

Leseprobe: **SQL mit MySQL - Band 1 & 2 Einführung
mit Online-DB zum Üben**

im Internet: www.SQL-Schule.de

...

6.2 WHERE – Klausel

Ihr lasst euch zunächst alle Datensätze der Tabelle *berg* ausgeben.

```
SELECT * FROM berg
```

Anschließend wollt ihr euch alle in der Datenbank vorhandenen Berge Deutschlands zeigen lassen und hierbei nur die Spalten *id_land*, *name* und *hoehe*. Dafür ändert ihr die Select-Anweisung und integrieren eine WHERE-Klausel wie folgt:

```
SELECT land, name, hoehe AS Gipfelhöhe FROM berg WHERE id_land =  
'de'
```

id_land	name	Gipfelhöhe
de	Großer Arber	1456
de	Großer Beerberg	983
de	Brocken	1142
de	Erbeskopf	816
de	Feldberg	1493
de	Großer Feldberg	882
de	Fichtelberg	1215
de	Fuchskaute	657
de	Hesselberg	689
de	Hohe Acht	747
de	Langenberg	843
de	Lemberg	1015
de	Gibacht	938
de	Schneefenerkopf	2874
de	Wasserkuppe	950
de	Watzmann	2713
de	Zugspitze	2962

Für eine Ausgabe aller Berge von Deutschland und Österreich ist folgende Anweisung einzugeben:

```
SELECT id_land, name, hoehe AS Gipfelhöhe FROM berg
      WHERE id_land = 'de' OR id_land = 'a'
```

Ausgabe aller Berge von Deutschland und Österreich mit einer Gipfelhöhe mit mehr als 2.000 Meter :

```
SELECT id_land, name, hoehe AS Gipfelhöhe FROM berg
      WHERE ( id_land = 'de' OR id_land = 'a' ) AND hoehe > 2000
```

In Deutschland und Österreich gibt es selbstverständlich noch viel mehr Berge mit einer Gipfelhöhe von über 2.000 Metern. Es können jedoch von einem Datenbanksystem prinzipiell immer nur die Daten angezeigt werden, die auch tatsächlich in einer Datenbank vorhanden (gespeichert) sind.

6.3 AND, OR, NOT - Vergleichsbedingungen und Prädikate

In der letzten Anfrage habt ihr eine AND-Verknüpfung in Klammern zusammen mit einer OR-Verknüpfung in der Where-Klausel eingesetzt.

OR: Oder-Verknüpfung von zwei Werten

AND: Und-Verknüpfung von zwei Werten

NOT: Negation eines Ausdrucks

Klammern: alle Ausdrücke in Klammern werden zuerst bewertet, immer von der inneren Klammer hin zur äußersten Klammer. Anschließend werden gegebenenfalls die noch verbliebenen äußeren Verknüpfungen ausgewertet.

Im nachstehenden Select suchen wir nach allen Bergen außer denen von Deutschland, Österreich und der Schweiz:

```
SELECT id_land, name, hoehe AS Gipfelhöhe FROM berg
      WHERE NOT ( id_land = 'de' OR id_land = 'a' OR
id_land = 'ch' )
```

id_land	name	Gipfelhöhe
ar	Aconcagua	6962
es	Pico de Aneto	3404
fr	Dôme du Goûter	4304
it	Dufourspitze	5634
us	Mount Elbert	4401
ru	Elbrus	5642
np	Mount Everest	8848
no	Galdhøpiggen	2469
sk	Gerlachovský štít	2655
a	Großglockner	3798
pk	K2	8611
in	Kangchendzönga	8586
tz	Kibo	5895
au	Mount Kosciuszko	2228
ca	Mount Logan	5965
us	Mount McKinley	6194
bg	Musala	2902
ru	Narodnaja	1895
	Nevado Huascarán	6768
		6634

Für den Vergleich von Inhalten könnt ihr nachstehende Bedingungen anwenden:

= Vergleich auf Gleichheit: $a = b$

< Vergleich kleiner $a < b$

> Vergleich größer $a > b$

<= Vergleich kleiner/gleich $a <= b$

>= Vergleich größer/gleich $a >= b$

<> Vergleich ungleich $a <> b$

!= identisch <>

IS [NOT] NULL überprüft, ob ein Wert Null ist

REGEXP, RLIKE Vergleichsoperatoren für reguläre Ausdrücke

Nachfolgende Prädikate (Bedingungen) können in Where, On- und Having-Klauseln verwendet werden:

LIKE	sucht Sätze teilqualifiziert mit Vorgabe-String-Muster
EXISTS	überprüft, ob Datensätze mit entsprechenden Werten vorhanden sind
IN	schaut nach, ob ein Ausdruck in einer vorgegebener Liste enthalten ist
BETWEEN	sucht Sätze mit Werten zwischen zwei Vorgabewerten

Beispiele mit Vergleichsoperatoren:

Ausgabe aller nicht-deutschen Berge:

```
SELECT * FROM berg WHERE id_land <> 'de'
```

Ausgabe aller deutschen, französischen und italienischen Berge mit IN-Operator :

```
SELECT * FROM berg WHERE id_land IN ( 'de', 'fr', 'it' )
```

Ausgabe aller Berge zwischen 2.000 und 4.000 Meter Höhe mit Between-Operator :

```
SELECT * FROM berg WHERE hoehe BETWEEN 2000 AND 4000
```

Das gleiche Ergebnis ist mit folgender Anfrage erreichbar:

```
SELECT * FROM berg WHERE hoehe >= 2000 AND hoehe <= 4000
```

Die Verwendung von LIKE findet ihr im nächsten Kapitel im Zusammenhang mit der Verwendung von Wildcards.