

Lösungen

Modul 1

1. Liste alle Einträge aus der Tabelle Customers auf.

```
SELECT * FROM Customers
```

2. Liste alle Kunden (CompanyName) und deren Ansprechpartner (ContactName) auf

```
SELECT CompanyName, ContactName FROM Customers
```

3. Liste zu allen Bestellungen den Netto- und Bruttowert der Frachtkosten auf. Ausgabe: OrderID, Freight, (Brutto) (MwST DE)

```
SELECT OrderID, Freight, Freight * 1.19 FROM Orders
```

4. Füge nun der Brutto-Spalte aus 3. Einen Alias-Namen hinzu

```
SELECT OrderID, Freight, Freight * 1.19 AS Brutto FROM Orders
```

5. In der Ausgabe sollen die letzten 3 Stellen der Telefonnummer der Kunden durch ,xxx' ersetzt werden

```
SELECT STUFF(Phone, LEN(Phone)-2, 3, 'xxx') FROM Customers
```

Modul 2

1. Alle Kunden, die aus Spanien sind (Ausgabe: CompanyName, Country)

```
SELECT CompanyName, Country FROM Customers WHERE Country = 'Spain'
```

2. Alle Kunden, die aus Spanien oder Deutschland sind (Ausgabe: CompanyName, Country)

```
SELECT CompanyName, Country FROM Customers WHERE Country = 'Spain' OR Country = 'Germany'
```

3. Alle Produkte, die nicht mehr verkauft werden (Tabelle Products, Discontinued = 1)

```
SELECT ProductName FROM Products WHERE Discontinued = 1
```

4. Alle Kunden, die aus Schweden und Mexiko sind

```
SELECT CompanyName FROM Customers WHERE Country = 'Sweden' AND Country = 'Mexico'
```

5. Alle Kunden, die einen Vokal als vorletzten Buchstaben im Firmennamen haben

```
SELECT CompanyName FROM Customers WHERE CompanyName LIKE '%[aeiou]_'
```

6. Suche alle Kunden, die im Firmennamen ein e haben, der 2te Buchstabe ein u ist und am Ende einen Buchstaben zwischen b-g haben

```
SELECT * FROM Customers WHERE CompanyName LIKE '_e%[b-g]' AND CompanyName LIKE '%e%'
```

7. Suche nach Kunden, die ein ' ' im Namen haben

```
SELECT * FROM Customers WHERE CompanyName LIKE '% ' %'
```

8. Alle Bestellungen mit Frachtkosten (Freight) zwischen 100 und 200

```
SELECT * FROM Orders WHERE Freight BETWEEN 100 AND 200
```

9. Alle Kunden aus Ländern, die mit A beginnen

```
SELECT * FROM Customers WHERE Country LIKE 'A%'
```

10. Liste alle Kunden auf, bei denen in der Regions-Spalte nichts angegeben ist

```
SELECT * FROM Customers WHERE Region IS NULL
```

Modul 3

1. Liste alle Bestellungen (OrderID) mit den dazugehörigen Kunden (CompanyName) auf

```
SELECT c.CompanyName, o.OrderID FROM Orders o
INNER JOIN Customers c ON o.CustomerID = c.CustomerID
```

2. Gib alle Bestellungen (OrderID) mit den zugewiesenen ProduktIDs (ProductID) und den AngestelltenIDs (EmployeeID) aus, die diese Bestellungen abgewickelt haben und sortiere die Ausgabe nach AngestelltenID und BestellungsID

```
SELECT o.EmployeeID, od.OrderID, od.ProductID FROM Orders o
INNER JOIN [Order Details] od ON o.OrderID = od.OrderID
ORDER BY o.EmployeeID, od.OrderID
```

3. Nun soll die Abfrage (Nr 2) so abgeändert werden, dass anstatt der IDs die Namen (LastName, FirstName) der Angestellten ausgegeben werden.

```
SELECT o.EmployeeID, od.OrderID, od.ProductID FROM Orders o
INNER JOIN [Order Details] od ON o.OrderID = od.OrderID
INNER JOIN Employees e ON o.EmployeeID = e.EmployeeID
ORDER BY o.EmployeeID, od.OrderID
```

```
SELECT e.LastName, e.FirstName, od.OrderID, od.ProductID FROM Employees e
INNER JOIN Orders o ON e.EmployeeID = o.EmployeeID
INNER JOIN [Order Details] od ON o.OrderID = od.OrderID
ORDER BY o.EmployeeID, od.OrderID
```

4. Nun geben sie noch anstelle der ProduktIDs die jeweiligen Namen (ProductName) der Produkte aus.

```
SELECT e.LastName, e.FirstName, od.OrderID, od.ProductID FROM Employees e
INNER JOIN Orders o ON e.EmployeeID = o.EmployeeID
INNER JOIN [Order Details] od ON o.OrderID = od.OrderID
INNER JOIN Products p ON od.ProductID = p.ProductID
ORDER BY o.EmployeeID, od.OrderID
```

5. Welche Produkte hat Nancy Davolio verkauft? Falls bestimmte Produkte mehrfach vorkommen, reduzieren sie die Liste um die Duplikate.

```
SELECT DISTINCT e.LastName, e.FirstName, p.ProductName FROM Employees e
INNER JOIN Orders o ON e.EmployeeID = o.EmployeeID
INNER JOIN [Order Details] od ON o.OrderID = od.OrderID
INNER JOIN Products p ON od.ProductID = p.ProductID
WHERE LastName = 'Davolio' AND FirstName = 'Nancy'
```

6. Welcher Angestellte verkaufte an QUICK-Stop Seafood oder Produce?

```
SELECT Distinct LastName, FirstName FROM Categories ca
INNER JOIN Products p ON ca.CategoryID = p.CategoryID
INNER JOIN [Order Details] od ON p.ProductID = od.ProductID
INNER JOIN Orders o ON od.OrderID = o.OrderID
INNER JOIN Employees e ON o.EmployeeID = e.EmployeeID
INNER JOIN Customers c ON o.CustomerID = c.CustomerID
WHERE CompanyName = 'QUICK-Stop' AND (CategoryName = 'Seafood' OR CategoryName = 'Produce')
```

Modul 4

Erstelle die Tabelle Person mit einer Spalte als Nachname, Vorname, Alter, Adresse, PLZ

```
CREATE TABLE Person (
    Nachname varchar(50),
    Vorname varchar(50),
    [Alter] smallint,
    Adresse varchar(50),
    PLZ varchar(10));
```

Füge der Tabelle die folgenden Werte hinzu:

- Becker, Michael, 23, Kuratorenweg 10, 59384
- Becker, Franziska, 26, Parkstraße 46, 49586
- Müller, Josef, 45, Theaterplatz 14, 39499

```
INSERT INTO Person VALUES
('Becker', 'Michael', 23, 'Kuratorenweg 10', 59384),
('Becker', 'Franziska', 26, 'Parkstraße 46', 49586),
('Müller', 'Josef', 45, 'Theaterplatz 14', 39499);
```

Alle Personen sollen um ein Jahr altern

```
UPDATE Person SET [Alter] += 1
```

Josef Müller zieht um. Seine neue Adresse ist: Parkallee, 67, 55874

```
UPDATE Person
SET Adresse = 'Parkallee 67', PLZ = 55874
WHERE Nachname = 'Müller'
```

Lösche Becker Michael aus der Tabelle

```
DELETE FROM Person WHERE Nachname = 'Becker'
```

Es sollen alle Employees auch in der Person-Tabelle stehen.

```
INSERT Person
SELECT e.LastName, e.FirstName, DATEDIFF(year, e.BirthDate, CURRENT_TIMESTAMP)
AS Age, e.Address, PostalCode FROM Employees e
```

Ändere die Frachtkosten um +10% bei allen Bestellungen die seafood bestellt haben

```
UPDATE Orders SET Freight *= 1.1
-- SELECT freight
FROM Categories c
INNER JOIN Products p ON c.CategoryID = p.CategoryID
INNER JOIN [Order Details] od ON p.ProductID = od.ProductID
INNER JOIN Orders o ON od.OrderID = o.OrderID
WHERE c.CategoryName = 'seafood'
```

Bonus-Modul

Erstelle eine Abfrage, bei der bei jeder Bestellung (OrderID) mit ausgegeben wird, ob die jeweiligen Frachtkosten größer, kleiner oder gleich 22 sind.

```
SELECT OrderID, CASE
    WHEN Freight < 22 THEN 'Frachtkosten sind kleiner als 22'
    WHEN Freight > 22 THEN 'Frachtkosten sind größer als 22'
    WHEN Freight = 22 THEN 'Frachtkosten sind genau 22'
END
FROM Orders
```

Berechne den Wert der vorrätigen Produkte (Products – UnitsInStock, UnitPrice) und den Wert der bestellten Produkte. Falls bei einem Produkt nichts bestellt wurde, soll „keine Bestellung“ ausgegeben werden.

```
SELECT (UnitPrice * UnitsInStock) AS WertVorrätig,
CASE
    WHEN (UnitPrice * UnitsOnOrder) = 0 THEN 'keine Bestellung'
    WHEN (UnitPrice * UnitsOnOrder) <> 0 THEN CAST((UnitPrice * UnitsOnOrder)
AS char)
END AS Bestellt
FROM Products
```