

7.3 Operationen auf Tabellen

H In SQL sind Befehle nicht von der Groß-/Kleinschreibung abhängig. Zur besseren Lesbarkeit werden jedoch die Schlüsselworte in Großbuchstaben geschrieben.

H Während Programmiersprachen wie Java oder C **prozedural** sind, handelt es sich bei SQL um eine **deklarative** Sprache. D. h. ein Programmierer muss nicht Schritt für Schritt alle Anweisungen angeben, die zum Erreichen eines Ziels durchgeführt werden müssen, sondern er formuliert, welches Ziel erreicht werden soll, und lässt das Datenbankmanagementsystem entscheiden, wie es dabei vorgeht.

Tabellen sind der integrale Bestandteil eines relationalen Datenbanksystems. Es geht allerdings nicht nur darum, Daten in Tabellen zu erfassen. Die wesentliche Funktion eines Datenbanksystems besteht darin, ausgewählte Informationen aus dem Datenbestand wieder zurückzugewinnen. Ein Benutzer eines Online-Buchhandels muss die Menge der Daten filtern, wenn er z. B. nach einem bestimmten Autor oder einer bestimmten Preiskategorie sucht. Dieses Filtern der Daten wird durch das Datenbankmanagementsystem übernommen und es ermöglicht vor allem bei großen Datenmengen mit vielen Attributen eine enorme Zeitersparnis gegenüber dem manuellen Suchen. Im Folgenden wird die Abfragesprache SQL (Structured Query Language) vorgestellt, die einen Standard für relationale Datenbanksysteme darstellt.

Der Grundaufbau einer SQL-Abfrage ist immer gleich:

```
SELECT Spalte (n)           ← Projektion
FROM Tabelle (n)
[WHERE Bedingung (en) ] ;  ← Selektion (kann entfallen)
```

Projektion

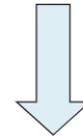
Eine Projektion filtert bestimmte Spalten für die Ausgabe. Sie blendet also alle anderen Spalten aus.

Buch			
ISBN	Titel	Autor	Preis
978-3-608-93544-8	Der Herr der Ringe	J.R.R. Tolkien	32,90
978-3-596-29433-6	Der Zauberberg	Thomas Mann	12,95
978-1-4012-2266-6	Watchmen	Alan Moore	13,95

Wähle die Spalte Autor ...

```
SELECT Autor
FROM Buch;
```

... aus der Tabelle Buch.



Autor
J.R.R. Tolkien
Thomas Mann
Alan Moore

Das Zeichen * bewirkt, dass alle Spalten einer Tabelle ausgegeben werden.

```
SELECT *
FROM Buch;
```

ISBN	Titel	Autor	Preis
978-3-608-93544-8	Der Herr der Ringe	J.R.R. Tolkien	32,90
978-3-596-29433-6	Der Zauberberg	Thomas Mann	12,95
978-1-4012-2266-6	Watchmen	Alan Moore	13,95

Selektion

Eine Selektion filtert jene Zeilen heraus, die einer bestimmten Bedingung genügen.

Buch			
ISBN	Titel	Autor	Preis
978-3-608-93544-8	Der Herr der Ringe	J.R.R. Tolkien	32,90
978-3-596-29433-6	Der Zauberberg	Thomas Mann	12,95
978-1-4012-2266-6	Watchmen	Alan Moore	13,95

... aus der Tabelle
Buch, ...

```
SELECT *
FROM Buch
WHERE Titel = 'Watchmen';
```

... deren Titel den Wert
„Watchmen“ hat.

ISBN	Titel	Autor	Preis
978-1-4012-2266-6	Watchmen	Alan Moore	13,95

Durch die Verbindung von Projektion und Selektion können die Daten einer Tabelle gezielt nach bestimmten Spalten und bestimmten Bedingungen gefiltert werden.

Buch			
ISBN	Titel	Autor	Preis
978-3-608-93544-8	Der Herr der Ringe	J.R.R. Tolkien	32,90
978-3-596-29433-6	Der Zauberberg	Thomas Mann	12,95
978-1-4012-2266-6	Watchmen	Alan Moore	13,95

```
SELECT Autor
FROM Buch
WHERE Titel = 'Watchmen';
```

Autor
Alan Moore

Vereinigung

Eine Vereinigung zweier Relationen führt alle Datensätze der beiden Relationen zu einer einzigen Relation zusammen. Dazu müssen die beiden Relationen dasselbe Schema (d.h. dieselben Attribute und Attributstypen) haben. Doppelt vorkommende Datensätze werden in der Ergebnisrelation nur einmal aufgeführt.

Buch			
ISBN	Titel	Autor	Preis
978-3-608-93544-8	Der Herr der Ringe	J.R.R. Tolkien	32,90
978-3-596-29433-6	Der Zauberberg	Thomas Mann	12,95
978-1-401-22266-6	Watchmen	Alan Moore	13,95

Buch1			
ISBN	Titel	Autor	Preis
978-3-596-29433-6	Der Zauberberg	Thomas Mann	12,95
978-3-257-24198-3	Fast genial	Benedict Wells	10,90

```
SELECT *
FROM Buch
UNION
SELECT *
FROM Buch1;
```

ISBN	Titel	Autor	Preis
978-3-608-93544-8	Der Herr der Ringe	J.R.R. Tolkien	32,90
978-3-596-29433-6	Der Zauberberg	Thomas Mann	12,95
978-1-401-22266-6	Watchmen	Alan Moore	13,95
978-3-257-24198-3	Fast genial	Benedict Wells	10,90

Verbund (Join)

Ein Datenbanksystem arbeitet in der Regel nicht nur auf einer Tabelle, sondern die zu speichernden Daten sind auf mehrere Relationen verteilt. Durch einen Join (dt.: Verbund) werden zwei Relationen zu einer neuen Tabelle verbunden. Natürlich sollen nur jene Datensätze miteinander verbunden werden, die auch von ihrer Bedeutung her zueinander passen. Voraussetzung dafür ist ein gemeinsames Attribut, hier im Beispiel die Verlagsnummer VID. Datensätze, für die kein passender „Partner“ in der anderen Tabelle vorliegt, werden in der Ergebnistabelle nicht aufgeführt.

Buch				
ISBN	Titel	Autor	Preis	VID
978-3-608-93544-8	Der Herr der Ringe	J.R.R. Tolkien	32,90	608
978-3-596-29433-6	Der Zauberberg	Thomas Mann	12,95	596
978-1-401-22266-6	Watchmen	Alan Moore	13,95	4012

Verlag		
VID	Name	Ort
608	Klett-Cotta	Stuttgart
596	Fischer Taschenbuch	Frankfurt a.M.
456	Schöningh	Paderborn

SELECT *
FROM Buch INNER JOIN Verlag
ON Buch.VID = Verlag.VID;

ISBN	Titel	Autor	Preis	Buch.VID	Verlag.VID	Name	Ort
978-3-6...	Der Herr der Ringe	J.R.R. Tolkien	32,90	608	608	Klett-Cotta	Stuttgart
978-3-5...	Der Zauberberg	Thomas Mann	12,95	596	596	Fischer Taschenbuch	Frankfurt a.M.

H Statt INNER JOIN wird auch der Befehl JOIN akzeptiert.

L Ein Join ist das Verbinden zweier Relationen zu einer neuen Relation. Dabei werden jene Datensätze miteinander verknüpft, die in einem (oder mehreren) gemeinsamen Attributen dieselben Werte haben. Diese gemeinsamen Attribute der beiden Relationen müssen dieselbe Bedeutung und denselben Wertebereich haben.

H Alternativ kann der INNER JOIN auch wie folgt umgesetzt werden.

```
SELECT *
FROM Buch, Verlag
WHERE Buch.VID = Verlag.VID;
```

Sollen auch solche Datensätze in der Ergebnistabelle vorkommen, für die kein passendes Gegenstück in der anderen Tabelle vorliegt, so bietet die Sprache SQL auch dafür Lösungen an. Mit dem Left Join werden alle Datensätze aus der linken, d. h. der ersten, Tabelle mit in die Ergebnisrelation aufgenommen, auch wenn keine entsprechenden Datensätze in der zweiten Tabelle existieren.

Buch				
ISBN	Titel	Autor	Preis	VID
978-3-608-93544-8	Der Herr der Ringe	J.R.R. Tolkien	32,90	608
978-3-596-29433-6	Der Zauberberg	Thomas Mann	12,95	596
978-1-401-22266-6	Watchmen	Alan Moore	13,95	4012

Verlag		
VID	Name	Ort
608	Klett-Cotta	Stuttgart
596	Fischer Taschenbuch	Frankfurt a.M.
456	Schöningh	Paderborn

SELECT *
FROM Buch LEFT JOIN Verlag
ON Buch.VID = Verlag.VID;

ISBN	Titel	Autor	Preis	Buch.VID	Verlag.VID	Name	Ort
978-3-6...	Der Herr der Ringe	J.R.R. Tolkien	32,90	608	608	Klett-Cotta	Stuttgart
978-3-5...	Der Zauberberg	Thomas Mann	12,95	596	596	Fischer Taschenbuch	Frankfurt a.M.
978-1-401-22266-6	Watchmen	Alan Moore	13,95	4012			

Umgekehrt fügt der Right Join alle Datensätze aus der rechten, d. h. der zweiten, Tabelle der Ergebnisrelation hinzu, unabhängig davon, ob sie mit passenden Daten aus der ersten Tabelle verbunden werden können oder nicht.

Buch				
ISBN	Titel	Autor	Preis	VID
978-3-608-93544-8	Der Herr der Ringe	J.R.R. Tolkien	32,90	608
978-3-596-29433-6	Der Zauberberg	Thomas Mann	12,95	596
978-1-401-22266-6	Watchmen	Alan Moore	13,95	4012

Verlag		
VID	Name	Ort
608	Klett-Cotta	Stuttgart
596	Fischer Taschenbuch	Frankfurt a.M.
456	Schöningh	Paderborn

SELECT *
FROM Buch RIGHT JOIN Verlag
ON Buch.VID = Verlag.VID;

ISBN	Titel	Autor	Preis	Buch.VID	Verlag.VID	Name	Ort
978-3-6...	Der Herr der Ringe	J.R.R. Tolkien	32,90	608	608	Klett-Cotta	Stuttgart
978-3-5...	Der Zauberberg	Thomas Mann	12,95	596	596	Fischer Taschenbuch	Frankfurt a.M.
					496	Schöningh	Paderborn

Aufgaben

1. Ermitteln Sie die Ergebnisrelationen der folgenden Abfragen auf den Tabellen Buch und Film.

Film				
EAN	Titel	FSK	Preis	zum_Buch
4042564141962	Der Zauberberg	16	11,99	978-3-596-29433-6
4011976887188	Fack Ju Göhte	12	7,99	
4010324200334	Rubinrot	12	7,99	

- a)

```
SELECT Titel, FSK
FROM Film
WHERE FSK = 12;
```
- b)

```
SELECT Titel, Preis
FROM Buch
UNION
SELECT Titel, Preis
FROM Film;
```
- c)

```
SELECT Buch.Titel, Buch.Preis, Film.Titel, Film.Preis
FROM Buch INNER JOIN Film
ON Buch.ISBN = Film.zum_Buch;
```
- d)

```
SELECT Buch.Titel, Buch.Preis, Film.Titel, Film.Preis
FROM Buch RIGHT JOIN Film
ON Buch.ISBN = Film.zum_Buch
WHERE Film.Preis < 10;
```
2. Es sind Abfrageergebnisse zu der obigen Tabelle Buch gegeben. Notieren Sie dazu jeweils zwei mögliche Abfragen in der Relationenalgebra und in SQL.

a)

ISBN
978-3-596-29433-6

b)

Titel
Der Herr der Ringe
Der Zauberberg
Watchmen

c)

Autor	Preis
J.R.R. Tolkien	32,90

3. Die deutsche Fußballnationalmannschaft muss für die Weltmeisterschaft neun Qualifikationsspiele spielen. Die dazugehörigen Daten sind in einer Tabelle zusammengefasst.



Recherchieren Sie für Aufgabe 3b)iii) nach dem SQL-Schlüsselwort DISTINCT.

H

Datum	Gegner	Heimspiel/Auswärtsspiel
07.09.12	Färöer	HS
11.09.12	Österreich	AS
12.10.12	Republik Irland	AS
16.10.12	Schweden	HS
26.03.13	Kasachstan	HS
06.09.13	Österreich	HS
10.09.13	Färöer	AS
11.10.13	Republik Irland	HS
15.10.13	Schweden	AS

- a) In der Tabelle ist kein Attribut als Schlüssel definiert. Diskutieren Sie, ob sich eines der vorhandenen Attribute dafür anbietet oder ob ein Schlüsselattribut eigens definiert werden soll.
- b) Geben Sie jeweils eine Abfrage in Relationenalgebra und SQL an.
- Ein Fan möchte alle Spiele in Deutschland sehen. Welche Spiele finden in Deutschland statt?
 - Ein Fan möchte für die Klassiker gegen Österreich Urlaub einplanen. Wann finden Spiele gegen Österreich statt?
 - Auf welche Gegner trifft Deutschland in der WM-Qualifikation?
4. Der Grundkurs im Fach Sozialwissenschaften in der Jahrgangsstufe Q1 besteht aus 23 Schülerinnen und Schülern und findet in Raum C 204 statt.
- Legen Sie eine passende Tabelle an, in der die Lehrerin des Kurses diesen und andere Kurse erfassen kann. Legen Sie geeignete Schlüsselattribute fest.
 - Die Lehrerin möchte auch zu jeder Schülerin und jedem Schüler relevante Daten notieren. Finden Sie zuerst solche Daten und entwickeln Sie dann ein passendes Tabellenschema.
 - Notieren Sie umgangssprachlich Abfragen für Ihre in b) entwickelten Tabellen. Lassen Sie diese von Mitschülern in Relationenalgebra und SQL formulieren.
5. Die Geschäftsführerin einer Pizzeria möchte alle Gerichte ihres Restaurants in einer zentralen Datenbank speichern. Einzelne Benutzergruppen (Geschäftsführerin, Mitarbeiter und Kunden) sollen unterschiedliche Ansichten auf die Daten erhalten.
Alle Daten werden in einer Tabelle Gericht organisiert:

Nächste

ID	Name ital.	Name dt.	Preis	Kosten	Ge...		Zube...
1	Rigatoni	Römernudeln	11,00	4,00	7,00	<input type="checkbox"/>	07:00
2	Pizza vegetale	Pizza vegetarisch	6,00	2,00	3,90	<input checked="" type="checkbox"/>	04:00
3	Lasagne Spinachi	Spinat Lasagne	9,00	3,00	5,70	<input type="checkbox"/>	12:00

Alle Daten für die Geschäftsführerin



Tabelle Gericht

Feldname	Feldtyp
ID	Integer [INTEGER]
Name ital.	Text [VARCHAR]
Name dt.	Text [VARCHAR]
Preis	Dezimal [DECIMAL]
Kosten	Dezimal [DECIMAL]
Gewinn	Text [VARCHAR]
Vegetarisch	Ja/Nein [BOOLEAN]
Zubereitungszeit	Zeit [TIME]



PIZZA	
Pizza vegetarisch	6,00
...	
LASAGNE	
Spinat Lasagne	9,00
...	
NUDELGERICHTE	
Römernudeln	11,00
...	

Speisekarte für den Gast

Tabelle Tisch

Feldname	Feldtyp
Nr	Integer [INTEGER]
Anzahl_Plätze	Integer [INTEGER]

Tabelle Bestellung

Feldname	Feldtyp
ID	Integer [INTEGER]
TischNr	Integer [INTEGER]
Gericht_ID	Integer [INTEGER]

Tabellenschemata der Rohdaten

**TISCH 1**

```

- Pizza vegetarisch           vegetarisch   6,00
- Spinat Lasagne             nicht vege.  9,00
...

```

Kassenprogramm der Bedienung

- a) Erstellen Sie die nötigen Abfragen, die erforderlich sind, um die jeweiligen Informationen zu einem Gericht der jeweiligen Benutzergruppe anzuzeigen.

- b) Diskutieren Sie, inwieweit folgende weitere Kategorien für die Rohdaten sinnvoll sein könnten: Kategorie (Lasagne, Pizza, ...), Im_Angebot, Beliebtheitsgrad und Kalorien. Finden Sie weitere Kategorien.
- c) Ermitteln Sie die Ergebnisrelation der folgenden SQL-Abfrage für die angegebenen Beispieltabellen. Formulieren Sie umgangssprachlich, was die Abfrage leistet.

```
SELECT Nr
FROM Tisch LEFT JOIN Bestellung
ON Tisch.Nr = Bestellung.TischNr
WHERE Gericht_ID IS NULL;
```

Beispieltabelle Tisch

	Nr	Anzahl_Plätze
	1	4
	2	2
	3	6

Beispieltabelle Bestellung

	ID	TischNr	Gericht_ID
	1	1	2
	2	1	3

H Mit dem Operator IS NULL wird geprüft, ob ein Wert vorhanden ist oder ob das Feld noch mit NULL belegt ist.

Nächst

6. Weiterführung der Projektaufgabe: World Factbook

Sie haben in einem DBMS bereits die Tabelle Country angelegt:

Country			
country	area	population	Internet users
...			

Formulieren Sie die folgenden Abfragen in SQL und überprüfen Sie das Ergebnis mithilfe des DBMS.

- a) Gesucht sind die Namen aller Länder.
- b) Gesucht sind alle Länder, deren Fläche größer als 8 000 000 km² ist.
- c) Gesucht ist die Einwohnerzahl von Frankreich.
- d) Entwickeln Sie weitere Abfragen auf der Tabelle Country. Formulieren Sie in SQL und testen Sie im DBMS.

7. Gegeben sind drei Relationen, die das Ausgehverhalten einiger Jugendlicher dokumentieren.

Hört_gerne	
Name	Musikrichtung
Tina	Pop
Lisa	Latin
Tina	Rock
Maxi	Pop

Geht_in	
Name	Lokal
Maxi	A
Maxi	C
Lisa	B
Tina	B
Flo	A

Spielt	
Lokal	Musikrichtung
A	Rock
A	Pop
B	Latin
D	Rock



- Bilden Sie den Verbund der Tabellen Hört_gerne und Spielt über das gemeinsame Attribut Musikrichtung und beschreiben Sie, welche Information Ihre Ergebnisrelation beinhaltet.
- Geben Sie alle Lokale an, die Musikrichtungen spielen, die Tina mag. Entwickeln Sie eine entsprechende Abfrage in SQL.
- Analysieren Sie die folgenden Abfragen, indem Sie jeweils die Ergebnisrelation und eine umgangssprachliche Formulierung für die Abfrage angeben.

- ```
SELECT Musikrichtung
FROM Hört_gerne INNER JOIN Geht_in ON Hört_gerne.Name
= Geht_in.Name
WHERE Lokal = 'A';
```
- ```
SELECT Musikrichtung
FROM Geht_in INNER JOIN Spielt ON Geht_in.Lokal =
Spielt.Lokal
WHERE Name = 'Maxi';
```
- ```
SELECT Name
FROM Hört_gerne INNER JOIN Spielt ON Hört_gerne.Musik-
richtung = Spielt.Musikrichtung
INNER JOIN Geht_in ON Hört_gerne.Name = Geht_in.Name
AND Spielt.Lokal = Geht_in.Lokal;
```

(Hinweis: in c) 3. werden alle drei Tabellen miteinander verbunden.)

## 7.4 Daten filtern mit SQL

Eine wesentliche Aufgabe von Datenbankmanagementsystemen ist es, Daten nach bestimmten Kriterien zu filtern. Das Beispiel des Online-Buchhandels macht dies deutlich:

Finde alle Medien, die „Coraline“ beinhalten und zur Kategorie Bücher gehören.

... die zwischen 7,50 und 9,99 € kosten.

The screenshot shows the website interface with search results for 'Coraline'. The results are sorted by price, and a callout bubble points to the 'Sortieren nach:' dropdown menu. Another callout bubble points to the price range filter on the left sidebar.

Sortiere alle Ergebnisse aufsteigend nach ihrem Preis.

Mit einfachen Projektionen und Selektionen sind diese Abfragen an die Datenbank nicht mehr möglich. Bei vielen Fragestellungen reicht es nicht aus, einfach nur gewisse Spalten oder Zeilen einer Tabelle herauszufiltern. Es müssen mehrere Bedingungen miteinander verknüpft, die Daten sortiert oder Berechnungen auf Zahlenwerten einer Tabelle durchgeführt werden.

Der SELECT-Befehl in SQL bietet zahlreiche Sprachkonstrukte an, um Abfragen differenziert zu formulieren und zu gestalten. Die wichtigsten werden nun vorgestellt.

### Vergleichsoperatoren

Bei einer Selektion werden die Datensätze nach einer bestimmten Bedingung gefiltert. Neben dem Gleichheitszeichen können die folgenden Operatoren verwendet werden, um in der WHERE-Klausel eine Bedingung zu formulieren.

| Operator | Erläuterung                                                                                                                                          |
|----------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| <>       | ungleich                                                                                                                                             |
| <        | kleiner                                                                                                                                              |
| >        | größer                                                                                                                                               |
| <=       | kleiner oder gleich                                                                                                                                  |
| >=       | größer oder gleich                                                                                                                                   |
| BETWEEN  | Es wird ein zusammenhängender Bereich durch Angabe eines Start- und eines Endwertes bestimmt. Der Start- und Endwert selbst zählen mit zum Ergebnis. |

| Operator | Erläuterung                                                                                                                                                                        |
|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| LIKE     | Es wird mit einem Muster verglichen. Jokerzeichen helfen bei der Formulierung des Musters. % steht für beliebig viele beliebige Zeichen, _ steht für genau ein beliebiges Zeichen. |
| IN       | Es wird mit einer Liste von Kriterien verglichen.                                                                                                                                  |
| IS NULL  | Es wird geprüft, ob ein Wert vorhanden ist oder ob das Feld noch mit NULL belegt ist.                                                                                              |

**H** In MS-Access werden die Platzhalter \* für beliebig viele beliebige Zeichen und ? für ein beliebiges Zeichen verwendet.

### Beispiele:

Welche Buchtitel beginnen mit einem Buchstaben von ‚A‘ bis ‚D‘?

```
SELECT Titel
FROM Buch
WHERE Titel < ,E`;
```

| Titel              |
|--------------------|
| Der Herr der Ringe |
| Der Zauberberg     |

Welche Bücher kosten 10 bis 20 €?

```
SELECT Titel
FROM Buch
WHERE Preis BETWEEN 10 AND 20;
```

| Titel          |
|----------------|
| Der Zauberberg |
| Watchmen       |

Suche alle Autoren, deren Name ‚Tolk‘ beinhaltet, sowie ihre Werke.

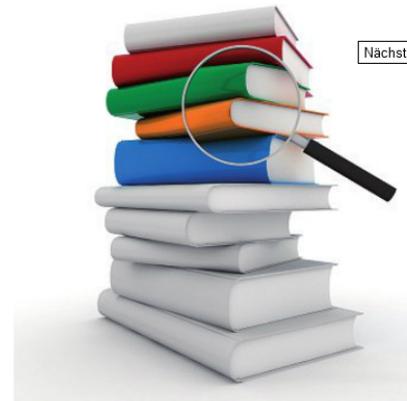
```
SELECT Autor, Titel
FROM Buch
WHERE Autor LIKE ,%Tolk%`;
```

| Autor          | Titel              |
|----------------|--------------------|
| J.R.R. Tolkien | Der Herr der Ringe |

Suche alle Werke, die für 12,95 € oder 13,95 € angeboten werden.

```
SELECT Titel, Preis
FROM Buch
WHERE Preis IN (12.95, 13.95);
```

| Titel          | Preis |
|----------------|-------|
| Der Zauberberg | 12,95 |
| Watchmen       | 13,95 |



Nächst

## Logische Operatoren

Ähnlich wie in anderen Programmiersprachen lassen sich auch in SQL einzelne Bedingungen mithilfe von logischen Operatoren (AND, OR, NOT) miteinander verknüpfen. So können in die WHERE-Klausel mehrere Bedingungen aufgenommen werden.

### Beispiel:

Welche Bücher von Thomas Mann oder Alan Moore werden angeboten?

```
SELECT Titel
FROM Buch
WHERE Autor = ,Thomas Mann' OR Autor = ,Alan Moore';
```



| Titel          |
|----------------|
| Der Zauberberg |
| Watchmen       |

## Verfeinerung der Ausgabe

SQL bietet einige Möglichkeiten, die auszugebenden Tabellen den eigenen Ansprüchen anzupassen. Spaltennamen können mit dem AS-Operator umbenannt werden. Zudem können die Datensätze für die Ausgabe sortiert dargestellt werden. Dies leistet der Befehl ORDER BY. Manchmal ist es wünschenswert, dass in der Ergebnistabelle keine Einträge doppelt vorkommen. Dann wird der DISTINCT-Befehl angewendet.

### Beispiele:

Die folgenden Abfragen beziehen sich auf eine erweiterte Buchtabelle:

| Buch              |                    |                |        |
|-------------------|--------------------|----------------|--------|
| ISBN              | Titel              | Autor          | Preis  |
| 978-3-608-93544-8 | Der Herr der Ringe | J.R.R. Tolkien | 32,90  |
| 978-3-596-29433-6 | Der Zauberberg     | Thomas Mann    | 12,95  |
| 978-1-401-22266-6 | Watchmen           | Alan Moore     | 13,95  |
| 978-3-608-93000-9 | Der Herr der Ringe | J.R.R. Tolkien | 260,00 |
| 978-3-423-59079-3 | Der kleine Hobbit  | J.R.R. Tolkien | 7,95   |
| 978-3-596-29431-2 | Buddenbrooks       | Thomas Mann    | 9,95   |

```
SELECT Titel AS Buchname
FROM Buch
WHERE Autor = ,Alan Moore';
```



| Buchname |
|----------|
| Watchmen |

```
SELECT Titel, Autor, Preis
FROM Buch
WHERE NOT Autor = 'Alan Moore'
ORDER BY Preis ASC;
```

| Titel              | Autor          | Preis  |
|--------------------|----------------|--------|
| Der kleine Hobbit  | J.R.R. Tolkien | 7,95   |
| Buddenbrooks       | Thomas Mann    | 9,95   |
| Der Zauberberg     | Thomas Mann    | 12,95  |
| Der Herr der Ringe | J.R.R. Tolkien | 32,90  |
| Der Herr der Ringe | J.R.R. Tolkien | 260,00 |

Der Zusatz ASC (von engl. ascending – aufsteigend) kann weggelassen werden. Um die Daten absteigend zu sortieren, muss der Zusatz DESC (von engl. descending – absteigend) angefügt werden.

```
SELECT DISTINCT Autor
FROM Buch;
```

| Autor          |
|----------------|
| J.R.R. Tolkien |
| Thomas Mann    |
| Alan Moore     |

→ Aufgabe 1a)

## Aggregatfunktionen und mathematische Operatoren

Mithilfe von Aggregatfunktionen können Auswertungen über alle Datensätze einer Tabelle gemacht werden. So kann mit der Funktion COUNT die Anzahl der Datensätze ermittelt werden. Auf numerischen Attributen können weitere Funktionen angewendet werden: SUM liefert die Summe aller Werte, MAX den größten und MIN den kleinsten Wert der Datenreihe. Zudem kann mit der Funktion AVG das arithmetische Mittel aller Zahlenwerte berechnet werden. Eine Aggregatfunktion liefert als Ergebnis einen Zahlenwert.

Die aus der Sprache Java bekannten Operatoren für die vier Grundrechenarten (+, -, \*, /) sowie die Klammerung von Rechenausdrücken werden auch von SQL unterstützt.

### Beispiele:

Wie viele Bücher von J.R.R. Tolkien werden angeboten?

```
SELECT COUNT (*) AS Anzahl_Tolkien
FROM Buch
WHERE Autor = 'J.R.R. Tolkien';
```

| Anzahl_Tolkien |
|----------------|
| 3              |

Mit dem AS-Befehl wird das Ergebnis der Aggregatfunktion sinnvoll benannt.

Wie viele Autoren sind im Angebot aufgenommen?

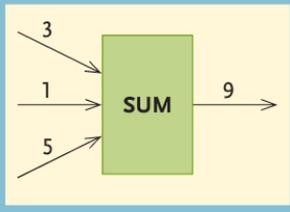
```
SELECT COUNT (DISTINCT Autor) AS Anzahl_Autoren
FROM Buch;
```

| Anzahl_Autoren |
|----------------|
| 3              |



Nächst

Aggregatfunktionen werten eine Gruppe von Werten aus und geben einen einzigen Wert zurück.



→ Aufgabe 1b), c)

Wie viel kostet das günstigste Buch von Tolkien?

```
SELECT MIN(Preis) AS günstigster_Tolkien
FROM Buch
WHERE Autor = ‚J.R.R. Tolkien‘;
```

| günstigster_Tolkien |      |
|---------------------|------|
|                     | 9,95 |

Wie hoch ist der Durchschnittspreis aller angebotenen Bücher?

```
SELECT AVG(Preis) AS Durchschnittspreis
FROM Buch;
```

| Durchschnittspreis |       |
|--------------------|-------|
|                    | 64,75 |

Wie hoch ist jeweils die Mehrwertsteuer der einzelnen Bücher?

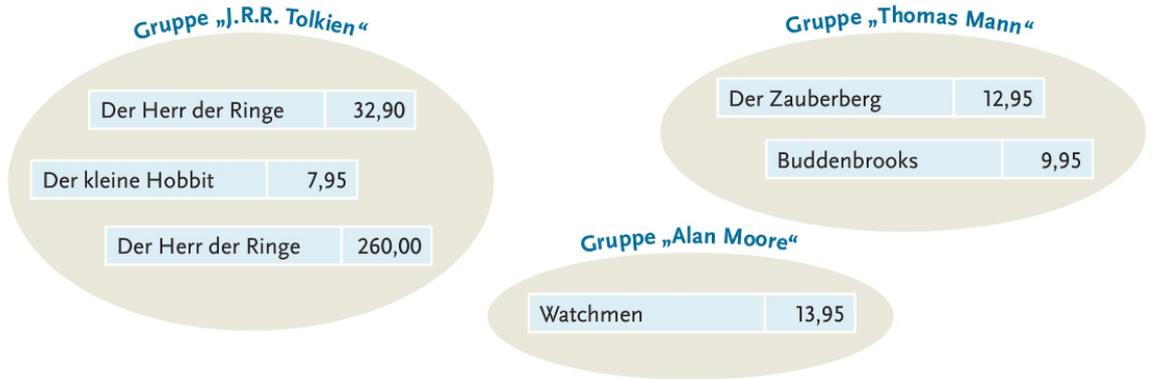
```
SELECT Titel, Preis, Preis * 7/100 AS Mehrwertsteuer
FROM Buch;
```

| Titel              | Preis | Mehrwertsteuer |
|--------------------|-------|----------------|
| Watchmen           | 13,95 | 0,98           |
| Der Zauberberg     | 12,95 | 0,91           |
| Der Herr der Ringe | 32,90 | 2,3            |

### Gruppieren von Datensätzen

Oftmals möchte man Aggregatfunktionen differenziert auf einzelne Gruppen von Datensätzen anwenden. Anstatt nach dem durchschnittlichen Preis aller angebotenen Bücher zu fragen, wäre es auch interessant zu wissen, wie teuer jeweils die Bücher eines bestimmten Autors im Schnitt sind. Dazu müssen die Datensätze zunächst nach dem Merkmal Autor gruppiert werden.

Der Befehl GROUP BY Autor bewirkt, dass die folgenden Gruppen unterschieden werden:



Nun kann die Aggregatfunktion AVG auf das Attribut Preis angewendet werden, um für jede Gruppe den jeweiligen Durchschnittspreis zu ermitteln. Da in der Ergebnistabelle jede der drei Gruppen einen Datensatz darstellen wird, ist dies sogar notwendig, um alle einzelnen Werte der ausgewählten Attribute zu einem einzigen Wert zusammenzuführen. Da dies für das Attribut Titel nicht sinnvoll ist, darf es auch nicht im SELECT-Teil der Abfrage auftauchen:

Wie teuer sind die Bücher der einzelnen Autoren im Schnitt?

```
SELECT Autor, AVG (Preis)
FROM Buch
GROUP BY Autor;
```

Falls gruppiert wird, müssen alle ausgewählten Attribute (außer dem, nach dem gruppiert wird) mit einer Aggregatfunktion versehen werden.

| Autor          | AVG (Preis) |
|----------------|-------------|
| J.R.R. Tolkien | 100,28      |
| Thomas Mann    | 11,45       |
| Alan Moore     | 13,95       |

→ Aufgabe 1d)

## Geschachtelte Select-Ausdrücke

SQL-Abfragen können wiederum selbst als Bedingung für eine weitere Abfrage dienen. Ein solcher geschachtelter SELECT-Ausdruck enthält in der WHERE-Klausel eine Unterabfrage. Unterabfragen werden in runde Klammern gesetzt und vom Interpreter zuerst ausgewertet.

### Beispiele:

Finde alle Bücher, die vom selben Autor geschrieben wurden wie „Der Kleine Hobbit“.

```
SELECT Titel, Autor
FROM Buch
WHERE Autor = (
 SELECT Autor
 FROM Buch
 WHERE Titel =
 ‚Der kleine Hobbit‘);
```

Ergebnis der Unterabfrage

Gesamtergebnis

| Autor          |
|----------------|
| J.R.R. Tolkien |

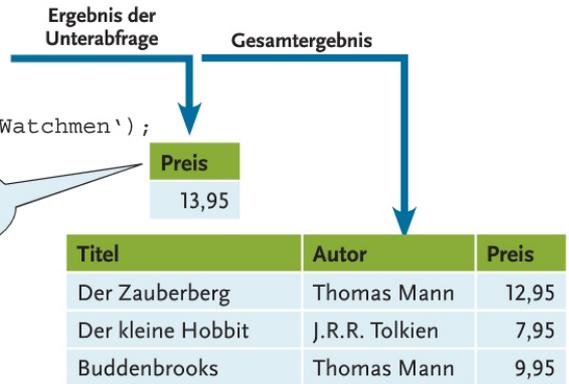
| Titel              | Autor          |
|--------------------|----------------|
| Der Herr der Ringe | J.R.R. Tolkien |
| Der Herr der Ringe | J.R.R. Tolkien |
| Der kleine Hobbit  | J.R.R. Tolkien |

Achtung: Die Unterabfrage darf nur einen einzigen Wert (hier: „J.R.R. Tolkien“) als Ergebnis haben. Ansonsten muss mit dem IN-Operator gearbeitet werden.

Finde alle Bücher, die günstiger sind als Watchmen.

```
SELECT Titel, Autor, Preis
FROM Buch
WHERE Preis < (
 SELECT Preis
 FROM Buch
 WHERE Titel = 'Watchmen');
```

Zudem muss die Unterabfrage immer auf ein einziges Attribut projizieren.



→ Aufgabe 1e), 3 und 4

### Übersicht der SQL-Befehle

Zusammenfassend ergibt sich für SQL-Abfragen folgender grundlegender Aufbau, bei dem die Reihenfolge der Schlüsselwörter fest vorgegeben ist.

| SQL-Befehl                | Reihenfolge bei der Analyse | Beschreibung                                                                                                         |
|---------------------------|-----------------------------|----------------------------------------------------------------------------------------------------------------------|
| SELECT [DISTINCT]         | 5                           | Projiziert auf die angegebenen Spalten. Hier werden auch die Aggregatfunktionen angegeben.                           |
| FROM                      | 1                           | Angabe der Datenquelle, ggf. mehrere Tabellen als Verbund miteinander verknüpft ([LEFT   RIGHT   INNER] JOIN ... ON) |
| WHERE                     | 2                           | Selektiert alle Datensätze, welche die Bedingung erfüllen. Hier können auch Unterabfragen angegeben werden.          |
| GROUP BY                  | 3                           | Gruppiert nach einem Attribut                                                                                        |
| ORDER BY ... [ASC   DESC] | 4                           | Sortiert auf- oder absteigend                                                                                        |

Bei der Analyse einer SQL-Abfrage ist es jedoch sinnvoll, die Befehle nicht sequentiell zu betrachten. Dies soll an einem Beispiel verdeutlicht werden.

1. Aus der Tabelle Buch ...

2. ... wähle alle Datensätze aus, die nicht „Alan Moore“ als Autor haben.

3. Bilde dabei Gruppen nach Autoren ...

```
SELECT Autor, AVG(Preis)
FROM Buch
WHERE Autor <> 'Alan Moore'
GROUP BY Autor
ORDER BY AVG(Preis) DESC;
```

4. ... und sortiere die Ausgabe absteigend nach dem Durchschnittspreis.

5. Projiziere schließlich auf die Spalten Autor und AVG(Preis).

| Autor          | AVG (Preis) |
|----------------|-------------|
| J.R.R. Tolkien | 100,28      |
| Thomas Mann    | 11,45       |

→ Aufgabe 2

## Aufgaben

### 1. Selbstlernen Online

Im Internet finden sich zahlreiche Tutorials zur Erarbeitung von SQL. Für schulische Zwecke eignet sich in besonderer Weise eine Einführung, die von einem Informatik Leistungskurs der Lichtenbergschule Darmstadt entwickelt wurde:

<https://luo-darmstadt.de/sqltutorial/>

Arbeiten Sie die einzelnen Lektionen sowie die dazu passenden Übungen durch und notieren Sie zu jedem benutzten SQL-Befehl seine Wirkung sowie ein Beispiel.

| SQL-SPRACHELEMENTE |            |          |
|--------------------|------------|----------|
| Befehl             | Auswirkung | Beispiel |
|                    |            |          |
|                    |            |          |
|                    |            |          |
|                    |            |          |
|                    |            |          |

- Lektion 1: SELECT-Anweisungen
- Lektion 2: Aggregatfunktionen
- Lektion 3: Joins
- Lektion 4: Gruppierung mit GROUP BY
- Lektion 5: Geschachtelte SELECT-Anweisungen

- Mittels eines elektronischen Komfortprogrammdienstes lässt sich das SiehFern Info (SFI) über eine Satellitenschüssel auf einen TV-Receiver laden und speichern. Mithilfe des SFI lassen sich unterschiedliche Ansichten wählen:



Die Hauptansicht



Die Ansicht kann durch die Auswahl eines Genres verfeinert werden.



Informationen zu einer Sendung

- Erläutern Sie die drei unterschiedlichen Ansichten des SFI. Erklären Sie, dass es möglich ist eine Datenbasis zu nutzen und trotzdem diese drei Ansichten darzustellen.
- Die Daten des SFI liegen in einer Tabelle mit dem folgenden Schema vor: sfi (SendungsID, Sender, Sendung, Datum, Anfang, Ende, Genre, Infotext). Geben Sie beispielhaft vier Datensätze dieser Tabelle an.
- Erläutern Sie die Wirkung der folgenden SQL-Abfragen und geben Sie jeweils die Ergebnistabelle an. Sie können davon ausgehen, dass nur die Sendungen, die in der Hauptansicht abgebildet sind, in der Tabelle sfi gespeichert sind.
  - ```
SELECT Sender
FROM sfi
WHERE Sendung = ,Tagesschau`;
```
 - ```
SELECT Sender, Sendung, Anfang
FROM sfi
WHERE Genre = ,Nachrichten`
AND Anfang >= 10.00
ORDER BY Sender;
```

- ```

iii) SELECT Sender
      FROM sfi
      WHERE Sendung LIKE ,%schau%`
           OR Sendung LIKE ,%nano%`;

iv) SELECT Genre, COUNT(Sendung)
      FROM sfi
      WHERE NOT Sender = ,ZDFHD`
      AND NOT (Genre='Show`
              OR Genre ='Sport`)
      GROUP BY Genre;

v) SELECT DISTINCT Genre
     FROM sfi;

vi) SELECT Sendung, Sender, Anfang
     FROM sfi
     WHERE Anfang > (
         SELECT Anfang
         FROM sfi
         WHERE Sendung = ,Tagesschau`);

```

3. Eine Schülerfirma hat an ihrer Schule einen kleinen Schreibwarenladen eröffnet, der die Schülerinnen und Schüler mit diversen Produkten für den täglichen Gebrauch in der Schule versorgt. Die Firma verwaltet ihr Sortiment in der folgenden Tabelle:



Sortiment					
Artikelnr	Artikelbezeichnung	Rubrik	Menge	Einkaufspreis	Verkaufspreis
10034	Papier A3	Bedruckbares	3	6,07	11,98
10044	Papier A4	Bedruckbares	16	2,91	5,99
10326	Schulheft A4/26	Hefte	30	0,44	0,79
10325	Schulheft A4/25	Hefte	26	0,44	0,79
10966	Hefter A4	Ordnung	39	0,50	0,99
10932	Ordner A4	Ordnung	21	2,04	3,89
22002	Fineliner	Schreibgeräte	80	0,27	0,59
22044	Bleistift	Schreibgeräte	33	0,78	1,09
20003	Kugelschreiber	Schreibgeräte	27	0,70	1,29
33993	Geodreieck	Zeichnen	11	1,80	3,19
33800	Zirkel	Zeichnen	4	5,85	7,79

Entwickeln Sie SQL-Abfragen, die die folgenden Informationen liefern. Geben Sie auch die Ergebnistabellen an.

- Welche Artikel kosten 1 bis 2 €?
- Bei welchen Schreib- oder Zeichengeräten ist der Bestand bereits auf unter 30 Stück zurückgegangen?

- c) Bei welchen Artikeln (Angabe von Artikelnr, Artikelbezeichnung, Verkaufspreis, Gewinn) verdient die Schülerfirma mehr als 1,00 €? Geben Sie die Artikel absteigend sortiert nach dem Gewinn aus.
- d) Welche Rubriken bietet der Schreibwarenladen an?
- e) Wie viele verschiedene Artikel werden angeboten?
- f) Welche Artikel sind im Einkauf günstiger als ein Schulheft?
- g) Bei welchem Artikel ist die Gewinnmarge am größten?
- h) In welcher Rubrik werden die meisten unterschiedlichen Artikel angeboten?

4. Weiterführung der Projektaufgabe: World Factbook

Entwickeln Sie weitere SQL-Abfragen für die Tabelle Country. Testen Sie Ihre Abfragen jeweils im DBMS.

- a) Geben Sie die Zahl der Internetnutzer in Deutschland, Frankreich und Österreich aus.
- b) In welchen Ländern ist der Anteil der Internetnutzer an der Gesamtbevölkerung kleiner als 20 %? Geben Sie diese Zahl als Anteil_Internetnutzer aus.
- c) Bestimmen Sie die durchschnittliche Bevölkerungsanzahl der Länder Deutschland, Frankreich und Spanien.
- d) Welche Länder sind flächenmäßig größer als Deutschland?
- e) Überlegen Sie sich weitere Abfragen an die Datenbank und formulieren Sie diese als SQL-Abfrage. Notieren Sie die umgangssprachliche Formulierung Ihrer Abfrage auf einem Zettel und geben Sie sie anderen Gruppen zur Bearbeitung.