

Name: Klasse: Datum:

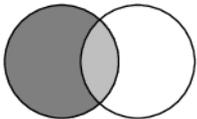
Schuljahr: 2022/23

Verwendete DB-Version:

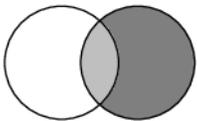
Version MariaDB	Version gm3
10.6.7-MariaDB-2ubuntu1.1	2022-08-12

LEFT/RIGHT JOIN

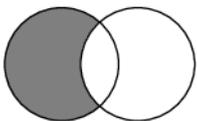
Beim `LEFT JOIN` (auch `LEFT OUTER JOIN`) werden alle Zeilen der „linken“ Tabelle in die Ergebnistabelle übernommen, auch wenn es zu ihnen keine entsprechenden Zeilen in der „rechten“ Tabelle gibt. Fehlende Werte werden mit `NULL` belegt.



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.

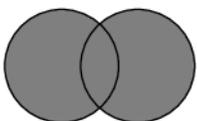
**LEFT JOIN mit Bedingung**

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-08-3-1

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.

Hierfür benötigt man eine andere Art von Joins, so genannte **OUTER JOINS**. Hiervon gibt es zwei Varianten, den **LEFT JOIN** und den **RIGHT JOIN**.

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.

2a. Erklären Sie anhand des Ergebnisses die Funktionsweise eines Left Join's bzw. Right Join's. Worin unterscheiden sich diese Outer Joins von einem Inner Join?

2b. Formulieren Sie obigen SQL-Befehl für das selbe Ergebnis mithilfe eines Right Join.



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

id	bez	pid	bid
6	Welde Export	nul	nul
17	Bitburger Pils	nul	nul
41	Andechser Weissbier hell	nul	nul
97	Sprite	nul	nul
115	Sprite	nul	nul
125	Odenwald Quelle Classic PET	nul	nul
152	Wiesentaler Medium	nul	nul
230	Bionade Holunder	nul	nul

Left- bzw. Right Join anwenden AUF-08-3-2

Wenden Sie nun den Left- bzw. Right Join an, um folgendes für die Personal- bzw. Verkaufsabteilung herauszufinden.

► **SQL-outer-1**



Abschließende Wiederholung AUF-08-3-3

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;`

id	name	id	wert
1	a	1	x
1	a	3	y
1	a	5	z
2	b	1	x
2	b	3	y
2	b	5	z
3	c	1	x
3	c	3	y
3	c	5	z

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

id	name	id	wert
1	a	1	x
3	c	3	y

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

id	name	id	wert
1	a	1	x
1	a		
3	c	3	y
3	c		

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

id	name	id	wert
		1	x
		3	y
		5	z



5. In die Tabellen T1 und T2 werden nun weitere Datensätze eingefügt, T1 besitzt nun **insgesamt 10 Datensätze** und T2 **insgesamt 20 Datensätze**.

Wie viel Datensätze besitzt das Ergebnis nun aus

1. Aufgabe 1 **genau**?

2. Aufgabe 2 **höchstens**?

3. Aufgabe 2 **mindestens**?

4. Aufgabe 3 **genau**?

5. Aufgabe 4 **genau**?

