Wirtschaftsinformatik 2

Datenbanken – Beziehungen zwischen Entitäten

Bernd Blümel, Christian Bockermann, Volker Klingspor

Hochschule Bochum

Wintersemester 2023/24

Inhalt

1 Wiederholung

2 Modell mit Beziehungen

3 Tabellen für Beziehungen

4 Suche über Beziehungen

5 Ausblick

Vom Geschäftsprozess zur Datenbank

Geschäftsprozess

"Kunde bestellt Artikel."

"Bestellungen haben mindestens 1 Artikel."

Interviews mit Fachabteilungen

Textuelle Beschreibungen, Use-Case Diagramme

Entity-Relationship Modell

Entitätstypen

Relationen

ER Diagramm
z.B. in der Sprache UML oder Chen-Notation

Datenbank

Bestellung

BestellNr	Datum	KundeNr
1	2016-05-01	1
2	2016-05-04	2
3	2016-05-09	3
4	2016-05-17	1

Artikel

ArtikelNr	Bezeichnung	Preis
1	Lasermaus	12.99
2	Tastatur	27.99
3	Festplatte	59.99
4	USB-Stick	9.99
5	WebCam	13.99

Kunde

KundeNr	Name	Vorname	PLZ	Ort
1	Maier	Ella	44801	Bochum
2	Schmidt	Jonas	44801	Bochum
3	Müller	Emma	44802	Bochum
4	Weber	Lukas	45127	Essen

SQL Schema Definition
Tabellen, Relationstabellen, Constraints/Regeln

Unglückliche Tabellenstrukturen

Kunden mit Bestellung

Name	*Adresse*	*Datum*	*Anzahl*	*Artikel*
Werner Müller	Wittener Str. 79, 44803 Bochum	2022-02-12	2	Sneaker Gazelle
Werner Müller	Wittener Str. 79, 44803 Bochum	2022-02-12	1	Sneaker Stan Smith
Thomas Walter	Lerchenweg 13b, 36272 Niederaula	2022-03-28	1	Fußballschuh King Ultimate
Ella Maier	53347 Alfter, Amselweg 28	2022-05-12	2	Fußballschuhe King Ult.
Werner Müller	Wittener Str. 79, 44789 Bochum	2022-08-01	3	Volleyballschuhe Upcourt 5
Maier, Ella	53347 Alfter, Amselweg 28	2022-09-01	1	Volleyballschuhe Upcourt

Kunden mit Bestellungen zusammen in einer Tabelle war eine schlechte Idee.

Separate Tabellen für Entitätstypen

Kunde

Tabelle Kunde

KundeNr	Nachname	Vorname	PLZ	Ort	Strasse
1	Müller	Werner	44789	Bochum	Wittener Str. 79
2	Thomas	Walter	36272	Niederaula	Lerchenweg 13b
3	Maier	Ella	53347	Alfter	Amselweg 28

Primärschlüssel

Attribut

Tupel

- Welche Entitätstypen gibt es in der Anwendung?
- Jeder Entitätstyp bekommt eigene Tabelle
- Je Tabelle eine Spalte zur eindeutigen Identifikation (Primärschlüssel)

Welche Kunden wohnen in Bochum?

Select * from Kunde where **Ort = 'Bochum'**

KundeNr	*Nachname*	*Vorname*	*PLZ*	*Ort*	*Strasse*
1	Müller	Werner	44789	Bochum	Wittener Str. 79
5	Meyer	Klaus	44801	Bochum	Wittener Str. 101
6	Maier	Heike	44789	Bochum	Wittener Str. 79

Allgemeine Form einer SELECT Abfrage:

```
Select * | Liste von Attributen | count(*)
From Tabelle
Where Bedingung
Order By Liste von Attributen (ggf. mit asc | desc)
```

Was machen wir mit den Bestellungen?

Kunde

Name
Werner Müller
Werner Müller
Thomas Walter
Ella Maier
Werner Müller
Maier, Ella

Bestellungen unserer Kunden

Datum	***Anzahl***	***Artikel***
2022-02-12	2	Sneaker Gazelle
2022-02-12	1	Sneaker Stan Smith
2022-03-28	1	Fußballschuh King Ultimate
2022-05-12	2	Fußballschuhe King Ult.
2022-08-01	3	Volleyballschuhe Upcourt 5
2022-09-01	1	Volleyballschuhe Upcourt

Beziehungen zwischen Entitätstypen

Beziehung (Relationship) zwischen Kunde und Bestellung

Beziehungen zwischen Entitätstypen

Beziehung (Relationship) zwischen Kunde und Bestellung

Kardinalitäten:

- Zu einer Bestellung gehört genau ein Kunde
- Ein Kunde kann beliebig viele Bestellungen haben

Kardinalitäten

Kardinalitäten legen für jeden Beziehungstyp fest, wie viele Entitäten eines Entitätstyps mit genau einer Entität des anderen am Beziehungstyp beteiligten Entitätstyps (und umgekehrt) in Beziehung stehen können oder müssen.

nach Wikipedia: https://de.wikipedia.org/wiki/Kardinalität_(Datenbankmodellierung)

Kardinalitäten

Kardinalitäten legen für jeden Beziehungstyp fest, wie viele Entitäten eines Entitätstyps mit genau einer Entität des anderen am Beziehungstyp beteiligten Entitätstyps (und umgekehrt) in Beziehung stehen können oder müssen.

nach Wikipedia: `https://de.wikipedia.org/wiki/Kardinalität_(Datenbankmodellierung)`

Schreibweise immer *von .. bis*

0 .. 1	höchstens ein
1 .. 1	genau ein (darf auch *1* geschrieben werden)
0 .. *	beliebig viele (darf auch * geschrieben werden)
1 .. *	mindestens ein

Bestellungen unserer Kunden

Datum	*Anzahl*	*Artikel*
2022-02-12	2	Sneaker Gazelle
2022-02-12	1	Sneaker Stan Smith
2022-03-28	1	Fußballschuh King Ultimate
2022-05-12	2	Fußballschuhe King Ult.
2022-08-01	3	Volleyballschuhe Upcourt 5
2022-09-01	1	Volleyballschuhe Upcourt

Gut so?

Bestellungen unserer Kunden

Datum	*Anzahl*	*Artikel*
2022-02-12	2	Sneaker Gazelle
2022-02-12	1	Sneaker Stan Smith
2022-03-28	1	Fußballschuh King Ultimate
2022-05-12	2	Fußballschuhe King Ult.
2022-08-01	3	Volleyballschuhe Upcourt 5
2022-09-01	1	Volleyballschuhe Upcourt

Gut so?

- Es gibt eine Bestellung mit mehreren Artikeln
- Es gibt Artikel, die mehrmals bestellt wurden

Bestellungen unserer Kunden

Datum	*Anzahl*	*Artikel*
2022-02-12	2	Sneaker Gazelle
2022-02-12	1	Sneaker Stan Smith
2022-03-28	1	Fußballschuh King Ultimate
2022-05-12	2	Fußballschuhe King Ult.
2022-08-01	3	Volleyballschuhe Upcourt 5
2022-09-01	1	Volleyballschuhe Upcourt

Gut so?

- Es gibt eine Bestellung mit mehreren Artikeln
- Es gibt Artikel, die mehrmals bestellt wurden

Die Artikeldaten gehören nicht in die Bestellung!

Tabelle Bestellung

BestellungNr	***Datum***
1	2022-02-12
2	2022-03-28
3	2022-05-12
4	2022-08-01
5	2022-09-01

- Wir betrachten zunächst nur die Daten zur eigentlichen Bestellung. Die bestellten Artikel schauen wir uns später an.

Tabelle Bestellung

BestellungNr	Datum
1	2022-02-12
2	2022-03-28
3	2022-05-12
4	2022-08-01
5	2022-09-01

- Wir betrachten zunächst nur die Daten zur eigentlichen Bestellung. Die bestellten Artikel schauen wir uns später an.
- Wie bekommt man jetzt die Bestellungen mit den Kunden zusammen?

Fremdschlüssel bei 1:n-Beziehung

Tabelle Kunde

KundeNr	*Vorname*	*Nachname*
1	Werner	Müller
2	Walter	Thomas
3	Ella	Maier
4	Ulrike	Schmidt
5	Klaus	Meyer
6	Heike	Maier

Tabelle Bestellung

BestellungNr	*Datum*
1	2022-02-12
2	2022-03-28
3	2022-05-12
4	2022-08-01
5	2022-09-01

Fremdschlüssel bei 1:n-Beziehung

Tabelle Kunde

KundeNr	Vorname	Nachname
1	Werner	Müller
2	Walter	Thomas
3	Ella	Maier
4	Ulrike	Schmidt
5	Klaus	Meyer
6	Heike	Maier

Tabelle Bestellung

BestellungNr	Datum	KundeNr
1	2022-02-12	1
2	2022-03-28	2
3	2022-05-12	3
4	2022-08-01	1
5	2022-09-01	3

Fremdschlüssel

Wir nehmen den Primärschlüssel der 1-Seite, und fügen ihn als Fremdschlüssel auf der n-Seite (*-Seite) hinzu.

Fremdschlüssel

Ein Fremdschlüssel ist ein Attribut oder eine Attributkombination einer Tabelle, welches auf einen Primärschlüssel (bzw. Schlüsselkandidaten) einer anderen oder der gleichen Tabelle verweist.

nach Wikipedia: `https://de.wikipedia.org/wiki/Schlüssel_(Datenbank)`

Anomalieproblem gelöst?

Tabelle Kunde

KundeNr	Vorname	Nachname
1	Werner	Müller
2	Walter	Thomas
3	Ella	Maier
4	Ulrike	Schmidt
5	Klaus	Meyer
6	Heike	Maier

Tabelle Bestellung

BestellungNr	Datum	KundeNr
1	2022-02-12	1
2	2022-03-28	2
3	2022-05-12	3
4	2022-08-01	1
5	2022-09-01	3

- Zu einem Kunden können mehrere Bestellungen gespeichert werden (ohne redundante Kundendaten)

Anomalieproblem gelöst?

Tabelle Kunde

KundeNr	Vorname	Nachname
1	Werner	Müller
2	Walter	Thomas
3	Ella	Maier
4	Ulrike	Schmidt
5	Klaus	Meyer
6	Heike	Maier

Tabelle Bestellung

BestellungNr	Datum	KundeNr
1	2022-02-12	1
2	2022-03-28	2
3	2022-05-12	3
4	2022-08-01	1
5	2022-09-01	3

- Zu einem Kunden können mehrere Bestellungen gespeichert werden (ohne redundante Kundendaten)
- Es können Kunden ohne Bestellungen angelegt werden

Anomalieproblem gelöst?

Tabelle Kunde

KundeNr	Vorname	Nachname
1	Werner	Müller
2	Walter	Thomas
3	Ella	Maier
4	Ulrike	Schmidt
5	Klaus	Meyer
6	Heike	Maier

Tabelle Bestellung

BestellungNr	Datum	KundeNr
1	2022-02-12	1
2	2022-03-28	2
3	2022-05-12	3
4	2022-08-01	1
5	2022-09-01	3

- Zu einem Kunden können mehrere Bestellungen gespeichert werden (ohne redundante Kundendaten)
- Es können Kunden ohne Bestellungen angelegt werden
- Es können Bestellungen gelöscht werden, ohne dass Kundendaten verloren gehen

Suche über Beziehungen

Bisher: Abfragen über einzelne Tabellen

```
SELECT * FROM Kunde
```

- Daten in unterschiedlichen Tabellen (Kunde, Bestellung) gespeichert
- Wie fragen wir Kunden und die zugehörigen Bestellungen ab?

Bisher: Abfragen über einzelne Tabellen

```
SELECT * FROM Kunde
```

- Daten in unterschiedlichen Tabellen (Kunde, Bestellung) gespeichert
- Wie fragen wir Kunden und die zugehörigen Bestellungen ab?

Zur Erinnerung:

```
SELECT Spalte(n)
  FROM tabelle(n)
  [WHERE Bedingung(en)]
```

Abfrage von mehreren Tabellen

```
SELECT * FROM Kunde, Bestellung
```

KundeNr	Name	Vorname	PLZ	Ort
1	Maier	Ella	44801	Bochum
2	Schmidt	Jonas	44801	Bochum
3	Müller	Emma	44802	Bochum
4	Weber	Lukas	45127	Essen

BestellNr	Datum	KundeNr
1	2016-05-01	1
2	2016-05-04	2
3	2016-05-09	3
4	2016-05-17	1

Abfrage von mehreren Tabellen

```
SELECT * FROM Kunde, Bestellung
```

KundeNr	Name	Vorname	PLZ	Ort
1	Maier	Ella	44801	Bochum
2	Schmidt	Jonas	44801	Bochum
3	Müller	Emma	44802	Bochum
4	Weber	Lukas	45127	Essen

BestellNr	Datum	KundeNr
1	2016-05-01	1
2	2016-05-04	2
3	2016-05-09	3
4	2016-05-17	1

KundeNr	Name	Vorname	PLZ	Ort	BestellNr	Datum	KundeNr
1	Maier	Ella	44801	Bochum	1	2016-05-01	1
1	Maier	Ella	44801	Bochum	2	2016-05-04	2
1	Maier	Ella	44801	Bochum	3	2016-05-09	3
1	Maier	Ella	44801	Bochum	4	2016-05-17	1
2	Schmidt	Jonas	44801	Bochum	1	2016-05-01	1

Tabellen als Mengen von Elementen

Kunde **K**

Bestellung **B**

Das Kreuzprodukt zweier Mengen K und B ist:

$$K \times B = \{(k, b) \mid k \in K, b \in B\}$$

Kreuzprodukt von *K* und *B*

KundeNr	Vorname	Nachname	BestellungNr	Datum	KundeNr
1	Werner	Müller	1	2022-02-12	1
1	Werner	Müller	2	2022-03-28	2
1	Werner	Müller	3	2022-05-12	3
1	Werner	Müller	4	2022-08-01	1
1	Werner	Müller	5	2022-09-01	3
2	Walter	Thomas	1	2022-02-12	1
2	Walter	Thomas	2	2022-03-28	2
2	Walter	Thomas	3	2022-05-12	3
2	Walter	Thomas	4	2022-08-01	1
2	Walter	Thomas	5	2022-09-01	3
3	Ella	Maier	1	2022-02-12	1
3	Ella	Maier	2	2022-03-28	2

Kunde **Bestellung**

Kreuzprodukt von *K* und *B*

KundeNr	Vorname	Nachname	BestellungNr	Datum	KundeNr
1	Werner	Müller	1	2022-02-12	1
1	Werner	Müller	2	2022-03-28	2
1	Werner	Müller	3	2022-05-12	3
1	Werner	Müller	4	2022-08-01	1
1	Werner	Müller	5	2022-09-01	3
2	Walter	Thomas	1	2022-02-12	1
2	Walter	Thomas	2	2022-03-28	2
2	Walter	Thomas	3	2022-05-12	3
2	Walter	Thomas	4	2022-08-01	1
2	Walter	Thomas	5	2022-09-01	3
3	Ella	Maier	1	2022-02-12	1
3	Ella	Maier	2	2022-03-28	2

Kunde **Bestellung**

Kreuzprodukt von *K* und *B*

KundeNr	Vorname	Nachname	BestellungNr	Datum	KundeNr
1	Werner	Müller	1	2022-02-12	1
1	Werner	Müller	2	2022-03-28	2
1	Werner	Müller	3	2022-05-12	3
1	Werner	Müller	4	2022-08-01	1
1	Werner	Müller	5	2022-09-01	3
2	Walter	Thomas	1	2022-02-12	1
2	Walter	Thomas	2	2022-03-28	2
2	Walter	Thomas	3	2022-05-12	3
2	Walter	Thomas	4	2022-08-01	1
2	Walter	Thomas	5	2022-09-01	3
3	Ella	Maier	1	2022-02-12	1
3	Ella	Maier	2	2022-03-28	2

Kunde **Bestellung**

Kreuzprodukt von *K* und *B*

KundeNr	Vorname	Nachname	BestellungNr	Datum	KundeNr
1	Werner	Müller	1	2022-02-12	1
1	Werner	Müller	2	2022-03-28	2
1	Werner	Müller	3	2022-05-12	3
1	Werner	Müller	4	2022-08-01	1
1	Werner	Müller	5	2022-09-01	3
2	Walter	Thomas	1	2022-02-12	1
2	Walter	Thomas	2	2022-03-28	2
2	Walter	Thomas	3	2022-05-12	3
2	Walter	Thomas	4	2022-08-01	1
2	Walter	Thomas	5	2022-09-01	3
3	Ella	Maier	1	2022-02-12	1
3	Ella	Maier	2	2022-03-28	2

Kunde **Bestellung**

Kreuzprodukt von *K* und *B*

KundeNr	Vorname	Nachname	BestellungNr	Datum	KundeNr
1	Werner	Müller	1	2022-02-12	1
1	Werner	Müller	2	2022-03-28	2
1	Werner	Müller	3	2022-05-12	3
1	Werner	Müller	4	2022-08-01	1
1	Werner	Müller	5	2022-09-01	3
2	Walter	Thomas	1	2022-02-12	1
2	Walter	Thomas	2	2022-03-28	2
2	Walter	Thomas	3	2022-05-12	3
2	Walter	Thomas	4	2022-08-01	1
2	Walter	Thomas	5	2022-09-01	3
3	Ella	Maier	1	2022-02-12	1
3	Ella	Maier	2	2022-03-28	2

Kunde **Bestellung**

Vorsicht: Ggf. Spalten mit gleichen Namen!

```
SELECT * FROM Kunde, Bestellung
```

KundeNr	Vorname	Nachname	BestellungNr	Datum	KundeNr
1	Werner	Müller	1	2022-02-12	1
1	Werner	Müller	2	2022-03-28	2
1	Werner	Müller	3	2022-05-12	3
1	Werner	Müller	4	2022-08-01	1

```
SELECT KundeNr FROM Kunde, Bestellung
```

```
ERROR:  column reference "KundeNr" is ambiguous
LINE 1: select KundeNr from Kunde, Bestellung
```

Eindeutige Spalten durch TABELLE.SPALTE

```
SELECT Kunde.KundeNr,Vorname,Nachname,Datum
  FROM Kunde, Bestellung
```

KundeId	Vorname	Nachname	Datum
1	Werner	Müller	2022-02-12
1	Werner	Müller	2022-03-28
1	Werner	Müller	2022-05-12
1	Werner	Müller	2022-08-01
1	Werner	Müller	2022-09-01
2	Walter	Thomas	2022-02-12
2	Walter	Thomas	2022-03-28
2	Walter	Thomas	2022-05-12
2	Walter	Thomas	2022-08-01
2	Walter	Thomas	2022-09-01
3	Ella	Maier	2022-02-12
3	Ella	Maier	2022-03-28
3	Ella	Maier	2022-05-12

JOIN zweier Tabellen

```
SELECT * FROM Kunde, Bestellung
   WHERE Kunde.KundeNr = Bestellung.KundeNr
```

KundeId	Vorname	Nachname	BestellungNr	Datum	KundeNr
1	Werner	Müller	1	2022-02-12	1
2	Walter	Thomas	2	2022-03-28	2
3	Ella	Maier	3	2022-05-12	3
1	Werner	Müller	4	2022-08-01	1
3	Ella	Maier	5	2022-09-01	3

JOIN zweier Tabellen

```
SELECT * FROM Kunde, Bestellung
   WHERE Kunde.KundeNr = Bestellung.KundeNr
```

Ergibt die Menge **aller** gültigen Kunde/Bestellung Paare.

JOIN zweier Tabellen

```
SELECT * FROM Kunde, Bestellung
   WHERE Kunde.KundeNr = Bestellung.KundeNr
```

Ergibt die Menge **aller** gültigen Kunde/Bestellung Paare.

Wir wollen aber die Bestellung von *Ella Maier*, also:

```
SELECT * FROM Kunde, Bestellung
   WHERE Kunde.KundeNr = Bestellung.KundeNr
     AND Kunde.Nachname = "Maier" AND Kunde.Vorname = "Ella"
```

Alternative Syntax

Vermischung von Verbindungs-Bedingung und Such-Bedingung:

```
SELECT * FROM Kunde, Bestellung
   WHERE Kunde.KundeNr = Bestellung.KundeNr
     AND Kunde.Nachname = "Maier" AND Kunde.Vorname = "Ella"
```

Alternative Syntax

Vermischung von Verbindungs-Bedingung und Such-Bedingung:

```
SELECT * FROM Kunde, Bestellung
   WHERE Kunde.KundeNr = Bestellung.KundeNr
     AND Kunde.Nachname = "Maier" AND Kunde.Vorname = "Ella"
```

Alternativ (JOIN Syntax):

```
SELECT * FROM Kunde
    JOIN Bestellung ON Kunde.KundeNr = Bestellung.KundeNr
   WHERE Kunde.Nachname = "Maier" AND Kunde.Vorname = "Ella"
```

Problem:

Kreuzprodukt erzeugt nur Paare aus existierenden Kombinationen (Kunde, Bestellung).

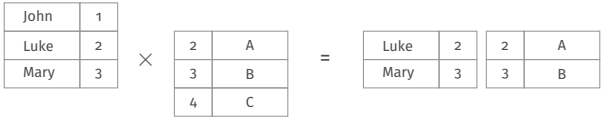
Es gibt Kunden ohne Bestellung, daher wird für dies auch kein Paar gebildet.

Mit LEFT JOIN erzwingen wir auch Paare mit fehlender Bestellung:

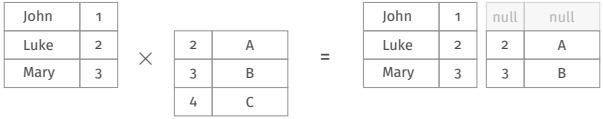
```
SELECT * FROM Kunde
    LEFT JOIN Bestellung ON Bestellung.KundeNr = Kunde.KundeNr
```

Überblick: JOIN von Tabellen

INNER JOIN

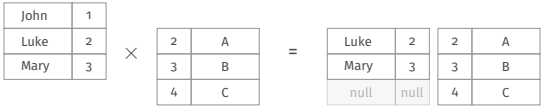


LEFT JOIN

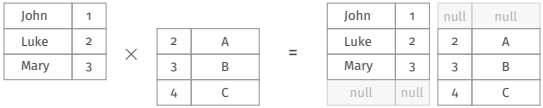


Überblick: JOIN von Tabellen

RIGHT JOIN



FULL JOIN



Welche Bestellungen hat Ella Maier aufgegeben?

Tabelle Kunde

KundeNr	Nachname	Vorname	PLZ	Ort
1	Müller	Werner	44789	Bochum
2	Thomas	Walter	36272	Niederaula
3	Maier	Ella	53347	Alfter
4	Schmidt	Ulrike	53347	Alfter

Tabelle Bestellung

BestellungNr	Datum	KundeNr
1	2022-02-12	1
2	2022-03-28	2
3	2022-05-12	3
4	2022-08-01	1
5	2022-09-01	3

```
SELECT *
   FROM Kunde, Bestellung
   WHERE Kunde.KundeNr = Bestellung.KundeNr AND
         Kunde.Vorname = 'Ella' AND Kunde.Nachname = 'Maier'
```

Wann hat Ella Maier Bestellungen aufgegeben?

```
Select Bestellung.Datum
From Kunde join Bestellung on (Kunde.KundeNr = Bestellung.KundeNr)
Where Kunde.Vorname = 'Ella' and Kunde.Nachname = 'Maier'
```

```
Select Bestellung.Datum
From Kunde, Bestellung
Where Kunde.Vorname = 'Ella' and Kunde.Nachname = 'Maier'
and Kunde.KundeNr = Bestellung.KundeNr
```

Ergebnis

Datum
2022-05-12
2022-09-01

Wann Bestellungen hat Ella Maier aufgegeben? Sortiere nach Bestelldatum!

```
Select * From Kunde join Bestellung on (Kunde.KundeNr = Bestellung.KundeNr)
Where Kunde.Vorname = 'Ella' and Kunde.Nachname = 'Maier'
order by Bestellung.Bestelldatum desc
```

```
Select * From Kunde, Bestellung
Where Kunde.Vorname = 'Ella' and Kunde.Nachname = 'Maier'
and Kunde.KundeNr = Bestellung.KundeNr
order by Bestellung.Bestelldatum desc
```

Ergebnis

KundeNr	Nachname	Vorname	PLZ	Ort	Strasse	BestellungNr	Datum	KundeNr
3	Maier	Ella	53347	Alfter	Amselweg 28	5	2022-09-01	3
3	Maier	Ella	53347	Alfter	Amselweg 28	3	2022-05-12	3

Welche Kunden habe ich und welche haben Bestellungen?

```
Select *
From Kunde left join Bestellung on (Kunde.KundeNr = Bestellung.KundeNr)
```

KundeNr	Nachname	Vorname	PLZ	Ort	Strasse	BestellungNr	Datum	KundeNr
1	Müller	Werner	44789	Bochum	Wittener Str. 79	1.0	2022-02-12	1.0
1	Müller	Werner	44789	Bochum	Wittener Str. 79	4.0	2022-08-01	1.0
2	Thomas	Walter	36272	Niederaula	Lerchenweg	2.0	2022-03-28	2.0
3	Maier	Ella	53347	Alfter	Amselweg	3.0	2022-05-12	3.0
3	Maier	Ella	53347	Alfter	Amselweg	5.0	2022-09-01	3.0
4	Schmidt	Ulrike	53347	Alfter	Höhenweg 3	NULL	NULL	NULL
5	Meyer	Klaus	44801	Bochum	Wittener Str. 79	NULL	NULL	NULL
6	Maier	Heike	44789	Bochum	Wittener Str. 79	NULL	NULL	NULL

Kunden ohne Bestellungen

Left Join über zwei Tabellen 2

- Der left join füllt den Join mit allen Datensätzen der linken Tabelle.
- Datensätze werden mit den dazugehörigen Daten der rechten Tabelle aufgefüllt.
- Datensätze, zu denen es keinen Datensatz aus der rechten Tabelle gibt, werden mit NULL aufgefüllt.
- NULL kennzeichnet, dass kein Wert gespeichert ist.

- Umgekehrt gibt es auch einen right join.

Welche Kunden haben keine Bestellungen?

```
Select *
From Kunde join Bestellung on (Kunde.KundeNr = Bestellung.KundeNr)
Where Bestellung.BestellungNr is Null
```

KundeNr	Nachname	Vorname	PLZ	Ort	Strasse	BestellungNr	Datum	KundeNr
4	Schmidt	Ulrike	53347	Alfter	Höhenweg 3	NULL	NULL	NULL
5	Meyer	Klaus	44801	Bochum	Wittener Str. 79	NULL	NULL	NULL
6	Maier	Heike	44789	Bochum	Wittener Str.	NULL	NULL	NULL

- Genauso kann mit is not Null geprüft werden, ob eine Attribut einen Wert besitzt.

Wie viele Kunden haben keine Bestellungen?

```
Select count(*)
From Kunde join Bestellung on (Kunde.KundeNr = Bestellung.KundeNr)
Where Bestellung.BestellungNr is Null
```

count(*)
3

Allgemeine Form einer SQL-Abfrage über zwei Tabellen

```
Select * | Liste von Attributen | count(*)
From Tabelle1 join Tabelle2
        on (Tabelle1.PK = Tabelle2.FK)
Where Bedingung
Order By Liste von Attributen (ggf. mit asc | desc)
```

Allgemeine Form einer SQL-Abfrage mit *distinct*

```
Select distinct * | Liste von Attributen
From Tabelle1 join Tabelle2
        on (Tabelle1.PK = Tabelle2.FK)
Where Bedingung
Order By Liste von Attributen (ggf. mit asc | desc)
```

Ausblick – Was kommt als nächstes?

- Wie bekommen wir die Artikel zu den Bestellungen?
- Wie speichern wir dies in Tabellen?
- Wie suchen wir Kunden mit ihren bestellten Artikeln?