

Einführung in die Datenbank-Abfragesprache SQL

Datenbanken kann man sich wie eine Sammlung von Tabellen vorstellen. Eine Übersicht deutscher Städte könnte z. B. so aussehen:

Aber wie bekommt man jetzt möglichst schnell eine Übersicht über alle Städte *in Niedersachsen*, die in der Datenbank enthalten sind?

Dafür gibt es die Datenbank-Abfragesprache SQL, die im Folgenden erläutert werden soll.

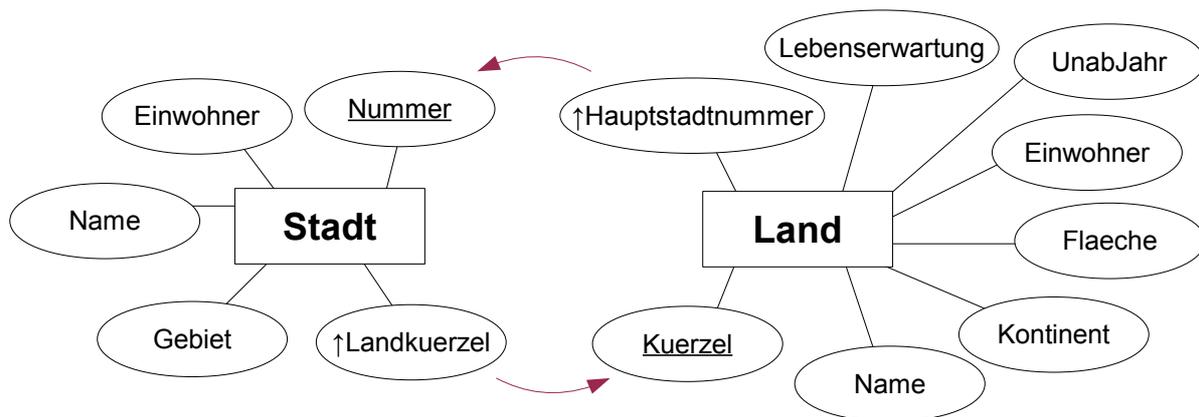
<i>Name</i>	<i>Bundesland</i>	<i>Einwohner</i>
Aachen	Nordrhein-Westfalen	243.825
Augsburg	Bayern	254.867
Bergisch Gladbach	Nordrhein-Westfalen	106.150
Berlin	Berlin	3.386.667
Bielefeld	Nordrhein-Westfalen	321.125
Bochum	Nordrhein-Westfalen	392.830
Bonn	Nordrhein-Westfalen	301.048
Bottrop	Nordrhein-Westfalen	121.097
Braunschweig	Niedersachsen	246.322

Vorbemerkungen zum Query-Browser:

- ✓ Die SQL-Anweisungen müssen in das Eingabefeld des Query-Browsers eingegeben werden. Dabei ist es egal, ob Groß- oder Kleinschreibung verwendet wird. (Großbuchstaben verwende ich nur der Übersichtlichkeit halber.)
- ✓ Die SQL-Anweisungen sollten auf mehrere Zeilen verteilt werden (das habe ich der Übersichtlichkeit halber getan), sie können aber auch in einer einzigen Zeile stehen.
- ✓ Nach der Eingabe betätigt man mit der Maus den „Blitz“-Button unterhalb von „Ansicht“ oder drückt auf der Tastatur „Strg+Enter“.
- ✓ Wenn Du mehrere Anweisungen parallel durchführen und betrachten willst, kannst Du Dir einen neuen Tabulator aufmachen, indem Du auf den Ordner neben „Ergebnismenge 1“ klickst.
- ✓ Bisher ausgeführte SQL-Anweisungen kannst Du rechts unter „Chronik“ anzeigen lassen.
- ✓ Unter „Schemata“ (rechts) kannst Du Dir die Namen der Tabellenspalten anzeigen lassen.

Die Welt-Datenbank

In dieser Einführung werden wir die Welt-Datenbank verwenden, die folgende Struktur hat:



Es gibt also eine Tabelle „Stadt“ mit den Städten und eine Tabelle „Land“ mit den Ländern der Welt. (Die dritte, nicht dargestellte Tabelle „Muttersprache“ wird später verwendet.)

Zu jeder Stadt gehören die Eigenschaften, die in den Kringeln rund um „Stadt“ stehen. So hat jede Stadt z. B. eine eindeutige Nummer. Die Eigenschaft „Landkuerzel“ hat denselben Wert wie das „Kuerzel“ des Landes, in dem die Stadt liegt. Umgekehrt hat die Eigenschaft „Hauptstadtnummer“ bei einem Land denselben Wert wie die „Nummer“ der zugehörigen Hauptstadt.

Auflistungen einer Tabelle – die ersten SQL-Abfragen

So, nun soll es aber wirklich losgehen. Wir wollen alle in der Datenbank stehenden Städte anzeigen lassen. Dazu gib bitte Folgendes ein: „**SELECT * FROM Stadt;**“. (Zur Erinnerung: Du darfst alles klein schreiben; die Anweisung wird mit dem „Blitz“-Button ausgeführt.)

Ganz unten links im Fenster kannst Du dann übrigens sehen, wie viele Städte in der Datenbank verzeichnet sind: „4079 Zeilen geholt“ bedeutet, diese Abfrage ergab 4079 Datensätze. Wenn Du bis zur Nummer 3068 nach unten scrollst, stößt Du endlich auf die deutschen Städte. Das müsste doch einfacher gehen, oder? Doch vorher die erste Aufgabe:

1. Wie sieht die Anweisung aus, wenn Du alle *Länder* anzeigen lassen möchtest? Wie viele Länder gibt es in der Datenbank?

Irgendwie ist die Liste noch zu breit. Was macht man, wenn nur einige der Informationen angezeigt werden sollen? Anstelle von „*“ gibst Du einfach die Namen der gewünschten Spalten getrennt durch Kommas ein, z. B. „**SELECT Name, Kontinent, Kuerzel FROM Land;**“.

Fassen wir noch einmal zusammen: Einfache SQL-Abfragen haben folgende Struktur:

```
SELECT Auswahl FROM Tabelle;
```

Die zu verwendende Tabelle ist (bis jetzt) „Stadt“ oder „Land“. Die Auswahl ist entweder ein „*“ oder eine Liste der gewünschten Spalten, deren Namen in den Kringeln um die gewählte Tabelle stehen. Jede SQL-Anweisung schließt mit einem Semikolon ab.

2. Es sollen alle Länder mit ihrem Namen, ihrem Kontinent und ihrer Bevölkerungszahl angezeigt werden.
3. Welches Land hat die kleinste Fläche? (Zur Lösung der Aufgabe musst Du durch die ausgegebene Liste scrollen.)

„Ordnung ist das halbe Leben.“ hast Du bestimmt schon häufig gehört. Eine ungeordnete Tabelle muss man erst lange durchsuchen, damit man den entsprechenden Eintrag findet (siehe Aufgabe 3). Die Arbeit kann man sich auch abnehmen lassen: Wenn man an das Ende der SQL-Anweisung „**ORDER BY**“, gefolgt von dem Spaltennamen, nach dem sortiert werden soll, schreibt, erfolgt eine aufsteigende Sortierung, für eine absteigende Sortierung (also größter Eintrag am Anfang) muss zusätzlich „**DESC**“ (Abkürzung für „descending“ = absteigend) angehängt werden. Beispiel: „**SELECT * FROM Land ORDER BY Einwohner DESC;**“ listet alle Länder auf, wobei das Land mit den meisten Einwohnern oben steht.

Wenn man nach zwei Kriterien gleichzeitig sortieren möchte (z. B. zuerst nach dem Kontinent und dann absteigend nach der Fläche) verwendet man eine Liste aus Spaltennamen, die jeweils durch ein Komma getrennt sind: „**ORDER BY Kontinent, Flaechе DESC**“.

```
SELECT Auswahl
```

```
FROM Tabelle
```

```
[ORDER BY Spaltenname [DESC] [, Spaltenname2 [DESC]]]
```

◀ neu

```
[LIMIT Anzahl];
```

◀ neu

Die eckigen Klammern „[“ und „]“ bedeuten, dass alles zwischen diesen Klammern weggelassen werden darf, wenn man es nicht verwenden möchte. Eine aufsteigende Sortierung verwendet z. B. kein „**DESC**“; bei einer Auflistung ohne Sortierung kann man auf die ganze Anweisungszeile „**ORDER BY** ...“ verzichten. Die eckigen Klammern werden grundsätzlich nicht mit eingegeben!

Die Limit-Angabe ermöglicht es, nur die ersten Zeilen einer vielleicht sehr umfangreichen Tabelle anzeigen zu lassen: „**LIMIT 15**“ zeigt z. B. nur die ersten 15 Zeilen an.

Die Reihenfolge der Bestandteile einer SQL-Abfrage ist von Bedeutung: erst „**FROM** ...“, dann „**ORDER BY** ...“, dann „**LIMIT** ...“!

4. Es sollen die fünf Städte der Welt mit den meisten Einwohnern angezeigt werden.
5. In welchem Land der Erde ist die Lebenserwartung am geringsten? (Hinweis: „NULL“ als Eintrag bedeutet, dass der zugehörige Wert nicht vorliegt.)
6. Welches Land ist laut Datenbank das älteste?
7. Es sollen alle Länder zuerst nach dem Kontinent und dann absteigend nach der Einwohnerzahl angezeigt werden. Wie heißt das Land Europas mit den meisten Einwohnern?

Die letzte Aufgabe macht deutlich, dass es doch möglich sein müsste, nur die Länder Europas anzeigen zu lassen. Davon handelt der nächste Abschnitt.

Auswahl von Datensätzen – SQL-Abfragen mit „WHERE“

Wenn man nur bestimmte Datensätze angezeigt haben möchte, die eine bestimmte Bedingung erfüllen, muss man „**WHERE**“, gefolgt von dieser Bedingung, verwenden:

```
SELECT Auswahl
FROM Tabelle
  [WHERE Bedingung]
  [ORDER BY Spaltenname [DESC] [, Spaltenname2 [DESC]]]
  [LIMIT Anzahl];
```

◀ neu

Die Bedingung vergleicht in der Regel zwei Werte, z. B. „Einwohner > 1000“. Als Vergleichsoperatoren stehen zur Verfügung: „=“ (gleich), „>“ (ungleich), „<“ (kleiner als), „>“ (größer als), „<=“ (kleiner oder gleich) und „>=“ (größer oder gleich).

Wörter (so genannte Zeichenketten) müssen in einfache Anführungszeichen (findest Du neben dem „Ä“ auf der Tastatur) eingeschlossen werden: „Kuerzel = 'DEU'“. Zahlenwerte werden nie in Anführungszeichen eingeschlossen.

Wenn mehrere Bedingungen *alle gleichzeitig* zutreffen sollen (z. B. Kontinent soll Europa sein und die Bevölkerungszahl soll mindestens 20 Millionen betragen), muss man die Bedingungen aufschreiben und zwischen diese „**AND**“ setzen. Wenn von mehreren Bedingungen *mindestens eine* zutreffen soll, muss zwischen den Bedingungen „**OR**“ stehen.

Bei Zeichenketten ist häufig nicht von Interesse, dass sie exakt einen bestimmten Wert haben, sondern dass sie z. B. mit „Me“ anfangen. Dafür schreibt man als Bedingung „*Zeichenkette* **LIKE** *Muster*“, wobei „Muster“ selbst wieder eine Zeichenkette ist. Im „Muster“ steht das Zeichen „%“ für beliebig viele Zeichen und „_“ (Unterstrich) für genau ein Zeichen. Das Muster „_a%“ findet also alle Zeichenketten, bei denen der zweite Buchstabe ein „a“ ist; das Muster „Me%“ findet alle Zeichenketten, die mit „Me“ beginnen.

Wenn man genau das Gegenteil einer Bedingung als neue Bedingung haben möchte, schreibt man „**NOT** (*Bedingung*)“.

8. Suche in der Datenbank alle Städte, die mindestens eine Million Einwohner haben. Wie viele solcher Städte gibt es insgesamt?
9. Wie viele europäische Länder stehen in der Datenbank? (Achtung: englische Bezeichnung für „Europa“ verwenden!)
10. Wie lauten die fünf bevölkerungsreichsten Länder Europas?

11. Welche Städte Niedersachsens stehen in der Datenbank?
12. In welchen afrikanischen Ländern haben die Einwohner eine Lebenserwartung von mehr als 70 Jahren. (Tipp: Verwende „AND“.)
13. Eine Liste aller Länder des amerikanischen Kontinents (Nord- und Südamerika) soll ausgegeben werden. (Tipp: Verwende „OR“.)
14. Welche außer-europäischen Länder erreichten vor 1800 ihre Unabhängigkeit? (Tipp: Verwende „<>“ und „AND“.)
15. Suche (z. B. in der Ergebnistabelle von Aufgabe 9 oder 10) das Kürzel für Großbritannien heraus. Welche Millionenstädte gibt es dort?
16. Welche Länder haben in ihrem Namen (in englischer Schreibung) als zweiten Buchstaben ein „z“?
17. Welche Ländernamen enden auf „au“?
18. Wie viele Ländernamen enthalten kein „a“? (Tipp: Verwende „NOT“.)
19. Gesucht sind alle Städte in Deutschland und Österreich, die mehr als 800.000 Einwohner haben. (Diese Aufgabe ist schwer. Tipp: Du darfst Teil-Bedingungen einklammern: „(... OR ...) AND ...“.)

Abfragen aus zwei Tabellen – der Verbund (Join)

Wenn man alle Millionenstädte Europas anzeigen lassen will, wäre es sehr mühsam, wenn man sich zuerst alle Kürzel der Länder Europas heraussuchen müsste, um sie dann wie in Aufgabe 19 alle einzeln einzugeben. Das kann auch die Datenbank für uns machen.

Wenn wir mehr als eine Tabelle hinter „FROM“ angeben, gibt es aber ein Problem: Die SQL-Anweisung „SELECT * FROM Stadt, Land;“ produziert fast eine Million Datensätze! Der Grund liegt darin, dass jede Stadt mit jedem Land zusammen angezeigt wird, also Hamburg mit Brasilien, Hamburg mit Australien, London mit Ägypten etc., und Anzahl der Länder multipliziert mit Anzahl der Städte gibt fast eine Million.

Wir müssen also sagen, *wie* wir die beiden Tabellen miteinander in Beziehung setzen wollen. Vielleicht erinnerst Du Dich noch daran, dass die Eigenschaft „Landkuerzel“ in der Stadt-Tabelle mit der Eigenschaft „Kuerzel“ bei der Land-Tabelle übereinstimmen muss. Wir schreiben in der SQL-Abfrage: „FROM Stadt INNER JOIN Land ON Landkuerzel = Kuerzel“. Hinter „ON“ steht also die Bedingung, über die die beiden Tabellen verknüpft werden. Jetzt werden also nur die Datensätze angezeigt, bei denen die beiden Kürzel übereinstimmen, also z. B. Hamburg mit Deutschland usw.

```
SELECT Auswahl
FROM Tabelle [INNER JOIN Tabelle2 ON Bedingung]
[WHERE Bedingung]
[ORDER BY Spaltenname [DESC] [, Spaltenname2 [DESC]]]
[LIMIT Anzahl];
```

Die Verknüpfungsbedingung hinter „ON“ bezieht sich auf die beiden genannten Tabellen. Die zu verwendenden Spaltennamen erkennt man daran, dass einer einen Pfeil (↑) vor dem Namen hat und der andere unterstrichen ist.

Wenn ein Spaltenname in *beiden* verwendeten Tabellen vorkommt, wie z. B. „Name“ und „Einwohner“, *muss* in allen Teilen der SQL-Abfrage vor dem Spaltennamen der zugehörige Tabellenname genannt werden; zwischen beiden steht ein Punkt: „Stadt.Name“ bzw. „Land.Einwohner“. Bei allen anderen Spaltennamen *kann* der Tabellenname genannt werden (z. B. „Land.Kuerzel“), er kann aber auch weggelassen werden.

Beispiel: Gesucht sind die fünf bevölkerungsreichsten Städte Europas.

```
SELECT Stadt.Name, Stadt.Einwohner, Land.Name
FROM Stadt INNER JOIN Land ON Stadt.Landkuerzel = Land.Kuerzel
WHERE Kontinent = 'Europe'
ORDER BY Stadt.Einwohner DESC
LIMIT 5;
```

20. Gesucht sind die fünf bevölkerungsreichsten Städte europäischer Länder, die nicht Hauptstadt ihres Landes sind. (Tipp: Gegenüber dem Beispiel musst Du nur noch zusätzlich überprüfen, dass die Stadt nicht Hauptstadt ist.)
21. Es sollen alle Millionenstädte der Welt mit ihren Ländern ausgegeben werden.
22. Wie viele Millionenstädte gibt es in Europa?
23. Bei welchen Städten in der Datenbank stimmt der Name der Stadt mit dem Namen des Landes überein?
24. Es sollen alle Länder mit ihren Hauptstädten ausgegeben werden. (Anmerkung: Du kannst ohne „WHERE“ auskommen!)

Wir lassen rechnen

Manchmal ist es sinnvoll, von MySQL Berechnungen durchführen zu lassen. Hat Dich nicht schon gestört, dass die Bevölkerungszahlen bei den Ergebnissen immer so schwer zu lesen sind, weil sie so lang sind? Betrachten wir folgendes Beispiel:

```
SELECT Name, Einwohner/1000000 AS `Einwohner in Mio.`
FROM Land
ORDER BY `Einwohner in Mio.` DESC;
```

Anstelle der Einwohnerzahl geben wir die Einwohnerzahl dividiert durch eine Million aus. Damit oben in der Tabelle nicht der Berechnungsterm „Einwohner/1000000“ steht, sagen wir noch, dass dort „Einwohner in Mio.“ stehen soll. Eine solche Ersatzbezeichnung nennt man „(Spalten-)Alias“ und wird durch „AS“ eingeleitet. Wenn der Alias Zeichen enthält, die keine Buchstaben sind, also z. B. Leerzeichen, musst Du ihn in besondere Anführungszeichen setzen. Dieses Anführungszeichen erhältst Du, indem Du die Umschalttaste und die Taste links neben der Backspace-Taste drückst. Zuerst siehst Du nichts, weil es sich bei diesem Zeichen strenggenommen um einen Akzent-Zeichen handelt. Durch anschließendes Drücken der Leertaste kommt das Zeichen auf den Bildschirm.

Der Alias kann jetzt anstelle eines Spaltennamens hinter „ORDER BY“ verwendet werden. In einer Bedingung hinter „WHERE“ ist die Verwendung eines Alias nicht möglich; dort muss ggf. der originale Berechnungsterm angegeben werden.

Natürlich ist auch die Berechnung mit den anderen Grundrechenarten möglich.

Einen Alias kann man auch verwenden, um die Tabellenbeschriftung eindeutiger zu gestalten:

```
SELECT Stadt.Name AS `Name (Stadt)`, Stadt.Einwohner AS `Einwohner (Stadt)`,
Land.Name AS `Name (Land)`, Land.Einwohner AS `Einwohner (Land)`
FROM Stadt INNER JOIN Land ON Stadt.Landkuerzel = Land.Kuerzel;
```

25. Die deutschen Städte sollen mit ihrer Einwohnerzahl (in Tausend) ausgegeben werden. Sie sollen dabei absteigend nach der Einwohnerzahl sortiert sein.
26. Wie viele Städte gibt es in der Welt mit mehr als 5 Millionen Einwohnern? Die Städte sollen absteigend sortiert mit ihrer Einwohnerzahl in Millionen ausgegeben werden.

27. Welches Land hat die größte Bevölkerungsdichte (also Einwohnerzahl dividiert durch Fläche)?
28. Es sollen alle Städte Deutschlands mit ihrem Anteil in Prozent an der Gesamtbevölkerung ausgegeben werden. (Tipp: Du benötigst beide Tabellen, weil Du die Einwohnerzahl Deutschlands mit einbeziehen musst. Die Prozentangabe erhältst Du, wenn Du den Anteil mit 100 multiplizierst. Wenn Du alles richtig gemacht hast, hat Berlin einen Anteil von etwas über 4 %.)
29. Welche Städte der Welt mit mehr als 20.000 Einwohnern haben mehr als 50 % der Einwohner ihres Landes? Gib die Liste der Städte und Länder mit ihren Namen und Einwohnerzahlen sortiert nach dem Anteil in Prozent aus. (Anmerkungen: Diese Aufgabe ist schwer. Die Daten in der Datenbank können nicht ganz korrekt sein. Wie wird das im Ergebnis deutlich?)
30. In welchen Ländern mit mehr als 1.000.000 Einwohnern hat die Hauptstadt mehr als 30 % der Einwohner des Landes? (Tipp: Verwende evtl. deine Lösung von Aufgabe 24.)

Wir lassen rechnen und zählen

Die Summe der Einwohner der deutschen Städte in der Datenbank könnte man zwar mühsam von Hand errechnen, dies kann einem aber MySQL abnehmen:

```
SELECT SUM(Einwohner)
FROM Stadt
WHERE Landkuerzel = 'DEU';
```

Es wird also bei dieser Abfrage nur die Summe angezeigt. Selbstverständlich könnte man auch wieder mit dieser Summe rechnen: „`SUM(Einwohner)/1000000 AS `Summe der Einwohner in Mio.``“

Neben der Summe mithilfe von „`SUM`“ gibt es noch weitere, vergleichbare Funktionen: „`AVG`“ berechnet den Durchschnitt (engl. „average“), „`MAX`“ das Maximum und „`MIN`“ das Minimum einer Spalte. Der jeweilige Spaltenname muss in Klammern stehen; zwischen dem Funktionsnamen und der Klammer darf kein Leerzeichen stehen!

Darüber hinaus gibt es noch die Funktion „`COUNT`“, die die Anzahl der Datensätze ermittelt. Dabei werden nur die Datensätze gezählt, die in der angegebenen Spalte nicht den Wert „`NULL`“ haben (vgl. Hinweis zur Aufgabe 5). „`COUNT (*)`“ zählt alle Datensätze.

Achtung: Die genannten Funktionen dürfen nicht in einer Bedingung hinter „`WHERE`“ stehen! Außerdem dürfen sie im Moment noch nicht zusammen mit normalen Spaltennamen in einer `SELECT`-Auswahl stehen; das wird erst im nächsten Abschnitt erklärt.

31. Wie viele Einwohner haben alle Länder Europas zusammen?
32. Wie groß ist die Fläche des größten Landes der Erde? (Anmerkung: *Welches Land das ist, soll in dieser Aufgabe nicht ermittelt werden.*)
33. Wie viele Städte Niedersachsens stehen in der Datenbank?
34. Wie groß ist die Gesamtfläche aller Länder Europas zusammen?
35. Wie hoch ist die durchschnittliche Lebenserwartung der Länder Afrikas?

Gruppierung

Die letzte Frage führt auf das nächste Problem: Es ist nicht schwierig, die durchschnittliche Lebenserwartung für jeden Kontinent *einzel*n ausgeben zu lassen. Es müsste aber doch möglich sein, eine *gesam*te Liste der Kontinente mit der jeweils durchschnittlichen Lebenserwartung erstellen zu lassen. Folgende SQL-Anweisung ist allerdings nicht zulässig:

```
SELECT Kontinent, AVG(Lebenserwartung)
FROM Land;
```

Was sollte die Datenbank hier tun? Es soll ein Durchschnitt berechnet werden. Aber woher weiß die Datenbank, dass das für die einzelnen Kontinente durchzuführen ist? Wir müssen der Datenbank „helfen“, indem wir sagen, dass zuerst alle Datensätze nach dem Kontinent zusammengefasst bzw. gruppiert werden sollen, also alle europäischen Länder, alle afrikanischen Länder usw., und anschließend von diesen Zusammenfassungen der Durchschnitt berechnet werden soll:

```
SELECT Kontinent, AVG(Lebenserwartung)
FROM Land
GROUP BY Kontinent;
```

Insgesamt ergibt sich damit folgende Struktur für SQL-Anweisungen:

```
SELECT Auswahl
FROM Tabelle [INNER JOIN Tabelle2 ON Bedingung]
[WHERE Bedingung]
[GROUP BY Spaltenname [DESC]]
[ORDER BY Spaltenname [DESC] [, Spaltenname2 [DESC]]]
[LIMIT Anzahl];
```

◀ neu

„GROUP BY“ schließt automatisch eine Sortierung wie bei „ORDER BY“ mit ein. Damit diese absteigend durchgeführt wird, muss ein „DESC“ angehängt werden. Ein gleichzeitiger Einsatz von „GROUP BY“ und „ORDER BY“ kann sinnvoll sein (siehe Beispiel weiter unten). In diesem Fall findet die Sortierung gemäß den Angaben hinter „ORDER BY“ statt.

Als Spaltenname hinter „GROUP BY“ darf auch ein Alias verwendet werden.

Bei der Eingabe der einzelnen Teile der SQL-Anweisung unbedingt die Reihenfolge beachten!

Beispiel für den gleichzeitigen Einsatz von „ORDER BY“:

```
SELECT Kontinent, AVG(Lebenserwartung) AS `Durchschnitt LE`
FROM Land
GROUP BY Kontinent
ORDER BY `Durchschnitt LE` DESC;
```

Hier wird also die durchschnittliche Lebenserwartung für jeden Kontinent bestimmt und anschließend nach dieser durchschnittlichen Lebenserwartung sortiert.

36. Erstelle eine Liste der gesamten Einwohnerzahl der einzelnen Kontinente. (Wenn möglich sollte die Liste die Einwohnerzahl in Mio. enthalten und nach dieser absteigend sortiert sein.)
37. Bestimme die auf die Bundesländer bezogene durchschnittliche Einwohnerzahl der in der Datenbank verzeichneten deutschen Städte. (Tipp: Überlege Dir zuerst, wonach gruppiert werden soll; dieser Spaltenname sollte auch in der SELECT-Auswahl stehen. Welche Bedingung hinter „WHERE“ müssen alle diese Städte erfüllen?)
38. Erstelle eine Liste der Bundesländer mit der Anzahl der verzeichneten Städte.

39. Erstelle eine Liste der Länder mit der Anzahl der zu diesen Ländern in der Datenbank verzeichneten Städte. (Tipp: Du musst beide Tabellen verwenden. Von welcher Spalte sollte die Anzahl bestimmt werden?) Welches Land hat die meisten Städte verzeichnet?
40. Erstelle eine Liste der Länder mit der Anzahl der Einwohner der bevölkerungsreichsten Stadt dieses Landes.

Subqueries – nur etwas für Spezialisten

Einige Fragestellungen lassen sich mit den bisherigen Mitteln noch nicht beantworten. Betrachten wir dazu folgendes Beispiel: Es sollen alle Städte aufgelistet werden, die mindestens ein Drittel der Bevölkerungszahl der größten Stadt der Welt haben. Das Ganze in einer einzigen Abfrage funktioniert folgendermaßen nicht, da „**MAX**“ nicht in einer Bedingung hinter „**WHERE**“ verwendet werden darf:

```
SELECT Name, Einwohner
FROM Stadt
WHERE Einwohner >= MAX(Einwohner)/3
ORDER BY Einwohner DESC;
```

„Von Hand“ würde man das Problem so lösen: Zuerst lässt man sich das Maximum der Einwohnerzahl der Städte anzeigen, notiert diese Zahl und ersetzt in der falschen SQL-Anweisung den Eintrag „**MAX**(Einwohner)“ durch diese Zahl. Diesen Vorgang kann man auch automatisieren:

```
SELECT Name, Einwohner
FROM Stadt
WHERE Einwohner >= (SELECT MAX(Einwohner) FROM Stadt)/3
ORDER BY Einwohner DESC;
```

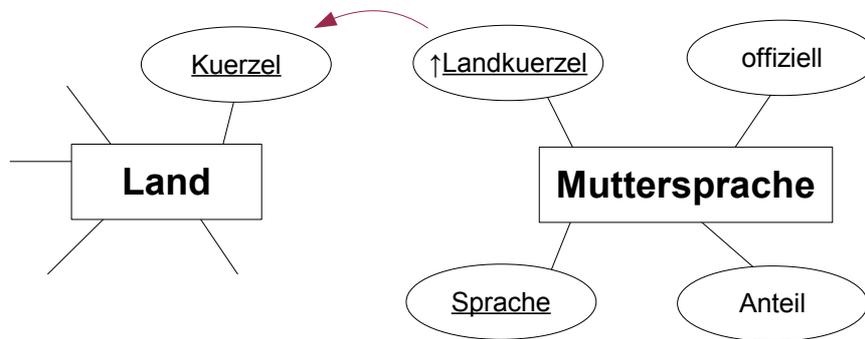
In den Klammern steht jetzt eine eigene SQL-Anweisung (allerdings ohne das abschließende Semikolon), die nur einen einzigen Zahlenwert liefert, nämlich gerade die Einwohnerzahl der bevölkerungsreichsten Stadt der Welt. Mit diesem Zahlenwert wird jetzt verglichen, und das gewünschte Ergebnis wird angezeigt. Eine solche eingeschachtelte SQL-Anweisung heißt „Subquery“ (zu Deutsch: Unterabfrage).

41. Es sollen alle Länder aufgelistet werden, die mindestens ein Fünftel der Größe des größten Landes haben.
42. In welchen Ländern ist die Lebenserwartung mindestens so hoch wie in Deutschland? (Tipp: Überlege Dir erst, wie die Subquery aussehen soll, und probiere sie vorher einzeln aus.) Beschränke die Liste auf europäische Länder und lass diese sortiert ausgeben.
43. In welchen Ländern ist die Lebenserwartung um mindestens fünf Jahre höher als der Durchschnitt in Europa? (Tipp: Überlege Dir zuerst die Subquery. Addiere *anschließend* zu deren Wert die Zahl „5“.)

Es gibt noch wesentlich kompliziertere Subqueries, aber deren Anwendung wird erst in späteren Jahrgängen unterrichtet.

Aufgabensammlung zur Tabelle „Muttersprache“

Da war doch noch was? Ach ja, die dritte Tabelle „Muttersprache“. Sie hat folgende Struktur:



Das „Landkuerzel“ hat wieder denselben Wert wie das „Kuerzel“ des zugehörigen Landes. Die Eigenschaft „offiziell“ gibt an, ob die jeweilige Sprache eine Amtssprache ist oder nicht; die Werte lauten also „ja“ bzw. „nein“; sie müssen in SQL-Abfragen in einfache Anführungszeichen gesetzt werden. Die Eigenschaft „Anteil“ gibt an, wie viel Prozent der Bevölkerung in dem jeweiligen Land die genannte Sprache als Muttersprache haben.

Auch wenn der Aufbau der Tabelle auf den ersten Blick recht einfach erscheint: Ein paar Informationen sollten noch gegeben werden.

Es gibt nämlich mehrere Einträge mit dem Landkürzel „DEU“, denn in Deutschland leben z. B. Immigranten, die nicht Deutsch als Muttersprache haben. Umgekehrt gibt es Deutsche, die im Ausland leben. Ihre Muttersprache bleibt natürlich Deutsch, auch wenn sie zurzeit in den USA oder in Argentinien wohnen. Außerdem gibt es noch sprachliche Minderheiten wie z. B. die so genannten Russland-Deutschen, deren Vorfahren vor Jahren ins Ausland gegangen sind, deren Nachfahren aber heute noch zu Hause Deutsch sprechen.

Wenn ein Datensatz nicht vorhanden ist, bedeutet das nicht, dass in diesem Land die Sprache nicht gesprochen wird; vielmehr ist der Anteil zu gering, als dass er aufgeführt werden würde.

Die folgenden Aufgaben sollen Dir helfen zu überprüfen, ob Du den bisherigen Stoff verstanden hast:

44. Erstelle eine Liste der in Deutschland verwendeten Muttersprachen mit deren Anteil, nach dem sortiert werden soll.
45. Erstelle eine Liste der Landkürzel der Länder, in denen es Bevölkerungsanteile mit Deutsch als Muttersprache gibt.
46. Erweitere die Aufgabe so, dass auch die Namen der jeweiligen Länder mit angezeigt werden. (Tipp: Du benötigst zwei Tabellen.)
47. Bilde die Summe der Muttersprachen-Anteile für Deutschland.
48. Erstelle eine Liste aller Länder mit deren Anteil-Summen (wie in Aufgabe 47). Woran kann man erkennen, dass die Datenbank falsche Daten enthalten muss?
49. Versuche herauszufinden, wie viele Sprachen in der Datenbank verzeichnet sind. (Tipp: Verwende eine Gruppierung.)
50. Erstelle eine Liste mit den Landkürzeln und der Anzahl der diesem Land zugeordneten Muttersprachen. Erweitere die Lösung so, dass das Land mit seinem vollen Namen zu sehen ist. Welche Länder haben die meisten Muttersprachen im eigenen Land?

51. Erstelle eine Liste der Sprachen mit der Anzahl der Länder, in denen diese Sprache als Muttersprache auftaucht. Welche Sprache wird am häufigsten als Muttersprache in den Ländern genannt?
52. Erstelle eine Liste der Sprachen mit der zugehörigen Anzahl von Personen, die diese Sprache als Muttersprache haben. (Diese Aufgabe ist schwer. Tipp: Du musst den jeweiligen Anteil mit der Einwohnerzahl des zugehörigen Landes multiplizieren, so dass Du z. B. die Anzahl der Personen erhältst, die in der Schweiz Deutsch als Muttersprache haben. Anschließend muss über die Länder summiert werden.) Welche Sprache ist bei den meisten Menschen der Erde Muttersprache? Wie viele Menschen haben Deutsch als Muttersprache?
53. Welche Länder haben die meisten Amtssprachen?