

Vortrag 4 Joins

Datenbanken

Lukas Wais

Codersbay

Version: 19. April 2023

Inhaltsverzeichnis

Mengenlehre

- Venn Diagramme

- Relationenalgebra

- Relationenalgebra in SQL

Joins SQL

- Joins in SQL Syntax

Einige Symbole

- ▶ \wedge und
- ▶ \vee oder
- ▶ \Rightarrow impliziert; daraus folgt
- ▶ \Leftrightarrow logische Äquivalenz; beide Seiten haben den gleichen Wahrheitswert.
- ▶ \in ist Element von; $x \in M$ bedeutet: x ist in der Menge M .
- ▶ \notin ist nicht Element von
- ▶ $:$ für das gilt
- ▶ \forall für alle
- ▶ \exists es existiert ein

Mengen

Definition

Eine Menge ist eine Zusammenfassung von Objekte, es muss klar sein ob ein Objekt zur Menge gehört oder eben nicht. $a \in A$ vs. $a \notin A$.

Mengen können in aufgezählter Form

$$\{1, 23, 57, 66\}$$

in beschreibender Form

$$\{x : \text{ist_prim}(x)\}$$

und auch in durch das Erzeugungsprinzip angegeben werden.

Tupel

Definition

Tupel sind Listen die Objekte zusammenfassen. Diese Objekte müssen nicht unterschiedlich sein. Im Gegensatz zu Mengen spielt hier die Reihenfolge Objekte eine Rolle. Beispiel:

$$(1, 1, 1, 2) \neq (1, 2) \quad (1, 2) \neq (2, 1) \quad (2, 1) \neq (2, 1, 2)$$

In einer relationalen Datenbank ist eine Zeile ein Tupel.

Teilmenge und Gleichheit

Teilmenge

Eine Menge A heißt Teilmenge der Menge B , geschrieben $A \subseteq B$, genau dann, wenn jedes Element der Menge A auch ein Element der Menge B ist, also

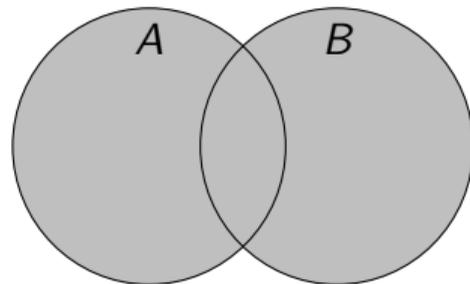
$$\forall a : a \in A \Rightarrow a \in B.$$

Gleichheit

Zwei Mengen, A, B sind gleich, geschrieben $A = B$, genau dann, wenn $x \in A$ genau dann gilt, wenn $x \in B$ ist, also

$$\forall a : a \in A \Leftrightarrow a \in B.$$

Venn Diagramme



Was ist die Relationenalgebra?

Was ist die Relationenalgebra?

Die Relationale Algebra ist eine formale Sprache, damit lassen sich Abfragen über ein relationales Schema ganz präzise definieren. Sie ist die theoretische Grundlage für Abfragesprachen.

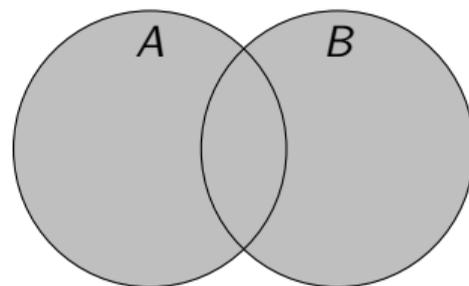
Vereinigung

Definition

$$A \cup B := \{x : x \in A \vee x \in B\}$$

	<u>X</u>	<u>Y</u>	<u>Z</u>		<u>X</u>	<u>Y</u>	<u>Z</u>
A:	1	2	3	B:	7	8	9
	4	5	6		4	5	6

	<u>X</u>	<u>Y</u>	<u>Z</u>
A ∪ B:	1	2	3
	7	8	9
	4	5	6



SQL - Vereinigung

```
SELECT * FROM A  
UNION  
SELECT * FROM B;
```

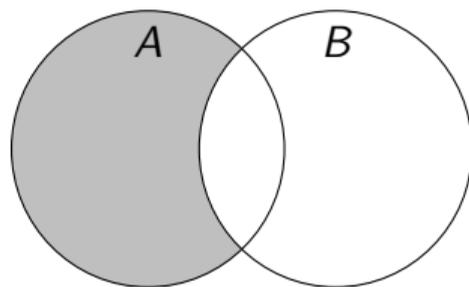
Differenz

Definition

$$A \setminus B := \{x : x \in A \wedge x \notin B\}$$

$A:$	<table><thead><tr><th>X</th><th>Y</th><th>Z</th></tr></thead><tbody><tr><td>1</td><td>2</td><td>3</td></tr><tr><td>4</td><td>5</td><td>6</td></tr></tbody></table>	X	Y	Z	1	2	3	4	5	6	$B:$	<table><thead><tr><th>X</th><th>Y</th><th>Z</th></tr></thead><tbody><tr><td>7</td><td>8</td><td>9</td></tr><tr><td>4</td><td>5</td><td>6</td></tr></tbody></table>	X	Y	Z	7	8	9	4	5	6
X	Y	Z																			
1	2	3																			
4	5	6																			
X	Y	Z																			
7	8	9																			
4	5	6																			

$A \setminus B:$	<table><thead><tr><th>X</th><th>Y</th><th>Z</th></tr></thead><tbody><tr><td>1</td><td>2</td><td>3</td></tr></tbody></table>	X	Y	Z	1	2	3
X	Y	Z					
1	2	3					



SQL - Differenz

```
SELECT * FROM A  
MINUS  
SELECT * FROM B;
```

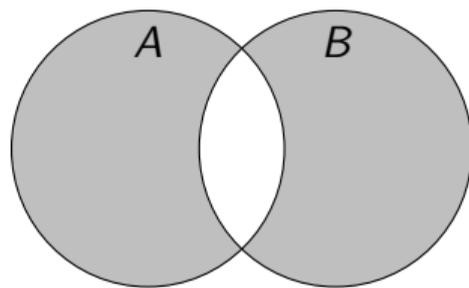
Symmetrische Differenz

Definition

$$A \triangle B := \{x : (x \in A \vee x \in B) \wedge x \notin A \cap B\}$$

	<u>X</u>	<u>Y</u>	<u>Z</u>		<u>X</u>	<u>Y</u>	<u>Z</u>
A:	1	2	3	B:	7	8	9
	4	5	6		4	5	6

	<u>X</u>	<u>Y</u>	<u>Z</u>
A \triangle B:	1	2	3
	7	8	9



SQL - Symmetrische Differenz

```
(SELECT * FROM A  
UNION  
SELECT * FROM B)  
MINUS  
(SELECT * FROM B  
INTERSECT  
SELECT * FROM A);
```

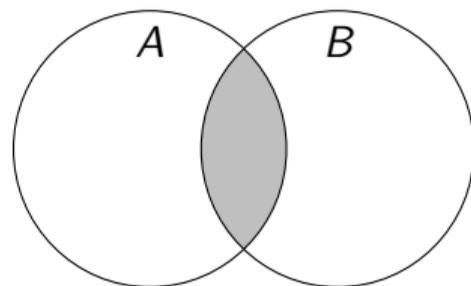
Durchschnitt

Definition

$$A \cap B := \{x : x \in A \wedge x \in B\}$$

$A:$	<table><thead><tr><th>X</th><th>Y</th><th>Z</th></tr></thead><tbody><tr><td>1</td><td>2</td><td>3</td></tr><tr><td>4</td><td>5</td><td>6</td></tr></tbody></table>	X	Y	Z	1	2	3	4	5	6	$B:$	<table><thead><tr><th>X</th><th>Y</th><th>Z</th></tr></thead><tbody><tr><td>7</td><td>8</td><td>9</td></tr><tr><td>4</td><td>5</td><td>6</td></tr></tbody></table>	X	Y	Z	7	8	9	4	5	6
X	Y	Z																			
1	2	3																			
4	5	6																			
X	Y	Z																			
7	8	9																			
4	5	6																			

$A \cap B:$	<table><thead><tr><th>X</th><th>Y</th><th>Z</th></tr></thead><tbody><tr><td>4</td><td>5</td><td>6</td></tr></tbody></table>	X	Y	Z	4	5	6
X	Y	Z					
4	5	6					



SQL - Durchschnitt

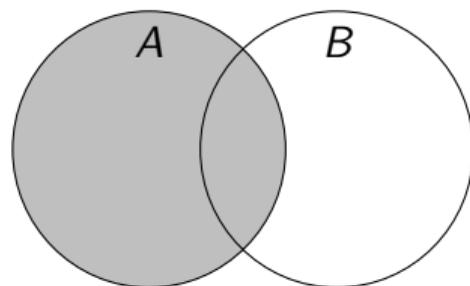
```
SELECT * FROM A  
INTERSECT  
SELECT * FROM B;
```

Joins in SQL

Beispiele an Eltern und Kindern

Wir haben zwei Tabellen: Paare (also Eltern) und Kinder. Es gibt kinderlose Paare, Paare mit Kindern und Waisenkinder. Wir wollen die Eltern und Kinder in Abfragen verknüpfen; bei den Symbolen steht der linke Kreis A für die Tabelle Paare und der rechte Kreis B für die Tabelle Kinder.

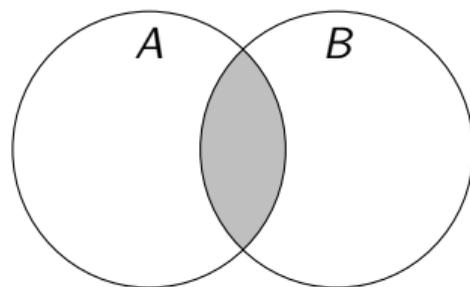
LEFT JOIN



Alle Paare und (falls es Kinder gibt) auch diese.

```
SELECT * FROM Paare  
LEFT JOIN Kinder ON Paare.Key = Kinder.Key
```

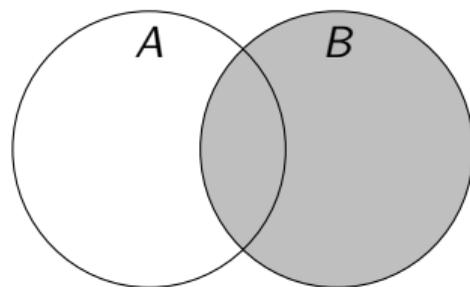
INNER JOIN



Nur Paare, die Kinder haben

```
SELECT * FROM Paare  
INNER JOIN Kinder ON Paare.Key = Kinder.Key
```

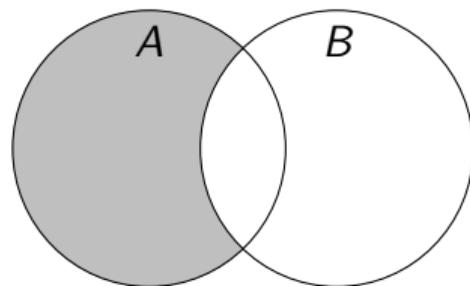
RIGHT JOIN



Alle Kinder und (falls es Eltern gibt) auch diese

```
SELECT * FROM Paare  
RIGHT JOIN Kinder ON Paare.Key = Kinder.Key
```

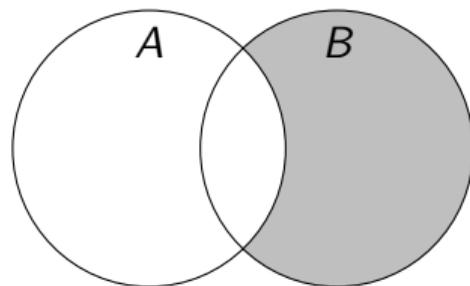
LEFT JOIN IS NULL



Nur Paare, die keine Kinder haben

```
SELECT * FROM Paare
  LEFT JOIN Kinder ON Paare.Key = Kinder.Key
 WHERE Kinder.Key IS NULL
```

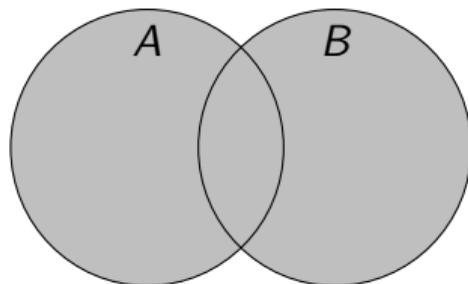
Right JOIN IS NULL



Nur Waisenkinder

```
SELECT * FROM Paare  
  RIGHT JOIN Kinder ON Paare.Key = Kinder.Key  
 WHERE Paare.Key IS NULL
```

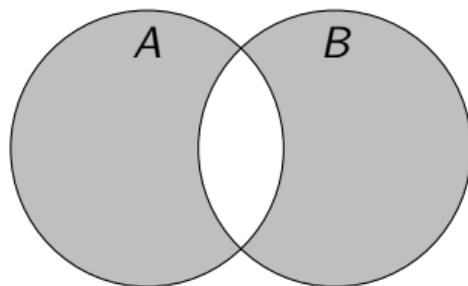
FULL JOIN



Alle Paare und alle Kinder

```
SELECT * FROM Paare  
FULL JOIN Kinder ON Paare.Key = Kinder.Key
```

FULL JOIN IS NULL

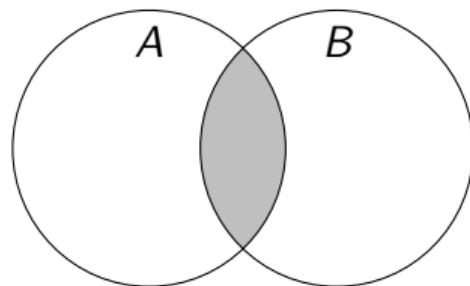


Alle kinderlosen Paare und alle Waisenkinder

```
SELECT * FROM Paare
FULL JOIN Kinder ON Paare.Key = Kinder.Key
WHERE Kinder.Key IS NULL OR Paare.Key IS NULL
```

LEFT JOIN IS NOT NULL

Alternativ zum INNER JOIN



Alle Paare und (falls es Kinder gibt) auch diese, wobei es ein Kind geben muss

```
SELECT * FROM Paare
LEFT JOIN Kinder ON Paare.Key = Kinder.Key
WHERE Kinder.Key IS NOT NULL
```

Varianten

Achtung: oft gibt es mehrere mögliche Wege Abfragen durchzuführen. Auch gibt es syntaktische Unterschiede in den jeweiligen DBMS.

Join clauses für die MariaDB:

<https://mariadb.com/kb/en/joining-tables-with-join-clauses/>