

6 SQL-Abfragen: Logik und Aggregate

6.1 Logische Operatoren

Aufgabe 6.1 Analyse

Überlegen Sie zunächst, welche Ergebnisse die folgenden SELECT-Befehle liefern könnten! Beschreiben Sie, welche Datensätze in der Ergebnistabelle enthalten sein werden!

```
-- Beispiel 1:
SELECT username, city
FROM users
WHERE city = "Berlin" AND name LIKE "Fabian%"

-- Beispiel 2:
SELECT username, city
FROM users
WHERE city = "Berlin" OR city = "Hamburg"

-- Beispiel 3:
SELECT username, city
FROM users
WHERE city = "Berlin" AND NOT gender LIKE "female"
```

Die folgende Tabelle zeigt die Bedeutung der logischen Operatoren:

A	B	NOT A	A AND B	A OR B	NOT A AND B	NOT(A OR B)
FALSE	FALSE	TRUE	FALSE	FALSE		
FALSE	TRUE	TRUE	FALSE	TRUE		
TRUE	FALSE	FALSE	FALSE	TRUE		
TRUE	TRUE	FALSE	TRUE	TRUE		

Aufgabe 6.2 Logikübung

Vervollständigen Sie die beiden letzten Spalten in der obigen Tabelle!

Aufgabe 6.3 Mehrere logische Operatoren

Erstellen Sie je eine SQL-Abfrage, die

1. alle Berliner auflistet, die *Marc* heißen,
2. alle Leipziger Frauen auflistet,
3. alle Linas und Lorenas auflistet,
4. alle Männer nach ihrer Körpergröße auflistet, die mindestens 16 Jahre alt sind und

5. alle Nutzer liefert, die weder männlich noch weiblich sind.

Aufgabe 6.4 Logische Operatoren mit Klammern

Erstellen Sie je eine SQL-Abfrage (Tipp: Verwenden Sie Klammern wie beim Taschenrechner), die

1. alle Benutzer mit dem Namen "Xaver" auflistet, die nicht in München oder Nürnberg leben und
2. alle Benutzer*Innen auflistet, die 10 cm größer als der Durchschnitt ihrer Geschlechtsgenossen sind. (Die Durchschnittsgröße von Männern beträgt in Deutschland 179,9 cm, die von Frauen 165,9 cm)

6.2 Zusammenfassungen (Aggregate)

Bisher haben Sie bei Abfragen

- die Spalten, die angezeigt werden sollen, ausgewählt (Projektion) oder
- die Zeilen, die im Ergebnis stehen sollen, ausgewählt (Selektion).

Aber was, wenn wir wissen wollen, wie groß unser größtes Mitglied in jeder Stadt ist? Oder wir für jede Stadt wissen wollen, wie viele Mitglieder wir dort haben?

Die Lösung hierfür heißt Aggregat (aggregieren: lat. *anhäufen*). Die Idee ist in der folgenden Abbildung dargestellt:

The diagram illustrates the concept of aggregation. On the left, a source table with columns 'username' and 'city' contains six rows of user data. On the right, an aggregate table with columns 'city' and 'COUNT(city)' shows the count of users for each city. Lines connect the rows in the source table to their corresponding counts in the aggregate table: four rows for 'München' and two rows for 'Dresden'.

username	city
max383	München
jen84	München
malte231	München
karl2	München
karl261	Dresden
brigitte94	Dresden

city	COUNT(city)
München	4
Dresden	2

Abbildung 1: Aggregate fassen mehrere Datensätze zusammen

Aggregate fassen also mehrere Zeilen der Quelltabellen zu einer einzigen Zeile in der Ergebnistabelle zusammen.

Aufgabe 6.4 Unser erstes Aggregat

Der folgende SELECT-Befehl liefert für jede Stadt den größten und kleinsten Benutzer:

```
SELECT  city AS "Stadt", MIN(centimeters) AS "kleinster",  
        MAX(centimeters) AS "groesster"  
FROM    users  
GROUP BY city
```

Probieren Sie den Befehl aus und erklären Sie, wie er funktioniert. Beachten Sie dabei *alle* neuen SQL-Wörter.

MAX ist eine **Aggregatsfunktion**. Weitere Aggregatsfunktionen sind beispielsweise COUNT, SUM und AVG (Durchschnitt, engl. *average*).

Aufgabe 6.5 Aggregatsfunktionen

Erstellen Sie je eine SQL-Abfrage, die

1. alle Werte des Feldes gender liefert und angibt, wie oft diese auftreten,
2. die durchschnittliche Größe aller Mitglieder in Dresden liefert und
3. das Geburtsdatum des jüngsten männlichen und des jüngsten weiblichen Mitglieds liefert.