

# GRUNDLAGEN WIRTSCHAFTSIN- FORMATIK

DATENBANKEN: SQL

PROF. DR. CHRISTIAN BOCKERMANN

HOCHSCHULE BOCHUM

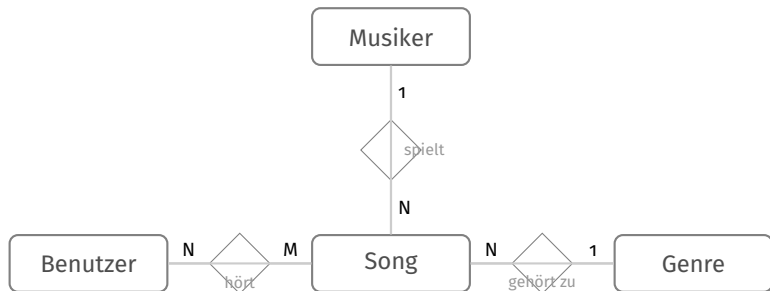
SOMMERSEMESTER 2024

## Inhalt

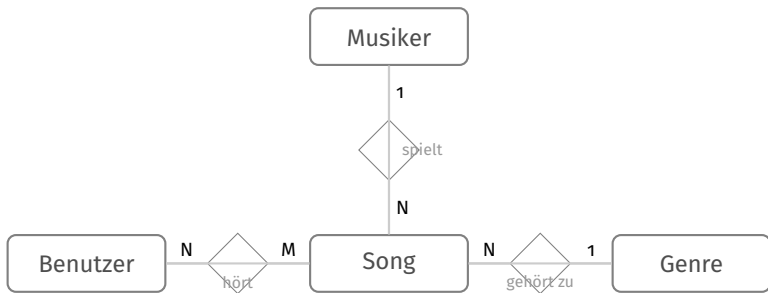
### 1 Abfragen über mehrere Tabellen

# Abfragen über mehrere Tabellen

## Zur Erinnerung: Datenbank eines Streaming-Dienstes



## Zur Erinnerung: Datenbank eines Streaming-Dienstes



**Welche Songs gehören zum Genre *Rock* ?**

## Frage: Welche Songs gehören zum Genre Rock ?

id	titel	genre	musiker
1	Supersonic Speed	3	1
2	Enter Sandman	6	2
3	Layla	5	3
4	Ayo Technology	1	4
5	Afterglow	1	5
6	Wer wenn nicht wir	1	7
7	Blinding Lights	1	6
8	Bad Habits	1	5
9	Neuanfang	1	8

id	name
1	Pop
2	Klassik
3	Rock
4	Electronic
5	Blues
6	Heavy Metal

## Frage: Welche Songs gehören zum Genre Rock ?

id	titel	genre	musiker
1	Supersonic Speed	3	1
2	Enter Sandman	6	2
3	Layla	5	3
4	Ayo Technology	1	4
5	Afterglow	1	5
6	Wer wenn nicht wir	1	7
7	Blinding Lights	1	6
8	Bad Habits	1	5
9	Neuanfang	1	8

id	name
1	Pop
2	Klassik
3	Rock
4	Electronic
5	Blues
6	Heavy Metal

```
SELECT id FROM Genre WHERE name = 'Rock'
```

id
3

## Frage: Welche Songs gehören zum Genre Rock ?

```
SELECT * FROM Songs WHERE genre = 3
```

id	titel	genre	musiker
1	Supersonic Speed	3	1



## Frage: Welche Songs gehören zum Genre Rock ?

```
SELECT * FROM Songs WHERE genre = 3
```

id	titel	genre	musiker
1	Supersonic Speed	3	1

Oder kombiniert:

```
SELECT * FROM Song  
WHERE id IN  
(SELECT id FROM Genre WHERE name = 'Rock')
```

## Frage: Welche Songs gehören zum Genre Rock ?

```
SELECT * FROM Songs WHERE genre = 3
```

id	titel	genre	musiker
1	Supersonic Speed	3	1

Oder kombiniert:

```
SELECT * FROM Song  
WHERE id IN  
(SELECT id FROM Genre WHERE name = 'Rock')
```

**Brauchen wir dazu wirklich zwei Abfragen?**

## SELECT Abfrage über zwei Tabellen

```
SELECT * FROM Song, Genre
```

id	titel	genre	musiker	id	name
1	Supersonic Speed	3	1	1	Pop
1	Supersonic Speed	3	1	2	Klassik
1	Supersonic Speed	3	1	3	Rock
1	Supersonic Speed	3	1	4	Electronic
1	Supersonic Speed	3	1	5	Blues
1	Supersonic Speed	3	1	6	Heavy Metal
2	Enter Sandman	6	2	1	Pop
2	Enter Sandman	6	2	2	Klassik
2	Enter Sandman	6	2	3	Rock
⋮	⋮	⋮	⋮	⋮	⋮

## SELECT Abfrage über zwei Tabellen

```
SELECT * FROM Song, Genre
```

id	titel	genre	musiker	id	name
1	Supersonic Speed	3	1	1	Pop
1	Supersonic Speed	3	1	2	Klassik
1	Supersonic Speed	3	1	3	Rock
1	Supersonic Speed	3	1	4	Electronic
1	Supersonic Speed	3	1	5	Blues
1	Supersonic Speed	3	1	6	Heavy Metal
2	Enter Sandman	6	2	1	Pop
2	Enter Sandman	6	2	2	Klassik
2	Enter Sandman	6	2	3	Rock
⋮	⋮	⋮	⋮	⋮	⋮

Song

Genre

## SELECT Abfrage über zwei Tabellen

```
SELECT * FROM Song, Genre
```

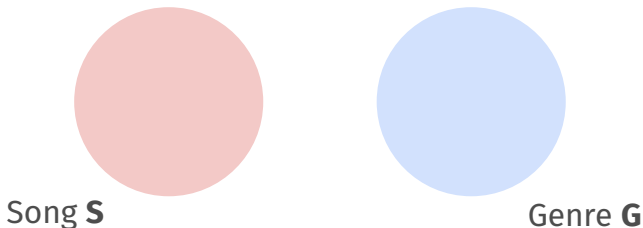
id	titel	genre	musiker	id	name
1	Supersonic Speed	3	1	1	Pop
1	Supersonic Speed	3	1	2	Klassik
1	Supersonic Speed	3	1	3	Rock
1	Supersonic Speed	3	1	4	Electronic
1	Supersonic Speed	3	1	5	Blues
1	Supersonic Speed	3	1	6	Heavy Metal
2	Enter Sandman	6	2	1	Pop
2	Enter Sandman	6	2	2	Klassik
2	Enter Sandman	6	2	3	Rock
⋮	⋮	⋮	⋮	⋮	⋮

Song

Genre



## Tabellen als Mengen von Elementen



Das **Kreuzprodukt** zweier Mengen  $S$  und  $G$  ist:

$$S \times G = \{(s, g) \mid s \in S, g \in G\}$$

## Kreuzprodukt von S und G

id	titel	genre	musiker	id	name
1	Supersonic Speed	3	1	1	Pop
1	Supersonic Speed	3	1	2	Klassik
1	Supersonic Speed	3	1	3	Rock
1	Supersonic Speed	3	1	4	Electronic
1	Supersonic Speed	3	1	5	Blues
1	Supersonic Speed	3	1	6	Heavy Metal
2	Enter Sandman	6	2	1	Pop
2	Enter Sandman	6	2	2	Klassik
2	Enter Sandman	6	2	3	Rock
⋮	⋮	⋮	⋮	⋮	⋮

Song

Genre

## Kreuzprodukt von S und G

id	titel	genre	musiker	id	name
1	Supersonic Speed	3	1	1	Pop
1	Supersonic Speed	3	1	2	Klassik
1	Supersonic Speed	3	1	3	Rock
1	Supersonic Speed	3	1	4	Electronic
1	Supersonic Speed	3	1	5	Blues
1	Supersonic Speed	3	1	6	Heavy Metal
2	Enter Sandman	6	2	1	Pop
2	Enter Sandman	6	2	2	Klassik
2	Enter Sandman	6	2	3	Rock
⋮	⋮	⋮	⋮	⋮	⋮

Song

Genre



## Kreuzprodukt von S und G

id	titel	genre	musiker	id	name
1	Supersonic Speed	3	1	1	Pop
1	Supersonic Speed	3	1	2	Klassik
1	Supersonic Speed	3	1	3	Rock
1	Supersonic Speed	3	1	4	Electronic
1	Supersonic Speed	3	1	5	Blues
1	Supersonic Speed	3	1	6	Heavy Metal
2	Enter Sandman	6	2	1	Pop
2	Enter Sandman	6	2	2	Klassik
2	Enter Sandman	6	2	3	Rock
⋮	⋮	⋮	⋮	⋮	⋮

Song

Genre

## Kreuzprodukt von S und G

id	titel	genre	musiker	id	name
1	Supersonic Speed	3	1	1	Pop
1	Supersonic Speed	3	1	2	Klassik
1	Supersonic Speed	3	1	3	Rock
1	Supersonic Speed	3	1	4	Electronic
1	Supersonic Speed	3	1	5	Blues
1	Supersonic Speed	3	1	6	Heavy Metal
2	Enter Sandman	6	2	1	Pop
2	Enter Sandman	6	2	2	Klassik
2	Enter Sandman	6	2	3	Rock
⋮	⋮	⋮	⋮	⋮	⋮

Song

Genre

## Kreuzprodukt von S und G

id	titel	genre	musiker	id	name
1	Supersonic Speed	3	1	1	Pop
1	Supersonic Speed	3	1	2	Klassik
1	Supersonic Speed	3	1	3	Rock
1	Supersonic Speed	3	1	4	Electronic
1	Supersonic Speed	3	1	5	Blues
1	Supersonic Speed	3	1	6	Heavy Metal
2	Enter Sandman	6	2	1	Pop
2	Enter Sandman	6	2	2	Klassik
2	Enter Sandman	6	2	3	Rock
⋮	⋮	⋮	⋮	⋮	⋮

Song

Genre

## Vorsicht: Ggf. Spalten mit gleichen Namen!

```
SELECT * FROM Song, Genre
```

id	titel	genre	musiker	id	name
1	Supersonic Speed	3	1	1	Pop
1	Supersonic Speed	3	1	2	Klassik
1	Supersonic Speed	3	1	3	Rock
1	Supersonic Speed	3	1	5	Blues

```
SELECT id FROM Song, Genre
```

```
ERROR: column reference "id" is ambiguous  
LINE 1: select id from Song, Genre
```

## Eindeutige Spalten durch TABELLE.SPALTE

```
SELECT titel, Genre.id, genre, name  
FROM Song, Genre
```

titel	id	genre	name
Supersonic Speed	1	3	Pop
Supersonic Speed	2	3	Klassik
Supersonic Speed	3	3	Rock
Supersonic Speed	5	3	Blues

## JOIN zweier Tabellen

```
SELECT * FROM Song, Genre  
WHERE Song.genre = Genre.id
```

id	titel	genre	musiker	id	name
1	Supersonic Speed	3	1	3	Rock
2	Enter Sandman	6	2	6	Heavy Metal
3	Layla	5	3	5	Blues
4	Ayo Technology	1	4	1	Pop
5	Afterglow	1	5	1	Pop
6	Wer wenn nicht wir	1	7	1	Pop
7	Blinding Lights	1	6	1	Pop
8	Bad Habits	1	5	1	Pop
9	Neuanfang	1	8	1	Pop

## JOIN zweier Tabellen

```
SELECT * FROM Song, Genre  
WHERE Song.genre = Genre.id
```

Ergibt die Menge **aller** gültigen Song/Genre Paare.

## JOIN zweier Tabellen

```
SELECT * FROM Song, Genre
WHERE Song.genre = Genre.id
```

Ergibt die Menge **aller** gültigen Song/Genre Paare.

Wir wollen aber die Songs des Genre *Rock*, also:

```
SELECT * FROM Song, Genre
WHERE Song.genre = Genre.id
AND Genre.name = 'Rock'
```



## JOIN Syntax

Statt SELECT mit 2 Tabellen, existiert das Schlüsselwort JOIN, um eine Verbindung mit einer anderen zu verknüpfen:

```
SELECT * FROM Song  
JOIN Genre ON Song.genre = Genre.id
```

ist das gleiche wie:

```
SELECT * FROM Song, Genre  
WHERE Song.genre = Genre.id
```

## JOIN Syntax – Vorteil:

JOIN Syntax erlaubt klare Trennung zwischen Fremdschlüssel-Bedingungen und Filter-Bedingungen (WHERE):

```
SELECT * FROM Song
  JOIN Genre ON Song.genre = Genre.id
 WHERE Genre.name = 'Rock'
```

## Wieviel Songs gibt es für jedes Genre?

```
SELECT Genre.name, count(Song.id)
FROM Song, Genre
JOIN Genre ON Song.genre = Genre.id
GROUP BY Genre.name
```

name	count
Heavy Metal	1
Pop	7
Rock	1
Blues	1

Was ist mit den Genres,  
für die es noch keinen Song gibt?



## Problem:

Kreuzprodukt erzeugt nur Paare aus existierenden Kombinationen (Genre, Song).

Das Genre *Klassik* hat keinen Song zugeordnet, daher wird auch kein Paar gebildet.

Mit LEFT JOIN erzwingen wir auch Paare mit fehlendem Song:

```
SELECT * FROM Genre  
LEFT JOIN Song ON Song.genre = Genre.id
```

## Beispiel: LEFT JOIN

```
SELECT * FROM Genre  
LEFT JOIN Song ON Song.genre = Genre.id
```

id	name	id	titel	genre	musiker
1	Pop	6	Wer wenn nicht wir	1	7
1	Pop	7	Blinding Lights	1	6
1	Pop	8	Bad Habits	1	5
1	Pop	9	Neuanfang	1	8
1	Pop	10	Flugmodus	1	8
2	Klassik	NULL	None	NULL	NULL
4	Electronic	NULL	None	NULL	NULL

**Frage: Wieviele Songs gibt es für jedes Genre?**

```
SELECT Genre.name, COUNT(song.id) FROM Genre
LEFT JOIN Song ON Song.genre = Genre.id
GROUP BY Genre.name
ORDER BY COUNT(song.id) DESC
```

name	count
Pop	7
Heavy Metal	1
Rock	1
Blues	1
Electronic	0
Klassik	0

## Überblick: JOIN von Tabellen

### INNER JOIN

John	1
Luke	2
Mary	3

 × 

2	A
3	B
4	C

 = 

Luke	2	2	A
Mary	3	3	B

### LEFT JOIN

John	1
Luke	2
Mary	3

 × 

2	A
3	B
4	C

 = 

John	1	null	null
Luke	2	2	A
Mary	3	3	B

## Überblick: JOIN von Tabellen

### RIGHT JOIN

John	1
Luke	2
Mary	3

 × 

2	A
3	B
4	C

 = 

Luke	2	2	A
Mary	3	3	B
null	null	4	C

### FULL JOIN

John	1
Luke	2
Mary	3

 × 

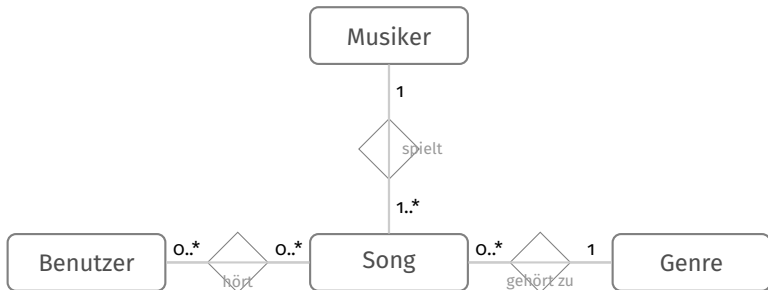
2	A
3	B
4	C

 = 

John	1	null	null
Luke	2	2	A
Mary	3	3	B
null	null	4	C



## Zur Erinnerung: Datenbank eines Streaming-Dienstes



**Welche Musiker spielen Songs aus dem Genre *Pop*?**

## Frage: Welche Musiker spielen Songs aus dem Genre Pop?

**Musiker**

id	name
1	Die Happy
2	Metallica
3	Eric Clapton
4	Milow
5	Ed Sheeran
6	The Weeknd
7	Vincent weiss
8	Clueso

**Song**

id	titel	genre	musiker
1	Supersonic Speed	3	1
2	Enter Sandman	6	2
3	Layla	5	3
4	Ayo Technology	1	4
5	Afterglow	1	5
6	Wer wenn nicht wir	1	7
7	Blinding Lights	1	6
8	Bad Habits	1	5
9	Neuanfang	1	8

**Genre**

id	name
1	Pop
2	Klassik
3	Rock
4	Electronic
5	Blues
6	Heavy Metal

## Frage: Welche Musiker spielen Songs aus dem Genre Pop?

**Musiker**

id	name
1	Die Happy
2	Metallica
3	Eric Clapton
4	Milow
5	Ed Sheeran
6	The Weeknd
7	Vincent weiss
8	Clueso

**Song**

id	titel	genre	musiker
1	Supersonic Speed	3	1
2	Enter Sandman	6	2
3	Layla	5	3
4	Ayo Technology	1	4
5	Afterglow	1	5
6	Wer wenn nicht wir	1	7
7	Blinding Lights	1	6
8	Bad Habits	1	5
9	Neuanfang	1	8

**Genre**

id	name
1	Pop
2	Klassik
3	Rock
4	Electronic
5	Blues
6	Heavy Metal

```
SELECT Musiker.name FROM Musiker
JOIN Song ON Song.musiker = Musiker.id
JOIN Genre ON Song.genre = Genre.id
WHERE Genre.name = 'Pop'
```

## Frage: Welche Musiker spielen Songs aus dem Genre Pop?

```
SELECT Musiker.name FROM Musiker
JOIN Song ON Song.musiker = Musiker.id
JOIN Genre ON Song.genre = Genre.id
WHERE Genre.name = 'Pop'
```

name
Die Happy
Metallica
Eric Clapton
Milow
Ed Sheeran
Vincent weiss
The Weeknd
Ed Sheeran
Clueso

**Frage: Welche Musiker spielen Songs aus dem Genre *Pop*?**

```
SELECT DISTINCT Musiker.name  
  FROM Musiker  
 JOIN Song ON Song.musiker = Musiker.id  
 JOIN Genre ON Song.genre = Genre.id  
 WHERE Genre.name = 'Pop'
```

name
Clueso
Ed Sheeran
Milow
The Weeknd
Vincent weiss

## Frage: Wie ist das Höchstalter der Hörer für Rock?

### Benutzer

id	name	alter	geschlecht	plz	ort
1	Maier	27	m	44801	Bochum
2	Weber	29	w	44801	Bochum
3	Schmidt	19	m	44225	Dortmund
4	Blumenfeld	47	w	45127	Essen
5	Mustermann	58	w	45128	Essen
6	Blumenfeld	63	m	44789	Bochum
7	Dampf	59	w	44135	Dortmund
8	Rolland	51	w	44269	Dortmund
9	Farnsworth	35	m	44225	Dortmund

### BenutzerSong

benutzer	song	datum
1	1	2022-04-04
2	1	2022-04-08
7	7	2022-04-03
5	9	2022-04-17
7	5	2022-04-17
6	4	2022-04-18
3	5	2022-04-06
2	5	2022-04-19
3	5	2022-04-05

### Song

id	titel	genre	musiker
1	Supersonic Speed	3	1
2	Enter Sandman	6	2
3	Layla	5	3
4	Ayo Technology	1	4
5	Afterglow	1	5
6	Wer wenn nicht wir	1	7
7	Blinding Lights	1	6
8	Bad Habits	1	5
9	Neuanfang	1	8

### Song

id	name
1	Pop
2	Klassik
3	Rock
4	Electronic
5	Blues
6	Heavy Metal

## Frage: **Wie ist das Höchstalter der Hörer für Rock?**

```
SELECT MAX(Benutzer.alter)
  FROM Benutzer, BenutzerSong, Song, Genre
 WHERE
       BenutzerSong.benutzer = Benutzer.id
 AND  BenutzerSong.song = Song.id
 AND  Song.genre = Genre.id
 AND  Genre.name = 'Rock'
```

## Frage: Wie ist das Höchstalter der Hörer je Genre?

```
SELECT Genre.name, MAX(Benutzer.alter)
FROM Benutzer, BenutzerSong, Song, Genre
WHERE
    BenutzerSong.benutzer = Benutzer.id
    AND BenutzerSong.song = Song.id
    AND Song.genre = Genre.id
    AND Genre.name = 'Rock'
GROUP BY Genre.name
```



## Frage: Welches Genre hat die meisten gespielten Songs?

```
SELECT Genre.name, COUNT(Benutzer.id)
  FROM Benutzer, BenutzerSong, Song, Genre
 WHERE
       BenutzerSong.benutzer = Benutzer.id
    AND BenutzerSong.song = Song.id
    AND Song.genre = Genre.id
 GROUP BY Genre.name
```

## Frage: Welches Genre hat die meisten Hörer?

```
SELECT Genre.name, COUNT(DISTINCT Benutzer.id)
  FROM Benutzer, BenutzerSong, Song, Genre
 WHERE
       BenutzerSong.benutzer = Benutzer.id
  AND BenutzerSong.song = Song.id
  AND Song.genre = Genre.id
 GROUP BY Genre.name
```

## Frage: Welches Genre hat die meisten Hörer im März?

```
SELECT Genre.name, COUNT(DISTINCT Benutzer.id)
  FROM Benutzer, BenutzerSong, Song, Genre
 WHERE
       BenutzerSong.benutzer = Benutzer.id
 AND  BenutzerSong.song = Song.id
 AND  Song.genre = Genre.id
 AND  BenutzerSong.datum >= '2022-03-01'
 AND  BenutzerSong.datum <= '2022-03-31'
 GROUP BY Genre.name
```