

Möchte man die Daten aus mehreren Tabellen abfragen, so wäre ein naiver erster Ansatz, einfach aus beiden Tabellen alles auszugeben.

Beispiel: Die Tabellen „hobbits“ und „gerichte“

hobbitNr	name	alter	lieblingssessen
1	Frodo Beutlin	33	3
2	Samweis Gamdschie	26	1
3	Meriadoc Brandybock	28	5
4	Peregrin Tuck	22	2

gerichtNr	name	vegetarisch
1	Kartoffelbrei	1
2	Apfel-Tarte	1
3	Brot	1
4	Honigschinken	0
5	Rührei	1

Das Ziel wird natürlich sein, zu jedem Hobbit zusätzlich die Informationen des Lieblingsessens auszugeben. Ein erster Versuch der Ausgabe könnte sein: `SELECT * FROM `hobbits`, `gerichte``

gerichtNr	name	vegetarisch	hobbitNr	name	alter	lieblingssessen
1	Kartoffelbrei	1	1	Frodo Beutlin	33	3
1	Kartoffelbrei	1	2	Samweis Gamdschie	26	1
1	Kartoffelbrei	1	3	Meriadoc Brandybock	28	5
1	Kartoffelbrei	1	4	Peregrin Tuck	22	2
2	Apfel-Tarte	1	1	Frodo Beutlin	33	3
2	Apfel-Tarte	1	2	Samweis Gamdschie	26	1
2	Apfel-Tarte	1	3	Meriadoc Brandybock	28	5
2	Apfel-Tarte	1	4	Peregrin Tuck	22	2

Die obige Tabelle ist nur ein Ausschnitt des Ergebnisses dieser Abfrage. Jeder **einzelne** Eintrag in der Tabelle „gerichte“ wird mit **allen** Hobbits kombiniert! Es wird das sogenannte **Kreuzprodukt** gebildet. (abgekürzt mit:  $\times$ , hier also: *gerichte*  $\times$  *hobbits*). Besonders bei großen Tabellen nimmt die Größe des Kreuzprodukts rapide zu. Bezeichnet man die Anzahl der Einträge der ersten Tabelle mit  $n$  und die der zweiten Tabelle mit  $m$ , so hat das Kreuzprodukt  $m \cdot n$  Einträge.

In dieser Fülle an erzeugten Zeilen befinden sich viele, die nicht für unser ursprüngliches Ziel interessant sind. Um „sinnvolle“ Ergebnisse zu erhalten, müssen aus dieser **Grundmenge** an Zeilen die relevanten ausgewählt werden.

Im konkreten Beispiel sind dies die Zeilen, bei denen das Lieblingsessen mit der Nummer des entsprechenden Gerichts übereinstimmt. Wir können dazu mit WHERE auch über Tabellen hinweg arbeiten:

```
SELECT * FROM `hobbits`, `gerichte` WHERE lieblingssessen = gerichtNr
```

hobbitNr	name	alter	lieblingssessen	gerichtNr	name	vegetarisch
1	Frodo Beutlin	33	3	3	Brot	1
2	Samweis Gamdschie	26	1	1	Kartoffelbrei	1
3	Meriadoc Brandybock	28	5	5	Rührei	1
4	Peregrin Tuck	22	2	2	Apfel-Tarte	1

Die Ausgabe der verschiedenen Nummern ist für einen Nutzer der Datenbank nicht notwendig. Aus den ausgegebenen Spalten kann noch ausgewählt werden:

```
SELECT name, name, vegetarisch FROM `hobbits`, `gerichte` WHERE lieblingsessen = gerichtNR
```

Versuchen wir diesen Befehl auszuführen teilt uns MySQL z.B. mit:

```
#1052 - Feld 'name' in field list ist nicht eindeutig
```

Damit die Ausgabe eindeutig sein kann, muss bei gleicher Benennung klar sein aus welcher Tabelle die Spalte kommt:

```
SELECT hobbits.name, gerichte.name, vegetarisch FROM `hobbits`, `gerichte`
```

```
WHERE lieblingsessen = gerichtNR
```

name	name	vegetarisch
Frodo Beutlin	Brot	1
Samweis Gamdschie	Kartoffelbrei	1
Meriadoc Brandybock	Rührei	1
Peregrin Tuck	Apfel-Tarte	1

Die Ausführung eines Kreuzprodukts mit anschließender Abfrage wird **Join** genannt. Um nicht immer die vollständigen Namen der Tabellen benutzen zu müssen können Abkürzungen definiert werden. Außerdem können nach wie vor Spalten mit AS umbenannt werden, um Verwirrung bei der Benennung vorzubeugen:

```
SELECT h.name as "Hobbitname", g.name as "Lieblingsessen", vegetarisch
```

```
FROM `hobbits` h , `gerichte` g
```

```
WHERE lieblingsessen = gerichtNR
```

Hobbitname	Lieblingsessen	vegetarisch
Frodo Beutlin	Brot	1
Samweis Gamdschie	Kartoffelbrei	1
Meriadoc Brandybock	Rührei	1
Peregrin Tuck	Apfel-Tarte	1

So erhalten wir eine für den Endbenutzer schön formatierte Ausgabe.

Alternative Schreibweise:

```
SELECT h.name as "Hobbitname", g.name as "Lieblingsessen", vegetarisch
```

```
FROM `hobbits` h JOIN `gerichte` g ON lieblingsessen = gerichtNR
```

Sind Primärschlüssel und Fremdschlüssel gleich benannt und ist dies die einzige Namensdopplung in den beiden Tabellen, so kann auch ein NATURAL JOIN verwendet werden.

```
SELECT name as "Hobbitname", g.gerichtname, vegetarisch
```

```
FROM `hobbits` NATURAL JOIN `gerichte`
```

Dazu musste in diesem Beispiel aber der „name“ in „gerichte“ durch „gerichtname“ ersetzt werden.