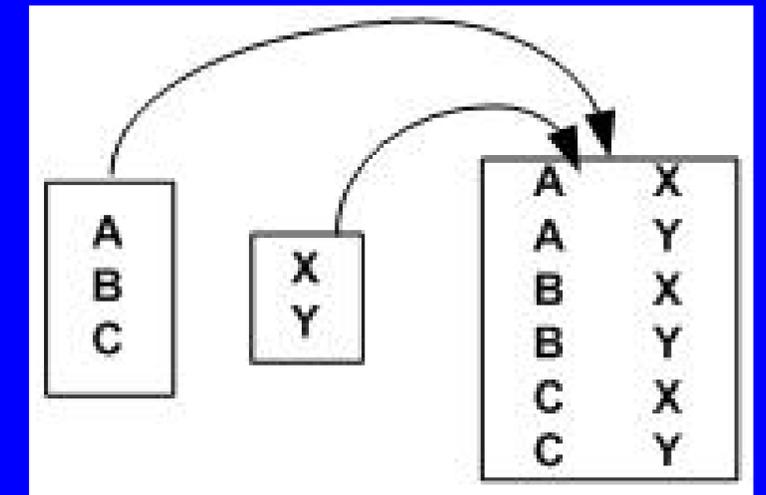


**Relationenalgebra:
Weitere Grundoperatoren**





**Grundoperator:
Das kartesische Produkt \times**

Was ist das Kartesische Produkt bzw. Kreuzprodukt bzw. Cross Join?

- Das kartesische Produkt $R \times S$ ist eine Operation, welche dem kartesischen Produkt aus der Mengenlehre ähnelt.
- Das Resultat des kartesischen Produkts ist die Menge aller Kombinationen der Tupel aus den Relationen R und S
 - Das bedeutet, dass jede Zeile der einen Tabelle mit jeder Zeile der anderen Tabelle kombiniert wird.
- Wenn alle Spalten verschieden sind, so umfasst die Resultatstabelle die Summe der Spalten der Ausgangstabellen.
 - Gleichnamige Spalten der zwei Tabellen werden durch Voranstellen des Tabellennamens referenziert.
 - Die Anzahl der Zeilen (Tupel) in der Resultatstabelle ist das Ergebnis der Multiplikation der Zeilenanzahlen der Ausgangstabellen.

Das kartesische Produkt

- R und S sind zwei Relationen (Datenbanktabellen)
- a1...n und b1...m sind alle Spalten von R und S.

$$R \times S := \{(a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_m) \mid (a_1, a_2, \dots, a_n) \in R \wedge (b_1, b_2, \dots, b_m) \in S\}$$

R:			
A	B	C	D
1	2	3	4
4	5	6	7
7	8	9	0

S:		
E	F	G
1	2	3
7	8	9

R × S:						
A	B	C	D	E	F	G
1	2	3	4	1	2	3
4	5	6	7	1	2	3
7	8	9	0	1	2	3
1	2	3	4	7	8	9
4	5	6	7	7	8	9
7	8	9	0	7	8	9

Das kartesische Produkt in MariaDB DRL

foods

ITEM_ID	ITEM_NAME	ITEM_UNIT	COMPANY_ID
1	Chex Mix	Pcs	16
6	Cheez-It	Pcs	15
2	BN Biscuit	Pcs	15
3	Mighty Munch	Pcs	17
4	Pot Rice	Pcs	15
5	Jaffa Cakes	Pcs	18
7	Salt n Shake	Pcs	

company

COMPANY_ID	COMPANY_NAME	COMPANY_CITY
18	Order All	Boston
15	Jack Hill Ltd	London
16	Akas Foods	Delhi
17	Foodies.	London
19	sip-n-Bite.	New York

SELECT

foods.item_name, foods.item_unit, company.company_name,

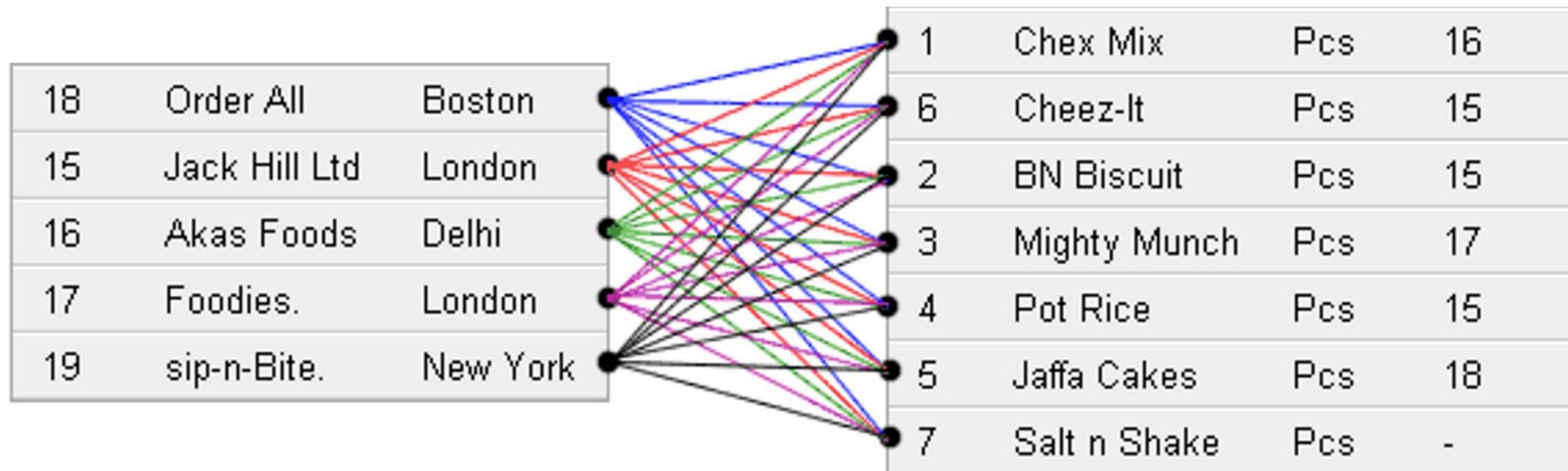
company.company_city

FROM foods

CROSS JOIN company;

Das kartesische Produkt in MariaDB DRL

```
SELECT  
foods.item_name, foods.item_unit, company.company_name,  
company.company_city  
FROM foods  
CROSS JOIN company;
```

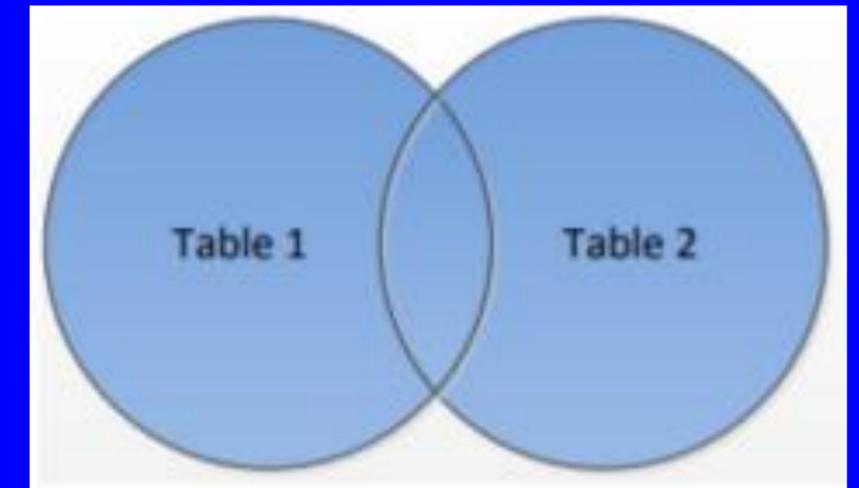


Das kartesische Produkt

MariaDB DRL vs. Relationenalgebra

```
SELECT  
foods.item_name, foods.item_unit, company.company_name,  
company.company_city  
FROM foods  
CROSS JOIN company;
```

- $R := \text{foods}$
- $S := \text{company}$
- $a1 := \text{foods.item_name}$
- $a2 := \text{foods.item_unit}$
- $b1 := \text{company.company_name}$
- $b2 := \text{company.company_city}$



**Grundoperator:
Die Vereinigung, Union \cup**

Was ist die Vereinigung bzw. Union?

- Bei der Vereinigung $R \cup S$ werden alle Tupel der Relation R mit allen Tupeln der Relation S zu einer einzigen Relation vereint.
- Voraussetzung dafür ist, dass R und S das gleiche Relationenschema haben:
 - Das heißt, sie haben gleiche Attribute und Attributtypen.
- Duplikate werden bei der Vereinigung gelöscht.

Vereinigung bzw. Union

- Seien R und S zwei Relationen (Datenbanktabellen)...

$$R \cup S := \{t \mid t \in R \vee t \in S\}$$

R:

A	B	C
1	2	3
4	5	6

S:

A	B	C
7	8	9
4	5	6

R U S:

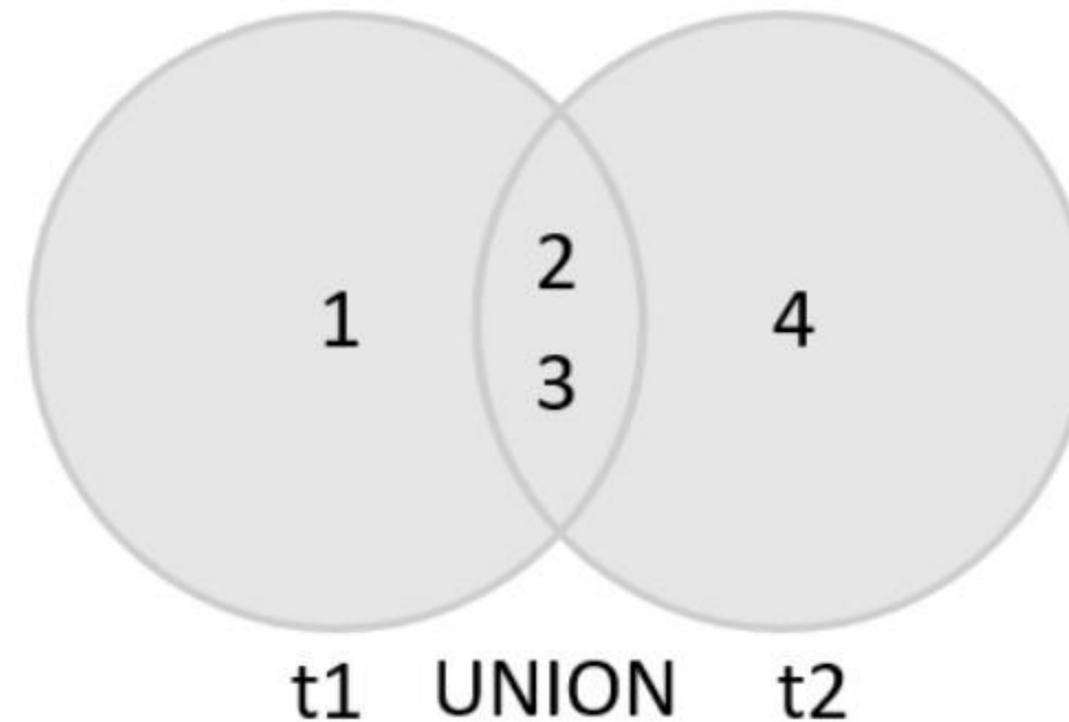
A	B	C
7	8	9
4	5	6
1	2	3

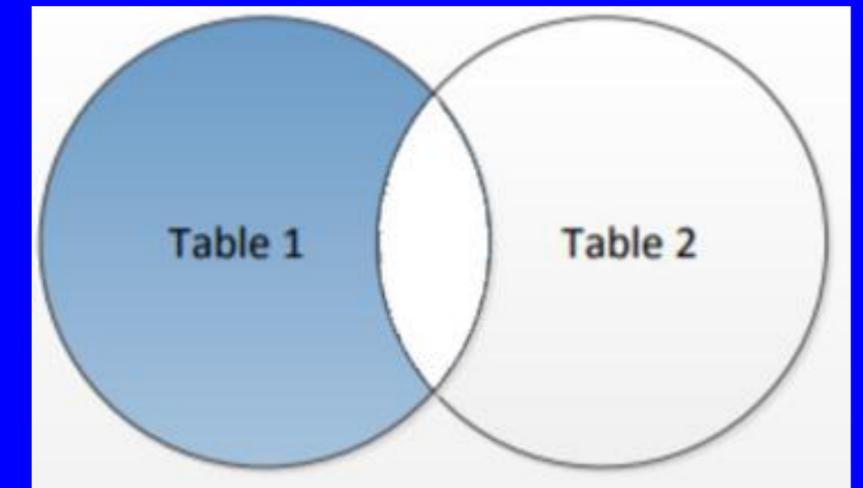
Vereinigung bzw. Union in MariaDB

```
CREATE TABLE t1 (id INT PRIMARY KEY);  
CREATE TABLE t2 (id INT PRIMARY KEY);  
INSERT INTO t1 VALUES (1), (2), (3);  
INSERT INTO t2 VALUES (2), (3), (4);  
SELECT id FROM t1 UNION SELECT id FROM t2;
```

- R := t1
- S := t2

id
1
2
3
4





**Grundoperator:
Die Differenz –**

Was ist die Differenz?

- Bei der Differenz-Operation $R \setminus S$ oder $R - S$ werden aus der ersten Relation R alle Tupel entfernt, die auch in der zweiten Relation S vorhanden sind.
- Seien R und S zwei Relationen (Datenbanktabellen)...

$$R - S := R \setminus S := \{t | t \in R \wedge t \notin S\}$$

R:			S:			R \ S:		
A	B	C	A	B	C	A	B	C
1	2	3	7	8	9	1	2	3
4	5	6	4	5	6			

- Im SQL-Standard:

SELECT * FROM R EXCEPT SELECT * FROM S;

oder

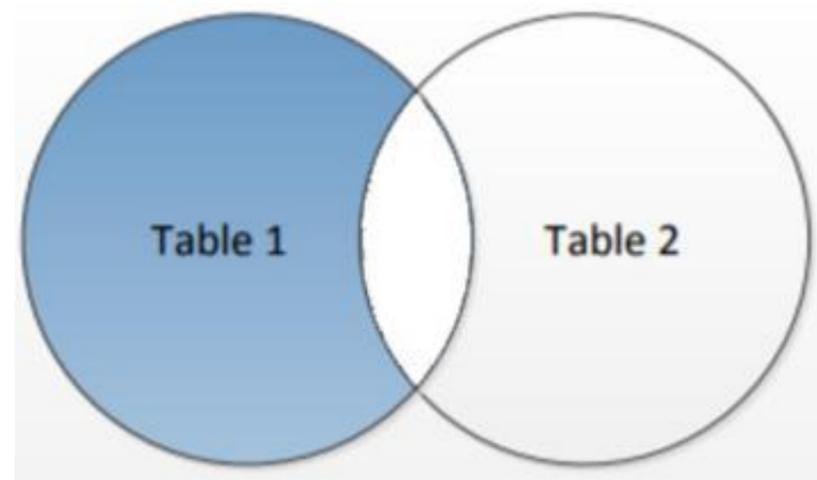
SELECT * FROM R MINUS SELECT * FROM S;

- MariaDB unterstützt die Operatoren EXCEPT und MINUS aber leider nicht...
- Also muss die Differenz anders hergeleitet werden...

Differenz in MariaDB

R:			S:			R \ S:		
A	B	C	A	B	C	A	B	C
1	2	3	7	8	9	1	2	3
4	5	6	4	5	6			

```
SELECT * FROM R WHERE NOT EXISTS  
(SELECT * FROM S WHERE R.ID = S.ID);
```



Grundoperator: Umbenennung ρ (Rho)

Wozu Umbenennungen?

- Diese Operation ist wichtig, um
 - Joins von unterschiedlichen benannten Relationen zu ermöglichen,
 - kartesische Produkte zu ermöglichen, wo es gleiche Attributnamen gibt, insbesondere auch mit der gleichen Relation
 - Mengenoperationen zwischen Relationen mit unterschiedlichen Attributen zu ermöglichen.

Umbenennungen in Relationenalgebra und in MariaDB

- Die Schreibweise ist allgemein $\rho_{[\text{neu} \leftarrow \text{alt}]}(R)$.
- Wir konstruieren dabei eine neue Tupelmeng t' aus der alten Tupelmeng:

$$\rho_{[\text{neu} \leftarrow \text{alt}]}(R) := \{t' \mid t'(R - \text{alt}) = t(R - \text{alt}) \wedge t'(\text{neu}) = t(\text{alt})\}$$

R		
A	B	C
1	2	3
4	5	6

$\rho_{[X \leftarrow B]}(R)$		
A	X	C
1	2	3
4	5	6

- In MariaDB: **ALTER TABLE R CHANGE B X;**