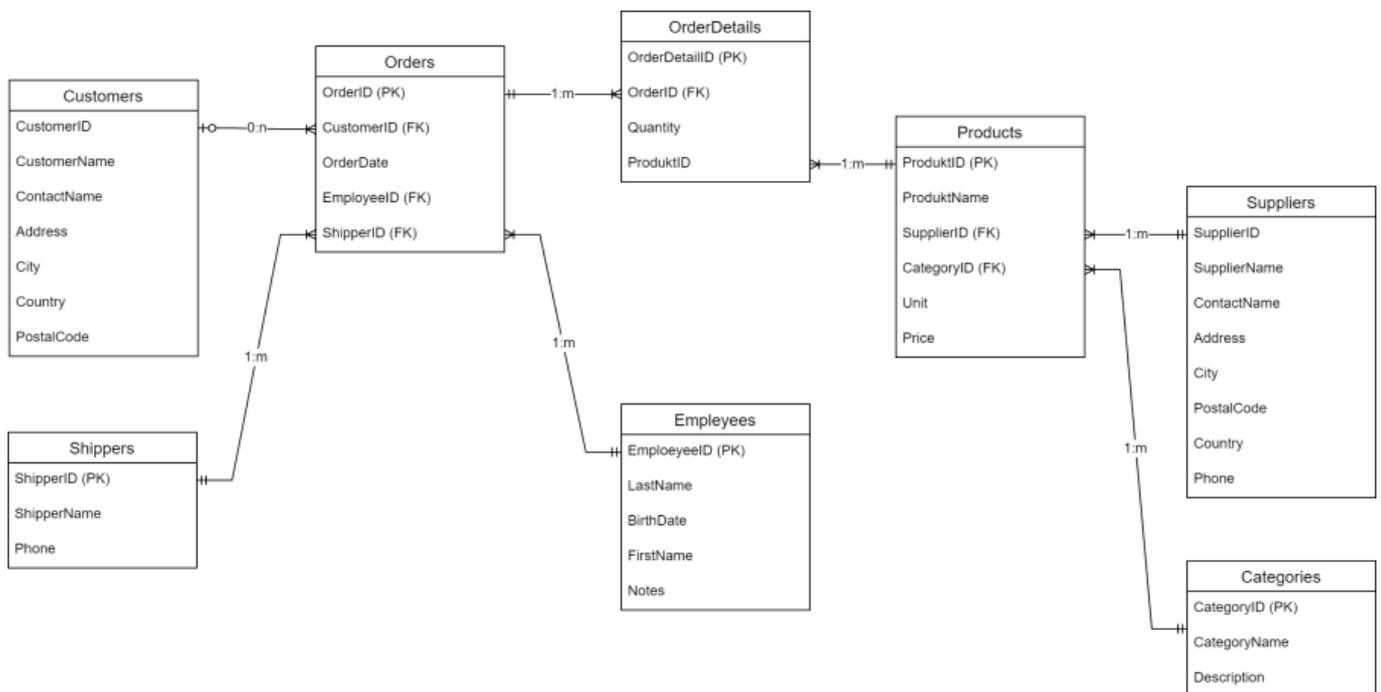


LU05.S02 - SQL-DQL: Select from multiple tables

Case studies / Assignments

The following ERD describes a order database for a a shop.



Assignments

The general assignment is to develop DQL commands that matches the requirements below:

Assignment A

We need a list with product details as following: Name of the supplier, Supplier phone, Product ID, Name of the product, units, price. The list must be sorted by SupplierName in descending order.

```

SELECT s.SupplierName, s.SupplierID, s.phone, p.ProductID, p.ProductName,
p.Unit, p.Price
FROM Suppliers s, Products p
WHERE p.SupplierID = s.SupplierID
ORDER BY s.SupplierName DESC;
  
```

Assignment B

Extend the the result of the Select statement from assignment a), so that it must contain only products of the supplier „Leka Trading“.

There are two possible solutions.

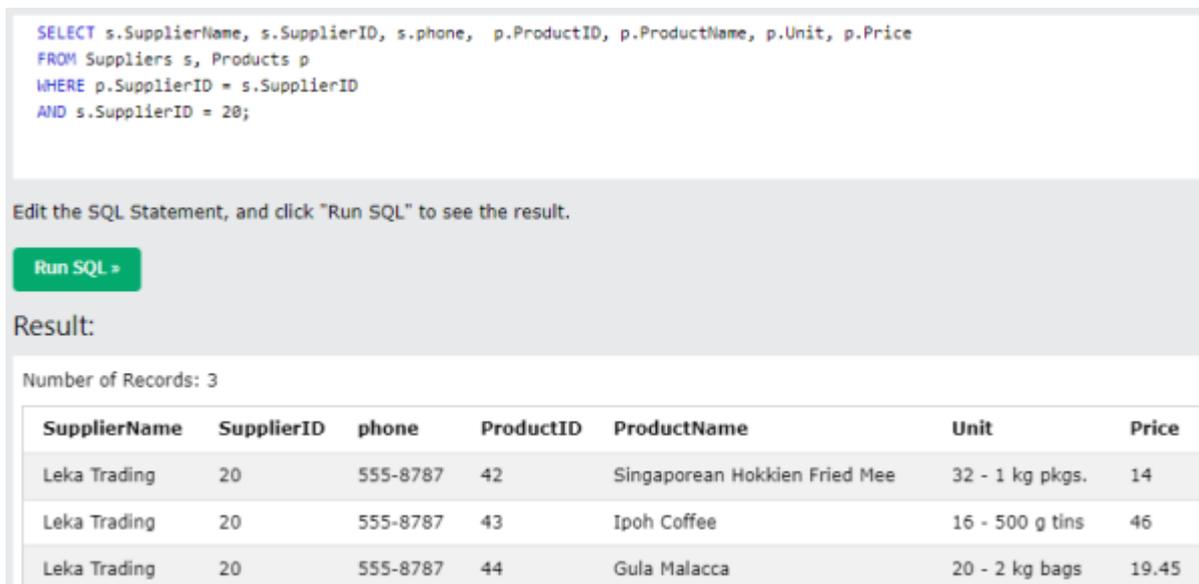
B1) The first assignment is to search for the ID (SupplierID = 20) of the supplier and take that ID into our SELECT as follows:

```
SELECT s.SupplierName, s.SupplierID, s.phone, p.ProductID, p.ProductName,
p.Unit, p.Price
FROM Suppliers s, Products p
WHERE p.SupplierID = s.SupplierID
AND s.SupplierID = 20;
```

B2) The second solution takes the name directly into the SELECT statement:

```
SELECT s.SupplierName, s.SupplierID, s.phone, p.ProductID, p.ProductName,
p.Unit, p.Price
FROM Suppliers s, Products p
WHERE p.SupplierID = s.SupplierID
AND s.SupplierName = 'Leka Trading';
```

The result set is in both cases, as the figure below shows:



The screenshot shows a SQL query execution interface. At the top, the SQL query is displayed: `SELECT s.SupplierName, s.SupplierID, s.phone, p.ProductID, p.ProductName, p.Unit, p.Price FROM Suppliers s, Products p WHERE p.SupplierID = s.SupplierID AND s.SupplierID = 20;`. Below the query, there is a button labeled "Run SQL +". Underneath the button, the text "Result:" is followed by "Number of Records: 3". A table with 7 columns (SupplierName, SupplierID, phone, ProductID, ProductName, Unit, Price) and 3 rows of data is shown. The rows represent products from Leka Trading with SupplierID 20.

| SupplierName | SupplierID | phone | ProductID | ProductName | Unit | Price |
|--------------|------------|----------|-----------|-------------------------------|-----------------|-------|
| Leka Trading | 20 | 555-8787 | 42 | Singaporean Hokkien Fried Mee | 32 - 1 kg pkgs. | 14 |
| Leka Trading | 20 | 555-8787 | 43 | Ipoh Coffee | 16 - 500 g tins | 46 |
| Leka Trading | 20 | 555-8787 | 44 | Gula Malacca | 20 - 2 kg bags | 19.45 |

Assignment C

We would like to know what products the customer „Hanari Carnes“ has ordered in the past. Sort the list by the quantity. In detail we require the following data: CustomerID, CustomerName, OrderID, OrderDate, Quantity, ProductName, CategoryName

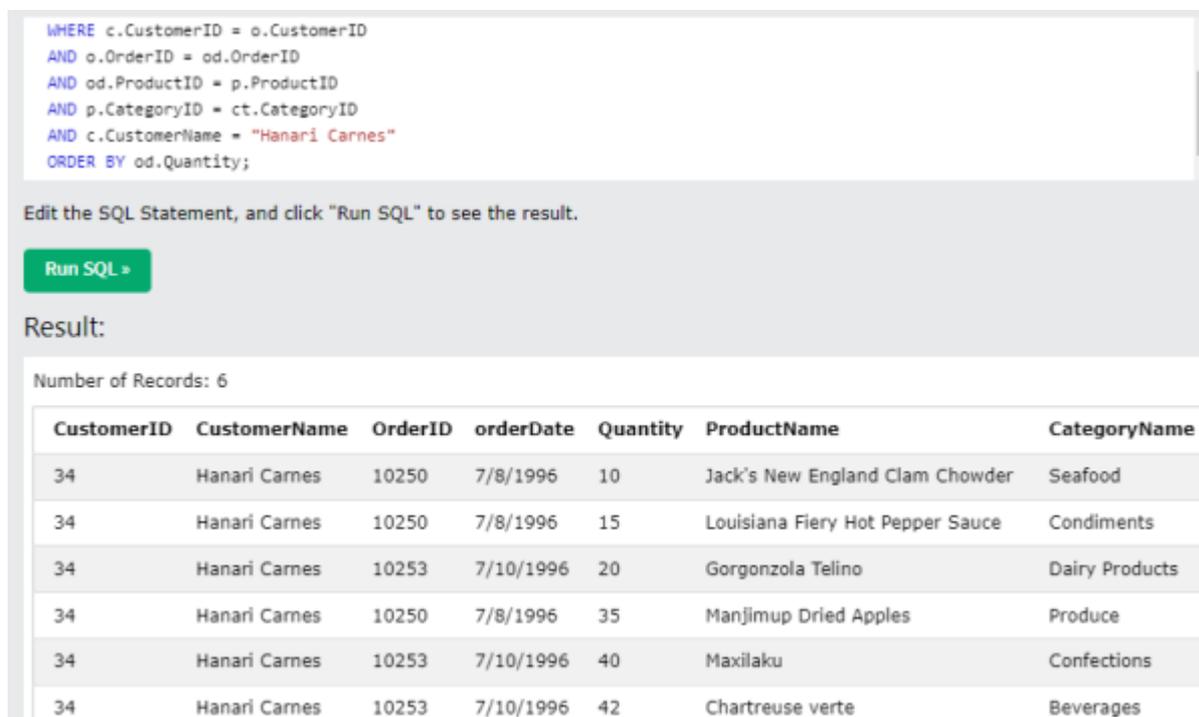
C1) Like in assignment b) here a two approaches possible: 1. find the customerID (34) and select according to the customer id.

```
SELECT c.CustomerID, c.CustomerName, o.OrderID, o.orderDate, od.Quantity,
p.ProductName, ct.CategoryName
FROM customers c, orders o, OrderDetails od, Products p, Categories ct
WHERE c.CustomerID = o.CustomerID
AND o.OrderID = od.OrderID
AND od.ProductID = p.ProductID
AND p.CategoryID = ct.CategoryID
AND c.CustomerID = 34
ORDER BY od.Quantity;
```

C2) The second approach, as in B2), is to filter the result set directly by customer name.

```
SELECT c.CustomerID, c.CustomerName, o.OrderID, o.orderDate, od.Quantity,
p.ProductName, ct.CategoryName
FROM customers c, orders o, OrderDetails od, Products p, Categories ct
WHERE c.CustomerID = o.CustomerID
AND o.OrderID = od.OrderID
AND od.ProductID = p.ProductID
AND p.CategoryID = ct.CategoryID
AND c.CustomerName = "Hanari Carnes"
ORDER BY od.Quantity;
```

The result set is in both cases as shown in the following figure.



The screenshot shows a SQL query execution interface. The query is:

```
WHERE c.CustomerID = o.CustomerID
AND o.OrderID = od.OrderID
AND od.ProductID = p.ProductID
AND p.CategoryID = ct.CategoryID
AND c.CustomerName = "Hanari Carnes"
ORDER BY od.Quantity;
```

Below the query is a button labeled "Run SQL >".

The result is displayed as a table with 6 records:

| CustomerID | CustomerName | OrderID | orderDate | Quantity | ProductName | CategoryName |
|------------|---------------|---------|-----------|----------|----------------------------------|----------------|
| 34 | Hanari Carnes | 10250 | 7/8/1996 | 10 | Jack's New England Clam Chowder | Seafood |
| 34 | Hanari Carnes | 10250 | 7/8/1996 | 15 | Louisiana Fiery Hot Pepper Sauce | Condiments |
| 34 | Hanari Carnes | 10253 | 7/10/1996 | 20 | Gorgonzola Telino | Dairy Products |
| 34 | Hanari Carnes | 10250 | 7/8/1996 | 35 | Manjimup Dried Apples | Produce |
| 34 | Hanari Carnes | 10253 | 7/10/1996 | 40 | Maxilaku | Confections |
| 34 | Hanari Carnes | 10253 | 7/10/1996 | 42 | Chartreuse verte | Beverages |

Solution

[Lösung](#)

Vocabulary

| English | German |
|--------------|-----------------|
| respectively | beziehungsweise |
| assignment | Auftrag |



Volkan Demir

From:
<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:
<https://wiki.bzz.ch/modul/m290/learningunits/lu04/loesungen/l02?rev=1727435001>

Last update: **2024/09/27 13:03**

