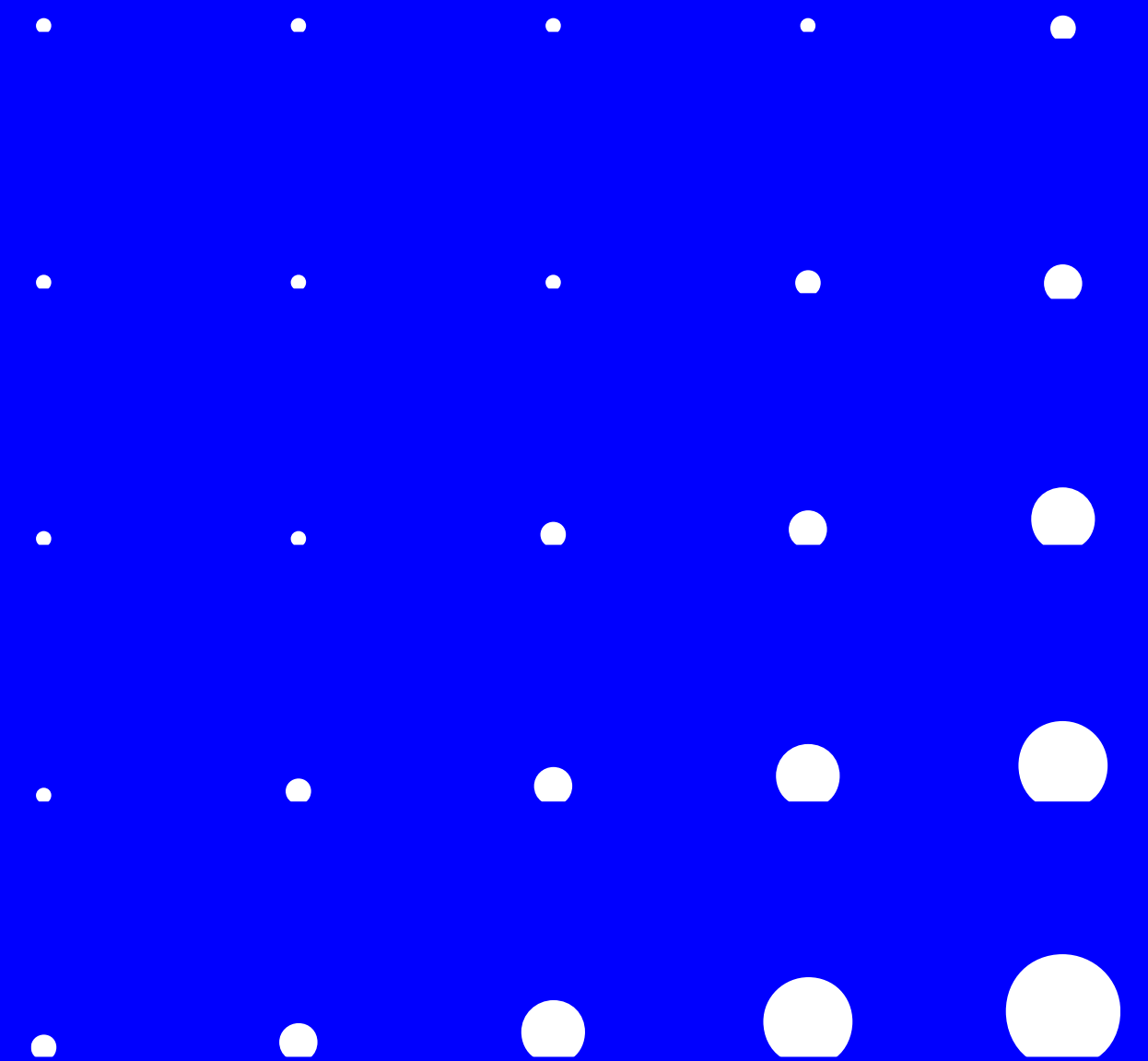
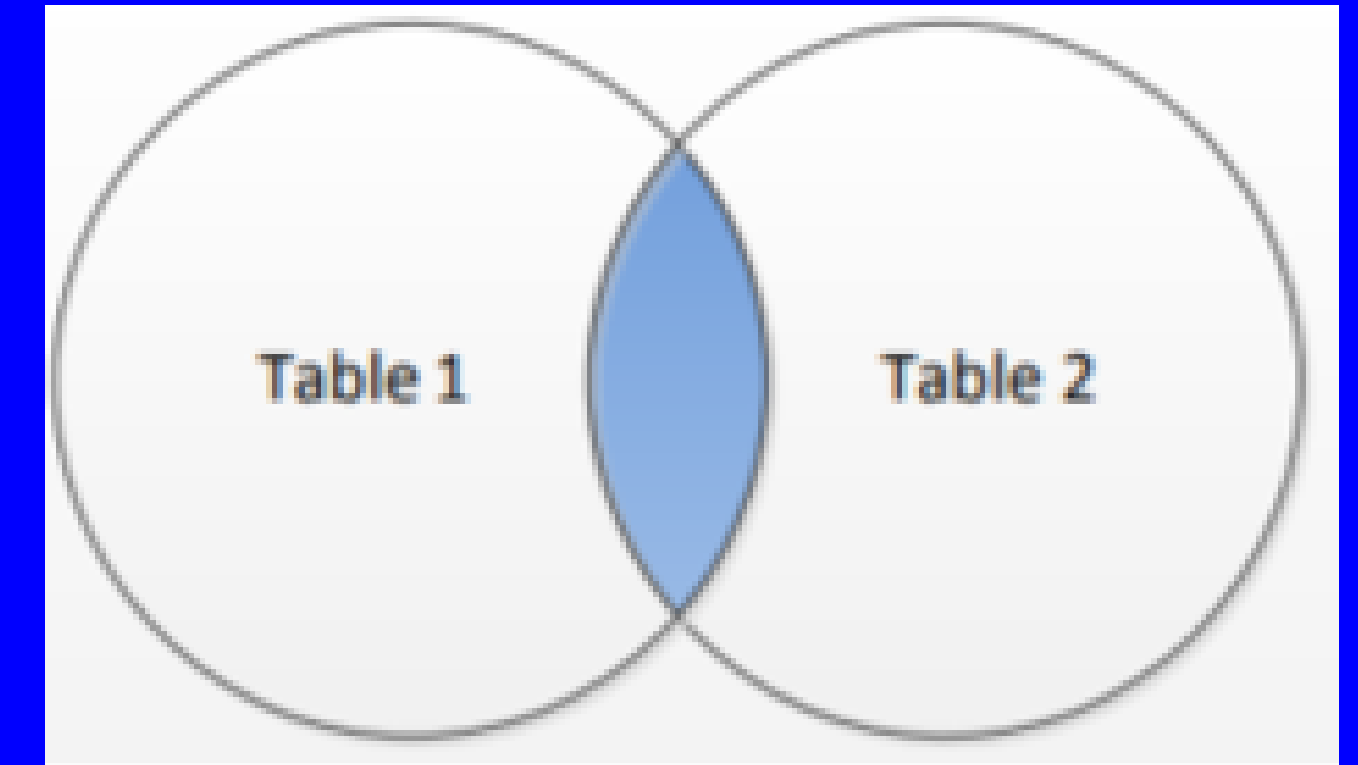


Relationenalgebra: Join



Die Schnittmenge Inner Join \cap



Was ist die Schnittmenge?

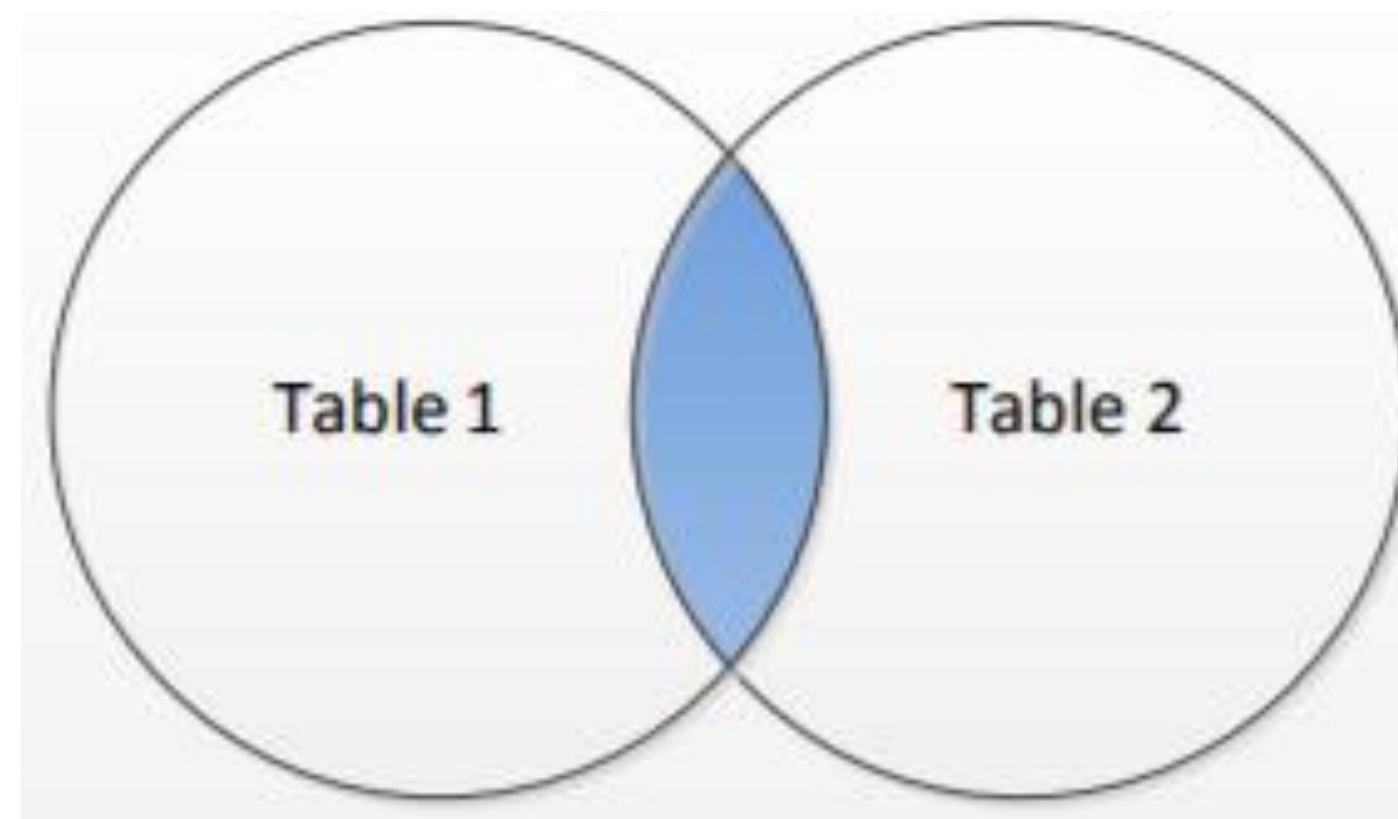
- Das Ergebnis der Durchschnittsoperation $R \cap S$ sind all die Tupel, die sich sowohl in der Relation R, als auch in der Relation S finden lassen.
- Eine Schnittmenge nennt man auch Durchschnitt.
- Seien R und S zwei Relationen (Datenbanktabellen)...

$$R \cap S := \{t | t \in R \wedge t \in S\}$$

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

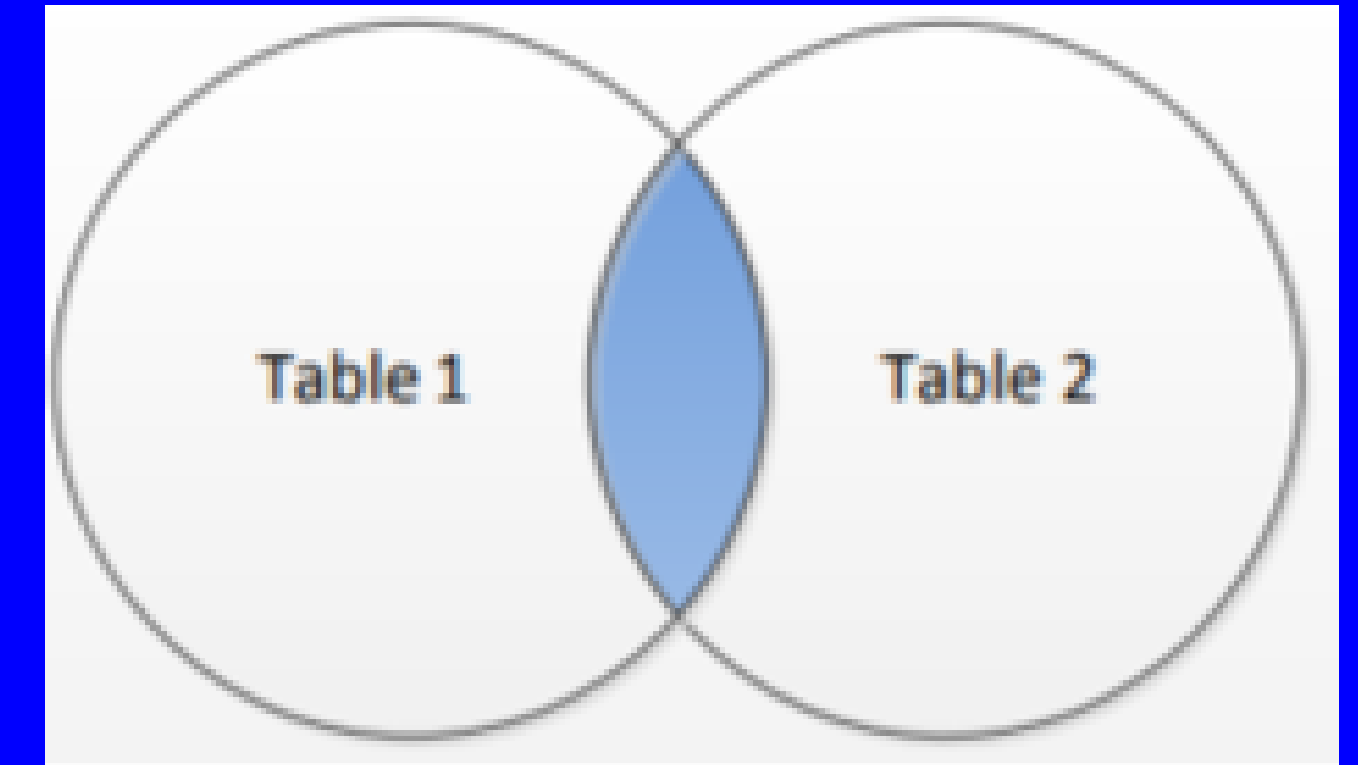
Schnittmenge in MariaDB

- R := Table_1 bzw. Alias t1
- S := Table_2 bzw. Alias t2
- Die eindeutigen Identifikatoren der einzelnen Tupel sind wie immer deren Primärschlüssel.



```
SELECT *  
  FROM Table_1 t1  
 INNER JOIN Table_2 t2  
    ON t1.id = t2.fk;
```

Natural Join ⋈



Wo ist der Unterschied zwischen einem Natural Join und einem Inner Join?

- Ein Natural Join („natürlicher Verbund“) verknüpft die beiden Tabellen über die Gleichheit aller gleichlautenden Spalten.
- Gleichlautende Spalten werden im Ergebnis nur einmal angezeigt.
- Haben die beiden Tabellen keine gleichlautenden Spalten, so wird der Natural Join zum Cross Join.
- Gibt es nur genau eine gleichlautende Spalte, so ist der Natural Join ein Inner Join mit anschließender Projektion, bei der gleichnamige Spalten ausgeblendet werden.
- Für den Natural Join gibt es keinen speziellen SQL92-Befehl. Er wird daher bei Bedarf aus einem Inner Join mit anschließender Projektion erzeugt.

Natural Join in Relationenalgebra

- Für zwei Relationen $R(A_1, \dots, A_n, B_1, \dots, B_n)$ und $S(B_1, \dots, B_n, C_1, \dots, C_n)$ ist das Ergebnis des natürlichen Verbundes

$$R \bowtie S := \{r \cup s_{[C_1, \dots, C_n]} \mid r \in R \wedge s \in S \wedge r_{[B_1, \dots, B_n]} = s_{[B_1, \dots, B_n]}\}$$

R:

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

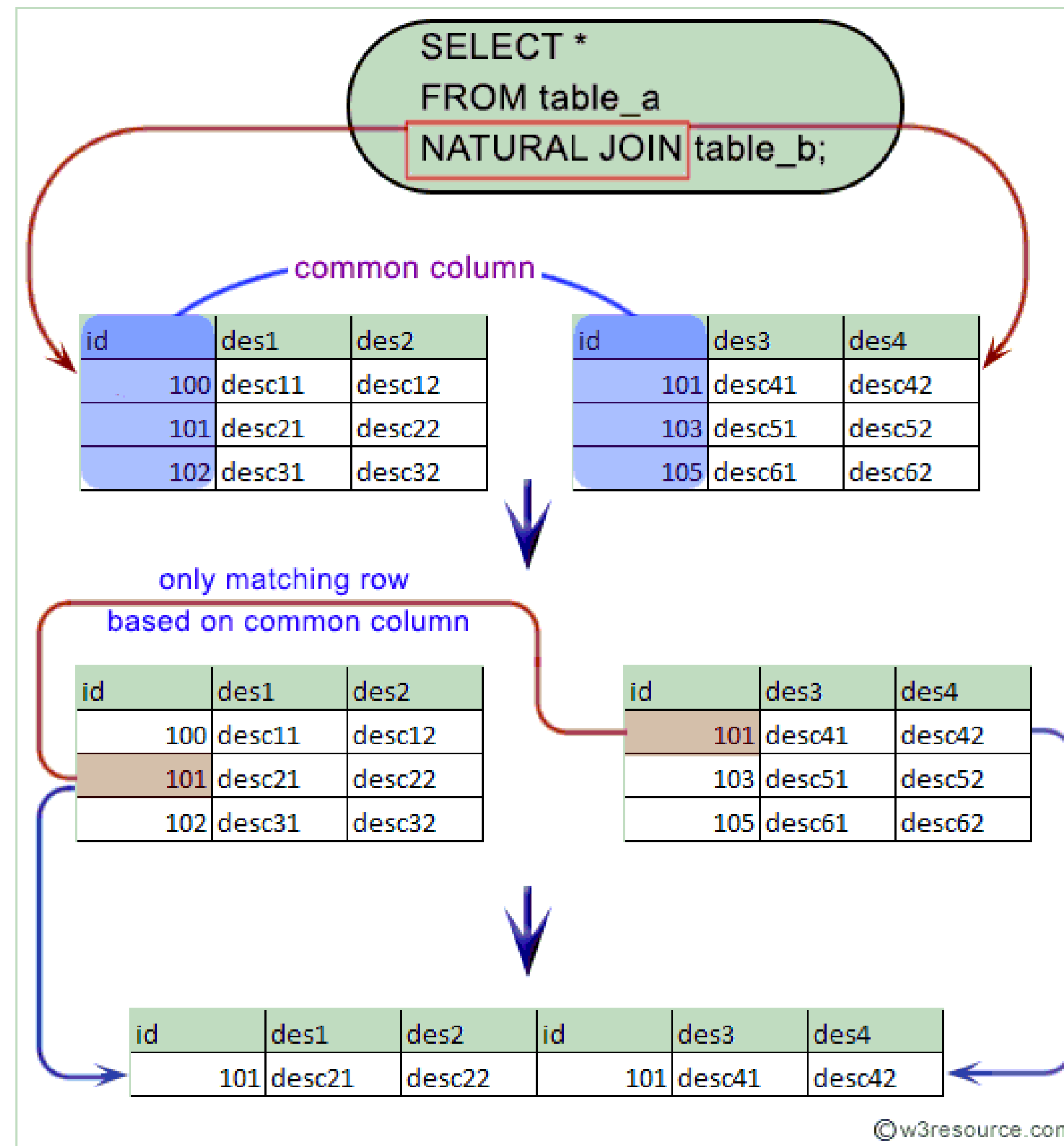
S:

A	F	G
1	2	3
7	8	9

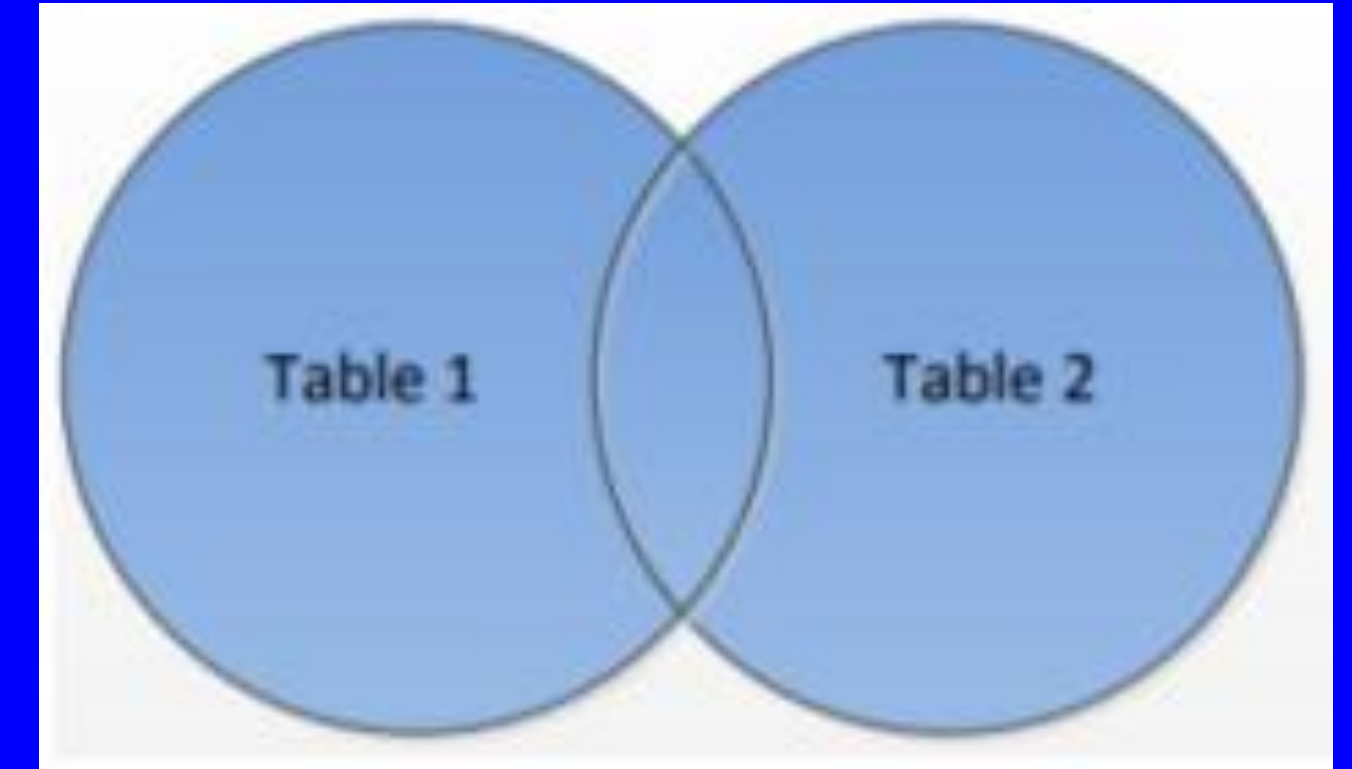
$R \bowtie S$

A	B	C	D	F	G
1	2	3	4	2	3
7	8	9	0	8	9

Natural Join in MariaDB



Full Outer Join \bowtie



Full Outer Join vs. Vereinigung/Union

- Die Mengenabbildung eines Full Outer Joins entspricht derselben Mengenabbildung einer Vereinigung bzw. Union. Ist das also dasselbe?
 - NEIN!
 - Während bei einer Union die Ergebniszeilen einfach hintereinandergesetzt werden, bildet ein Full Outer Join zunächst das kartesische Produkt und danach (meist) die Teilmengen anhand einer Bedingung:

```
mysql> SELECT * FROM
-> (SELECT 23 AS bah) AS foo
-> JOIN
-> (SELECT 45 AS bah) AS bar
-> ON (33=33);
```

foo	bar	
23	45	

```
mysql> SELECT 23 AS bah
-> UNION
-> SELECT 45 AS bah;
```

bah
23
45

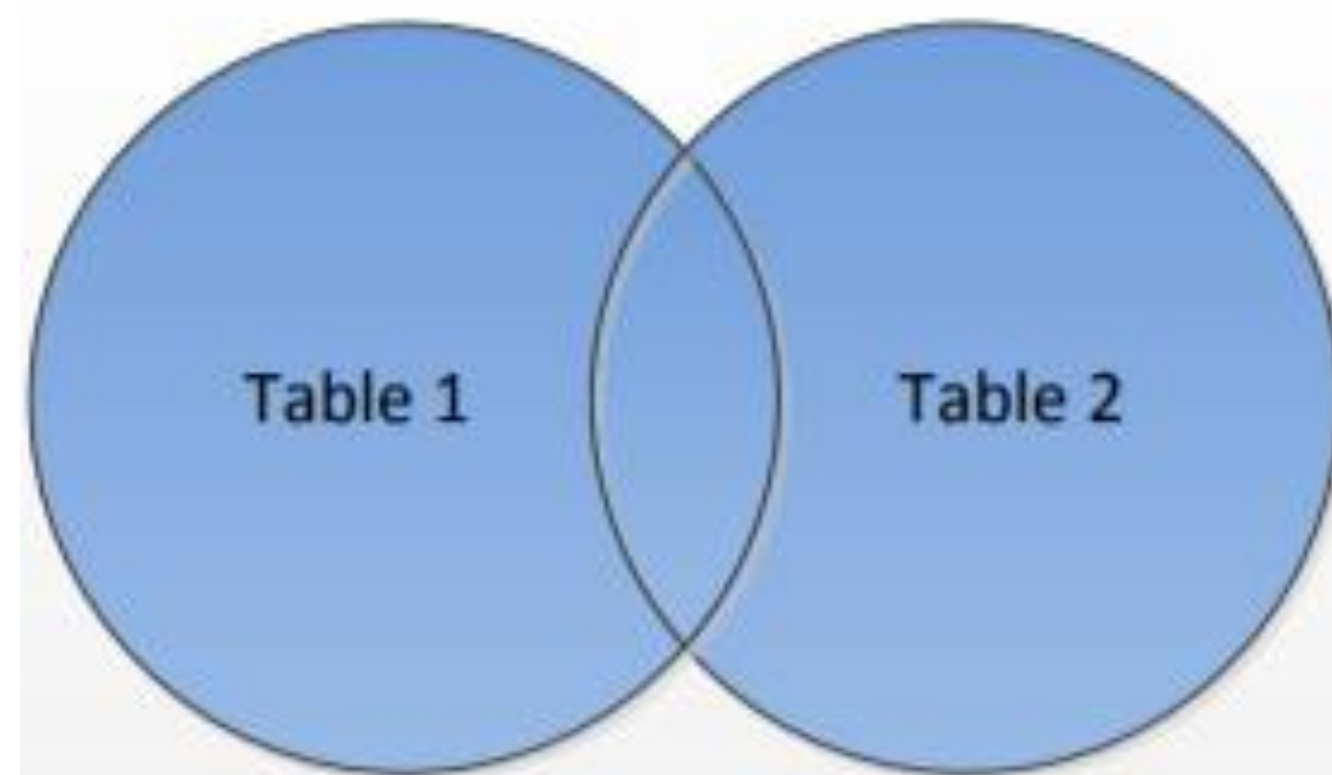
Full Outer Join in Relationenalgebra

- $r_1 \bowtie r_2$
- Bei unvollständigen Tupeln können auf beiden Seiten, also links und rechts, NULL-Werte entstehen.

r_1			r_2			$r_1 \bowtie r_2$				
A	B	C	C	D	E	A	B	C	D	E
a_1	b_1	c_1	c_1	d_1	e_1	a_1	b_1	c_1	d_1	e_1
a_2	b_2	c_2	c_3	d_2	e_2	a_2	b_2	c_2	\perp	\perp
						\perp	\perp	c_3	d_2	e_2

Full Outer Join in MariaDB

- Leider unterstützt MariaDB kein FULL OUTER JOIN, daher muss man hier ein wenig tricksen, indem man per UNION das Ergebnis zweier Abfragen miteinander verknüpft.
- Diese beiden Abfragen sind eben
 - ein LEFT JOIN und
 - ein RIGHT JOIN,deren Ergebnisse dann mit einem UNION verknüpft werden.



```
SELECT *  
  FROM Table_1 t1  
FULL OUTER JOIN Table_2 t2  
    ON t1.id = t2.fk;
```

DRL – FULL OUTER JOIN: Beispiel in MariaDB

Tabelle 'hersteller'

id	zulieferer
1	Hilti
2	Hoch & Tief
3	Eisen-Karl
4	Stahl AG
5	Gähm & Söhne

pk

Tabelle 'produkt'

id	produkt	hs_link
1	Schlagbohrer	1
2	Zement	2
3	Kneifzange	3
4	Brecheisen	3
5	Hammer	7

fk

- Gesucht werden alle Zulieferer und alle Produkte.
- Werden Produkte von bestimmten Zulieferern geliefert, so möchte man dies erkennen.
- Liefern Zulieferer bestimmte Produkte, so möchte man dies auch erkennen...

DRL – FULL OUTER JOIN: Beispiel in MariaDB und als Relationenalgebra

Tabelle 'hersteller'

id	zulieferer
1	Hilti
2	Hoch & Tief
3	Eisen-Karl
4	Stahl AG
5	Gähm & Söhne

pk

Tabelle 'produkt'

id	produkt	hs_link
1	Schlagbohrer	1
2	Zement	2
3	Kneifzange	3
4	Brecheisen	3
5	Hammer	7

fk

```
SELECT h.zulieferer, p.produkt
FROM hersteller h
LEFT JOIN produkt p ON (h.id = p.hs_link)
```

UNION

```
SELECT h.zulieferer, p.produkt
FROM hersteller h
RIGHT JOIN produkt p ON (h.id = p.hs_link);
```

- $\pi_{h.zulieferer, p.produkt} (h \bowtie_{h.id = p.hs_link} p)$

DRL – FULL OUTER JOIN: Beispiel in MariaDB

Tabelle 'hersteller'

id	zulieferer
1	Hilti
2	Hoch & Tief
3	Eisen-Karl
4	Stahl AG
5	Gähn & Söhne

pk

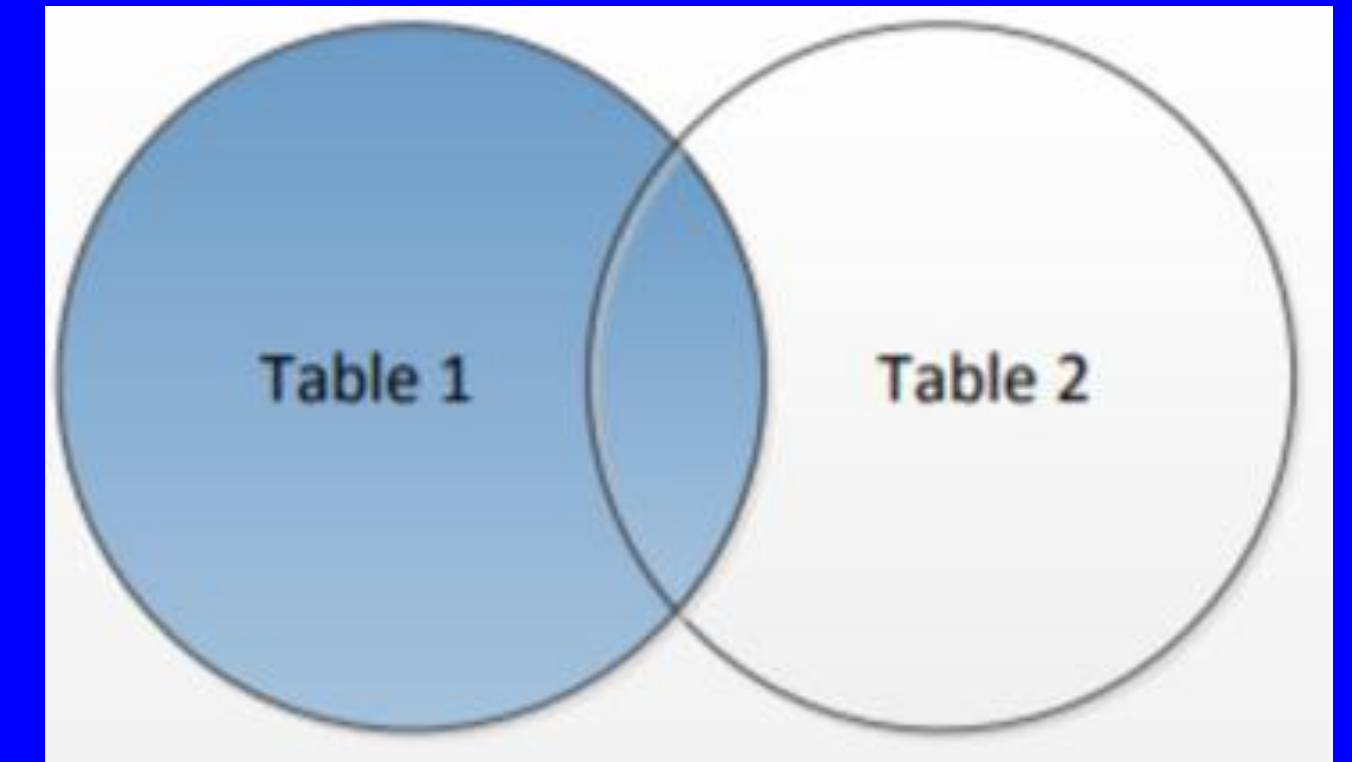
Tabelle 'produkt'

id	produkt	hs_link
1	Schlagbohrer	1
2	Zement	2
3	Kneifzange	3
4	Brecheisen	3
5	Hammer	7

fk

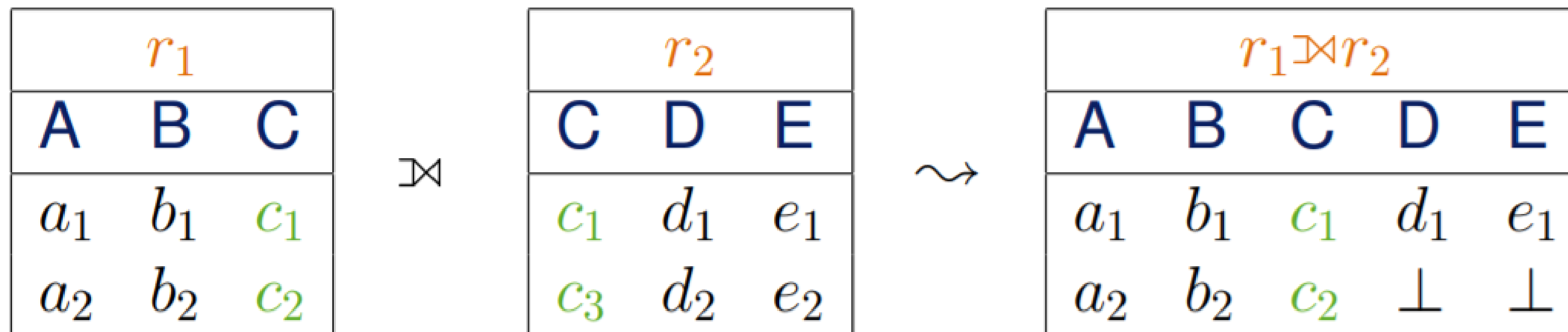
zulieferer	produkt
Hilti	Schlagbohrer
Hoch & Tief	Zement
Eisen-Karl	Kneifzange
Eisen-Karl	Brecheisen
Stahl AG	NULL
Gähn & Söhne	NULL
NULL	Hammer

Left Outer Join ⋈



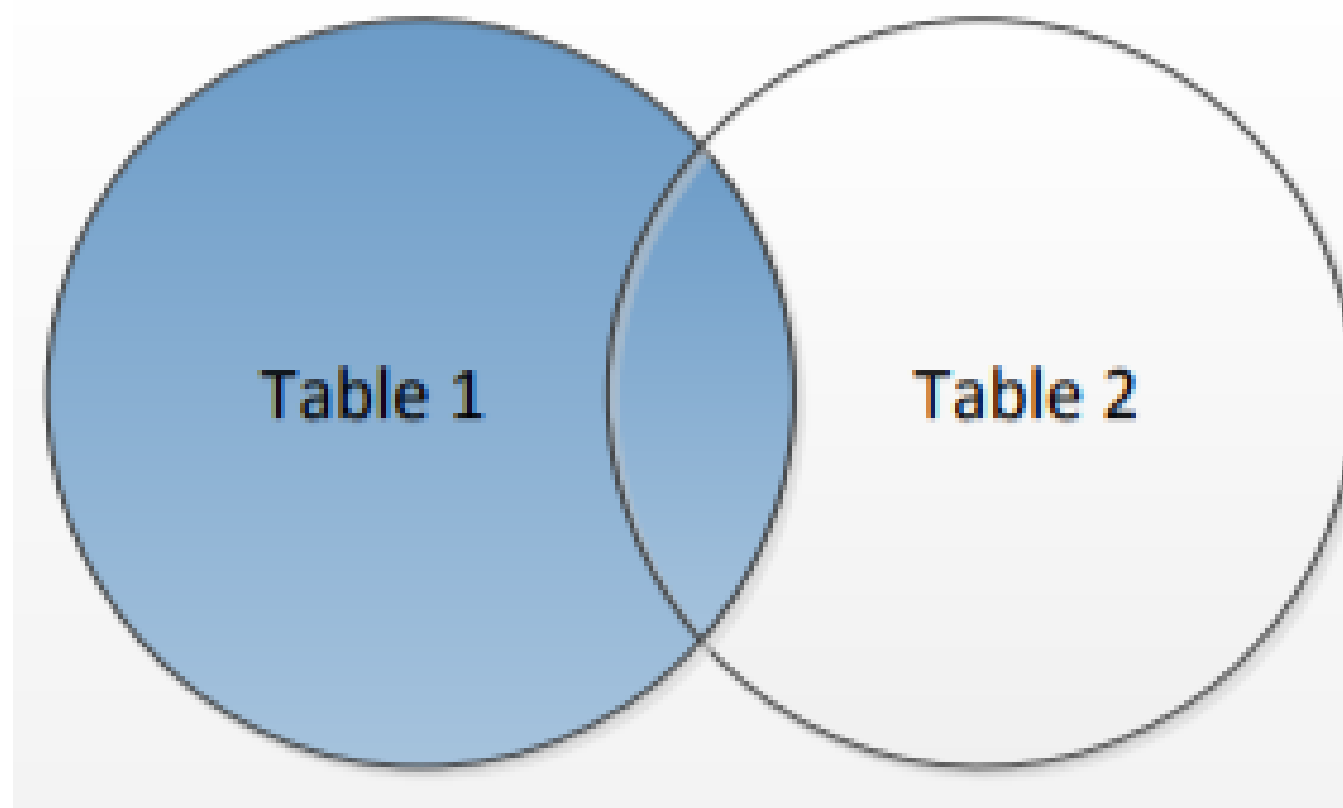
Left Outer Join als Relationenalgebra

- $r_1 \bowtie r_2$
- Bei unvollständigen Tupeln können auf der rechten Seite NULL-Werte entstehen.



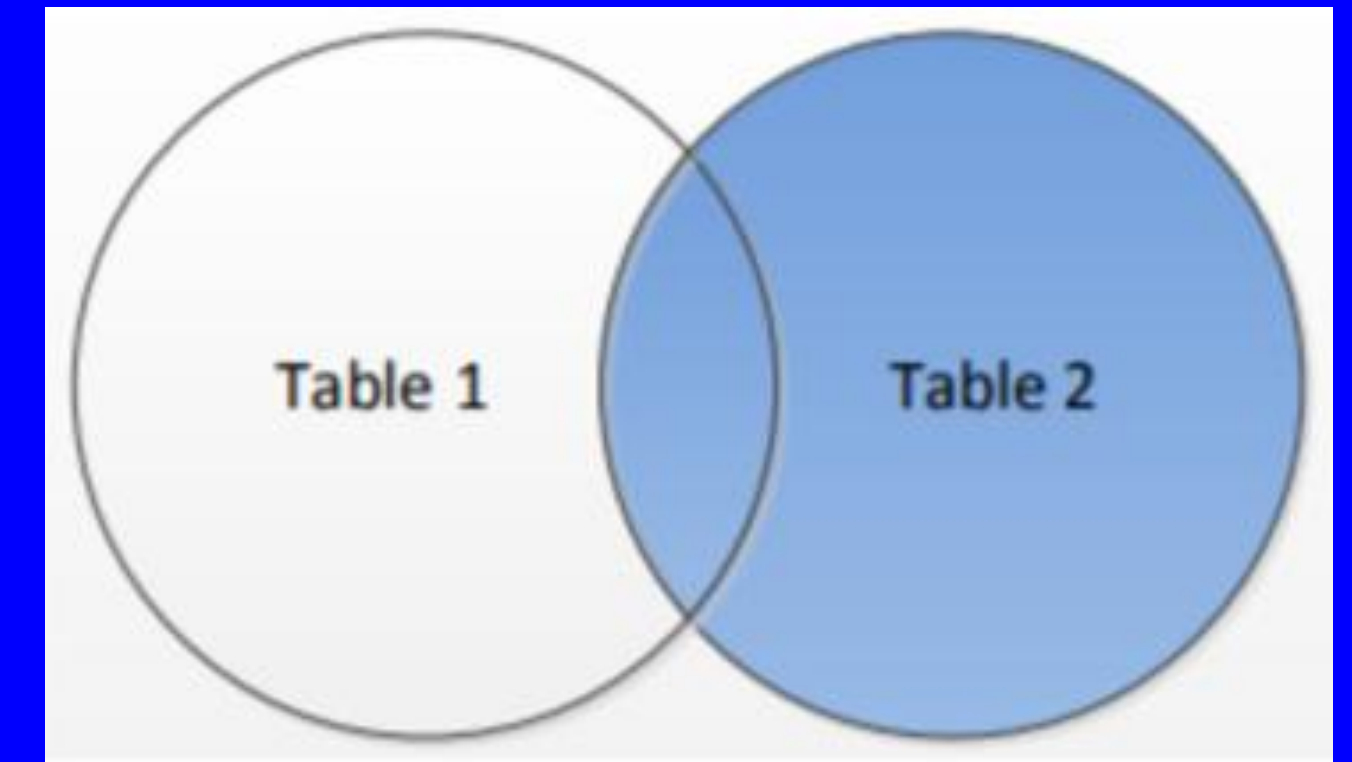
Left Outer Join in MariaDB

- $t1 \bowtie_{t1.id=t2.fk} t2$



```
SELECT *  
  FROM Table_1 t1  
LEFT JOIN Table_2 t2  
      ON t1.id = t2.fk;
```

Right Outer Join ⚡



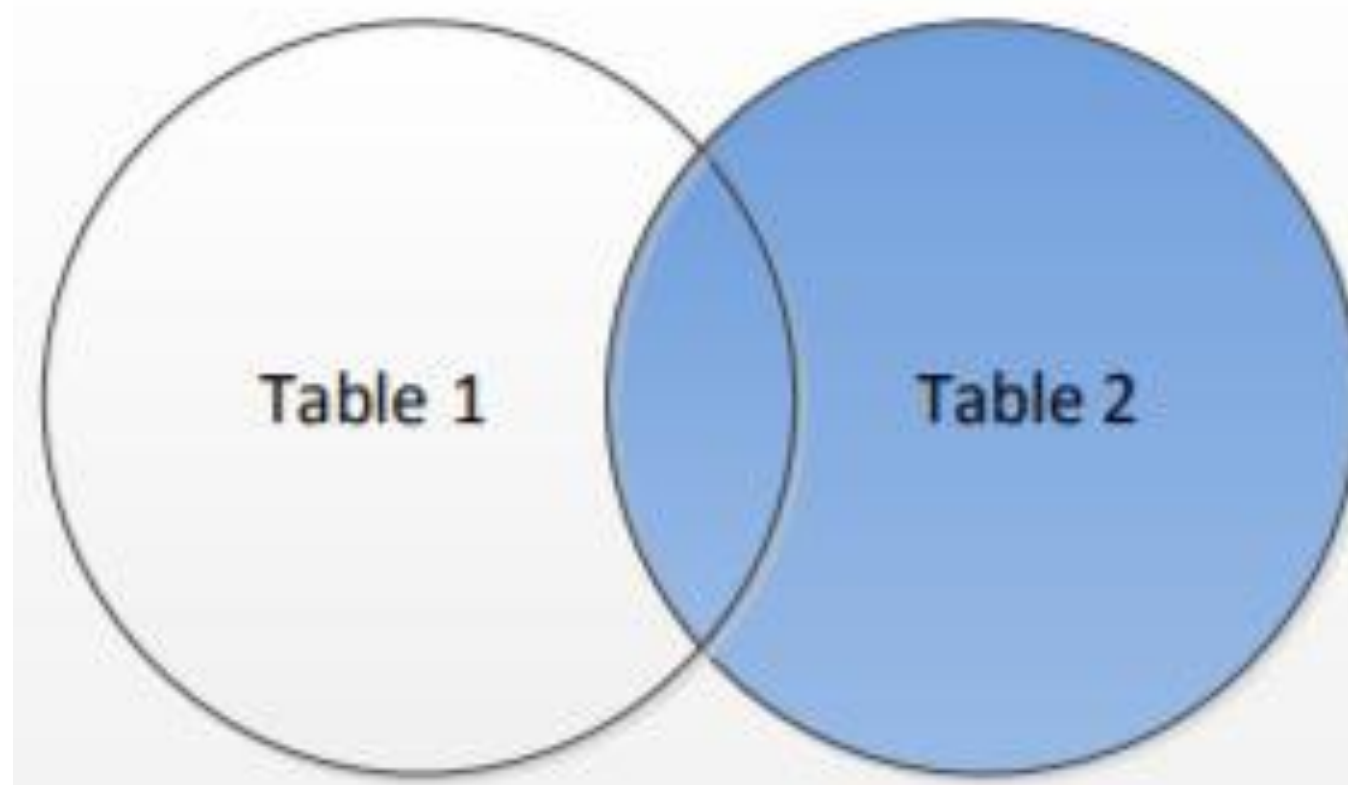
Right Outer Join als Relationenalgebra

- $r_1 \bowtie r_2$
- Bei unvollständigen Tupeln können auf der linken Seite NULL-Werte entstehen.

r_1			r_2			$r_1 \bowtie r_2$				
A	B	C	C	D	E	A	B	C	D	E
a_1	b_1	c_1	c_1	d_1	e_1	a_1	b_1	c_1	d_1	e_1
a_2	b_2	c_2	c_3	d_2	e_2	\perp	\perp	c_3	d_2	e_2

Right Outer Join in MariaDB

- $t1 \bowtie_{t1.id=t2.fk} t2$



```
SELECT *  
  FROM Table_1 t1  
 RIGHT JOIN Table_2 t2  
    ON t1.id = t2.fk;
```