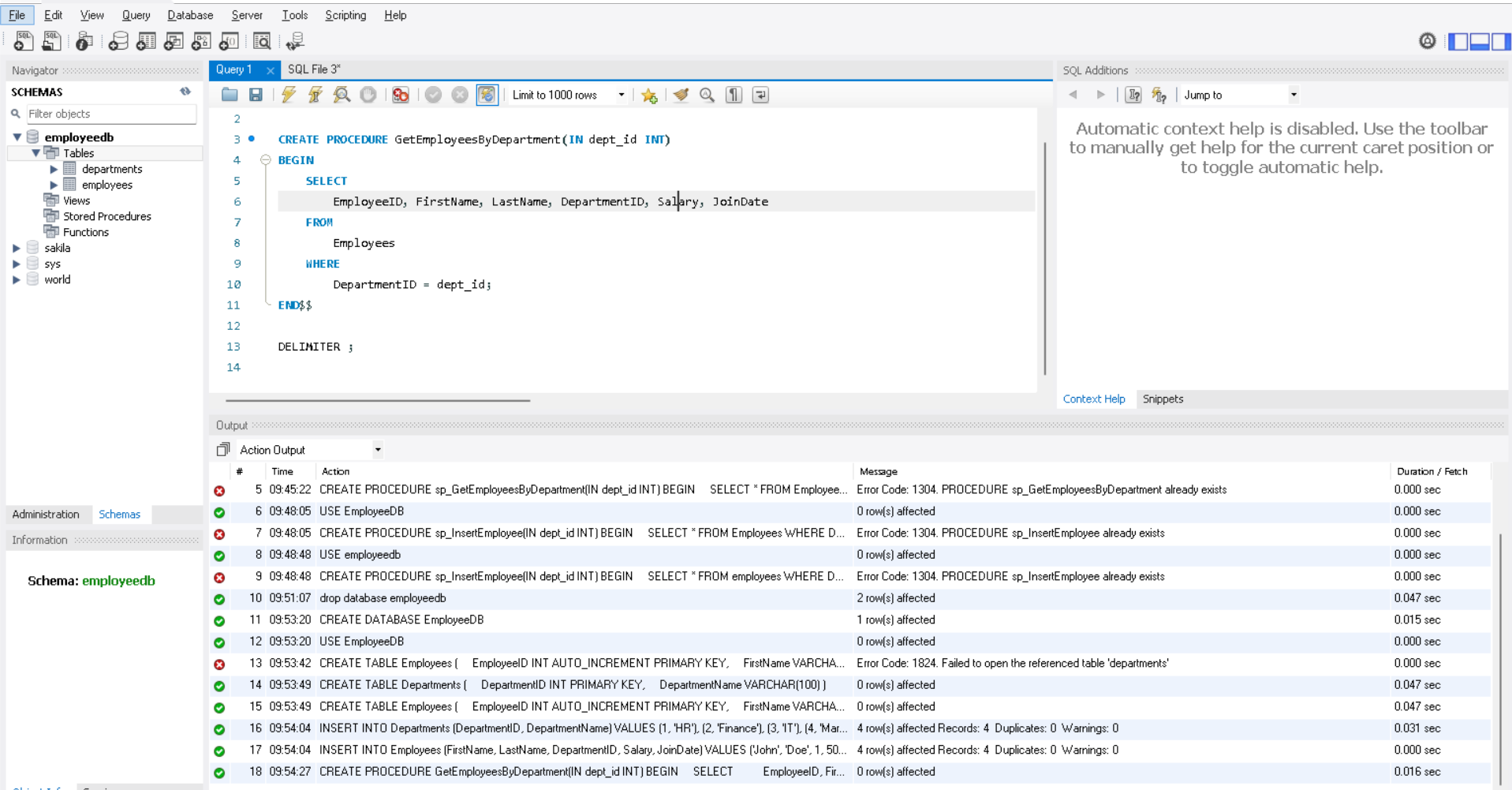
Employees

WHERE

DepartmentID = dept\_id;

END$$

DELIMITER ;



**Create a Stored Procedure to Insert an Employee**

DELIMITER $$

CREATE PROCEDURE sp\_InsertEmployee(

IN p\_FirstName VARCHAR(50),

IN p\_LastName VARCHAR(50),

IN p\_DepartmentID INT,

IN p\_Salary DECIMAL(10,2),

IN p\_JoinDate DATE

)

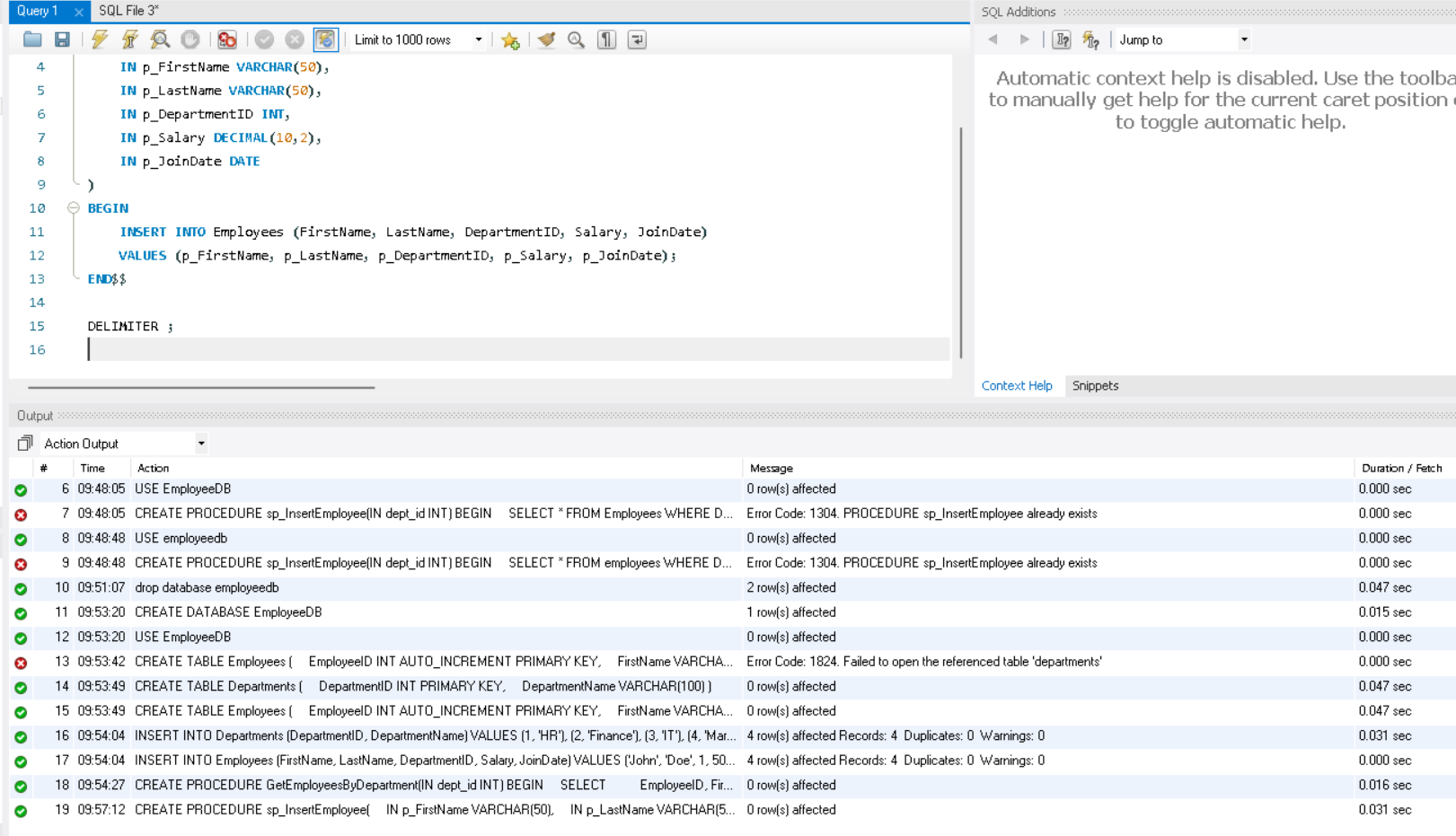
BEGIN

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

VALUES (p\_FirstName, p\_LastName, p\_DepartmentID, p\_Salary, p\_JoinDate);

END$$

DELIMITER ;



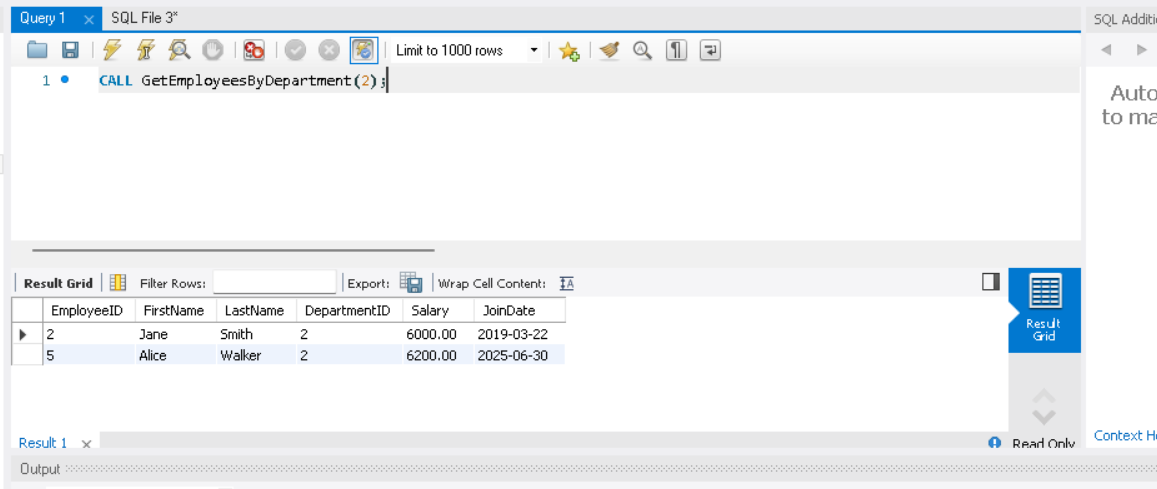
**Insert Employee**

CALL sp\_InsertEmployee('Alice', 'Walker', 2, 6200.00, '2025-06-30');



**Get Employees by Department**

CALL GetEmployeesByDepartment(2);

****

**3.Exercise 5: Return Data from a Stored Procedure**

**Stored Procedure: Count Employees by Department**

DELIMITER $$

CREATE PROCEDURE CountEmployeesByDepartment(IN dept\_id INT)

BEGIN

SELECT

COUNT(\*) AS TotalEmployees

FROM

Employees

WHERE

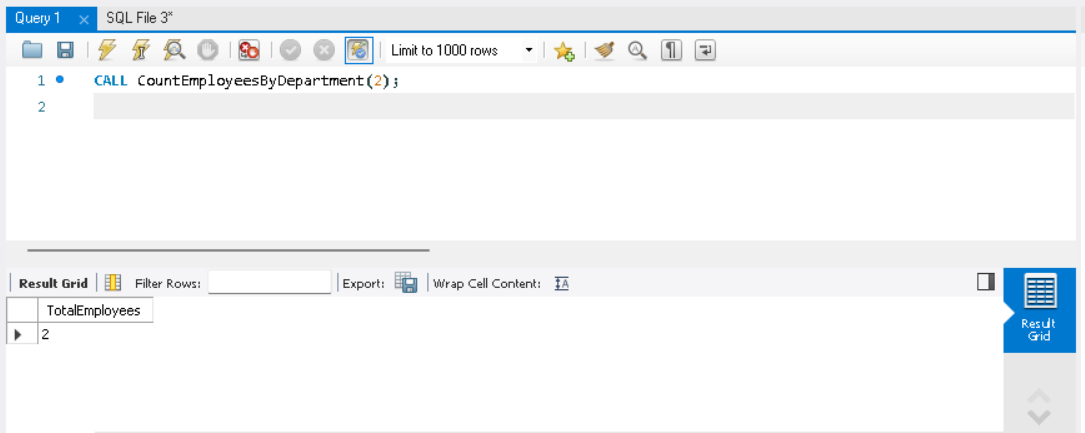
DepartmentID = dept\_id;

END$$

DELIMITER ;

****

CALL CountEmployeesByDepartment(2);

****