SQL

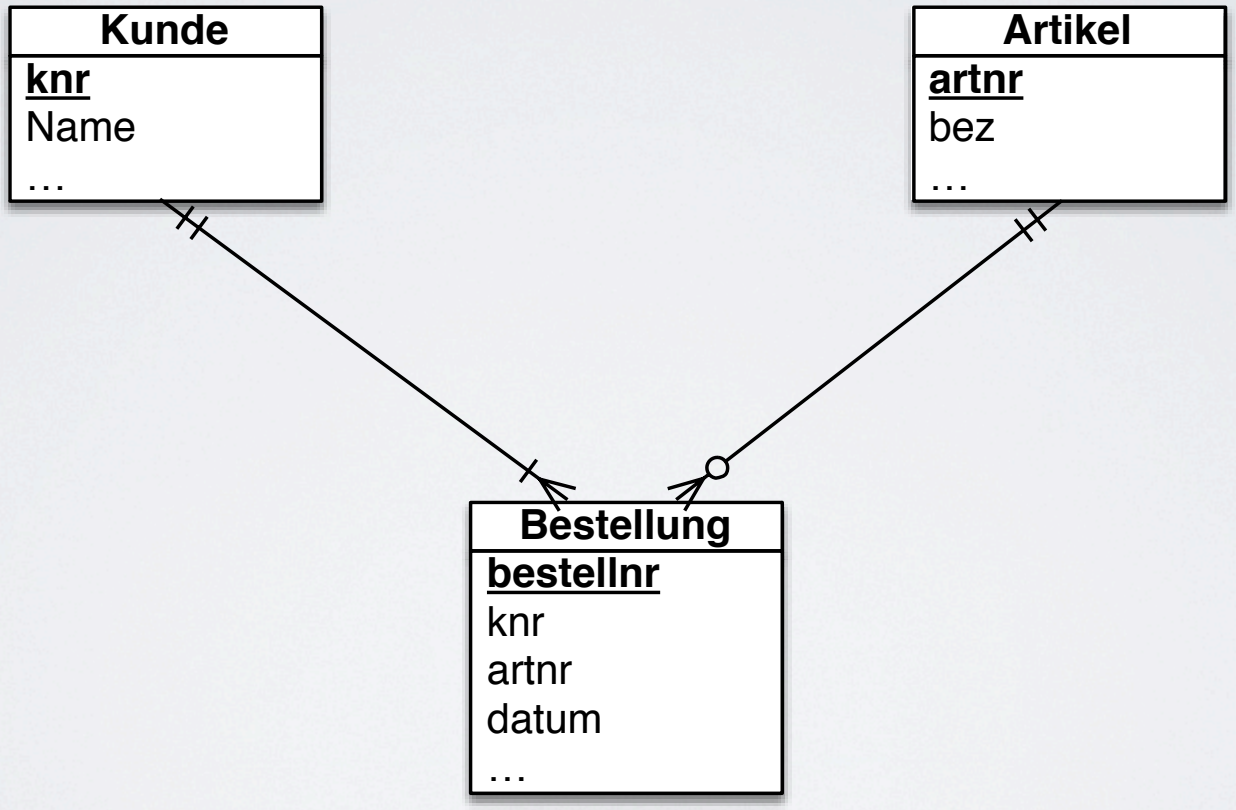
Joins

Vereinigung: union

- Alle Datensätze aus allen Tabellen

```
select * from tab_1
union
select * from tab_2;
```

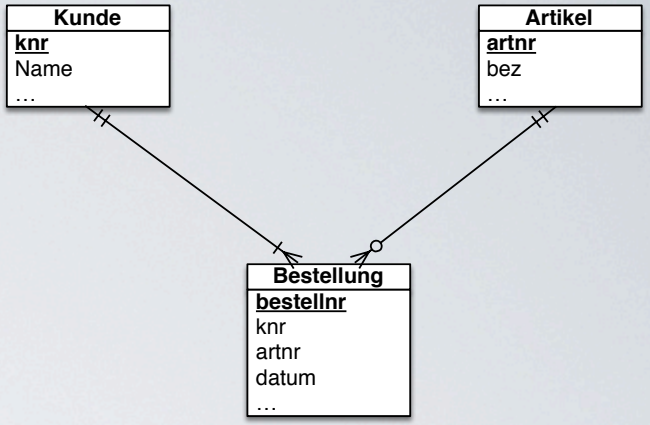
TABELLEN VERBINDEN



Verknüpfen via Schlüssel

- Selektion von Daten aus mehr als einer Tabelle gleichzeitig
- Prinzip:
  Verknüpfen von Primär- und Fremdschlüsseln

Einfache Selektion




```
select name, datum, bez
from kunde, bestellung, artikel
where kunde.knr = bestellung.knr
and bestellung.artnr = artikel.artnr;
```

- Die Verknüpfung erfolgt durch Attributvergleich in der where-Klausel
- Notwendig: Attribute mit Tabellenname (oder Aliasname für Tabelle) angeben, wenn mehrdeutig

Ein Problem

- Selektieren Sie alle Kunden, die in Basel wohnen und zeigen sie dabei auch eventuell getätigte Einkäufe im letzten Monat an
  (*Jeder Kunde soll angezeigt werden, aber nicht jeder hat auch aktuelle Käufe*)

- Dies lässt sich nicht durch eine einfache Verknüpfung lösen
- SQL kennt eine eigene Syntax, um beliebige Verknüpfungen zwischen Tabellen zu machen.

```
mysql> select * from person;
+-----+----------+---------+
| tel | name     | vorname |
+-----+----------+---------+
|   1 | Naumann  | Karl    |
|   2 | Baumann  | Beate   |
|   3 | Hartmann | Heike   |
|   4 | Naumann  | Jens    |
+-----+----------+---------+

mysql> select * from telefon;
+----+---------+
| nr | telefon |
+----+---------+
|  1 | 12345   |
|  7 | 98765   |
| 10 | 24680   |
+----+---------+
```

Die wichtigsten Joins

- Inner-Join / Equi-Join
- Left-Outer-Join
- Right-Outer-Join
- Full-Outer-Join

Inner Join

- Es werden alle Datensätze miteinander verbunden, bei denen gemeinsame Felder dieselben Werte haben
- Diese Form entspricht der klassischen Verbindung via where-Klausel

Inner Join Beispiel

```
mysql> select * from person;
+-----+----------+---------+
| tel | name     | vorname |
+-----+----------+---------+
|   1 | Naumann  | Karl    |
|   2 | Baumann  | Beate   |
|   3 | Hartmann | Heike   |
|   4 | Naumann  | Jens    |
+-----+----------+---------+

mysql> select * from telefon;
+----+---------+
| nr | telefon |
+----+---------+
|  1 | 12345   |
|  7 | 98765   |
| 10 | 24680   |
+----+---------+
```

Mit where-Klausel gelöst

```
mysql> select p.tel, name, vorname, t.nr, telefon
    -> from person p, telefon t
    -> where p.tel = t.nr;
+-----+---------+---------+----+---------+
| tel | name    | vorname | nr | telefon |
+-----+---------+---------+----+---------+
|   1 | Naumann | Karl    |  1 | 12345   |
+-----+---------+---------+----+---------+
```

Mit inner join gelöst

```
mysql> select p.tel, name, vorname, t.nr, telefon
    -> from person p
    -> inner join telefon t
    -> on p.tel = t.nr;
```

Left Outer Join

- Von der ersten Tabelle (links neben JOIN) werden alle Datensätze genommen. Von der zweiten Tabelle die jeweils dazu passenden, sofern sie existieren.
- Dies wäre eine Lösungsmöglichkeit des Beispiels mit den Kunden von vorher.

BEISPIEL LEFT OUTER JOIN

```
mysql> select * from person;
+-----+----------+---------+
| tel | name     | vorname |
+-----+----------+---------+
|   1 | Naumann  | Karl    |
|   2 | Baumann  | Beate   |
|   3 | Hartmann | Heike   |
|   4 | Naumann  | Jens    |
+-----+----------+---------+

mysql> select * from telefon;
+----+---------+
| nr | telefon |
+----+---------+
|  1 | 12345   |
|  7 | 98765   |
| 10 | 24680   |
+----+---------+
```

```
mysql> select p.tel, name, vorname, t.nr, telefon
    -> from person p
    -> left outer join telefon t
    -> on p.tel = t.nr;
+-----+----------+---------+------+---------+
| tel | name     | vorname | nr   | telefon |
+-----+----------+---------+------+---------+
|   1 | Naumann  | Karl    |    1 | 12345   |
|   2 | Baumann  | Beate   | NULL | NULL    |
|   3 | Hartmann | Heike   | NULL | NULL    |
|   4 | Naumann  | Jens    | NULL | NULL    |
+-----+----------+---------+------+---------+
```

Right Outer Join

- Analog zum left-outer-join, aber nun werden alle Datensätze aus der Tabelle rechts von JOIN gewählt. Aus der linken Tabelle werden die entsprechenden Daten ergänzt sofern vorhanden.
- In der Praxis beschränkt man sich wegen der Übersicht meist auf den left-outer-join (einfach linke und rechte Tabelle vertauschen)

Übungen

- 5, Union und inner join
  ca 20 Min
- 6, mehrere Tabellen verbinden
  ca 30 Min
- 7, komplexere Abfragen über mehrere Tabellen mit Aggregation (1)
  ca 40 Min
- 8, komplexere Abfragen über mehrere Tabellen mit Aggregation (2)
  ca 50 Min