



Datenbanken und SQL

(Woche 4 - Tag 1)

Agenda

- UNION

- Definition + Motivation
- Beispiele + Syntax
- Differenzierung zu UNION ALL
- Anwendung bzgl. FULL JOIN

UNION

Definition + Motivation

- Jeder SELECT-Befehl produziert eine Ausgabetabelle, die - mengentheoretisch betrachtet - als eine „**Menge von Datensätzen**“ betrachtet werden kann.
- Der UNION zweier SELECT-Befehle entspricht dann der mengentheoretischen **Vereinigung** der beiden Ausgabetafellen.
- Ein solcher UNION kann im Einzelfall auch inhaltlich interessant sein. Etwa, wenn wir die Entitäten unterschiedlicher Tabellen unter einer **gemeinsamen Überschrift** (Beziehungsweise: bezüglich eines gemeinsamen Oberbegriffs) ausgeben lassen wollen.
- Im Rahmen von MySQL besteht die zentrale Motivation für den Einsatz eines UNION aber sicherlich in der Möglichkeit, mit diesem den **FULL JOIN** zu „simulieren“.

Beispiel(1) + Syntax

Aufgabenstellung:

Alle Hersteller (Hersteller-ID und Hersteller-Name) sollen zusammen mit allen Speditionen (Spedition-ID und Spedition-Name) unter den Überschriften „ID“ und „Firmenname“ ausgegeben werden.

Folgender Code vereinigt beide Ausgabetablellen:

```
SELECT
    Hersteller_ID AS "ID",
    Hersteller_Name AS "Firmenname"
FROM Hersteller
```

UNION

```
SELECT
    Spedition_ID ,
    Spedition_Name
FROM Spedition;
```

ID	Firmenname
1	Contrabit
2	AntiByte
3	UltraBug
4	Hatnix 1992
5	Ladenhut AG
1	Speedvan GmbH
2	RocketLogistic AG
3	Turbo Transport
4	Parktnur
5	Kriegtnix
6	Ganzal Lein

Beispiel(2) + Syntax

Aufgabenstellung:

Alle Kunden (Kunde-ID und Vorname) sollen zusammen mit allen Produkten (Produkt-ID und Preis) unter den Überschriften „ID“ und „Häh ??“ ausgegeben werden.

Folgender Code vereinigt beide Ausgabetabellen:

```
SELECT
    Kunde_ID AS "ID",
    Vorname AS "Häh ??",
FROM Kunde

UNION

SELECT
    Produkt_ID ,
    Euro_Preis
FROM Produkt;
```

ID	Häh ??
2	Vera
3	Witali
4	Rita
5	Eva
6	Gala
7	Peter
10	Elli
1	15.979999542236328
2	22.75
3	31.690000534057617
4	45.04999923706055
5	98
6	1000

Gemeinsame Ausgabe (obwohl unterschiedlichen Typs!)
=> daher existiert auch keine sinnvolle gemeinsame Überschrift

Beispiel(3) + Syntax

Aufgabenstellung:

Alle Kunden (Kunde-ID und Vorname) sollen zusammen mit allen Produkten (Produkt-ID) unter den Überschriften „ID“ und „Rufname“ ausgegeben werden.

Folgender Code führt aber zu einer Fehlermeldung:

```
SELECT
    Kunde_ID AS "ID",
    Vorname AS "Rufname "
FROM Kunde
```

UNION

```
SELECT
    Produkt_ID
FROM Produkt;
```

MySQL meldet:

#1222 - Die verwendeten SELECT-Befehle liefern unterschiedliche Anzahlen von Feldern zurück

UNION ALL

Beispiel + Syntax

Den Unterschied zwischen **UNION** und **UNION ALL** erarbeiten wir uns schrittweise.

Zunächst betrachten wir die beiden Ausgabetabellen, die sich bei den folgenden **2 SELECT-Anweisungen** ergeben:

```
SELECT  
    Kunde_ID AS "ID"  
FROM Kunde;
```

ID
1
2
3
4
5
6
7

Beispiel + Syntax

Den Unterschied zwischen **UNION** und **UNION ALL** erarbeiten wir uns schrittweise.

Zunächst betrachten wir die beiden Ausgabetabellen, die sich bei den folgenden **2 SELECT-Anweisungen** ergeben:

```
SELECT  
    Kunde_ID AS "ID"  
FROM Kunde;
```

ID
1
2
3
4
5
6
7

```
SELECT  
    Produkt_ID  
FROM Produkt;
```

Produkt_ID
3
4
1
2
5
6

Beispiel + Syntax

Wir betrachten nun die Ausgabe, die sich bei dem **UNION** beider SELECT-Anweisungen ergibt:

```
SELECT  
    Kunde_ID AS "ID"  
FROM Kunde;
```

UNION

```
SELECT  
    Produkt_ID  
FROM Produkt;
```

Beispiel + Syntax

Wir betrachten nun die Ausgabe, die sich bei dem **UNION** beider SELECT-Anweisungen ergibt:

SELECT

Kunde_ID AS "ID"

FROM Kunde;

Semikolon **muss gelöscht** werden!

UNION

SELECT

Produkt_ID

FROM Produkt;

Beispiel + Syntax

Wir betrachten nun die Ausgabe, die sich bei dem **UNION** beider SELECT-Anweisungen ergibt:

```
SELECT
    Kunde_ID AS "ID"
FROM Kunde
```

UNION

```
SELECT
    Produkt_ID
FROM Produkt;
```

ID
1
2
3
4
5
6
7

Dubletten werden vermieden!

(wie auch bei der mengentheoretischen Vereinigung)

⇒ Die beiden Primärschlüsselwerte:

1 (Kunde-ID) und

1(Produkt-ID) werden nicht doppelt ausgegeben.

Beispiel + Syntax

Wir vergleichen dies mit der Ausgabe, die sich bei dem **UNION ALL** ergibt:

```
SELECT  
    Kunde_ID AS "ID"  
FROM Kunde
```

UNION ALL

```
SELECT  
    Produkt_ID  
FROM Produkt;
```

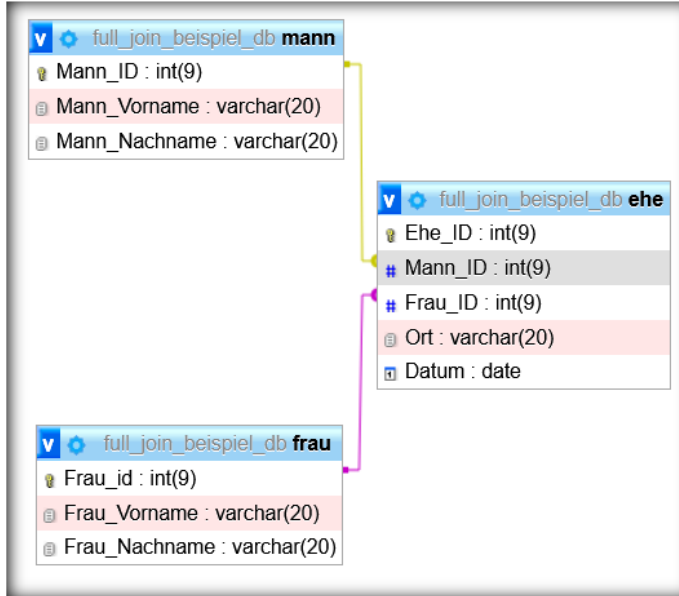
ID
1
2
3
4
5
6
7
3
4
1
2
5
6

Dubletten werden zugelassen!

FULL JOIN

Aufgabenstellung:

Datenbankschema



Die eingepflegten Daten

Mann_ID	Mann_Vorname	Mann_Nachname
1	Peter	Mayer
2	Klaus	Schmidt

Ehe_ID	Mann_ID	Frau_ID	Ort	Datum
1	1	2	Berlin	2012-12-12

Frau_id	Frau_Vorname	Frau_Nachname
1	Petra	Mayer
2	Claudia	Boll

Gewünschte Ausgabe:

mann_vorname	mann_nachname	frau_vorname	frau_nachname
Peter	Müller	Carina	Boll
Klaus	Schmidt	NULL	NULL
NULL	NULL	Petra	Müller

Teillösung – LEFT JOIN:

```
SELECT Mann_Vorname, Mann_Nachname, Frau_Vorname, Frau_Nachname  
FROM Mann LEFT JOIN Ehe ON Mann.Mann_ID=Ehe.Mann_ID  
LEFT JOIN Frau ON Ehe.Frau_ID=Frau.Frau_id;
```

Mann_Vorname	Mann_Nachname	Frau_Vorname	Frau_Nachname
Peter	Mayer	Claudia	Boll
Klaus	Schmidt	NULL	NULL

Teillösung – RIGHT JOIN:

```
SELECT Mann_Vorname, Mann_Nachname, Frau_Vorname, Frau_Nachname  
FROM Mann RIGHT JOIN Ehe ON Mann.Mann_ID=Ehe.Mann_ID  
      RIGHT JOIN Frau ON Ehe.Frau_ID=Frau.Frau_id;
```

Mann_Vorname	Mann_Nachname	Frau_Vorname	Frau_Nachname
NULL	NULL	Petra	Mayer
Peter	Mayer	Claudia	Boll

Vereinigung beider Teillösungen mittels **UNION**:


```
SELECT Mann_Vorname, Mann_Nachname, Frau_Vorname, Frau_Nachname  
FROM Mann LEFT JOIN Ehe ON Mann.Mann_ID=Ehe.Mann_ID  
      LEFT JOIN Frau ON Ehe.Frau_ID=Frau.Frau_id
```

UNION

```
SELECT Mann_Vorname, Mann_Nachname, Frau_Vorname, Frau_Nachname  
FROM Mann RIGHT JOIN Ehe ON Mann.Mann_ID=Ehe.Mann_ID  
      RIGHT JOIN Frau ON Ehe.Frau_ID=Frau.Frau_id;
```

Mann_Vorname	Mann_Nachname	Frau_Vorname	Frau_Nachname
Peter	Mayer	Claudia	Boll
Klaus	Schmidt	NULL	NULL
NULL	NULL	Petra	Mayer

Gemeinsame Übung („Live-Coding“) -> A_04_01_01



Aufgabe_04_01_01

Formulieren Sie bitte entsprechende SQL-Anweisungen für folgende Aufgabstellungen:

- Der Besitzer von „Geld_her“ wollte eine Spedition löschen, erhielt allerdings eine Fehlermeldung. Ändern Sie bitte die Struktur der Datenbank, damit dies zukünftig ohne Verletzung der Referentiellen Integrität möglich ist.
- Alle Kunden (Kunde-ID und Vorname), alle Produkte (Produkt-ID und Produktnamen) und alle Hersteller (Hersteller-ID und Herstellername) sollen unter den Unterschriften „ID“ und „Entitäten-Name“ ausgegeben werden.
- Für alle Kunden (Vorname und Nachname) sollen alle Speditionen (Name) ausgegeben werden, die ihn bereits mindestens 1-mal belieferten. Es sollen aber auch Kunden ausgegeben werden, die bisher noch nichts bestellten, so wie auch Speditionen aufgeführt werden sollen, die bisher noch kein Produkt von „Geld_her“ ausgeliefert haben.


Zur Erläuterung der Aufgabenstellung zeigen wir die gewünschte Ausgabetable:

Vorname	Nachname	Spedition_name
Eli	Rut	Speedvan GmbH
Eli	Rut	Rockett,ologic AG
Eli	Rut	Turbo Transport
Vera	Deise	Speedvan GmbH
Vera	Deise	Turbo Transport
Vera	Deise	Rockett,ologic AG
Witali	Mymrow	Rockett,ologic AG
Witali	Mymrow	Turbo Transport
Witali	Mymrow	Speedvan GmbH
Rita	Mymrow	Rockett,ologic AG
Rita	Mymrow	Speedvan GmbH
Rita	Mymrow	Turbo Transport
Eva	Hahn	Speedvan GmbH
Gala	Nieds	NULL
Peter	Kaufnix	NULL
NULL	NULL	Parktrur
NULL	NULL	Kriegnix
NULL	NULL	Ganzal Lein

WBS TRAINING AG
Lorenzweg 5
D-12099 Berlin
Amtsgericht Berlin HBR 68531
Sitz der Gesellschaft: Berlin

Vorstand:
Heinrich Krombichler,
Joachim Gliese
Aufsichtsrat (Vorsitz): Dr. Daniel Stadler
USt-IDNr.: DE 209 768 248

GLS Gemeinschaftsbank eG
IBAN: DE18 4306 0967 1146 1814 00
BIC: GENODEM3GLS



GLS ist ein Markenname
© 2014 GLS. Alle Rechte vorbehalten.
GLS ist ein Markenname
© 2014 GLS. Alle Rechte vorbehalten.