

Beziehungen in SQL



Photo by Tobias Fischer on Unsplash

Abfragen über mehrere Tabellen

```
SELECT * FROM `lehrer`, `schueler`
```

✓ Zeige Datensätze 0 - 24 (72 insgesamt, Die Abfrage dauerte 0.0006 Sekunden.)

```
select * from schueler,lehrer
```

1 > >> | Alles anzeigen | Anzahl der Datensätze: 25 | Zeilen filtern: Diese Tabelle

+ Optionen

name	vorname	klasse	telefon	name	vorname	klehrer_in	fach
Margarine	Carl	8a	7625653	Einstein	Albert	7a	Physik
Margarine	Carl	8a	7625653	Newton	Isaac	6a	Mathe
Margarine	Carl	8a	7625653	Lovelace	Ada	6a	Informatik
Margarine	Carl	8a	7625653	Noether	Emmi	6b	Mathe
Margarine	Carl	8a	7625653	Gauss	Carl-Friedrich	6a	Mathe
Margarine	Carl	8a	7625653	Freud	Sigmund	6a	Psychologie
Schockokeks	Marie	7a	87216625	Einstein	Albert	7a	Physik
Schockokeks	Marie	7a	87216625	Newton	Isaac	6a	Mathe
Schockokeks	Marie	7a	87216625	Lovelace	Ada	6a	Informatik
Schockokeks	Marie	7a	87216625	Noether	Emmi	6b	Mathe
Schockokeks	Marie	7a	87216625	Gauss	Carl-Friedrich	6a	Mathe
Schockokeks	Marie	7a	87216625	Freud	Sigmund	6a	Psychologie
Erdbeermarmelade	Alan	7b	454261	Einstein	Albert	7a	Physik
Erdbeermarmelade	Alan	7b	454261	Newton	Isaac	6a	Mathe
Erdbeermarmelade	Alan	7b	454261	Lovelace	Ada	6a	Informatik
Erdbeermarmelade	Alan	7b	454261	Noether	Emmi	6b	Mathe
Erdbeermarmelade	Alan	7b	454261	Gauss	Carl-Friedrich	6a	Mathe
Erdbeermarmelade	Alan	7b	454261	Freud	Sigmund	6a	Psychologie
Schinkenwurst	Hermann	6a	872662625	Einstein	Albert	7a	Physik

Alle Kombinationen die möglich sind (kartesisches Produkt)

$72 = 6 * 12$ weitgehend sinnfreie Datensätze.

Beziehungen berücksichtigen mit WHERE

```
SELECT * FROM `lehrer`, `schueler`
WHERE schueler.klassenlehrer_id=lehrer.lehrer_id
```

✔ Zeige Datensätze 0 - 11 (12 insgesamt, Die Abfrage dauerte 0.0005 Sekunden.)

```
SELECT * FROM `lehrer`, `schueler` WHERE schueler.klassenlehrer_id=lehrer.lehrer_id
```

Mes

Alles anzeigen | Anzahl der Datensätze: Zeilen filtern:

+ Optionen

lehrer_id	name	vorname	klehrer_in	fach	schueler_id	name	vorname	klasse	telefon	klassenlehrer_id
3	Lovelace	Ada	6a	Informatik	1	Margarine	Carl	8a	7625653	3
2	Newton	Isaac	6a	Mathe	2	Schockokeks	Marie	7a	87216625	2
2	Newton	Isaac	6a	Mathe	3	Erdbeermarmelade	Alan	7b	454261	2
1	Einstein	Albert	7a	Physik	4	Schinkenwurst	Hermann	6a	872662625	1
1	Einstein	Albert	7a	Physik	5	Gummibärchen	Karl	7a	78265635	1
2	Newton	Isaac	6a	Mathe	6	Eisbein	Hildegard	7a	7288393	2
3	Lovelace	Ada	6a	Informatik	7	Kartoffelchips	Gabriele	6a	928893484	3
4	Noether	Emmi	6b	Mathe	8	Apfelsaft	Lieschen	7b	991823764	4
4	Noether	Emmi	6b	Mathe	9	Aprikosenjoghurt	Klaus-Gustav	7b	9288374	4
4	Noether	Emmi	6b	Mathe	10	Kaffepulver	Ingeborg	6a	209384	4
2	Newton	Isaac	6a	Mathe	11	Schokocreme	Lisa-Marie	6a	923867346	2
2	Newton	Isaac	6a	Mathe	12	Cornfkakes	Crunchie	7b	92848724t36	2

Beziehungen berücksichtigen mit WHERE

Die Verbindungs-Bedingung wird immer „mitgeschleppt“

✔ Zeige Datensätze 0 - 1 (2 insgesamt, Die Abfrage dauerte 0.0006 Sekunden.)

```
SELECT * FROM `lehrer`, `schueler` WHERE schueler.klassenlehrer_id=lehrer.lehrer_id AND lehrer.name="Lovelace"
```

Alles anzeigen | Anzahl der Datensätze: Zeilen filtern:

+ Optionen

lehrer_id	name	vorname	klehrer_in	fach	schueler_id	name	vorname	klasse	telefon	klassenlehrer_id
3	Lovelace	Ada	6a	Informatik	1	Margarine	Carl	8a	7625653	3
3	Lovelace	Ada	6a	Informatik	7	Kartoffelchips	Gabriele	6a	928893484	3

✔ Zeige Datensätze 0 - 1 (2 insgesamt, Die Abfrage dauerte 0.0006 Sekunden.)

```
SELECT * FROM `lehrer`, `schueler` WHERE schueler.klassenlehrer_id=lehrer.lehrer_id AND schueler.telefon LIKE "8%"
```

Alles anzeigen | Anzahl der Datensätze: Zeilen filtern:

+ Optionen

lehrer_id	name	vorname	klehrer_in	fach	schueler_id	name	vorname	klasse	telefon	klassenlehrer_id
2	Newton	Isaac	6a	Mathe	2	Schockokeks	Marie	7a	87216625	2
1	Einstein	Albert	7a	Physik	4	Schinkenwurst	Hermann	6a	872662625	1

Tabellenbeziehungen direkt mit JOIN herstellen

Mit dem JOIN-Statement lassen sich Werte aus mehreren Tabellen direkt kombinieren, ohne zunächst durch die Abfrage mehrerer Tabellen zunächst das Tabellenprodukt zu bilden und dieses anschliessend zu filtern.

MySQL hat mehrere JOIN-Typen: LEFT JOIN, RIGHT JOIN, INNER JOIN.
(Unvollständig, die wichtigsten)

LEFT JOIN

Die Syntax für einen LEFT JOIN ist wie folgt:

```
SELECT * FROM tabelle1
LEFT JOIN tabelle2 ON tabelle1.SpaltenameA = tabelle2.Spaltename
LEFT JOIN tabelle3 ON tabelle1.SpaltenameB = tabelle3.Spaltename
WHERE ...
```

LEFT JOIN es werden stets alle Zeilen der Tabelle zurückgegeben, die beim FROM aufgeführt ist - also gewissermaßen „links“ stehen. Diese Tabelle stellt die Basis für das Ergebnis dar.

Es kann jetzt aber sein, dass in der Tabelle die per LEFT JOIN verknüpft wird kein passender Eintrag gefunden wird, es gibt also keinen Datensatz in den beiden Tabellen, bei denen `tabelle1.SpaltenameA = tabelle2.Spaltename` ist. In diesem Fall erhalten diese Felder den NULL Wert, die Datenfelder der Ausgangstabelle werden aber auf jeden Fall ausgegeben.

```
SELECT * FROM lehrer LEFT JOIN schueler ON lehrer.id=schueler.KLID
```

id	name	vorname	klehrer_in	fach	id	name	vorname	klasse	telefon	KLID
1	Einstein	Albert	7a	Physik	1	Margarine	Carl	8a	7625653	1
1	Einstein	Albert	7a	Physik	2	Schockokeks	Marie	7a	87216625	1
3	Lovelace	Ada	6a	Informatik	3	Erdbeermarmelade	Alan	7b	454261	3
2	Newton	Isaac	6a	Mathe	4	Schinkenwurst	Hermann	6a	872662625	2
6	Freud	Sigmund	6a	Psychologie	5	Gummibärchen	Karl	7a	78265635	6
3	Lovelace	Ada	6a	Informatik	8	Apfelsaft	Lieschen	7b	991823764	3
2	Newton	Isaac	6a	Mathe	9	Aprikosenjoghurt	Klaus-Gustav	7b	9288374	2
5	Gauss	Carl-Friedrich	6a	Mathe	13	Cornfkakes	Crunchie	7b	92848724t36	5
4	Noether	Emmi	6b	Mathe	NULL	NULL	NULL	NULL	NULL	NULL

RIGHT JOIN

Die Syntax von RIGHT JOIN entspricht der von LEFT JOIN. Der Unterschied ist, dass hier die Tabelle die im JOIN hinzugefügt wird als Basis für die Datensätze dient - gibt es keine Treffer, werden die Felder der mit FROM selektierten Tabelle mit NULL-Werten gefüllt:

```
SELECT * FROM lehrer RIGHT JOIN schueler ON lehrer.id=schueler.KLID
```

id	name	vorname	klehrer_in	fach	id	name	vorname	klasse	telefon	KLID
2	Newton	Isaac	6a	Mathe	1	Margarine	Carl	8a	7625653	2
3	Lovelace	Ada	6a	Informatik	2	Schockokeks	Marie	7a	87216625	3
4	Noether	Emmi	6b	Mathe	3	Erdbeermarmelade	Alan	7b	454261	4
5	Gauss	Carl-Friedrich	6a	Mathe	4	Schinkenwurst	Hermann	6a	872662625	5
1	Einstein	Albert	7a	Physik	5	Gummibärchen	Karl	7a	78265635	1
2	Newton	Isaac	6a	Mathe	6	Eisbein	Hildegard	7a	7288393	2
NULL	NULL	NULL	NULL	NULL	7	Kartoffelchips	Gabriele	6a	928893484	0
3	Lovelace	Ada	6a	Informatik	8	Apfelsaft	Lieschen	7b	991823764	3
4	Noether	Emmi	6b	Mathe	9	Aprikosenjoghurt	Klaus-Gustav	7b	9288374	4
2	Newton	Isaac	6a	Mathe	10	Kaffepulver	Ingeborg	6a	209384	2
NULL	NULL	NULL	NULL	NULL	11	Schokocreme	Lisa-Marie	6a	923867346	0
1	Einstein	Albert	7a	Physik	12	Cornfkakes	Crunchie	7b	92848724t36	1

INNER JOIN

Bei einem INNER JOIN muss eine passende Zeile in den Tabellen gefunden werden, Datensätze, die die JOIN-Bedingung nicht erfüllen werden nicht zurückgegeben.

```
SELECT * FROM lehrer INNER JOIN schueler ON lehrer.id=schueler.KLID
```

id	name	vorname	klehrer_in	fach	id	name	vorname	klasse	telefon	KLID
2	Newton	Isaac	6a	Mathe	1	Margarine	Carl	8a	7625653	2
3	Lovelace	Ada	6a	Informatik	2	Schockokeks	Marie	7a	87216625	3
4	Noether	Emmi	6b	Mathe	3	Erdbeermarmelade	Alan	7b	454261	4
5	Gauss	Carl-Friedrich	6a	Mathe	4	Schinkenwurst	Hermann	6a	872662625	5
1	Einstein	Albert	7a	Physik	5	Gummibärchen	Karl	7a	78265635	1
2	Newton	Isaac	6a	Mathe	6	Eisbein	Hildegard	7a	7288393	2
3	Lovelace	Ada	6a	Informatik	8	Apfelsaft	Lieschen	7b	991823764	3
4	Noether	Emmi	6b	Mathe	9	Aprikosenjoghurt	Klaus-Gustav	7b	9288374	4
2	Newton	Isaac	6a	Mathe	10	Kaffepulver	Ingeborg	6a	209384	2
1	Einstein	Albert	7a	Physik	12	Cornfkakes	Crunchie	7b	92848724t36	1

Das entspricht unserer bisherigen Praxis, zunächst das kartesische Produkt aller beteiligten Tabellen abzufragen und dann die passenden Datensätze mit WHERE herauszufiltern:

```
SELECT * FROM lehrer, schueler WHERE lehrer.id=schueler.KLID
```