

WIRTSCHAFTSINFORMATIK 1

DATENBANKEN – WEITERE BEZIEHUNGEN

PROF. DR. CHRISTIAN BOCKERMANN, PROF. DR. JULIAN GROTE, PROF. DR.
VOLKER KLINGSPOR

HOCHSCHULE BOCHUM

WINTERSEMESTER 2026/27

Inhalt

- 1 Wiederholung
- 2 Weitere Arten von Beziehungen
 - 1:1-Beziehungen
 - Rekursive Beziehungen
 - Beziehungen mit mehr als zwei Entitätstypen
 - Spezialisierungen
- 3 Zusammenfassung

Zunächst: **Aggregatfunktionen** in der Tabelle Artikel

ArtikelNr	Name	Preis
1	Sneaker Gazelle	109.9
2	Sneaker Stan Smith	159.9
3	Fußballschuh King Ultimate	209.95
4	Volleyballschuhe Upcourt 5	65.0
5	Volleyballschuhe Gel-Furtherup Damen	NULL
6	Laufschuh Electrify Nitro 2 Herren	69.9

Bestimme den niedrigsten, den höchsten und den durchschnittlichen Preis sowie die Summe aller Preise

```
Select min(Preis), max(Preis), avg(Preis), sum(Preis) from Artikel
```

min(Preis)	max(Preis)	avg(Preis)	sum(Preis)
65.0	209.95	122.93	614.65

ArtikelNr	Name	Preis
1	Sneaker Gazelle	109.9
2	Sneaker Stan Smith	159.9
3	Fußballschuh King Ultimate	209.95
4	Volleyballschuhe Upcourt 5	65.0
5	Volleyballschuhe Gel-Furtherup Damen	NULL
6	Laufschuh Electrify Nitro 2 Herren	69.9

Wie viele Artikel habe ich?

Wie viele unterschiedliche Preise habe die Artikel?

Wie viele Artikel haben einen Preis?

```
Select count(*), count(distinct Preis), count(Preis) from Artikel
```

count(*)	count(distinct Preis)	count(Preis)
6	5	5

Ort	count(*)
Alfter	2
Bochum	3
Niederaula	1

KundeNr	Nach-	Vorname	PLZ	Ort	Strasse
3	Maier	Ella	53347	Alfter	Amselweg 28
4	Schmidt	Ulrike	53347	Alfter	Höhenweg 3
1	Müller	Werner	44789	Bochum	Wittener Str. 79
5	Meyer	Klaus	44801	Bochum	Wittener Str. 101
6	Maier	Heike	44789	Bochum	Wittener Str. 79
2	Thomas	Walter	36272	Niederaula	Lerchenweg 13b

Diagram illustrating the grouping of data by location (Ort) using the count(*) function. The left table shows the result of the aggregation, and the right table shows the original data. Brackets indicate that the count(*) function is applied to the 'Ort' column for each group.

```
Select Ort, count(*) from Kunde group by Ort
```

Ort	count(*)
Alfter	2
Bochum	3
Niederaula	1

KundeNr	Nach-	Vorname	PLZ	Ort	Strasse
3	Maier	Ella	53347	Alfter	Amselweg 28
4	Schmidt	Ulrike	53347	Alfter	Höhenweg 3
1	Müller	Werner	44789	Bochum	Wittener Str. 79
5	Meyer	Klaus	44801	Bochum	Wittener Str. 101
6	Maier	Heike	44789	Bochum	Wittener Str. 79
2	Thomas	Walter	36272	Niederaula	Lerchenweg 13b

Diagram illustrating the grouping of data by location (Ort) using the count(*) function. The left table shows the aggregated results, and the right table shows the original data. Brackets and arrows indicate the mapping from the original data rows to the aggregated counts.

```
Select Ort, count(*) from Kunde group by Ort
```

- **group by** fasst alle Tupel mit demselben Attributwert zu einer Teilmenge zusammen.

Ort	count(*)
Alfter	2
Bochum	3
Niederaula	1

KundeNr	Nach-	Vorname	PLZ	Ort	Strasse
3	Maier	Ella	53347	Alfter	Amselweg 28
4	Schmidt	Ulrike	53347	Alfter	Höhenweg 3
1	Müller	Werner	44789	Bochum	Wittener Str. 79
5	Meyer	Klaus	44801	Bochum	Wittener Str. 101
6	Maier	Heike	44789	Bochum	Wittener Str. 79
2	Thomas	Walter	36272	Niederaula	Lerchenweg 13b

Diagram illustrating the grouping of data by location (Ort) and the application of the count(*) aggregate function. The left table shows the result of the aggregation, and the right table shows the original data. Brackets and arrows indicate the mapping from the original data rows to the aggregated counts.

```
Select Ort, count(*) from Kunde group by Ort
```

- **group by** fasst alle Tupel mit demselben Attributwert zu einer Teilmenge zusammen.
- Die Aggregatfunktion wird dann auf jede Teilmenge getrennt angewendet.

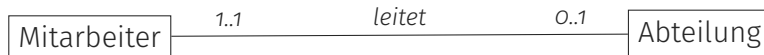
Ort	count(*)	
Alfter	2	← count(*)
Bochum	3	← count(*)
Niederaula	1	← count(*)

KundeNr	Nach-	Vorname	PLZ	Ort	Strasse
3	Maier	Ella	53347	Alfter	Amselweg 28
4	Schmidt	Ulrike	53347	Alfter	Höhenweg 3
1	Müller	Werner	44789	Bochum	Wittener Str. 79
5	Meyer	Klaus	44801	Bochum	Wittener Str. 101
6	Maier	Heike	44789	Bochum	Wittener Str. 79
2	Thomas	Walter	36272	Niederaula	Lerchenweg 13b

```
Select Ort, count(*) from Kunde group by Ort
```

- **group by** fasst alle Tupel mit demselben Attributwert zu einer Teilmenge zusammen.
- Die Aggregatfunktion wird dann auf jede Teilmenge getrennt angewendet.
- Das Attribut, nach dem gruppiert wird, kann mit ausgegeben werden.

WEITERE ARTEN VON BEZIEHUNGEN



Mitarbeiter		
MitarbeiterNr	Nach-	AbteilungNr
1	Maier	1
2	Müller	NULL
3	Schulte	2
4	Schmitt	NULL

Abteilung	
AbteilungNr	Name
1	Personal
2	Controlling

- **1:1-Beziehungen** können genauso wie 1:n-Beziehungen mit einem Fremdschlüssel aufgelöst werden.
- Beide Tabellen könnten den Fremdschlüssel der jeweils anderen Tabelle aufnehmen.
- Namen von Fremdschlüsseln dürfen geändert werden.



Mitarbeiter		
MitarbeiterNr	Nach-	AbteilungNr
1	Maier	1
2	Müller	NULL
3	Schulte	2
4	Schmitt	NULL

Abteilung	
AbteilungNr	Name
1	Personal
2	Controlling

- Wie wird geprüft, ob jede Abteilung durch einen Mitarbeiter geleitet wird?



Mitarbeiter		
MitarbeiterNr	Nach-	AbteilungNr
1	Maier	1
2	Müller	NULL
3	Schulte	2
4	Schmitt	NULL

Abteilung	
AbteilungNr	Name
1	Personal
2	Controlling

- Wie wird geprüft, ob jede Abteilung durch einen Mitarbeiter geleitet wird?
- Ist die Verwendung einer nur teilweise gefüllten Spalte sinnvoll?



Mitarbeiter	
MitarbeiterNr	Nach-
1	Maier
2	Müller
3	Schulte
4	Schmitt

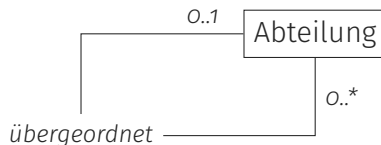
Abteilung		
AbteilungNr	Name	LeitungNr
1	Personal	1
2	Controlling	3

- Jede Abteilung wird geleitet, wenn keine NULL-Werte auftreten.

Welcher Mitarbeiter leitetet welche Abteilung?

```
Select Mitarbeiter.Nachname, Abteilung.Name  
from Mitarbeiter  
join Abteilung on (Mitarbeiter.MitarbeiterNr = Abteilung.LeitungNr)
```

Nach-	Name
Maier	Personal
Schulte	Controlling



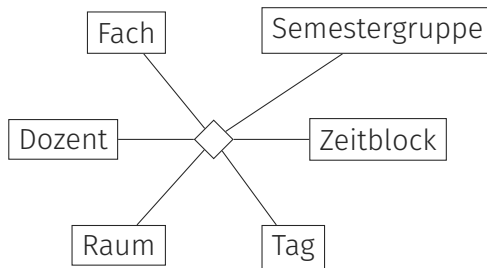
Abteilung		
AbteilungNr	Name	UebergeordnetNr
1	Personal	NULL
2	Controlling	NULL
3	Internes Personal	1
4	Externes Personal	1

- **Rekursive** Beziehungen werden genauso behandelt wie *normale* Beziehungen.
- Hier wird die Regel für 1:n-Beziehungen angewendet.
- Der Fremdschlüssel muss dabei umbenannt werden.

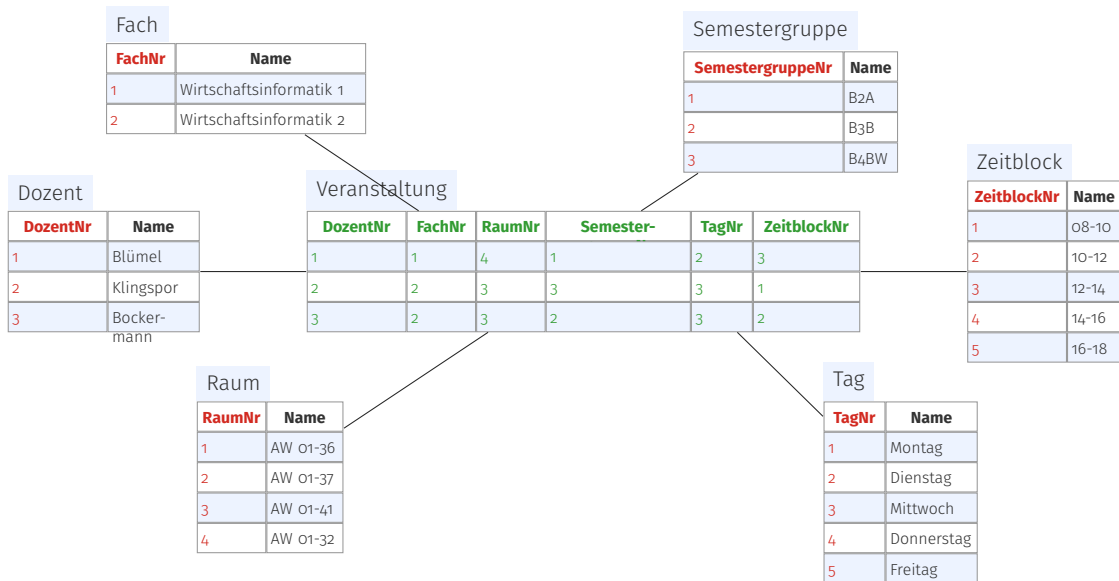
Welcher Abteilungen unterstehenden der Abteilung Personal?

```
Select Abteilung.Name, Abteilung2.Name
from Abteilung
join Abteilung as Abteilung2 on (Abteilung.AbteilungNr = Abteilung2.
                                UebergeordnetNr)
where Abteilung.Name = "Personal"
```

Name	Name
Personal	Internes Personal
Personal	Externes Personal



- Die Anzahl der Entitäten, die an einer Beziehung teilnehmen, nennt man **Grad der Beziehung**
- Bei einem Grad größer als 2 wird eine Raute gezeichnet.



- Bei der Auflösung einer Beziehung zwischen mehr als 2 Entitätstypen:
 - Erstellen Sie eine weitere Verbindungstabelle
 - Die Tabelle enthält die Primärschlüssel aller beteiligten Tabellen als Fremdschlüssel
 - Der Primärschlüssel setzt sich aus diesen Fremdschlüsseln zusammen
 - Die Tabelle kann weitere Attribute enthalten
 - Attribute, die nicht vom Primärschlüssel abhängig sind, müssen in den Primärschlüssel aufgenommen werden
- Das Vorgehen ist analog zu dem Vorgehen bei n:m-Beziehungen

Welche Dozenten lehren das Fach Wirtschaftsinformatik 2?

```
Select * from Fach
join Veranstaltung on (Fach.FachNr = Veranstaltung.FachNr)
join Dozent on (Veranstaltung.DozentNr = Dozent.DozentNr)
where Fach.Name='Wirtschaftsinformatik 2'
```

FachNr	Name	DozentNr	FachNr	RaumNr	Semester-	TagNr	ZeitblockNr	DozentNr	Name
2	Wirtschaftsinformatik 2	2	2	3	3	3	1	2	Klingspor
2	Wirtschaftsinformatik 2	3	2	3	2	3	2	3	Bocker- mann

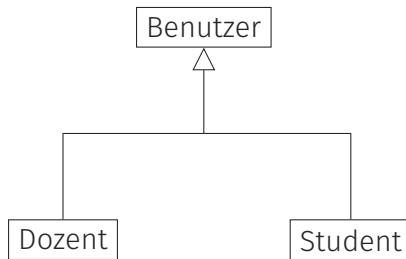
- Sie suchen genauso über die Verbindungstabelle wie bei n:m-Beziehungen

Welche Dozenten lehren das Fach Wirtschaftsinformatik 2?

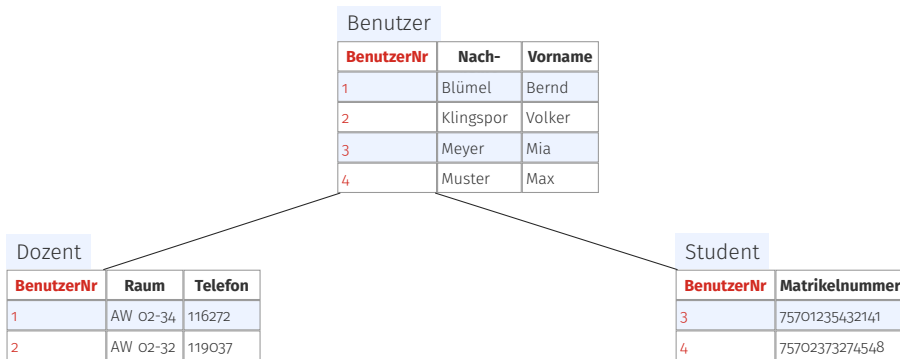
```
Select Fach.Name, Dozent.Name from Fach
join Veranstaltung on (Fach.FachNr = Veranstaltung.FachNr)
join Dozent on (Veranstaltung.DozentNr = Dozent.DozentNr)
where Fach.Name='Wirtschaftsinformatik 2'
```

Name	Name
Wirtschaftsinformatik 2	Klingspor
Wirtschaftsinformatik 2	Bocker- mann

- Sie können weiterhin alle gelernten SQL-Konstrukte verwenden



- Jede:r Dozent:in ist gleichzeitig auch ein:e Benutzer:in
- Jede:r Studierende ist gleichzeitig auch ein:e Benutzer:in
- Je nach Sichtweise reden wir von **Generalisierung** oder **Spezialisierung**
- Wegen der o.g. Eigenschaften reden wir auch von **is-a**-Beziehungen



- Sie erstellen für jeden Entitätstyp eine Tabelle.
- Alle beteiligten Tabellen erhalten als Primärschlüssel den Primärschlüssel der generellen Entität.
- Eigenschaften, die alle Entitäten besitzen, erscheinen in der Tabelle der generellen Entität.
- Eigenschaften, die die Spezialisierungen kennzeichnen, erscheinen in der Tabelle der jeweiligen Entität.

Welche Matrikelnummer hat die Studentin Mia Meyer?

```
Select * from Benutzer  
join Student on (Benutzer.BenutzerNr = Student.BenutzerNr)  
where Benutzer.Nachname='Meyer' and Benutzer.Vorname='Mia'
```

BenutzerNr	Nach-	Vorname	BenutzerNr	Matrikelnummer
3	Meyer	Mia	3	75701235432141

Welche Matrikelnummer hat die Studentin Mia Meyer?

```
Select * from Benutzer  
join Student on (Benutzer.BenutzerNr = Student.BenutzerNr)  
where Benutzer.Nachname='Meyer' and Benutzer.Vorname='Mia'
```

BenutzerNr	Nach-	Vorname	BenutzerNr	Matrikelnummer
3	Meyer	Mia	3	75701235432141

- Die Suche erfolgt über Gleichsetzung der gemeinsamen Primärschlüssel

Welche Matrikelnummer hat die Studentin Mia Meyer?

```
Select * from Benutzer
join Student on (Benutzer.BenutzerNr = Student.BenutzerNr)
where Benutzer.Nachname='Meyer' and Benutzer.Vorname='Mia'
```

BenutzerNr	Nach-	Vorname	BenutzerNr	Matrikelnummer
3	Meyer	Mia	3	75701235432141

- Die Suche erfolgt über Gleichsetzung der gemeinsamen Primärschlüssel

```
Select Student.Matrikelnummer from Benutzer
join Student on (Benutzer.BenutzerNr = Student.BenutzerNr)
where Benutzer.Nachname='Meyer' and Benutzer.Vorname='Mia'
```

Matrikelnummer
75701235432141

ZUSAMMENFASSUNG

- 1:1-Beziehungen zwischen Entitäten
- Rekursive Beziehungen
- Beziehungen zwischen mehr als 2 Entitäten
- Spezialisierungen