

Lösungen

Simple Select

- 1)

```
SELECT 100 AS Zahl
      , 'Donaudampfschiffahrtsgesellschaft' AS Text
      , 400/2 AS Rechnung
```

Datum

- 1)

```
SELECT DATEADD(dd, 38, getdate())
```
- 2)

```
SELECT DATENAME(dw, '1981-04-22')
```
- 3)

```
SELECT DATEDIFF(yyyy, getdate(), '1977-05-25')
```

In diesem Fall reicht es auch, nur das Jahr anzugeben, da wir ja nur nach den vergangenen Jahren fragen:

```
SELECT DATEDIFF(yyyy, getdate(), '1977')
```

Wir bekommen eine negative Zahl heraus; das Datum, das wir abfragen, liegt in der Vergangenheit. Abhängig von der Fragestellung wäre es auch möglich, Start- und Enddatum in der Abfrage zu vertauschen. (Dann kommt die gleiche Zahl heraus, aber ohne negatives Vorzeichen.)

- 4)

```
SELECT DATEPART(qq, '2019-10-26')
```
- 5)

```
SELECT  DATEPART(dd, '1981-04-22') AS Tag
      , DATEPART(mm, '1981-04-22') AS Monat
      , DATEPART(yyyy, '1981-04-22') AS Jahr
```

String Funktionen (Server Funktionen)

1) `SELECT REVERSE(STUFF(REVERSE('+49-8677988952'), 1, 3, 'xxx'))`

2)

.a) `SELECT TRIM(' Test ')`

.b) `SELECT DATALENGTH(' Test ') AS Vorher
 , DATALENGTH(TRIM(' Test ')) AS Nachher`

3)

```
SELECT  CONCAT(  
        SUBSTRING((STUFF('Wilham Shakesbeer', 4, 1, 'li')), 1, 14)  
        , SUBSTRING((STUFF('Wilham Shakesbeer', 14, 4, 'peare')), 14, 5)  
      )
```

Wenn nicht anders angegeben, verwenden wir für alle Abfragen ab hier die Northwind-Beispieldatenbank.

Tabellenabfragen

1) `SELECT CustomerID AS KundenID
 , CompanyName AS Firmenname
 , ContactName AS Kontaktperson
 , Phone AS Telefonnummer
FROM Customers`

Wir müssen hier kein „AS“ verwenden; wenn wir es weglassen, bekommen wir die ursprünglichen (englischen) Spaltenüberschriften CustomerID, CompanyName, ContactName und Phone.

2) `SELECT DATEDIFF(dd, RequiredDate, ShippedDate) AS Lieferverzögerung
FROM Orders`

3) `SELECT OrderID
 , ShippedDate
 , RequiredDate
 , DATEDIFF(dd, RequiredDate, ShippedDate) AS Lieferverzögerung
FROM Orders`

```
-- WHERE DATEDIFF(dd, RequiredDate, ShippedDate) IS NOT NULL
-- ORDER BY Lieferverzögerung DESC
```

Optional:

„WHERE DATEDIFF...“: Wir lassen Ergebnisse von den Bestellungen weg, die noch nicht geliefert wurden.

„ORDER BY...“: Wir ordnen die Ausgabe absteigend von der größten zur kleinsten Lieferverzögerung.

WHERE

- 1)

```
SELECT *
FROM Customers
WHERE Country = 'France'
```
- 2)

```
SELECT *
FROM Customers
WHERE Country = 'Argentina'
AND City = 'Buenos Aires'
```
- 3)

```
SELECT *
FROM Customers
WHERE Country = 'Germany'
OR Country = 'Austria'
```
- 4)

```
SELECT *
FROM Products
WHERE UnitsInStock > 100
```
- 5)

```
SELECT *
FROM Products
WHERE ProductID BETWEEN 10 AND 15
```
- 6)

```
SELECT *
FROM Products
WHERE SupplierID IN(2,7,15)
```
- 7)

```
SELECT *
FROM Products
WHERE ProductName LIKE '%coffee'
```
- 8)

```
SELECT *
FROM Products
WHERE ProductName LIKE 'L%'
```

- 9) `SELECT *`
`FROM Products`
`WHERE ProductName LIKE '%ost%'`
- 10) `SELECT *`
`FROM Products`
`WHERE SupplierID IN(5, 10, 15)`
`AND UnitsInStock > 10`
`AND UnitPrice < 100`
`ORDER BY UnitPrice DESC`
- 11) `SELECT COUNT(ProductID) --AS Anzahl`
`FROM Products`
`WHERE SupplierID = 17`

Optional können wir der Spalte mit „AS Spaltenname“ auch eine Überschrift geben. Die Ausgabe funktioniert auch ohne, dann bekommen wir aber eine Spalte mit der Überschrift „No column name“.

- 12) `SELECT *`
`FROM Products`
`WHERE ProductName LIKE '[d-l]%'[a-d|m-o]'`

INNER JOIN

- 1) `SELECT p.ProductName`
`, s.CompanyName`
`-- , s.ContactName`
`-- , s.Phone`
`FROM Products p INNER JOIN Suppliers s`
`ON p.SupplierID = s.SupplierID`
`WHERE p.ProductName LIKE 'Chai%'`
- 2) `SELECT c.CompanyName, o.Freight`
`FROM Customers c INNER JOIN Orders o`
`ON c.CustomerID = o.CustomerID`
`WHERE c.Country = 'USA'`
`ORDER BY o.Freight`
- 3) `SELECT`
`o.OrderID`
`, c.CompanyName`
`, od.Quantity`
`, p.ProductName`
`FROM Customers c INNER JOIN Orders o`
`ON c.CustomerID = o.CustomerID`
`INNER JOIN [Order Details] od`
`ON o.OrderID = od.OrderID`

```

            INNER JOIN Products p
                ON od.ProductID = p.ProductID
WHERE p.ProductName LIKE ('%Lager%')
OR p.ProductName LIKE ('%bier%')
OR p.ProductName LIKE ('%ale')
-- ORDER BY od.Quantity DESC, c.CompanyName

```

TOP

- 1)

```
SELECT TOP 1 ProductName, UnitPrice
FROM Products
ORDER BY UnitPrice DESC
```
- 2)

```
SELECT TOP 10 PERCENT WITH TIES p.ProductName, od.Quantity
FROM [Order Details] od INNER JOIN Products p
    ON od.ProductID = p.ProductID
ORDER BY od.Quantity DESC
```

Optional: „WITH TIES“

- 3)

```
SELECT TOP 3
    FirstName
    , LastName
    , Title
    , HireDate
FROM Employees
ORDER BY HireDate ASC
```

UNION

- 1)

```
SELECT CompanyName, ContactName, Phone, 'C' AS Category
FROM Customers
UNION
SELECT CompanyName, ContactName, Phone, 'S' AS Category
FROM Suppliers
```
- 2)

```
SELECT Region, 'C' AS Category
FROM Customers
UNION
SELECT Region, 'E' AS Category
FROM Employees
```

SUBSELECT

- 1)

```
SELECT *
FROM Customers
WHERE Country IN (SELECT ShipCountry FROM Orders)
```
- 2)

```
SELECT OrderID, Freight
FROM Orders
WHERE Freight < (SELECT AVG(Freight) FROM Orders)
-- ORDER BY Freight DESC
```

VIEW

```
CREATE VIEW Kundenbestellungen
AS
SELECT
    o.OrderID
  , c.CompanyName
  , p.ProductName
  , od.Quantity
  , SUM(od.UnitPrice*od.Quantity) AS Summe
FROM Customers c INNER JOIN Orders o
    ON c.CustomerID = o.CustomerID
    INNER JOIN [Order Details] od
        ON o.OrderID = od.OrderID
        INNER JOIN Products p
            ON od.ProductID = p.ProductID
GROUP BY o.OrderID, c.CompanyName, p.ProductName, od.Quantity
```

