Lab 10: SQL Sub Queries

CS355/CE373 Database Systems
Fall 2024

Dhanani School of Science and Engineering
Habib University

Contents

1 Instructions

- This lab will contribute 1% towards the final grade.
- The deadline to submit this lab is at the end of your lab.
- The lab must be submitted online via CANVAS. The SQL file should be named as *Lab_10_aa01234.sql* where *aa01234* will be replaced with your student id. ***Files which don't follow the appropriate naming convention will not be graded***.

1.1 Marking scheme

This lab will be marked out of 100.

- 50 Marks are for completion of the lab.
- 10 Marks are for filling the feedback form within the lab timings.
- 40 Marks are for progress and attendance during the lab.

1.2 Late submission policy

No late submissions are allowed.

2 Objective

This lab activity is prepared on Northwind Sample Database of SQL Server. The database will be analyzed for the following SQL constructs:

- Top
- Sub Queries

Note: You are only allowed to use Sub Queries for this lab.

3 Query Syntax Examples

- **Sub Queries**
  Select * From Orders
  Where EmployeeID in (
  Select Top 3 EmployeeID
  From Orders O
  Group By EmployeeID
  Order By Count(*) Desc)
- **SQL TOP**
  SELECT TOP 3 E.FirstName + ' ' + E.LastName AS EmployeeName, Year(O.OrderDate) AS [Year], count(*) AS 'Number of Orders'
  FROM Orders O
  INNER JOIN Employees E
  ON O.EmployeeID=E.EmployeeID
  GROUP BY E.FirstName + ' ' + E.LastName, Year(O.OrderDate)
  ORDER BY COUNT(*) DESC
- **SQL Case**

```
SELECT E.EmployeeID,
CASE
WHEN DateDiff(day,GETDATE(), E.HireDate) ¿ 10 THEN 'senior'
WHEN Datediff(day,E.HireDate, GETDATE()) between 10 and 5 THEN 'junior'
ELSE 'fresher'
END AS TimeHere
FROM Employees E
```

4 Exercises

The ERD Diagram for the Northwind Database is as shown in Figure 1.

1. **Find the employee who processed the first order placed in year 1998.**
   Output: Employee ID.
   Result contains 1 row.

2. **Select all employees who work directly under the top manager of the company.**
   Output: EmployeeID.
   Result contains 5 rows.

3. **Select all employees who are assigned to territories in 'Western' and 'Eastern' regions from Region Table.**
   Result contains 6 rows.

4. **Select all Customers and Suppliers belonging to 'Germany'.**
   Output: ContactName.
   Result contains 14 rows.

5. **Find the 3rd most expensive product in the database.**
   Output: ProductName.
   Result contains 1 row.

6. **Select all employees and their Seniority level**
   - Seniority level = 3 if employee has been with the company for more than 5 years.
   - Seniority level = 2 if employee has been with the company from 3-5 years.
   - Seniority level = 1 if employee has been with the company for < 3 years

   Output: EmployeeID, SeniorityLevel. *Result contains 9 rows.*

7. **List all products and their types which shows if they are 'Costly' (unit price $>$ 80), 'Economical' (unit price between 30 and 80) or 'Cheap' (Unit price $<$ 30).**
   Output: ProductName, Types.
   Result contains 77 rows.

8. **List all products and their trends based on the number of orders placed in the year 1997. If no. of orders >= 50 Trend = Customer favourite Else if 30 <= no. of orders <= 49 Trend = Trending. Else if 10 <= no. of orders <= 29 Trend = on the rise. Else trend = not popular.**
   Output: ProductName, Trend.
   Result contains 77 rows.

9. **Find the total number of orders placed by each customer.**
   Output: CustomerID, OrderCount.
   Result contains 91 rows.

10. **Retrieve customers who have placed orders for products with a price higher than the average price of all products.**
    Output: CustomerID.
    Result contains 86 rows.

11. **Find the customers who have placed orders for products from the same category as 'Chai'.**
    Output: Customers.ContactName
    Result contains 83 rows.

12. **Find the customer who has placed the highest total number of orders.**
    Output: ContactName, NumberOfOrders
    Result contains 1 row.

13. **List all the customers who have placed an order for the most expensive product.**
    Output: ContactName.
    Result contains 12 rows.

14. **Find the average number of products in each order.**
    Output: AverageProductsPerOrder.
    Result contains 1 row.

15. **Find the categories where the average product price is higher than the overall average product price.**
    Output: CategoryName.
    Result contains 3 rows.

16. **Find the product which has the second highest price.**
    Output: ProductName, UnitPrice.
    Result contains 1 row.

17. **Find the average order amount for customers from France.**
    Output: AverageOrderAmount
    Result contains 1 row.

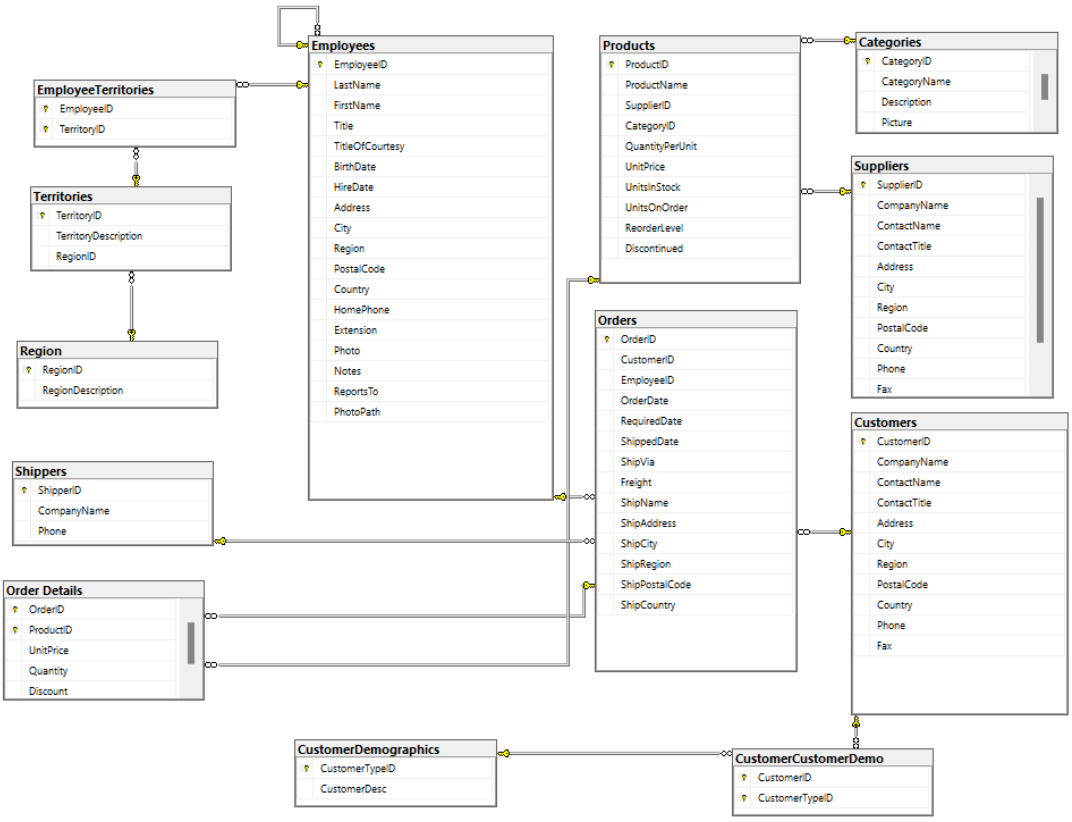

Figure 1: Northwind Database ERD