

Name:

Klasse:

Datum:

Schuljahr: 2021/22

**Wiederholung: Grundstruktur von einfachen SQL-Abfragen**

Für alle nachfolgenden Aufgaben verwenden Sie die Datenbank gm3 und spielen die Daten dump\_gm3\_start.sql ein!

Zur Wiederholung, die vereinfachte Ausführung sieht dabei wie folgt aus:

```
SELECT select_expr, ...  
  [FROM table_references]  
  [WHERE where_condition]  
  [ORDER BY {col_name | expr | position} [ASC | DESC], ...]  
  [LIMIT {[offset,] row_count | row_count OFFSET offset}]
```

Beispiel:

```
SELECT name FROM mitarbeiter ORDER BY 1 LIMIT 5;
```

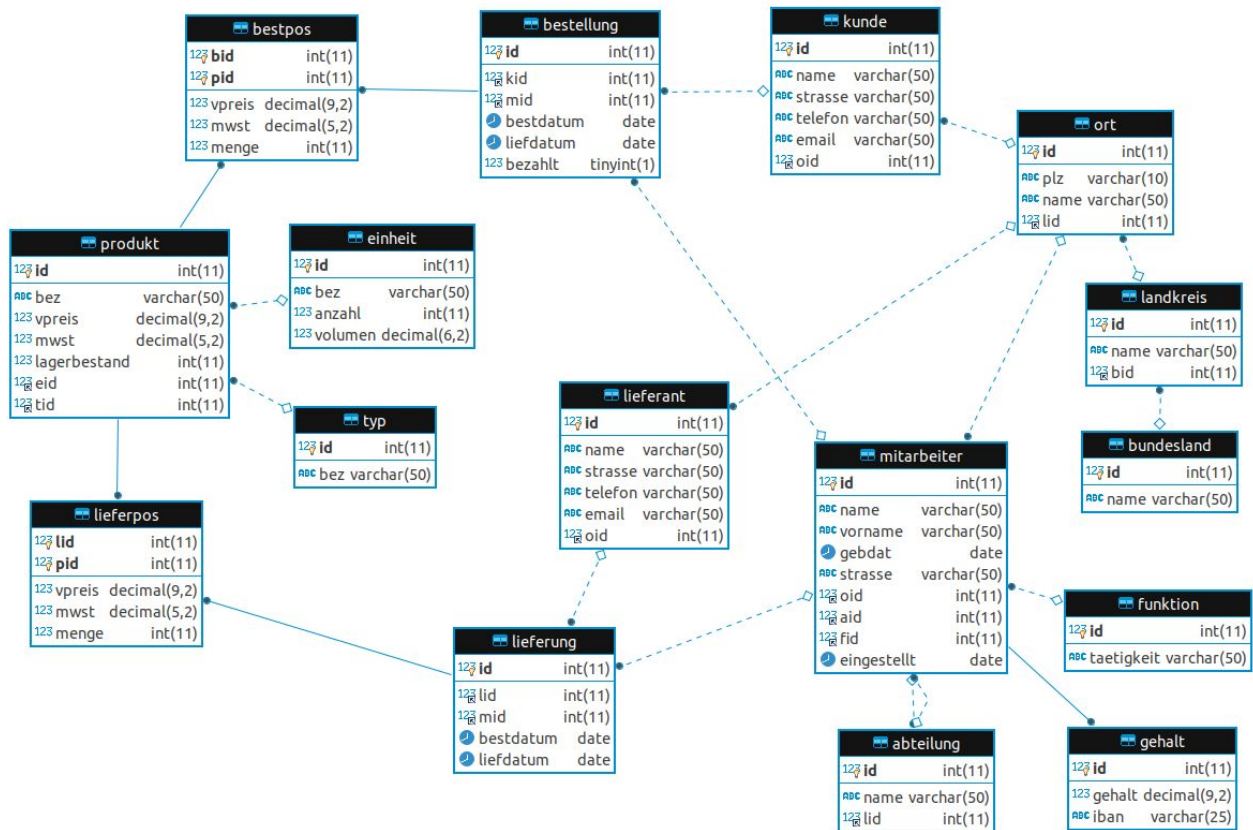
**Aufgaben**    **AUF-07-1-1**

1. Informieren Sie sich über die SQL-Schlüsselwörter SELECT, FROM, WHERE, ORDER BY und LIMIT. Gehen Sie dabei auf folgende Begriffe bzw. Schlüsselwörter ein: Selektion, Projektion, \*, AND, OR, ASC und DESC.

- SELECT
  
  
  
  
  
- FROM
  
  
  
  
  
- WHERE
  
  
  
  
  
- ORDER BY
  
  
  
  
  
- LIMIT
  
  
  
  
  
- Beschreiben Sie in eigenen Worten das Ergebnis der obigen Beispielabfrage.



## Tabellenmodell für den Getränkemarkt – gm3



MariaDB-Version:

```
SELECT version();
```

```

+-----+
| version() |
+-----+
| 10.3.34-MariaDB-0ubuntu0.20.04.1 |
+-----+
    
```



## SQL-Abfragen für die Personalabteilung AUF-07-1-2

Die Personalabteilung benötigt zur Organisation einige Informationen über die Mitarbeiter. Helfen Sie der Abteilung, indem Sie die benötigten SQL-Befehle herausfinden und diese notieren. Notieren Sie sich auch Informationen über neue Schlüsselwörter, die Sie benötigt haben.

1. In welcher Straße (incl. Hausnummer) wohnt Frau Sophia Lorenz?

### Kontrollergebnis

```
+-----+
| strasse |
+-----+
| Hammer Str. 349 |
+-----+
```

2. Gesucht ist das Einstellungsdatum von Sonja Kaufmann und Michael Wolff.

### Kontrollergebnis

```
+-----+-----+-----+
| vorname | name      | eingestellt |
+-----+-----+-----+
| Sonja   | Kaufmann  | 2001-08-01  |
| Michael | Wolff     | 2017-06-01  |
+-----+-----+-----+
```

3. Erstellen Sie eine alphabetisch sortierte Liste der Mitarbeiter (Name und Vorname). Dabei soll nach dem Nachnamen absteigend sortiert werden. Bei gleichen Nachnamen soll aufsteigend nach dem Vornamen sortiert werden.

### Kontrollergebnis (Auszug)

```
+-----+-----+
| Name    | Vorname   |
+-----+-----+
| Wolff   | Bettina   |
| Wolff   | Michael   |
| Wolff   | Theodor   |
| Wieland | Brunhilde |
| Weber   | Karl-Heinz |
+-----+-----+
```

4. Erstellen Sie eine Liste aller Mitarbeiter mit Name und Vorname. Dabei soll zwischen Vor- und Nachname ein „ „ gesetzt werden. Benennen Sie die Spalte entsprechend dem Kontrollergebnis. Hinweis: CONCAT, Schlüsselwort AS in SELECT



#### Kontrollergebnis (Auszug)

```
+-----+
| Nachname, Vorname |
+-----+
| Lorenz, Sophia    |
| Ritter, Tatjana   |
| Wolff, Theodor    |
+-----+
```

5. Erstellen Sie eine Liste mit Vor-, Nachname und Geburtsdatum der Mitarbeiter. Die Liste soll dabei nach ihrem Alter aufsteigend sortiert sein.  
Hinweis: Sie müssen dafür das Alter **nicht** berechnen.

#### Kontrollergebnis (Auszug)

```
+-----+-----+-----+
| name  | vorname | gebdat |
+-----+-----+-----+
| Huber | Sepp    | 2001-04-04 |
| Wolff | Michael | 2000-01-13 |
| Schlau | Susi    | 1999-10-14 |
+-----+-----+-----+
```

6. Welche Mitarbeiter (Id, Name, Vorname und eingestellt) wurden vor dem 01.01.2005 eingestellt?  
Hinweis: Datentyp DATE

#### Kontrollergebnis (Auszug)

```
+-----+-----+-----+-----+
| id | name  | vorname | eingestellt |
+-----+-----+-----+-----+
| 1  | Lorenz | Sophia  | 2000-01-01 |
| 2  | Ritter | Tatjana | 2004-03-01 |
| 3  | Wolff  | Theodor | 2002-04-15 |
| 4  | Richter | Hans-Otto | 2004-03-01 |
| 8  | Hagen  | Friedhelm | 2000-07-15 |
+-----+-----+-----+-----+
```

7. Welche Mitarbeiter (Id, Name, Vorname und eingestellt) wurden im Jahr 2006 eingestellt?  
Hinweis: YEAR)



### Kontrollergebnis (Auszug)

id	name	vorname	eingestellt
5	Wieland	Brunhilde	2006-12-01
6	Wolff	Bettina	2006-01-01
11	Hoelzer	Richard	2006-01-01
19	Hoffmann	Theresa	2006-02-01
26	Hannes	Klaus-Dieter	2006-08-01
28	Lindner	Dieter	2006-08-01

8. Welche Mitarbeiter (Id, Name, Vorname, eingestellt) wurden im Jahr 2001 zwischen April und Oktober eingestellt?

### Kontrollergebnis (Auszug)

id	name	vorname	eingestellt
18	Kaufmann	Sonja	2001-08-01
24	Lauterbach	Wilma	2001-05-01

9. Gesucht ist eine Liste aller Mitarbeiter (Id, Name und Vorname) und ihr dazugehöriges Alter. Achten Sie darauf, dass dieser SQL-Befehl immer das richtige Alter bestimmt und bei mehrfacher Ausführung zu unterschiedlichen Zeitpunkten nicht angepasst werden muss.  
Hinweis: TimeStampDiff

### Kontrollergebnis (Auszug)

id	name	vorname	Alter
1	Lorenz	Sophia	46
2	Ritter	Tatjana	43
3	Wolff	Theodor	37
4	Richter	Hans-Otto	55
5	Wieland	Brunhilde	33

10. **Optional:** für Experten

Die Funktion TimeStampDiff gibt es nicht in allen SQL-Dialekten. Folgender Befehl für die Altersberechnung funktioniert unabhängig für jeden SQL-Dialekt. Dabei wird verwendet, dass der boolesche Wert für die Bedingung intern als Zahl repräsentiert wird (1=TRUE, 0=FALSE).

```
USE gm3;
SELECT id, name, vorname,
       YEAR(CURDATE()) - YEAR(gebdat) -
       (DATE_FORMAT(CURDATE(), '%m%d') < DATE_FORMAT(gebdat, '%m%d'))
```



```
AS 'Alter'
FROM mitarbeiter LIMIT 5;
```

#### Kontrollergebnis (Auszug)

id	name	vorname	Alter
1	Lorenz	Sophia	46
2	Ritter	Tatjana	43
3	Wolff	Theodor	37
4	Richter	Hans-Otto	55
5	Wieland	Brunhilde	33

Informieren Sie sich über den den Befehl DATE\_FORMAT und erklären Sie diese Altersberechnung.

#### 11. **Optional:** für Experten

Eine einfachere Berechnung für das Alter, die meistens genau genug ist, ist folgender SQL-Befehl:

```
USE gm3;
SELECT id, name, vorname,
       TRUNCATE(DATEDIFF(CURDATE(), gebdat) / 365.25, 0) AS 'Alter'
FROM mitarbeiter LIMIT 5;
```

#### Kontrollergebnis (Auszug)

id	name	vorname	Alter
1	Lorenz	Sophia	46
2	Ritter	Tatjana	43
3	Wolff	Theodor	37
4	Richter	Hans-Otto	55
5	Wieland	Brunhilde	33

Informieren Sie sich über die Befehle TRUNCATE und DATEDIFF und erklären Sie diese Altersberechnung. Gehen Sie dabei auch auf die Zahl 365.25 ein.





## SQL-Abfragen für die Verkaufsabteilung AUF-07-1-3

Die Verkaufsabteilung hat von Ihrer erfolgreichen Unterstützung der Personalabteilung gehört und bittet Sie für folgende Informationen nun ebenfalls um Hilfe.

1. Erstellen Sie eine Liste aller Produkte (Id, Bezeichnung, Verkaufspreis, Einheit-Id, Typ-Id), die in der Bezeichnung das Wort „Wein“ enthalten.

Hinweis: Schlüsselwort LIKE)

### Kontrollergebnis

id	bez	vpreis	eid	tid
269	Alter Hochstädter Speyerling Apfelwein	10.95	20	36
270	Müller Odenwald Goldernte Apfelwein	10.95	20	36
271	Possmann Frankfurter Apfelwein	11.95	20	36
272	Rapp's Nr.1 Apfelwein alkoholfrei	11.95	20	37

2. Ein Kunde aus Dänemark möchte einen Überblick über die Produkte. Erstellen Sie eine Liste mit Produktbezeichnung und dem dazugehörigen Preis. Der Preis soll dabei in Dänische Kronen (DKK: Faktor zur Zeit 7,44) umgerechnet und auf zwei Nachkommastellen gerundet werden. Verwenden Sie dazu ROUND().

### Kontrollergebnis (Auszug)

bez	Preis in DKK
Binding Export	114.95
Dachsenfranz Kellerbier Bügelflasche	92.63

3. Gesucht sind Id, Bezeichnung und Verkaufspreis des teuersten Produkts. Verwenden Sie hierzu order by und limit.

Gibt es etwas worauf bei der Lösung geachtet werden muss?

### Kontrollergebnis

id	bez	vpreis
228	Alwa Brewed Soda Limette-Ingwer	25.95



#### 4. **Optional:** für Experten

Um einen besseren Überblick über den Lagerbestand der Produkte für den Einkauf zu bekommen, soll eine Liste aller Produkte (Id, Bezeichnung) zusammen mit folgender Information über den Lagerbestand angezeigt werden:

- Das Wort **Ausreichend** wenn der Lagerbestand mehr als 10 beträgt.
- Das Wort **Gering** wenn der Lagerbestand zwischen 1 und 10 beträgt.
- Das Wort **Ausverkauft** wenn der Lagerbestand 0 beträgt.

Sortieren Sie dabei das Ergebnis nach der Produkt-ID aufsteigend.

Hinweis: IF

#### Kontrollergebnis (Auszug)

id	bez	Lagerbestand
1	Binding Export	Ausreichend
2	Dachsenfranz Kellerbier Bügelflasche	Ausverkauft
3	Eichbaum Export	Ausreichend
4	Heidelberger Export	Ausreichend
5	Kurpfalz Bräu Kellerbier	Ausreichend
6	Welde Export	Gering





## Verknüpfung von Tabellen

Es gibt mehrere Möglichkeiten, zwei oder mehrere Tabellen miteinander zu verbinden. Dies wird in SQL als `JOIN` bezeichnet, allgemein als Verbund.

`INNER JOIN` Ist eine Verknüpfung „innerhalb“ zweier Tabellen, d. h. der Wert der Verknüpfung muss in beiden Tabellen vorhanden sein.

`OUTER JOIN` Ist eine Verknüpfung, bei der auch Datensätze geliefert werden, die die Auswahlbedingung nicht erfüllen. Das Wort „OUTER“ kann auch entfallen. Dazu zählen:

`LEFT JOIN` Alle Werte aus der linken Tabelle, verknüpft mit der rechten Tabelle. Fehlende Verknüpfungen werden mit „null“ aufgefüllt.

`RIGHT JOIN` Alle Werte aus der rechten Tabelle, verknüpft mit der linken Tabelle. Fehlende Verknüpfungen werden mit „null“ aufgefüllt.

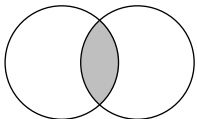
`FULL JOIN` Liefert alle Datensätze beider Tabellen. Fehlende Verknüpfungen werden mit „null“ aufgefüllt.

`SELF JOIN` Eine Tabelle wird mit sich selbst verknüpft.

### INNER JOIN

Ein `INNER JOIN` (auch natürlicher Verbund genannt – Equivalent Join), verknüpft zwei Tabellen über ein gemeinsames Attribut. Es werden jeweils Zeilen mit gleichen Attributwerten zu einer neuen verbunden.

Betrachtet man dies mit den Möglichkeiten der Mengenlehre, so wird die Schnittmenge gebildet.



### Aufgaben zu INNER JOIN [AUF-07-1-4](#)

Die Personalabteilung benötigt erneut Ihre Hilfe. Diesmal soll eine Übersicht über die Mitarbeiter und ihrer dazugehörigen Abteilung erstellt werden:

name	vorname	Abteilungsname
Gehrke	Anna-Marie	Buchhaltung
Reibach	Bernd	Hausmeister
Wolff	Bettina	Fahrdienst
Keller	Brigitte	Buchhaltung
Wieland	Brunhilde	Buchhaltung

Das Problem bei dieser Abfrage ist, dass für diese Liste Informationen aus zwei Tabellen benötigt werden. Betrachtet man folgende Auszüge aus den Tabellen `mitarbeiter` und `abteilung`, benötigt man die `aid`, um in der Tabelle `abteilung` den entsprechenden Namen herauszufinden.

id	name	vorname	aid
22	Gehrke	Anna-Marie	4
32	Reibach	Bernd	5

id	name
1	Einkauf
2	Verkauf
3	Fahrdienst



4	Buchhaltung	
5	Hausmeister	
6	Geschäftsleitung	
7	IT	
+-----+		

Folgender Befehl verknüpft die beiden Tabellen bzgl. der `id`. Damit bei gleichen Spaltennamen in den zwei Tabellen nicht immer der komplette Tabellename angegeben werden muss, wird jeweils ein Alias für die Tabelle definiert.

```
USE gm3;
SELECT m.name, vorname, a.name as 'Abteilungsname'
FROM mitarbeiter m INNER JOIN abteilung a ON a.id=m.aid
WHERE m.name='Gehrke' OR m.name='Reibach';
```

Als Ergebnis erhält man folgende Tabelle:

+-----+		
name	vorname	Abteilungsname
+-----+		
Gehrke	Anna-Marie	Buchhaltung
Reibach	Bernd	Hausmeister
+-----+		

Eine Alternative Möglichkeit ist die Verwendung eines Kreuzproduktes:

```
USE gm3;
SELECT m.id, m.name, m.vorname, m.aid, a.id, a.name
FROM mitarbeiter m, abteilung a WHERE m.name='Gehrke' OR m.name='Reibach';
```

+-----+						
id	name	vorname	aid	id	name	
+-----+						
22	Gehrke	Anna-Marie	4	1	Einkauf	
22	Gehrke	Anna-Marie	4	2	Verkauf	
22	Gehrke	Anna-Marie	4	3	Fahrdienst	
22	Gehrke	Anna-Marie	4	4	Buchhaltung	
22	Gehrke	Anna-Marie	4	5	Hausmeister	
22	Gehrke	Anna-Marie	4	6	Geschäftsleitung	
22	Gehrke	Anna-Marie	4	7	IT	
32	Reibach	Bernd	5	1	Einkauf	
32	Reibach	Bernd	5	2	Verkauf	
32	Reibach	Bernd	5	3	Fahrdienst	
32	Reibach	Bernd	5	4	Buchhaltung	
32	Reibach	Bernd	5	5	Hausmeister	
32	Reibach	Bernd	5	6	Geschäftsleitung	
32	Reibach	Bernd	5	7	IT	
+-----+						

1. Erklären Sie diese Ausgabe.  
Hinweis: Kreuzprodukt/Cross-Join)
2. Dieses Ergebnis kann jedoch mit einer einzigen Bedingung in der `WHERE`-Klausel benutzt werden, die mit `AND` „angehängt“ wird, um nun die korrekten Verbindungen zwischen den Tabellen `mitarbeiter` und `abteilung` zu bestimmen.  
Wie lautet die entsprechende Bedingung?



3. Da sich aktuell sehr viele Mitarbeiter im Home-Office befinden, möchte die Personalabteilung ebenso die Gehaltsabrechnungen per Post zu den Mitarbeitern nach Hause versenden. Sie benötigt dafür die Vor- und Nachnamen jedes Mitarbeiters, deren Gehalt und deren jeweilige Anschrift (Straße, PLZ und Ort).

Wie lautet der entsprechende SQL-Befehl für eine solche Liste?

#### Kontrollergebnis (Auszug)

vorname	name	gehalt	strasse	plz	name
Sophia	Lorenz	1250.00	Hammer Str. 349	48153	Münster
Tatjana	Ritter	2252.00	Austermannstraße 75	48149	Münster
Theodor	Wolff	1327.00	Telgenkamp 13	48268	Greven
Hans-Otto	Richter	3250.00	Königsberger Str. 41	48157	Münster



#### 4. **Optional:** für Experten

Der Getränkemarkt möchte die Kunden und Lieferanten über Neuigkeiten telefonisch informieren. Dazu soll eine gemeinsame Liste aller Kunden- und Lieferantennamen mit der dazugehörigen Telefonnummer erstellt werden. Erstellen Sie eine entsprechende Liste mit einem passenden SQL-Befehl.

Hinweis: UNION

Kontrollergebnis (Auszug)

name	telefon
Augustiner Brauerei GmbH	089519940
Red Bull AG	0891234567
Coca Cola Deutschland AG	08002223232
Winzer Franke KG	06713435667
Adelholzener Alpenquellen GmbH	08662 620
Diskotheek Blue	(06241)102222

#### 5. **Optional:** für Experten

Erweitern Sie die Liste um eine weitere Spalte `typ`, in welcher das Zeichen `L` steht, falls es sich bei dem entsprechenden Datensatz um einen Lieferanten handelt und das Zeichen `K`, falls es sich um einen Kunden handelt. Dabei soll die Liste nach dem Namen sortiert werden.

Kontrollergebnis (Auszug)

name	telefon	typ
Adelholzener Alpenquellen GmbH	08662 620	L
Augustiner Brauerei GmbH	089519940	L
BASIS und Partner	(04999)832147	K
Biergarten Waldesruh	(03569)723633	K
Café Maldaner	(07891)444111	K
Club Rock me	(05051)1212	K



## Übungen **AUF-07-1-5**

Da Sie nun die Grundlage über Verknüpfungen von Tabellen kennengelernt haben, sollen Sie einige Informationen für die Personal- bzw. Verkaufsabteilung herausfinden. Formulieren Sie dafür die verbal formulierten Anfragen in SQL.

1. Welche unterschiedlichen Typ-Bezeichnungen gibt es für Coca-Cola Produkte?

Hinweis: Schlüsselwort DISTINCT

### Kontrollergebnis

```
+-----+
| bez                |
+-----+
| Glasflasche 0,2 Liter |
| Glasflasche 0,33 Liter |
| Glasflasche 0,5 Liter |
| PEW Flasche 0,5 Liter |
| Pet Flasche 1,0 Liter |
+-----+
```

2. Gesucht sind alle Kunden (Id, Name) incl. Lieferdatum, die vor über einem Monat oder mehr ihre Bestellung erhalten haben, aber noch nicht bezahlt haben. Sortieren Sie dabei nach dem Kundennamen alphabetisch aufsteigend.

### Kontrollergebnis (Auszug)

```
+-----+-----+-----+
| id | name                | liefdatum |
+-----+-----+-----+
| 20 | BASIS und Partner    | 2021-10-28 |
| 20 | BASIS und Partner    | 2021-10-31 |
| 8  | Biergarten Waldesruh | 2021-11-25 |
| 15 | Café Maldaner        | 2021-09-10 |
+-----+-----+-----+
```

3. Welche unterschiedlichen Mitarbeiter (Id, Name und Vorname) betreuten Lieferungen vom Lieferanten „Red Bull AG“? Sortieren Sie dabei nach dem Nachnamen des Mitarbeiters alphabetisch aufsteigend.



#### Kontrollergebnis (Auszug)

id	name	vorname
13	Beck	Rafaela
23	Berger	Ludwig
14	Collmar-Schmidt	Nadine

4. Welche unterschiedlichen Mitarbeiter (Id, Name und Vorname) betreuen Bestellungen von Kunden aus Greven? Sortieren Sie dabei nach dem Nachnamen des Mitarbeiters alphabetisch aufsteigend.

#### Kontrollergebnis (Auszug)

id	name	vorname
21	Baehr	Konrad
16	Hellmeister	Sepp
11	Hoelzer	Richard

5. Geben Sie alle Lieferanten (Id, Name, Bundesland) an, die aus Bayern kommen.

#### Kontrollergebnis

id	name	Bundesland
1	Augustiner Brauerei GmbH	Bayern
2	Red Bull AG	Bayern
5	Adelholzener Alpenquellen GmbH	Bayern

6. Bei der Eingabe von Coca-Cola Bestellungen wurden Fehler gemacht. Finden Sie deswegen heraus, welche unterschiedlichen Lieferanten (Id, Name) laut System Coca-Cola Produkte in 0,5l Glasflaschen lieferten?





### Kontrollergebnis

id	name
2	Red Bull AG
4	Winzer Franke KG
5	Adelholzener Alpenquellen GmbH

7. Gesucht ist der Vor- und Nachname der Abteilungsleiterin von Mitarbeiterin Tatjana Ritter.  
Hinweis: Wird zweimal die selbe Tabelle in der FROM-Klausel verwendet, so müssen für die beiden Tabellen zwingend Aliase verwendet werden.

### Kontrollergebnis

vorname	name
Theresa	Hoffmann



## 8. **Optional:** für Experten (Anspruchsvoll!)

Für Produktanalysen möchte man herausfinden, welche Produkte häufig gegenseitig zusammen gekauft wurden. Dafür hat ein Kollege für jedes Produkt schon herausgefunden, welches andere Produkt am häufigsten mit diesem Produkt gekauft wurde. Dieses Ergebnis hat er in einer View Produktanalyse (temporäre bzw. virtuelle Tabelle, welche wie eine normale Tabelle in SQL-Abfragen verwendet werden kann.) festgehalten. Machen Sie sich einen Überblick über Inhalt und Aufbau der View Produktanalyse.

```
USE gm3;
CREATE OR REPLACE VIEW produktanalyse AS
  SELECT DISTINCT b1.pid AS 'pid',
    (SELECT b3.pid FROM
      (bestpos b2 JOIN bestpos b3)
      WHERE b2.bid=b3.bid AND b1.pid=b2.pid AND b2.pid<>b3.pid
      GROUP BY b2.pid, b3.pid
      ORDER BY count(0) DESC, 1
      LIMIT 1)
    AS 'HaeufigGekauftMit' FROM bestpos b1;
SELECT * FROM produktanalyse LIMIT 10;
```

Führen Sie dazu zuvor das SQL-Skript `create_view_gm3.sql` aus.

### Kontrollergebnis (Auszug)

pid	HaeufigGekauftMit
1	69
2	28
3	48
4	9
5	88
7	196
8	31
9	251
10	79
11	61

## 9. **Optional:** für Experten (Anspruchsvoll!)

Erstellen Sie nun eine Liste von Produkt-ID's (sortiert nach `pID1`), welche gegenseitig häufig zusammen gekauft worden sind. Achten Sie darauf, dass jedes Produktpaar nur einmal im Ergebnis steht (Hinweis: Verwenden Sie hierzu `p1.pid>p2.pid`).



### Kontrollergebnis (Auszug)

pID1	pID2
24	15
30	21
31	8
47	13
48	3
54	23
61	11
65	45
67	27
69	1

#### 10. **Optional:** für Experten (Anspruchsvoll!)

Passen Sie nun die Abfrage so an, dass anstelle der Produkt-ID's die Produktbezeichnungen angezeigt werden (sortiert nach Produkt1).

### Kontrollergebnis (Auszug)

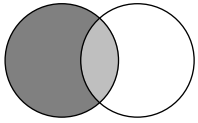
Produkt1	Produkt2
Alwa Naturelle PET	Coca-Cola
Bitburger Pils	Heidelberger Pils
Bitburger Pils Stubbi	Augustiner Hell
Clausthaler Premium alkoholfrei	Kurpfalzbräu Helles
Eichbaum Radler	Binding Export
Ensinger Sport Still	Rothaus Zäpfle Hefeweizen
Fanta	Kurpfalz Bräu Kellerbier
Fanta Orange	Fürstenberg Pils
Heidelberger Winterbier	Franziskaner Weißbier Hefe hell
Hirschquelle	Selters Classic PET

#### 11. Aus welchem Grund wurde für Produktanalyse eine View verwendet und keine Tabelle?

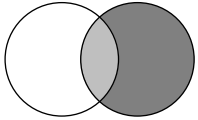


## LEFT/RIGHT JOIN

Beim `LEFT JOIN` (auch `LEFT OUTER JOIN`) werden alle Zeilen der „linken“ Tabelle in die Ergebnistabelle übernommen, auch wenn es zu ihnen keine entsprechenden Zeilen in der „rechten“ Tabelle gibt. Fehlende Werte werden mit `NULL` belegt.

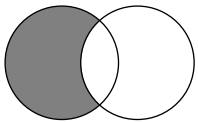


Beim `RIGHT JOIN` (auch `RIGHT OUTER JOIN`) werden alle Zeilen der „rechten“ Tabelle in die Ergebnistabelle übernommen, auch wenn es zu ihnen keine entsprechenden Zeilen in der „linken“ Tabelle gibt. Fehlende Werte werden mit `NULL` belegt. Ein `RIGHT JOIN` kann auch mit einem `LEFT JOIN` realisiert werden, indem man die beiden Tabellen vertauscht.



## LEFT JOIN mit Bedingung

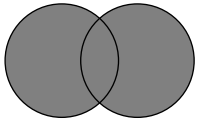
Damit man nun mit dem `LEFT JOIN` alle die Datensätze bekommt, die **nicht** der Verknüpfung entsprechen (also kein entsprechender Eintrag vorhanden ist), wird zusätzlich mit der `WHERE`-Bedingung auf `NULL` geprüft (Achtung: mit `IS NULL`).



## FULL JOIN

Beim `FULL JOIN` (auch `FULL OUTER JOIN`) werden alle Datensätze beider Tabellen verknüpft. Fehlende Verknüpfungen werden mit „null“ aufgefüllt. Ein `FULL JOIN` kann auch durch eine Kombination von `LEFT JOIN` und `RIGHT JOIN` mittels `UNION` erreicht werden.

MySQL/MariaDB unterstützen diesen Typ nicht; er wird auch in der Praxis nur sehr selten gebraucht.



## Outer Join AUF-07-1-6

Die Verkaufsabteilung möchte herausfinden, welche Produkte des Getränkemarkts noch nie von Kunden gekauft worden sind.

Für diese Problemstellung kann kein INNER JOIN verwendet werden, da diese gesuchten Produkt-ID's nicht in der Tabelle `bestpos` als FK's auftauchen.

Diese können zum Beispiel folgendermaßen verwendet werden:

```
SELECT p.id, p.bez, b.pid, b.bid
FROM produkt p LEFT JOIN bestpos b ON p.id=b.pid;
```

1. Dies führt beispielsweise zu folgendem Ergebnis:

Kontrollergebnis (Auszug):

id	bez	pid	bid
1	Binding Export	1	25
2	Dachsenfranz Kellerbier Bügelflasche	2	28
3	Eichbaum Export	3	95
6	Welde Export	nul	nul
17	Bitburger Pils	nul	nul
41	Andechser Weissbier hell	nul	nul

Was bedeuten die null-Werte in den Spalten `pid` und `bid`?

Hinweis: Suchen Sie in der Tabelle `bestpos` nach den entsprechenden Produkt-ID's.

2. Erklären Sie anhand des Ergebnisses die Funktionsweise eines Left bzw. Right Join's.

3. Mit welcher Bedingung kann der obige SQL-Befehl angepasst werden, so dass nur noch die Produkte angezeigt werden, die noch **nie** gekauft worden sind?

Wie lautet der SQL-Befehl dazu?

Kontrollergebnis (Auszug)

id	bez	pid	bid
6	Welde Export	nul	nul
17	Bitburger Pils	nul	nul
41	Andechser Weissbier hell	nul	nul



#### 4. **Optional:** für Experten (Anspruchsvoll!)

Gesucht ist eine Übersicht aller Mitarbeiter mit ihrer dazugehörigen Funktion (taetigkeit). Außerdem sollen alle Funktionen im Ergebnis zu finden sein.

Formulieren Sie einen entsprechenden SQL-Befehl. Informieren Sie sich dazu über den FULL OUTER JOIN. Warum wird dieser in MySQL bzw. MariaDB nicht direkt unterstützt, sondern bedarf eines „Workaround“?

##### Kontrollergebnis (Auszug)

id	name	taetigkeit
33	Walker	null
34	Schlau	null
nu	null	Praktikant
1	Lorenz	Einkaeufer
18	Kaufmann	Einkaeufer
39	Meier	Einkaeufer





## Left- bzw. Right Join anwenden AUF-07-1-7

1. Welche Kunden (Id, Name) haben noch nie eine Bestellung aufgegeben?

### Kontrollergebnis

id	name
22	Restaurant im Schloss Biebrich

2. Welche Mitarbeiter (Id, Name, Vorname) haben noch nie eine Bestellung betreut? Sortieren Sie das Ergebnis aufsteigend nach der Mitarbeiter-ID. Verwenden Sie für diese Aufgabe einen Right-Join.

### Kontrollergebnis (Auszug)

id	name	vorname
4	Richter	Hans-Otto
5	Wieland	Brunhilde
7	Weber	Karl-Heinz
8	Hagen	Friedhelm

3. Erstellen Sie eine Liste aller Mitarbeiter (Nachname, Vorname) mit ihrer dazugehörigen Funktionsbeschreibung (Tätigkeit). Benennen Sie die Spalte der Funktionsbeschreibung gemäß des Kontrollergebnisses. Sollte einem Mitarbeiter keine Funktion zugeordnet sein, so soll der Eintrag in der Tätigkeitsspalte leer bleiben. Sortieren Sie dabei das Ergebnis nach der Funktionsbeschreibung so, dass die Mitarbeiter ohne Funktionsbeschreibung ganz oben stehen.

Hinweis: ifnull()

### Kontrollergebnis (Auszug)

name	id	Aktuelle Tätigkeit
Schlau	34	
Walker	33	
Kaiser	31	CEO
Reibach	32	CEO
Meier	39	Einkaeufer
Mair	40	Einkaeufer



4. Gesucht sind alle Bundesländer (Name), in denen Kunden existieren. Sortieren Sie das Ergebnis nach dem Namen alphabetisch aufsteigend.

#### Kontrollergebnis

name
Baden-Württemberg
Bayern
Niedersachsen
Nordrhein-Westfalen

5. Gesucht sind alle Bundesländer (Name), in denen **noch keine** Kunden existieren. Sortieren Sie das Ergebnis nach dem Namen alphabetisch aufsteigend.

#### Kontrollergebnis

name
Berlin
Brandenburg
Bremen
Hamburg
Hessen
Mecklenburg-Vorpommern
Rheinland-Pfalz
Saarland
Sachsen
Sachsen-Anhalt
Schleswig-Holstein
Thüringen



## Abschließende Wiederholung **AUF-07-1-8**

Gegeben sind folgende Tabellen:

**T1**

id	name
1	a
2	b
3	c

**T2**

id	wert
1	x
3	y
5	z

Geben Sie zu den folgenden SQL-Abfragen die Ergebnistabellen (incl. Spaltenüberschriften) an.

1. `SELECT * FROM T1, T2;`

2. `SELECT * FROM T1 INNER JOIN T2 ON T1.id=T2.id;`

3. `SELECT * FROM T1 LEFT JOIN T2 ON T1.id=T2.id;`

4. `SELECT * FROM T1 RIGHT JOIN T2 ON T1.id=T2.id;`

