**Exercise 1: Ranking and Window Functions**

CODE:

SELECT

ProductID,

ProductName,

Category,

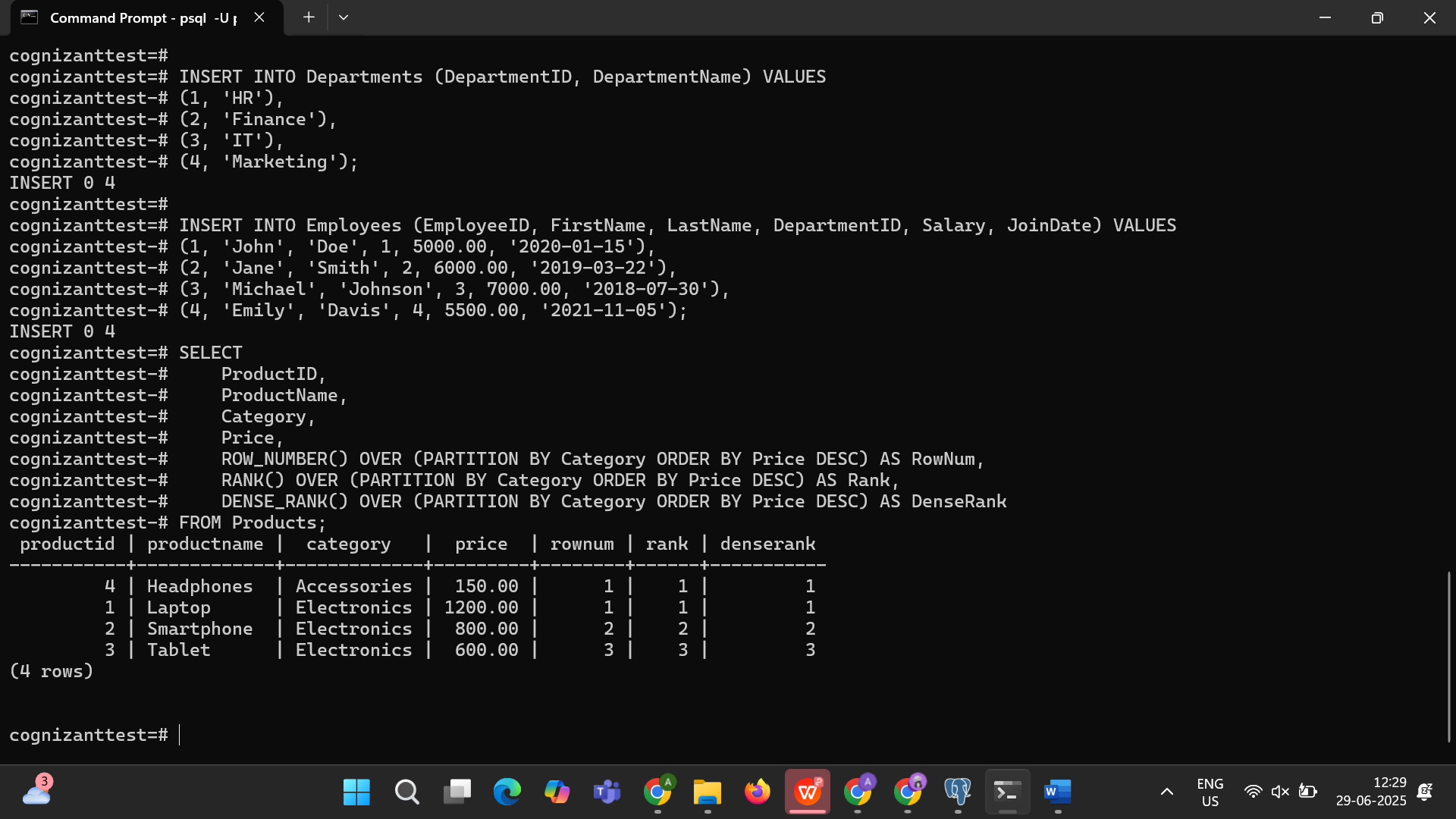
Price,

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

RANK() OVER (PARTITION BY Category ORDER BY Price DESC) AS Rank,

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

FROM Products;

OUTPUT:  


**Exercise 1: Stored Procedure — Get Employees by Department**

CODE:

CREATE OR REPLACE FUNCTION get\_employees\_by\_department(dept\_id INT)

RETURNS TABLE(EmployeeID INT, FirstName VARCHAR, LastName VARCHAR, Salary DECIMAL, JoinDate DATE) AS $$

BEGIN

RETURN QUERY

SELECT

Employees.EmployeeID,

Employees.FirstName,

Employees.LastName,

Employees.Salary,

Employees.JoinDate

FROM Employees

WHERE Employees.DepartmentID = dept\_id;

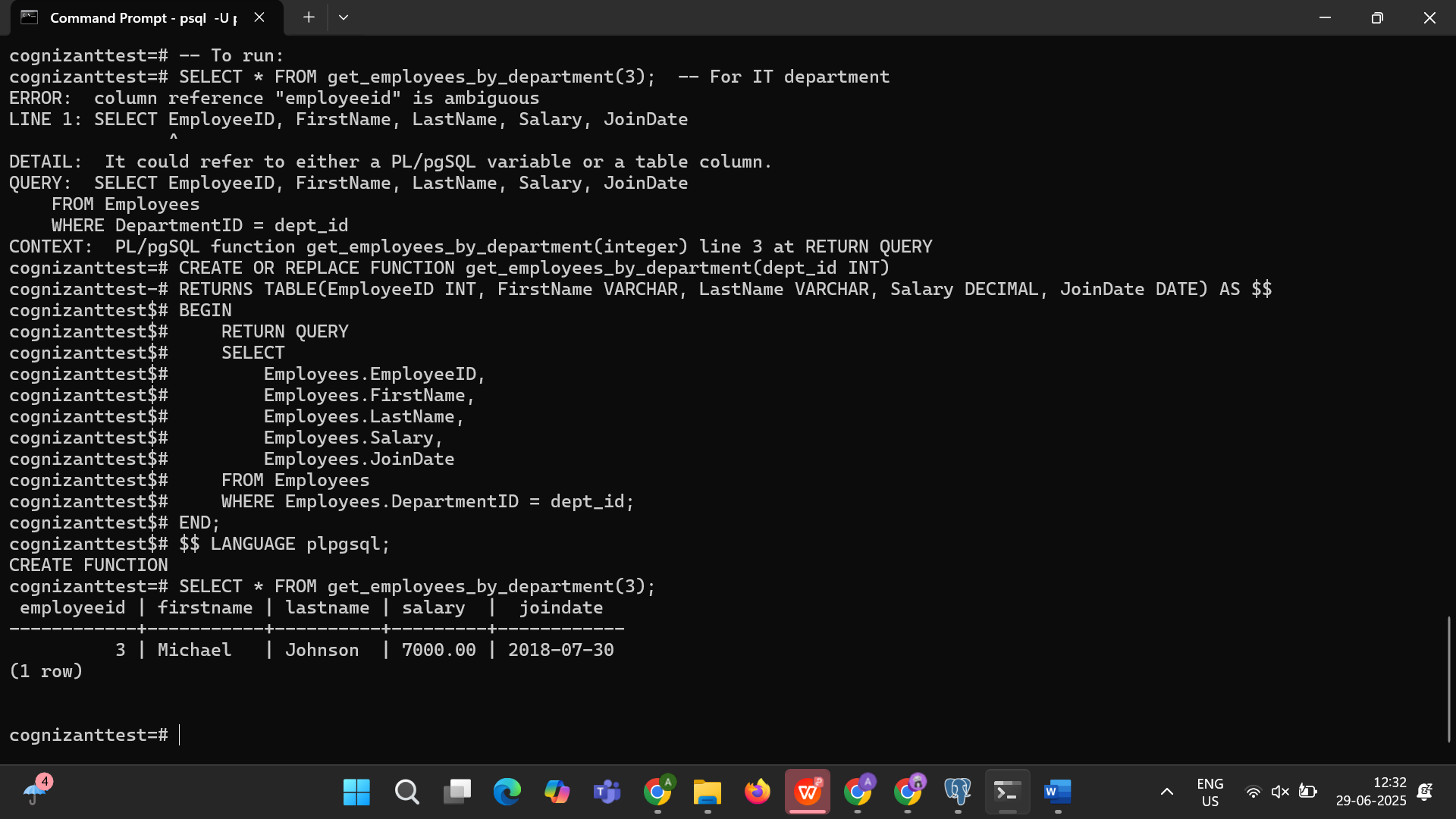
END;

$$ LANGUAGE plpgsql;

-- To run:

SELECT \* FROM get\_employees\_by\_department(3); -- For IT department

OUTPUT:



Exercise 5: Stored Procedure — Employee Count by Department

CODE:

CREATE OR REPLACE FUNCTION get\_department\_employee\_count(dept\_id INT)

RETURNS INT AS $$

DECLARE

emp\_count INT;

BEGIN

SELECT COUNT(\*) INTO emp\_count FROM Employees WHERE DepartmentID = dept\_id;

RETURN emp\_count;

END;

$$ LANGUAGE plpgsql;

-- To run:

SELECT get\_department\_employee\_count(2) AS employee\_count; -- For Finance department

OUTPUT:

