

Prof. DI Dr. Erich Gams

Datenmanipulation mit SQL

Joins und mehr.....

informationssysteme htl-wels



Übersicht ➡ Was lernen wir?

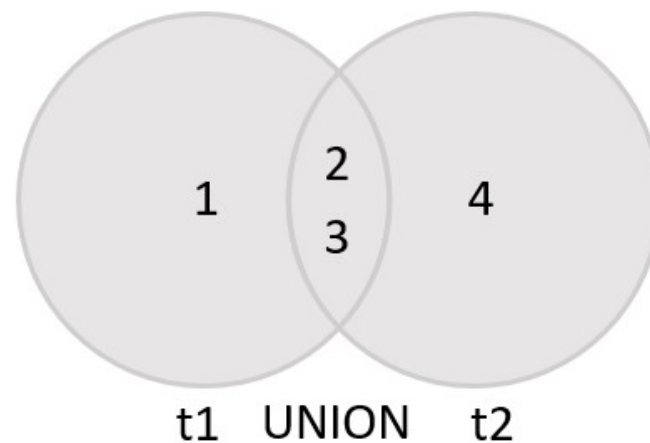
- ■ Elementare Operationen auf Relationen
 - Union
 - Intersection
 - Differenz
 - Join
- ■ daraus einfache Anfragen in SQL

Union oder Verbund

- Union verbindet Datensätze aus 2 Relationen.
- Dabei werden Tupeln ausgegeben, die entweder in der einen Relation oder in der anderen Relation enthalten sind.
- Entspricht der Mengenoperation „vereinigt“.
- Die Relationen müssen folgende Bedingungen erfüllen:
 - Beide Relationen haben dieselbe Anzahl von Spalten oder man selektiert die Spalten.
 - Die Spalten haben in ihrer Reihenfolge übereinstimmende Domänen.

Union SQL-Beispiel

- Fügt die Zeilen zweier Tabellen mit gleicher Spaltenzahl in einer Tabelle zusammen.
- Die Namen der jeweiligen Spalten der zwei Tabellen müssen nicht identisch sein, lediglich der Datentyp bzw. Wertebereich des Inhalts



Union SQL Beispiel

- Doppelte Zeilen werden unterdrückt!

```
SELECT id  
FROM t1  
UNION  
SELECT id  
FROM t2;
```

```
+-----+  
| id |  
+-----+  
| 1 |  
| 2 |  
| 3 |  
| 4 |  
+-----+  
4 rows in set (0.00 sec)
```

- Mit UNION ALL werden sie angezeigt.

UNION ALL

```
SELECT id  
FROM t1  
UNION ALL  
SELECT id  
FROM t2;
```

```
+-----+  
| id |  
+-----+  
| 1 |  
| 2 |  
| 3 |  
| 2 |  
| 3 |  
| 4 |  
+-----+
```

```
6 rows in set (0.00 sec)
```

Intersection (Schnittmenge)

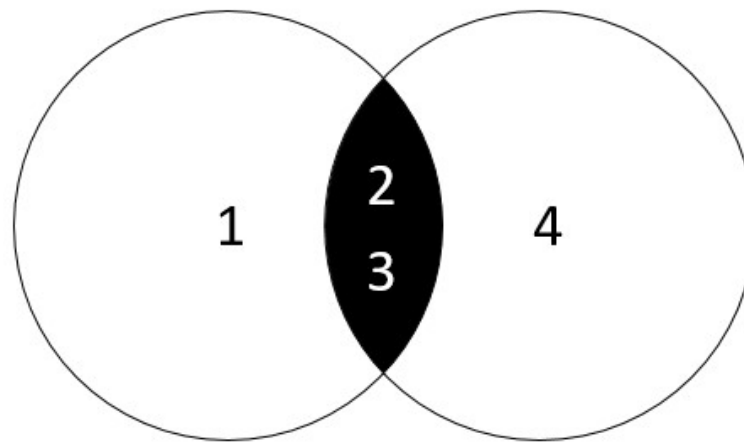
- Vereinigt 2 Relationen und gibt jene Tupel in die Ergebnismenge, welche in beiden Relationen vorkommen.
- Es gelten die Regeln der Union.

```
(SELECT column_list  
FROM table_1)  
INTERSECT  
(SELECT column_list  
FROM table_2);
```

Intersection (Schnittmenge)

- INTERSECT steht in MySQL nicht zur Verfügung, da es zum SQL 92 Intermediate Level gehört.
- Es kann aber nachgebildet werden, z. B. durch ein geschachteltes SELECT-Statement nachgebildet werden.

MySQL Intersect



id

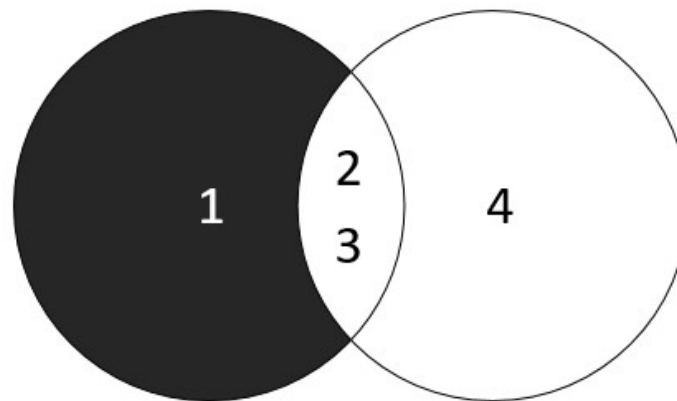
2

3

```
SELECT DISTINCT id
FROM t1
WHERE id IN (SELECT id FROM t2);
```

Differenz

- Liefert die Zeilen der ersten Tabelle, die in der zweiten Tabelle nicht enthalten sind.



SQL Differenz

```
1 SELECT id FROM t1
2 MINUS
3 SELECT id FROM t2;
```

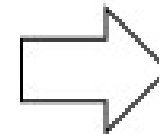
id
1
2
3

t1 table

MINUS

id
2
3
4

t2 table



id
1

MySQL Differenz

```
SELECT
    id
FROM
    t1
LEFT JOIN
    t2 USING (id)
WHERE
    t2.id IS NULL;
```

CROSS JOIN

Verlag (Verlag_ID, Name, Plz)
Postleitzahl (Plz, Ort)

- Die häufigste Form einer Abfrage, die mehrere Tabellen umfasst, ist jener Fall, in Tabellen wieder zusammengeführt werden sollen.
- Das Verknüpfen ("joining") von Tabellen wird als JOIN bezeichnet.

```
SELECT *  
FROM Verlag, Postleitzahl;
```

- Es werden alle möglichen Kombinationen der Zeilen dieser Tabellen untereinander gebildet (**kartesischen** Produkt).
- Verbindet jede Zeile der ersten Tabelle mit jeder Zeile der zweiten Tabelle

Join von Tabellen

Tabelle VERLAG

Verlag#	Name	plz
1	A	4020
2	B	4600

Tabelle POSTLEITZAHL

plz	Ort
4020	Linz
4040	Linz
4600	Wels

Ergebnistabelle

Verlag#	Name	plz	plz	ort
1	A	4020	4020	Linz
1	A	4020	4040	Linz
1	A	4020	4600	Wels
2	B	4600	4600	Wels
2	B	4600	4040	Linz
2	B	4600	4600	Wels

Achtung
Fehler!

Join von Tabellen

- Diese Auswahl von bestimmten Zeilen (**Selektion**) der Zwischentabelle muss noch in einer WHERE-Klausel angegeben werden:

... **WHERE** Verlag.plz = Postleitzahl.plz;

Ergebnistabelle

Verlag#	Name	plz	plz	ort
1	A	4020	4020	Linz
2	B	4600	4020	Linz

**Achtung
Fehler !**

Join von Tabellen (alte Schreibweise)

- Da das verknüpfende Attribut in den beiden Ausgangstabellen denselben Namen hat,
 - muss es noch *qualifiziert* werden,
 - d.h. durch Voranstellen des Tabellennamens wird es eindeutig.
- Die vollständige Query sieht also so aus:
SELECT *
FROM Verlag, Postleitzahl
WHERE Verlag.plz = Postleitzahl.plz;

Inner Join

- Eine andere (neuere) Schreibweise für die diese Verknüpfung:

```
SELECT * FROM Verlag  
INNER JOIN Postleitzahl ON Postleitzahl.plz  
=Verlag.plz;
```

- Verbindet Datensätze aus zwei Tabellen, sobald ein gemeinsames Feld dieselben Werte enthält.

Inner Join Example

■ Members und Committees

```
SELECT
    m.member_id,
    m.name member,
    c.committee_id,
    c.name committee
FROM
    members m
INNER JOIN committees c
    ON c.name = m.name;
```

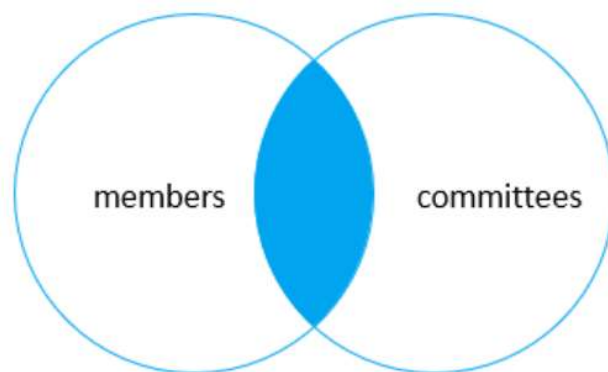
	member_id	name
▶	1	John
	2	Jane
	3	Mary
	4	David
	5	Amelia

	committee_id	name
▶	1	John
	2	Mary
	3	Amelia
	4	Joe

Inner Join Example

Result

	member_id	member	committee_id	committee
▶	1	John	1	John
	3	Mary	2	Mary
	5	Amelia	3	Amelia



Inner Join Examples

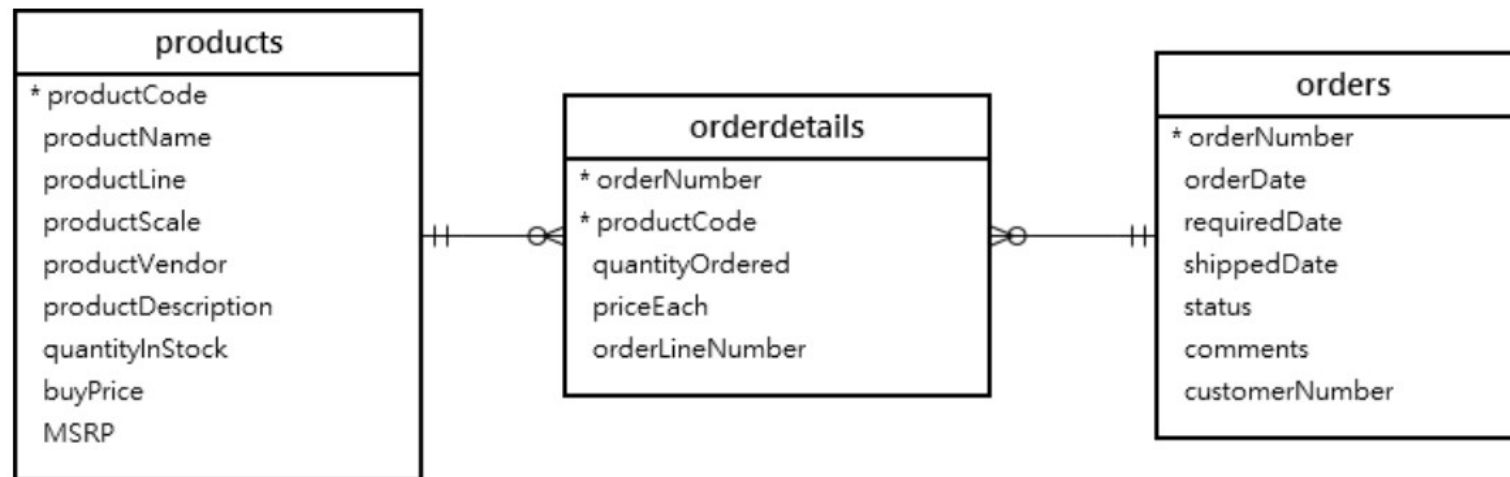
- Wenn die Spaltennamen in beiden Tabellen gleich heißen:

```
SELECT
    m.member_id,
    m.name member,
    c.committee_id,
    c.name committee
FROM
    members m
INNER JOIN committees c USING(name);
```

Inner Join mit mehreren Tabellen

```
SELECT
    select_list
FROM t1
INNER JOIN t2 ON join_condition1
INNER JOIN t3 ON join_condition2
...;
```

Beispiel: Inner Join über 3 Tabellen



Beispiel: Inner Join über 3 Tabellen

```
SELECT
    orderNumber,
    orderDate,
    orderLineNumber,
    productName,
    quantityOrdered,
    priceEach
FROM
    orders
INNER JOIN
    orderdetails USING (orderNumber)
INNER JOIN
    products USING (productCode)
ORDER BY
    orderNumber,
    orderLineNumber;
```

Beispiel: Inner Join über 3 Tabellen

	orderNumber	orderDate	orderLineNumber	productName	quantityOrdered	priceEach
►	10100	2003-01-06	1	1936 Mercedes Benz 500k Roadster	49	35.29
	10100	2003-01-06	2	1911 Ford Town Car	50	55.09
	10100	2003-01-06	3	1917 Grand Touring Sedan	30	136.00
	10100	2003-01-06	4	1932 Alfa Romeo 8C2300 Spider Sport	22	75.46
	10101	2003-01-09	1	1928 Mercedes-Benz SSK	26	167.06

.. and using Operators

```
SELECT
    orderNumber,
    productName,
    msrp,
    priceEach
FROM
    products p
INNER JOIN orderdetails o
    ON p.productcode = o.productcode
    AND p.msrp > o.priceEach
WHERE
    p.productcode = 'S10_1678';
```

Weitere Join Arten

- Beispiele mit den zwei Tabellen "Mitarbeiter" und "Projekte":

Mitarbeiter

Nachname	Vorname	Geburtsdatum
Milke	Lise	3.6.1934
Huber	Karl	16.12.1964
Trunstein	Helga	30.7.1956

Projekte

Projekt	Nachname	Vorname
Neubau	Huber	Anna
Werbung	Trunstein	Helga
Design	Kohlmeier	Johann

Left Outer Join = Left Join

- Mit einem Left Join wird eine sogenannte linke Inklusionsverknüpfung erstellen.
- Linke Inklusionsverknüpfungen schließen **alle Datensätze** aus **der ersten (linken) Tabelle** ein, auch wenn **keine** entsprechenden Werte für Datensätze in der zweiten Tabelle existiert.

Beispiel Left Join

■ Das SQL-Statement

```
SELECT * FROM Mitarbeiter LEFT JOIN Projekte ON  
(Mitarbeiter.Nachname = Projekte.Nachname AND  
Mitarbeiter.Vorname = Projekte.Vorname)
```

■ liefert:

Mitarbeiter. Nachname	Mitarbeiter. Vorname	Geburtsdatum	Projekt	Projekte. Nachname	Projekte. Vorname
Milke	Lise	3.6.1934	NULL	NULL	NULL
Huber	Karl	16.12.1964	NULL	NULL	NULL
Trunstein	Helga	30.7.1956	Werbung	Trunstein	Helga

Right Outer Join = Right Join

- Mit einem Right Join wird eine sogenannte rechte Inklusionsverknüpfung erstellen.
- Rechte Inklusionsverknüpfungen schließen **alle Datensätze** aus der zweiten (rechten) Tabelle ein, **auch wenn keine** entsprechenden Werte für Datensätze **in der ersten Tabelle** existieren.

Beispiel Right Join

■ Das SQL-Statement

```
SELECT * FROM Mitarbeiter RIGHT JOIN Projekte ON  
(Mitarbeiter.Nachname = Projekte.Nachname AND  
Mitarbeiter.Vorname = Projekte.Vorname)
```

■ liefert:

Mitarbeiter. Nachname	Mitarbeiter. Vorname	Geburtsdatum	Projekt	Projekte. Nachname	Projekte. Vorname
NULL	NULL	NULL	Neubau	Huber	Anna
Trunstein	Helga	30.7.1956	Werbung	Trunstein	Helga
NULL	NULL	NULL	Design	Kohlmeier	Johann

Full Outer Join = Full Join

- Eine Kombination von Left Outer Join und Right Outer Join.

- Beispiel:

```
SELECT * FROM Mitarbeiter FULL JOIN Projekte ON  
(Mitarbeiter.Nachname = Projekte.Nachname AND  
Mitarbeiter.Vorname = Projekte.Vorname)
```

Mitarbeiter. Nachname	Mitarbeiter. Vorname	Geburtsdatum	Projekt	Projekte. Nachname	Projekte. Vorname
Milke	Lise	3.6.1934	NULL	NULL	NULL
Huber	Karl	16.12.1964	NULL	NULL	NULL
Trunstein	Helga	30.7.1956	Werbung	Trunstein	Helga
NULL	NULL	NULL	Neubau	Huber	Anna
NULL	NULL	NULL	Design	Kohlmeier	Johann

Selektion

Tabelle 1

Projektion

Tabelle 1

Join

Tabelle 1

Tabelle 2