

WIRTSCHAFTSINFORMATIK 1

DATENBANKEN – AGGREGATFUNKTIONEN UND GRUPPIERUNG

PROF. DR. BERND BLÜMEL, PROF. DR. CHRISTIAN BOCKERMANN, PROF.
DR. VOLKER KLINGSPOR

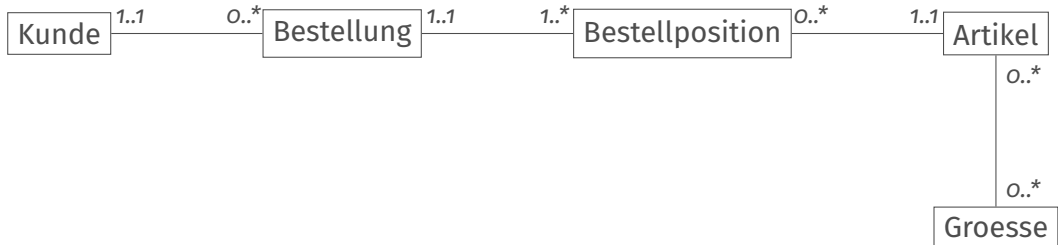
HOCHSCHULE BOCHUM

WINTERSEMESTER 2024/2025

Inhalt

- 1 Wiederholung
- 2 Betrachtungen zum Modell
- 3 Aggregatfunktionen in SQL
- 4 Gruppierung

Modell aus der Vorlesung



Modell aus der Übung





ArtikelNr	Name	Preis
1	Sneaker...	109.9
2	Sneaker...	159.9
3	Fußball...	209.95
4	Volleyb...	65.0
5	Volleyb...	NULL
6	Laufsch...	69.9

ArtikelNr	GroesseNr	Anzahl	GroesseNr	EU	US	UK	Laenge
1	1	12		35	2,5	3,5	21,8
1	2	3	2	36	3	5	22,5
1	3	0		37	4	6	23,2
2	2	5	4	38	4,5	6,5	23,8
2	3	12		39	5,5	7,5	24,5
3	5	4	6	40	6,5	8,5	25,2

GroesseNr	EU	US	UK	Laenge
1	35	2,5	3,5	21,8
2	36	3	5	22,5
3	37	4	6	23,2
4	38	4,5	6,5	23,8
5	39	5,5	7,5	24,5
6	40	6,5	8,5	25,2

- Wir erstellen eine **(Verbindungstabelle)**.
- Die Tabelle enthält die Primärschlüssel der Grundtabellen als **Fremdschlüssel**.



Artikel		
ArtikelNr	Name	Preis
1	Sneaker...	109.9
2	Sneaker...	159.9
3	Fußball...	209.95
4	Volleyb...	65.0
5	Volleyb...	NULL
6	Laufsch...	69.9

ArtikelGroesse		Groesse					
ArtikelNr	GroesseNr	Anzahl	GroesseNr	EU	US	UK	Laenge
1	1	12		35	2,5	3,5	21,8
1	2	3	2	36	3	5	22,5
1	3	0		37	4	6	23,2
2	2	5	4	38	4,5	6,5	23,8
2	3	12		39	5,5	7,5	24,5
3	5	4	6	40	6,5	8,5	25,2

- Wir erstellen eine (Verbindungstabelle).
- Die Tabelle enthält die Primärschlüssel der Grundtabellen als Fremdschlüssel.
- Diese Fremdschlüssel sind gleichzeitig der zusammengesetzte Primärschlüssel

Artikel

ArtikelNr	Name	Preis
1	Sneaker...	109.9
2	Sneaker...	159.9
3	Fußball...	209.95
4	Volleyb...	65.0
5	Volleyb...	NULL
6	Laufsch...	69.9

ArtikelGroesse

ArtikelNr	GroesseNr	Anzahl	GroesseNr	EU	US	UK	Laenge
1	1	12		35	2,5	3,5	21,8
2	2	5	2	36	3	5	22,5
1	2	3		37	4	6	23,2
1	3	0	4	38	4,5	6,5	23,8
2	3	12		39	5,5	7,5	24,5
3	5	4	6	40	6,5	8,5	25,2

Groesse

```
Select * from Artikel
join ArtikelGroesse on (Artikel.ArtikelNr = ArtikelGroesse.ArtikelNr)
join Groesse on (ArtikelGroesse.GroesseNr = Groesse.GroesseNr)
```

ArtikelNr	Name	Preis	ArtikelNr	GroesseNr	Anzahl	GroesseNr	EU	US	UK
1	Sneaker Gazelle	109.9	1	1	12	1	35	2,5	3,5
2	Sneaker Stan Smith	159.9	2	2	5	2	36	3	5
1	Sneaker Gazelle	109.9	1	2	3	2	36	3	5
1	Sneaker Gazelle	109.9	1	3	0	3	37	4	6
2	Sneaker Stan Smith	159.9	2	3	12	3	37	4	6

Was ist der Nettopreis der Artikel?

```
Select Name, Preis, Preis / 1.19 as Nettopreis from Artikel
```

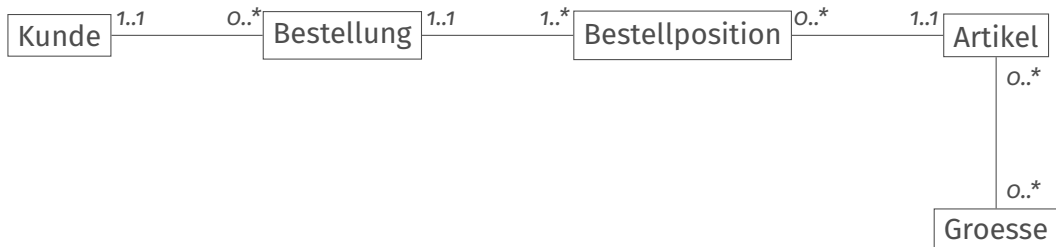
Name	Preis	Nettopreis
Sneaker Gazelle	109.9	92.3529411764706
Sneaker Stan Smith	159.9	134.36974789915968
Fußballschuh King Ultimate	209.95	176.42857142857142
Volleyballschuhe Upcourt 5	65.0	54.6218487394958
Volleyballschuhe Gel-Furtherup Damen	NULL	NULL
Laufschuh Electrify Nitro 2 Herren	69.9	58.73949579831933

Wie ist der Gesamtpreis jedes Artikels in den Bestellungen?

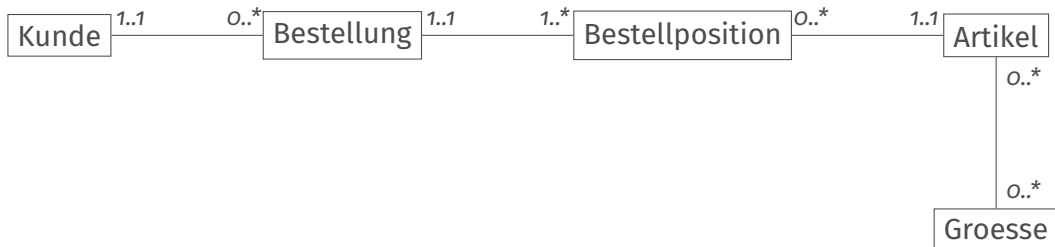
```
Select *, Bestellposition.Anzahl * Artikel.Preis As Gesamtpreis  
from Bestellposition  
join Artikel on (Bestellposition.ArtikelNr = Artikel.ArtikelNr)
```

BestellpositionNr	BestellungNr	ArtikelNr	Anzahl	ArtikelNr	Name	Preis	
1	1	1	2	1	Sneaker Gazelle	109.9	21
2	1	2	1	2	Sneaker Stan Smith	159.9	15
3	2	3	1	3	Fußballschuh King Ultimate	209.95	20
4	3	3	2	3	Fußballschuh King Ultimate	209.95	41
5	4	4	3	4	Volleyballschuhe Upcourt 5	65.0	19
6	5	4	1	4	Volleyballschuhe Upcourt 5	65.0	65

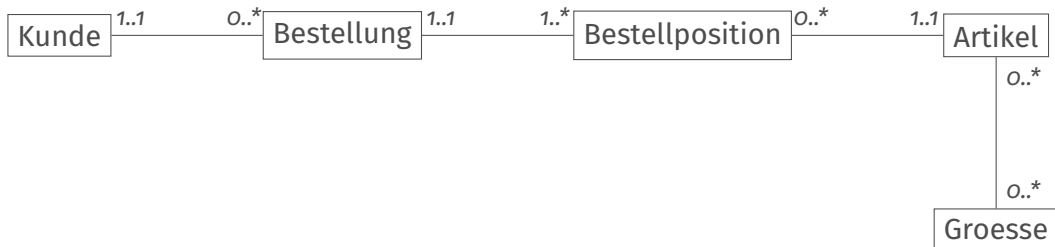
Betrachtungen zum Modell



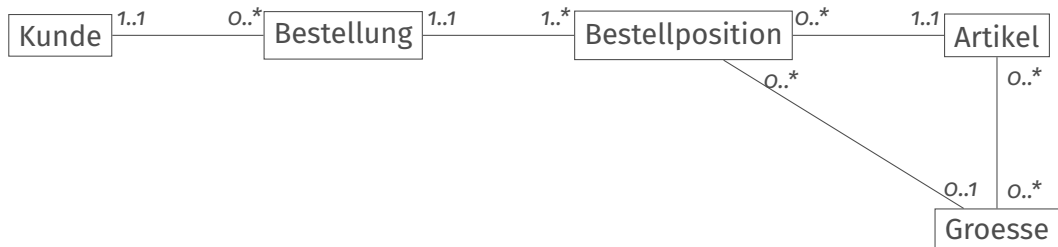
- Wird die Größe der bestellten Artikel gespeichert?



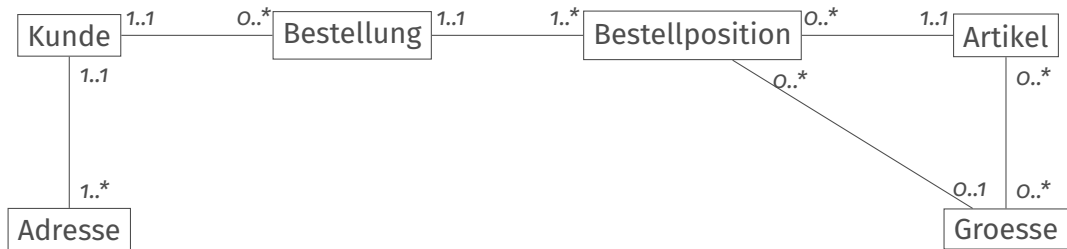
- Wird die Größe der bestellten Artikel gespeichert?
- Was passiert mit den Bestellungen, wenn sich Artikelpreise ändern?



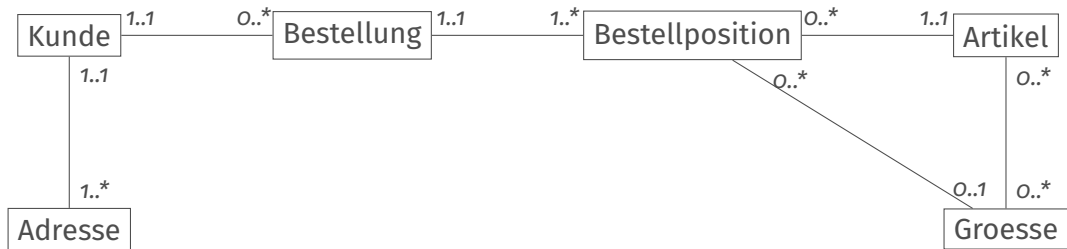
- Wird die Größe der bestellten Artikel gespeichert?
- Was passiert mit den Bestellungen, wenn sich Artikelpreise ändern?
- Haben Kunden nur eine Adresse und was passiert mit den Bestellungen, wenn Kunden umziehen?



- Speicherung der Größe einer Bestellposition als neue Beziehung



- Speicherung der Größe einer Bestellposition als neue Beziehung
- Speichern der Adresse als eigene Entität mit Beziehung zum Kunden



- Speicherung der Größe einer Bestellposition als neue Beziehung
- Speichern der Adresse als eigene Entität mit Beziehung zum Kunden
- Speichern aller zum Bestellzeitpunkts gültigen Daten (z.B. Preis und Liefer-/Rechnungsadresse) in der Bestellung (im Modell nicht sichtbar)

Aggregatfunktionen in SQL

Was können wir mit SQL bisher?

- Suche über verschiedene Tabellen
- Berechnung neuer Attribute basierend auf den Wert der aktuelle Zeile



Sneaker Shop

Hochschule Bochum
Bochum University
of Applied Sciences



Warenkorb: 3 Artikel, 672,00 €

Warenkorb

Warenkorb

Position	Artikel	Anzahl	Beschreibung	Einzelpreis	Preis	
1	1	2	Nike SB Dunk Low April Skateboards	186,00 €	372,00 €	X
2	2	1	Pharrell x NMD_S1 Mahbs 'Earth Strata'	300,00 €	300,00 €	X
				Gesamt:	672,00 €	

Zur Bestellung

Shop-Daten

shop.hsbo.de
Version: 0.0.1



Sneaker Shop

Hochschule Bochum
Bochum University
of Applied Sciences



Warenkorb: 3 Artikel, 672,00 €

Warenkorb

Warenkorb

Position	Artikel	Anzahl	Beschreibung	Einzelpreis	Preis	
1	1	2	Nike SB Dunk Low April Skateboards	186,00 €	372,00 €	X
2	2	1	Pharrell x NMD_S1 Mahbs 'Earth Strata'	300,00 €	300,00 €	X
				Gesamt:	672,00 €	

Zur Bestellung

Select Anzahl * Preis

Shop-Daten

shop.hsbo.de
Version: 0.0.1



Sneaker Shop

Hochschule Bochum
Bochum University
of Applied Sciences



Warenkorb: 3 Artikel, 672,00 €

Warenkorb

Warenkorb

Position	Artikel	Anzahl	Beschreibung	Einzelpreis	Preis	
1	1	2	Nike SB Dunk Low April Skateboards	186,00 €	372,00 €	X
2	2	1	Pharrell x NMD_S1 Mahbs 'Earth Strata'	300,00 €	300,00 €	X
				Gesamt:	672,00 €	

Zur Bestellung

Select ?

Shop-Daten

shop.hsbo.de
Version: 0.0.1

Es fehlt:

- Aggregation von Daten mehrerer Zeilen
- Beispiel: Berechne den Gesamtpreis einer Bestellung!

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	122.92999999999999	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

Wie ist der Gesamtpreis der Bestellung Nr. 1?

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

BestellPosition

BestellpositionNr	BestellungNr	ArtikelNr	Anzahl
1	1	1	2
2	1	2	1
3	2	3	1
4	3	3	2
5	4	4	3
6	5	4	1

Wie ist der Gesamtpreis der Bestellung Nr. 1?

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

BestellPosition

BestellpositionNr	BestellungNr	ArtikelNr	Anzahl
1	1	1	2
2	1	2	1
3	2	3	1
4	3	3	2
5	4	4	3
6	5	4	1

Die Tabelle Bestellung wird nicht benötigt!

Wie ist der Gesamtpreis der Bestellung Nr. 1?

BestellPosition

BestellpositionNr	BestellungNr	ArtikelNr	Anzahl
1	1	1	2
2	1	2	1
3	2	3	1
4	3	3	2
5	4	4	3
6	5	4	1

Zunächst: Welche Artikel sind in der Bestellung mit der Nr. 1?

```
Select * from  
BestellPosition join Artikel on (BestellPosition.ArtikelNr = Artikel.ArtikelNr)  
where BestellPosition.BestellungNr = 1
```

BestellpositionNr	BestellungNr	ArtikelNr	Anzahl	ArtikelNr	Name	Preis
1	1	1	2	1	Sneaker Gazelle	109.9
2	1	2	1	2	Sneaker Stan Smith	159.9

```
Select * from  
BestellPosition join Artikel on (BestellPosition.ArtikelNr = Artikel.ArtikelNr)  
where BestellPosition.BestellungNr = 1
```

BestellpositionNr	BestellungNr	ArtikelNr	Anzahl	ArtikelNr	Name	Preis
1	1	1	2	1	Sneaker Gazelle	109.9
2	1	2	1	2	Sneaker Stan Smith	159.9

Wie ist der Gesamtpreis der Bestellung Nr. 1?

(Anzahl * Preis) berechnen und das Produkt aufsummieren

```
Select * from  
BestellPosition join Artikel on (BestellPosition.ArtikelNr = Artikel.ArtikelNr)  
where BestellPosition.BestellungNr = 1
```

BestellpositionNr	BestellungNr	ArtikelNr	Anzahl	ArtikelNr	Name	Preis
1	1	1	2	1	Sneaker Gazelle	109.9
2	1	2	1	2	Sneaker Stan Smith	159.9

Wie ist der Gesamtpreis der Bestellung Nr. 1?

(Anzahl * Preis) berechnen und das Produkt aufsummieren

```
Select sum(BestellPosition.Anzahl * Artikel.Preis)  
from BestellPosition join Artikel on (BestellPosition.ArtikelNr = Artikel.ArtikelNr)  
where BestellPosition.BestellungNr = 1
```

sum(BestellPosition.Anzahl * Artikel.Preis)
379.70000000000005

Aggregatfunktionen

- berechnen einen Wert über alle gefundenen Datensätze/Tupel
- liefern als Ergebnis nur ein Tupel zurück
- können die Anzahl, das Minimum und Maximum, den Durchschnitt und die Summe einer Spalte (oder eines mathematischen Ausdrucks) berechnen

Gruppierung

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
4	Schmidt	Ulrike	53347	Alfter	Höhenweg 3
5	Meyer	Klaus	44801	Bochum	Wittener Str. 101
6	Maier	Heike	44789	Bochum	Wittener Str. 79

- Wie können wir die Anzahl der Kunden in den jeweiligen Orten berechnen?

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
4	Schmidt	Ulrike	53347	Alfter	Höhenweg 3
5	Meyer	Klaus	44801	Bochum	Wittener Str. 101
6	Maier	Heike	44789	Bochum	Wittener Str. 79

- Wie können wir die Anzahl der Kunden in den jeweiligen Orten berechnen?

Select count(*) from Kunde
where Ort = 'Bochum'

count(*)
3

Select count(*) from Kunde
where Ort = 'Niederaula'

count(*)
1

Select count(*) from Kunde
where Ort = 'Alfter'

count(*)
2

...

KundeNr	Nachname	Vorname	PLZ	Ort	
3	Maier	Ella	53347	Alfter	Amselweg
4	Schmidt	Ulrike	53347	Alfter	Höhenweg
1	Müller	Werner	44789	Bochum	Wittener S
5	Meyer	Klaus	44801	Bochum	Wittener S
6	Maier	Heike	44789	Bochum	Wittener S
2	Thomas	Walter	36272	Niederaula	Lerchenwe

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

	KundeNr	Nachname	Vorname	PLZ	Ort	
{	3	Maier	Ella	53347	Alfter	Amselweg
	4	Schmidt	Ulrike	53347	Alfter	Höhenweg
{	1	Müller	Werner	44789	Bochum	Wittener S
	5	Meyer	Klaus	44801	Bochum	Wittener S
{	6	Maier	Heike	44789	Bochum	Wittener S
	2	Thomas	Walter	36272	Niederaula	Lerchenwe

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

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

	KundeNr	Nachname	Vorname	PLZ	Ort	
count(*) {	3	Maier	Ella	53347	Alfter	Amselweg
	4	Schmidt	Ulrike	53347	Alfter	Höhenweg
count(*) {	1	Müller	Werner	44789	Bochum	Wittener S
	5	Meyer	Klaus	44801	Bochum	Wittener S
count(*) {	6	Maier	Heike	44789	Bochum	Wittener S
	2	Thomas	Walter	36272	Niederaula	Lerchenwe

```
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
Bochum	3
Niederaula	1

KundeNr	Nachname	Vorname	PLZ	Ort	
3	Maier	Ella	53347	Alfter	Amselweg
4	Schmidt	Ulrike	53347	Alfter	Höhenweg
1	Müller	Werner	44789	Bochum	Wittener S
5	Meyer	Klaus	44801	Bochum	Wittener S
6	Maier	Heike	44789	Bochum	Wittener S
2	Thomas	Walter	36272	Niederaula	Lerchenwe

Diagram illustrating the grouping process:

- count(*) { 3, 4 } → Alfter (2)
- count(*) { 1, 5, 6 } → Bochum (3)
- count(*) { 2 } → Niederaula (1)

```
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.

Ort	count(*)
Alfter	2
Bochum	3

Diagram illustrating the grouping of data by location (Ort) and the resulting count of records (count(*)).

The data is grouped by Ort, resulting in two groups: Alfter (count 2) and Bochum (count 3). The count(*) function is applied to each group. A red 'X' indicates that the count(*) function is not applied to the individual records within each group.

KundeNr	Nachname	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. 10
6	Maier	Heike	44789	Bochum	Wittener Str. 79
2	Thomas	Walter	36272	Niederaula	Lerchenweg 13b

```
Select Ort, count(*) from Kunde group by Ort having count(*) >= 2
```

Ort	count(*)
Alfter	2
Bochum	3

count(*) {

count(*) {

X

count(*) {

KundeNr	Nachname	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. 10
6	Maier	Heike	44789	Bochum	Wittener Str. 79
2	Thomas	Walter	36272	Niederaula	Lerchenweg 13b

```
Select Ort, count(*) from Kunde group by Ort having count(*) >= 2
```

- **having** filtert *nach* der Aggregation alle Ergebnistupel heraus, die nicht der Bedingung entsprechend.

Mit dem erlernten SQL-Befehlen können wir den Umsatz des Unternehmens aus verschiedenen Perspektiven betrachten.

- Umsatz der verschiedenen Bestellungen
- Umsatz der verschiedenen Kunden
- Umsatz der verschiedenen Artikel
- Umsatz in den verschiedenen Orten
- Umsatz in den verschiedenen Monaten

Mit dem erlernten SQL-Befehlen können wir den Umsatz des Unternehmens aus verschiedenen Perspektiven betrachten.

- Umsatz der verschiedenen Bestellungen
 - Umsatz der verschiedenen Kunden
 - Umsatz der verschiedenen Artikel
 - Umsatz in den verschiedenen Orten
 - Umsatz in den verschiedenen Monaten
-
- Die Berechnung des Umsatzes ist dabei immer gleich
 - Die Spalte, über die gruppiert wird, verändert sich

Bestellungen mit Artikeln

BestellpositionNr	BestellungNr	ArtikelNr	Anzahl	ArtikelNr	Name	Preis
1	1	1	2	1	Sneaker Gazelle	109.9
2	1	2	1	2	Sneaker Stan Smith	159.9
3	2	3	1	3	Fußballschuh King Ultimate	209.95
4	3	3	2	3	Fußballschuh King Ultimate	209.95
5	4	4	3	4	Volleyballschuhe Upcourt 5	65.0
6	5	4	1	4	Volleyballschuhe Upcourt 5	65.0

```
Select BestellPosition.BestellungNr, sum(BestellPosition.Anzahl * Artikel.Preis)
from BestellPosition join Artikel on (BestellPosition.ArtikelNr = Artikel.ArtikelNr)
group by BestellPosition.BestellungNr
```

BestellungNr	sum(BestellPosition.Anzahl * Artikel.Preis)
1	379.700000000000005
2	209.95
3	419.9
4	195.0
5	65.0

Kunden mit ihren Bestellungen und den dazugehörigen Artikeln

KundeNr	Nachname	Vorname	BestellungNr	Datum	KundeNr	BestellpositionNr	BestellungNr	
1	Müller	Werner	1	2022-02-12	1	1	1	1
1	Müller	Werner	1	2022-02-12	1	2	1	2
2	Thomas	Walter	2	2022-03-28	2	3	2	3
3	Maier	Ella	3	2022-05-12	3	4	3	3
1	Müller	Werner	4	2022-08-01	1	5	4	4
3	Maier	Ella	5	2022-09-01	3	6	5	4

```

Select Kunde.KundeNr, Kunde.Nachname, sum(BestellPosition.Anzahl * Artikel.Preis)
from Kunde join Bestellung on (Kunde.KundeNr = Bestellung.KundeNr)
join BestellPosition on (Bestellung.BestellungNr = BestellPosition.BestellungNr)
join Artikel on (BestellPosition.ArtikelNr = Artikel.ArtikelNr)
group by Kunde.KundeNr
    
```

KundeNr	Nachname	sum(BestellPosition.Anzahl * Artikel.Preis)
1	Müller	574.7
2	Thomas	209.95
3	Maier	484.9

Kunden mit ihren Bestellungen und den dazugehörigen Artikeln

KundeNr	Nachname	Vorname	BestellungNr	Datum	KundeNr	BestellpositionNr	BestellungNr	
1	Müller	Werner	1	2022-02-12	1	1	1	1
1	Müller	Werner	1	2022-02-12	1	2	1	2
2	Thomas	Walter	2	2022-03-28	2	3	2	3
3	Maier	Ella	3	2022-05-12	3	4	3	3
1	Müller	Werner	4	2022-08-01	1	5	4	4
3	Maier	Ella	5	2022-09-01	3	6	5	4

```

Select Kunde.Ort, sum(BestellPosition.Anzahl * Artikel.Preis)
from Kunde join Bestellung on (Kunde.KundeNr = Bestellung.KundeNr)
join BestellPosition on (Bestellung.BestellungNr = BestellPosition.BestellungNr)
join Artikel on (BestellPosition.ArtikelNr = Artikel.ArtikelNr)
group by Kunde.Ort
    
```

Ort	sum(BestellPosition.Anzahl * Artikel.Preis)
Alfter	484.9
Bochum	574.7
Niederaula	209.95

Kunden mit ihren Bestellungen und den dazugehörigen Artikeln

KundeNr	Nachname	Vorname	BestellungNr	Datum	KundeNr	BestellpositionNr	BestellungNr	
1	Müller	Werner	1	2022-02-12	1	1	1	1
1	Müller	Werner	1	2022-02-12	1	2	1	2
2	Thomas	Walter	2	2022-03-28	2	3	2	3
3	Maier	Ella	3	2022-05-12	3	4	3	3
1	Müller	Werner	4	2022-08-01	1	5	4	4
3	Maier	Ella	5	2022-09-01	3	6	5	4

```

Select Month(Bestellung.Datum) as Monat, sum(BestellPosition.Anzahl * Artikel.Preis)
from Kunde join Bestellung on (Kunde.KundeNr = Bestellung.KundeNr)
join BestellPosition on (Bestellung.BestellungNr = BestellPosition.BestellungNr)
join Artikel on (BestellPosition.ArtikelNr = Artikel.ArtikelNr)
group by Monat
    
```

Monat	sum(BestellPosition.Anzahl * Artikel.Preis)
02	379.70000000000005
03	209.95
05	419.9
08	195.0
09	65.0