

Einführung SQL

An abstract graphic consisting of multiple overlapping, wavy, horizontal bands in various shades of gray, creating a sense of depth and movement. The waves flow from left to right across the lower half of the slide.

BEDINGUNGEN VERKNÜPFEN

**Redundanzen
vermeiden**

Aufgabe

Führe folgende Abfragen aus. Vergleiche die Syntax und die Ergebnisse.

1.1 SELECT Country FROM Customers;	1.2 SELECT distinct Country FROM Customers;

Aufgabe

Führe folgende Abfragen aus. Vergleiche die Syntax und die Ergebnisse.

SELECT Country FROM Customers;	SELECT distinct Country FROM Customers;
Anzahl Datensätze: 91 Ausgabe aller Attribute Länder	Anzahl Datensätze: 21 Ausgabe der verschiedenen Länder

Aufgabe

Führe folgende Abfragen aus. Vergleiche die Syntax und die Ergebnisse.

SELECT Country FROM Customers;	SELECT distinct Country FROM Customers;
Anzahl Datensätze: 91 Ausgabe aller Attribute Länder	Anzahl Datensätze: 21 Ausgabe der verschiedenen Länder

Aufgabe

1.3 Anwendung TERRA Datenbank:

Welche Wüstenarten gibt es?

Aufgabe

1.3 Anwendung TERRA Datenbank:

Welche Wüstenarten gibt es?

SQL-Anfrage-Ergebnis an TERRA

Das Ergebnis der Anfrage

```
SELECT DISTINCT WUESTENART FROM WUESTE;
```

hat 6 Zeilen mit 1 Spalte:

WUESTENART
Wueste/Halbwueste
Sandwueste
Trockenwueste
Stein
Sand
Felswueste

DISTINCT

Merke: Redundanzen werden durch das
Schlüsselwort _____ in der SELECT-
Klausel vermieden

DISTINCT

Merke: Redundanzen werden durch das
Schlüsselwort `___DISTINCT___` in der SELECT-
Klausel vermieden

GROUP BY

Beispiel 2.1

Führe in der DB Northwind auf w3schools folgende
Abfrage aus:

```
SELECT COUNT(CustomerID), Country
```

```
FROM Customers
```

```
GROUP By Country;
```

GROUP BY

Beispiel 2.1 Ergebnis:

Number of Records: 21

Expr1000	Country
3	Argentina
2	Austria
2	Belgium
9	Brazil
3	Canada
2	Denmark
2	Finland
11	France

```
SELECT COUNT(CustomerID)
```

number of Records: 21

GROUP BY

Beispiel 2.1 Ergebnis:

Expr1000	Country
3	Argentina
2	Austria
2	Belgium
9	Brazil
3	Canada
2	Denmark
2	Finland
11	France

GROUP By Country

GROUP BY

2.2 Beispiel

Zeige, wieviele Wüsten es von jeder Wüstenart gibt!

GROUP BY

2.2 Beispiel

Zeige, wieviele Wüsten es von jeder Wüstenart gibt!

SQL-Anfrage-Ergebnis an TERRA

Das Ergebnis der Anfrage
SELECT COUNT(W_NAME), WUESTENART FROM WUESTE GROUP BY WUESTENART
hat 6 Zeilen mit 2 Spalten:

COUNT(W_NAME)	WUESTENART
2	Felswueste
7	Sand
13	Sandwueste
1	Stein
1	Trockenwueste
6	Wueste/Halbwueste

GROUP BY



COUNT(*)

2.2 Beispiel

Zeige, wieviele Wüsten es von jeder Wüstenart gibt!

SQL-Anfrage-Ergebnis an TERRA

Das Ergebnis der Anfrage
`SELECT COUNT(W_NAME), WUESTENART FROM WUESTE GROUP BY WUESTENART`
hat 6 Zeilen mit 2 Spalten:

COUNT(W_NAME)	WUESTENART
2	Felswueste
7	Sand
13	Sandwueste
1	Stein
1	Trockenwueste
6	Wueste/Halbwueste

GROUP BY

2.3 Beispiel

Gib die Anzahl der Kunden pro Land in absteigender Reihenfolge aus.

GROUP BY

2.3 Beispiel

Gib die Anzahl der Kunden pro Land in absteigender Reihenfolge aus.

SQL Statement:

```
SELECT COUNT(CustomerID), Country  
FROM Customers  
GROUP BY Country  
ORDER BY COUNT(CustomerID) DESC;
```

GROUP BY

2.3 Beispiel

SQL Statement:

```
SELECT COUNT(CustomerID), Country  
FROM Customers  
GROUP BY Country  
ORDER BY COUNT(CustomerID) DESC;
```

Number of Records: 21

Expr1000	Country
13	USA
11	France
11	Germany
9	Brazil
7	UK
5	Mexico

GROUP BY

2.3 Syntax

```
SELECT column_name(s)  
FROM table_name  
WHERE condition  
GROUP BY column_name(s)  
ORDER BY column_name(s);
```

Hausaufgabe

2.5. Hausaufgabe: Gib eine Liste mit der Anzahl der Wüsten pro Wüstenart in absteigender Reihenfolge aus !

HAVING

2.5. Führe folgende Abfrage aus. Welche Aufgabe verbirgt sich dahinter?

```
SELECT COUNT(CustomerID), Country  
FROM Customers  
GROUP BY Country  
HAVING COUNT(CustomerID) > 5;
```

HAVING

2.5. Führe folgende Abfrage aus. Welche Aufgabe verbirgt sich dahinter?

```
SELECT COUNT(CustomerID), Country  
FROM Customers  
GROUP BY Country  
HAVING COUNT(CustomerID) > 5;
```

Number of Records: 5

Expr1000	Country
9	Brazil
11	France
11	Germany
7	UK
13	USA

HAVING

Gib dieselbe Liste wie unter 3.1 aus, aber
sortiere sie nach Kundenanzahl.

HAVING

Gib dieselbe Liste wie unter 3.1 aus, aber sortiere sie nach Kundenanzahl.

```
SELECT COUNT(CustomerID), Country  
FROM Customers  
GROUP BY Country  
HAVING COUNT(CustomerID) > 5  
ORDER BY COUNT(CustomerID) DESC;
```


HAVING

Gib dieselbe Liste wie unter 3.1 aus, aber sortiere sie nach Kundenanzahl.

```
SELECT COUNT(CustomerID), Country  
FROM Customers  
GROUP BY Country  
HAVING COUNT(CustomerID) > 5  
ORDER BY COUNT(CustomerID) DESC;
```

Number of Records: 5

Expr1000	Country
13	USA
11	Germany
11	France
9	Brazil
7	UK

Having

Syntax

```
SELECT column_name(s)  
FROM table_name  
WHERE condition  
GROUP BY column_name(s)  
HAVING condition  
ORDER BY column_name(s);
```