

# WIRTSCHAFTSINFORMATIK 2

DATENBANKEN: SQL

BERND BLÜMEL, CHRISTIAN BOCKERMANN,  
CHRISTIAN METZGER

HOCHSCHULE BOCHUM

SOMMERSEMESTER 2023

## Vom Konzept zur Implementierung

- SQL ist **standardisierte** Sprache für Datenbanken
- kleine Unterschiede / Dialekte zwischen Datenbanken
- SQL erlaubt Abfragen durch Benutzer + Programme



A	B	C	D

Abfrage (SQL)



Datenbank



Ergebnis (Tabelle)

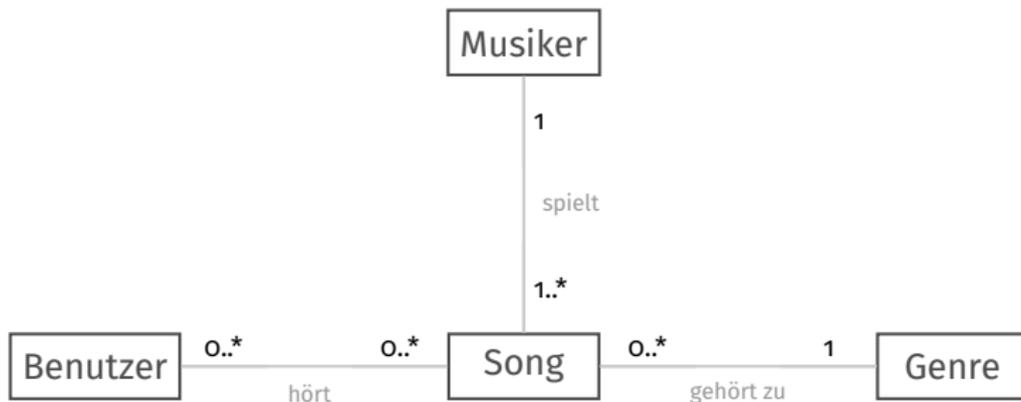
## SELECT zum Abfragen von Datensätzen

```
SELECT [DISTINCT | ALL] spalten  
FROM tabelle(n)  
[WHERE bedingung]  
[GROUP BY spalten]  
[HAVING bedingung]  
[ORDER BY spalten]
```

## SELECT mit JOIN zum Abfragen von Datensätzen

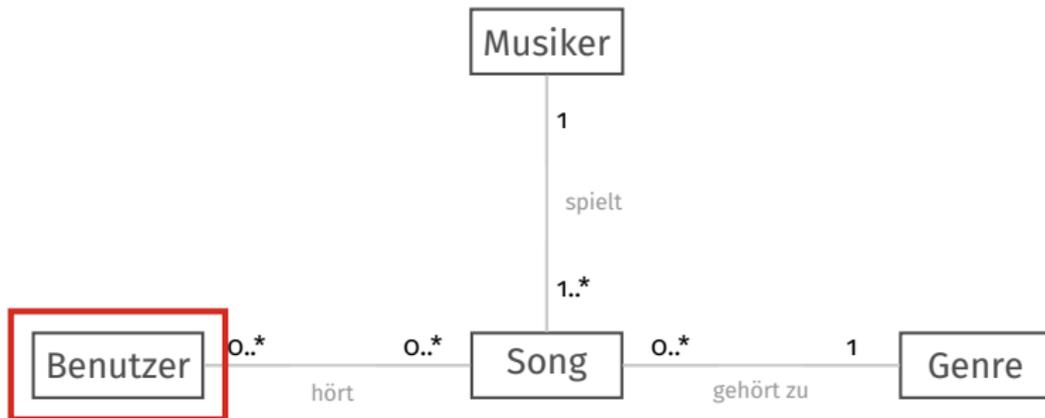
```
SELECT [DISTINCT | ALL] spalten  
FROM tabelle(n)  
[[LEFT | RIGHT |..] JOIN tabelle ON bedingung]  
[WHERE bedingung]  
[GROUP BY spalten]  
[HAVING bedingung]  
[ORDER BY spalten]
```

## Beispiel: Datenbank eines Streaming-Dienstes



**Wieviele Benutzer kommen aus den verschiedenen Städten?**

## Beispiel: Datenbank eines Streaming-Dienstes



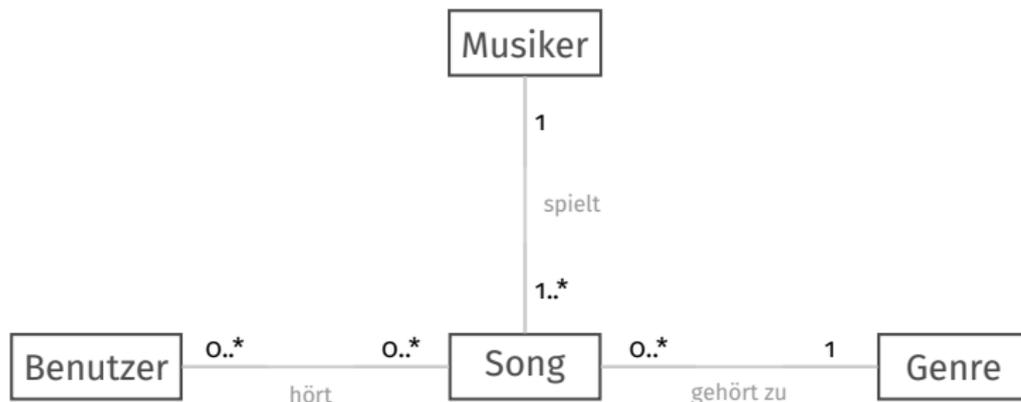
**Wieviele Benutzer kommen aus den verschiedenen Städten?**

**Frage:** Wie viele Benutzer kommen aus den verschiedenen Städten?

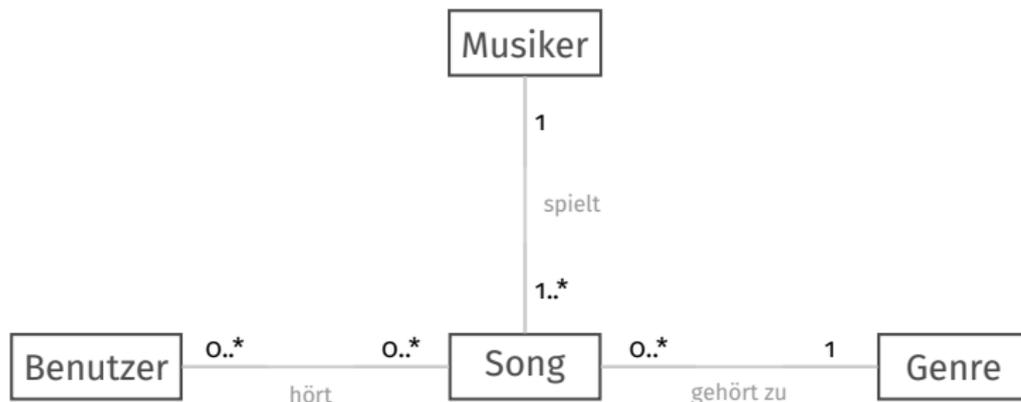
```
SELECT ort, count(*) FROM Benutzer  
GROUP BY ort  
HAVING count(*) > 2
```

ort	count(*)
Bochum	3
Dortmund	4

## Beispiel: Datenbank eines Streaming-Dienstes

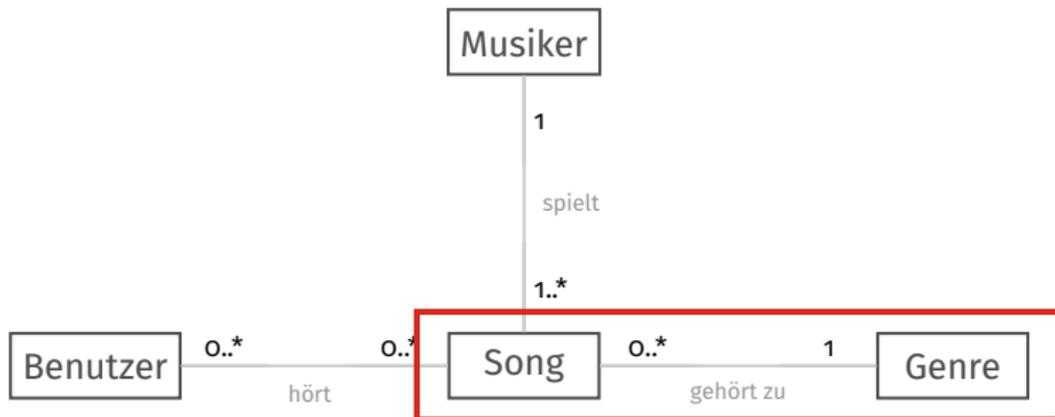


## Beispiel: Datenbank eines Streaming-Dienstes



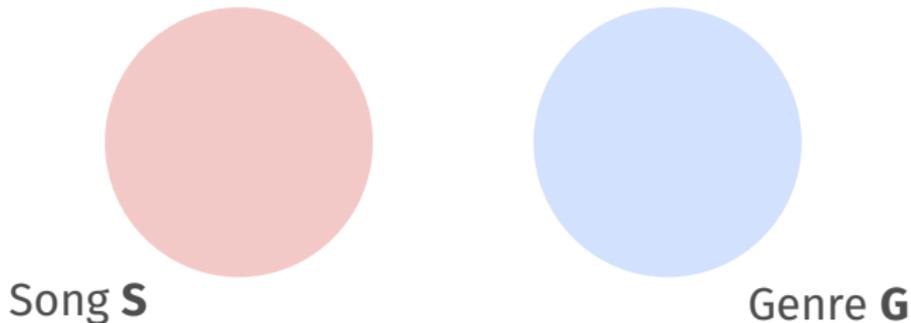
**Welche Songs gehören zum Genre Rock ?**

## Beispiel: Datenbank eines Streaming-Dienstes



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

## 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**

## 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'
```

## Wieviel Songs gibt es für jedes Genre?

```
SELECT Genre.name, count(Song.id)
FROM Song, Genre
WHERE 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?



## 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

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

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

name
Clueso
Ed Sheeran
Milow
The Weeknd
Vincent weiss

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

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

name
Clueso
Ed Sheeran
Milow
The Weeknd
Vincent weiss

## Ändern von Daten mit **UPDATE**

```
UPDATE tabelle SET spalte1=wert1, ...  
WHERE bedingung
```

- Ohne Bedingung werden alle Zeilen einer Tabelle geändert
- Gezielte Änderungen z.B. über den *Primärschlüssel*

## Daten löschen mit **DELETE**

```
DELETE FROM tabelle WHERE bedingung
```

- Ohne Bedingung werden alle Daten einer Tabelle gelöscht
- Datenbank überprüft auch hier Constraints!

# Übungen zu SQL

## Beispiel: Datenbank Medienausstattung

**Datei:** [weitereBeispiele.pdf](#)

(auf Sciebo oder [datascience.hs-bochum.de](https://datascience.hs-bochum.de))

Gebaeude

Raum

Dozent

Servicepartner

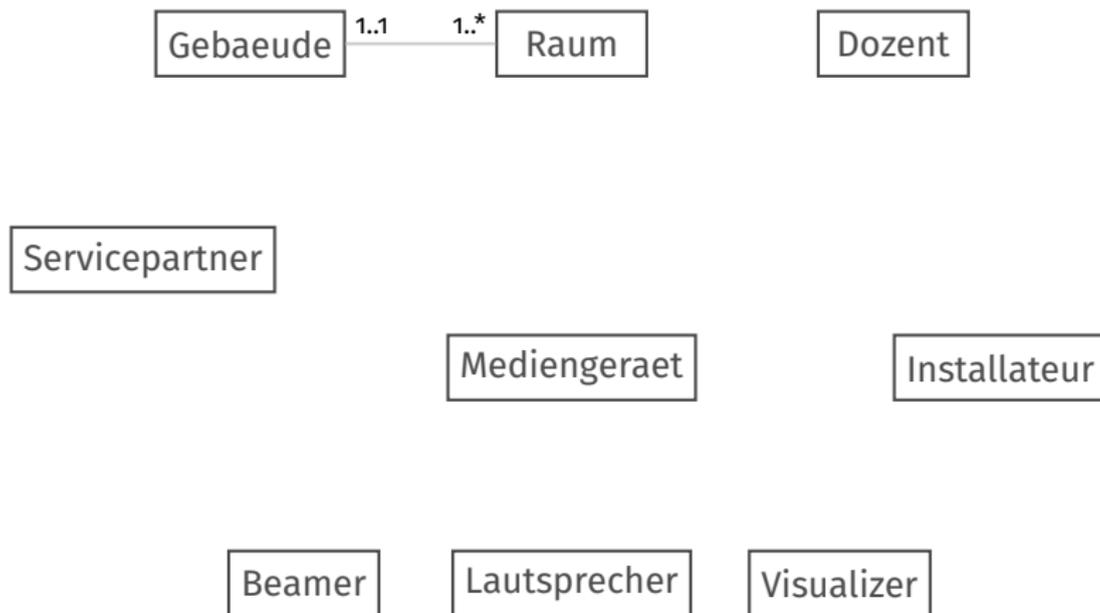
Mediengeräet

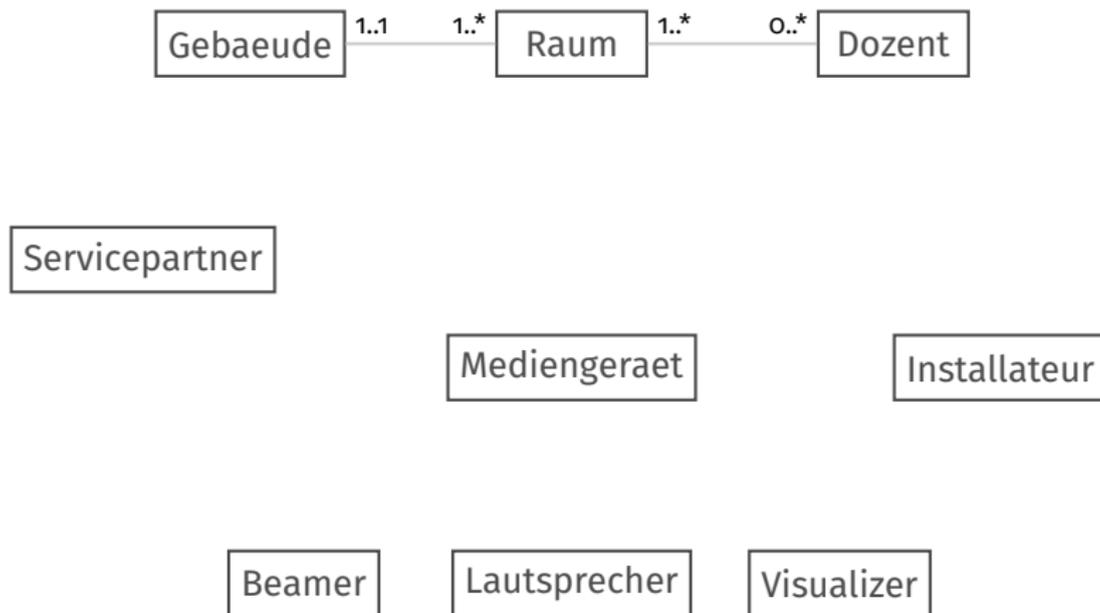
Installateur

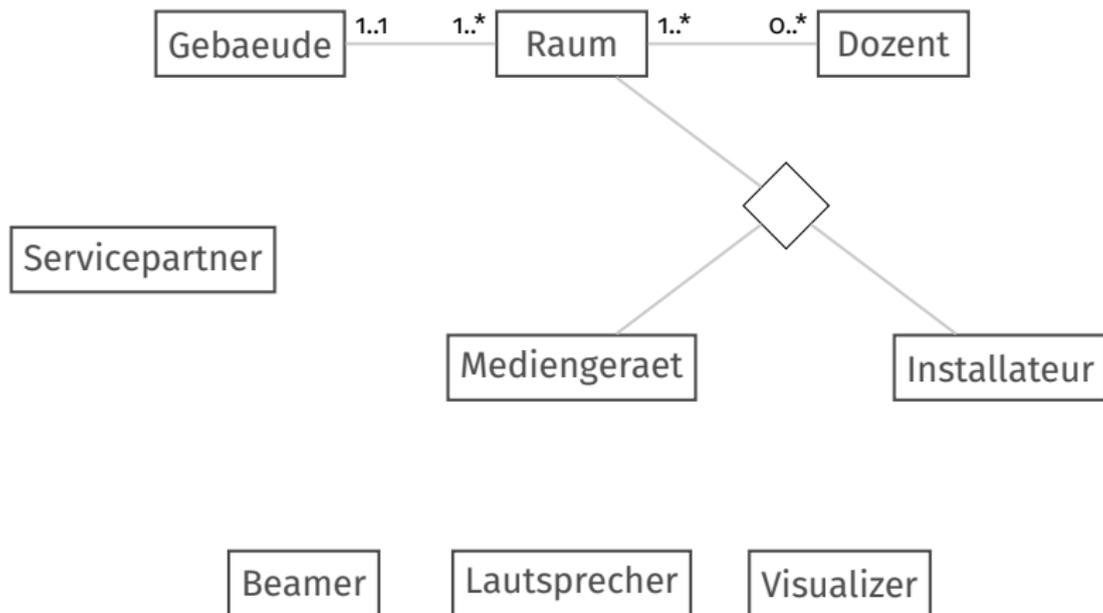
Beamer

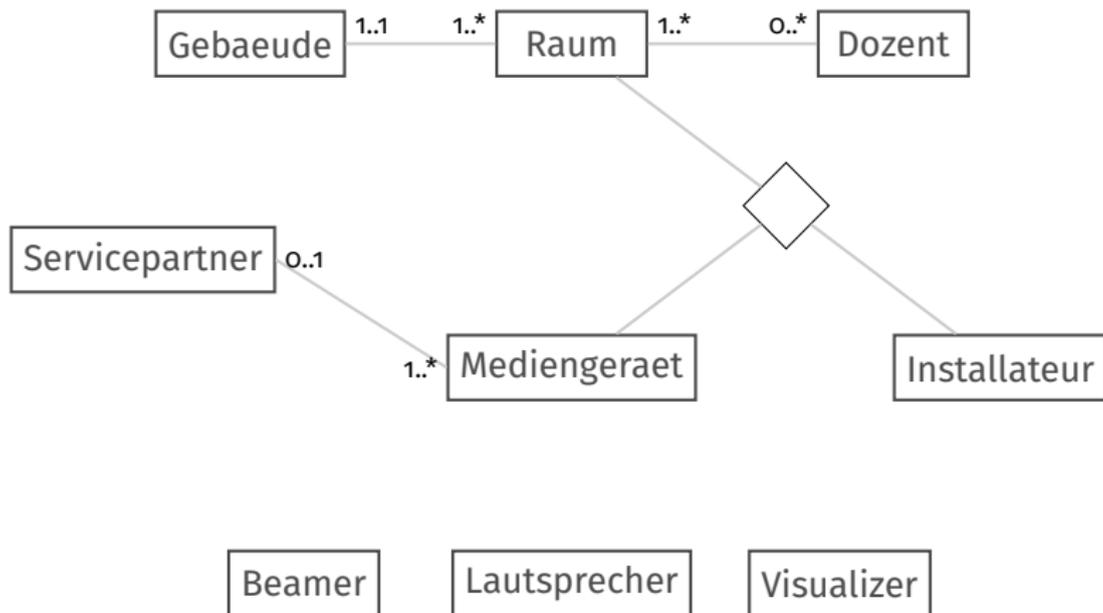
Lautsprecher

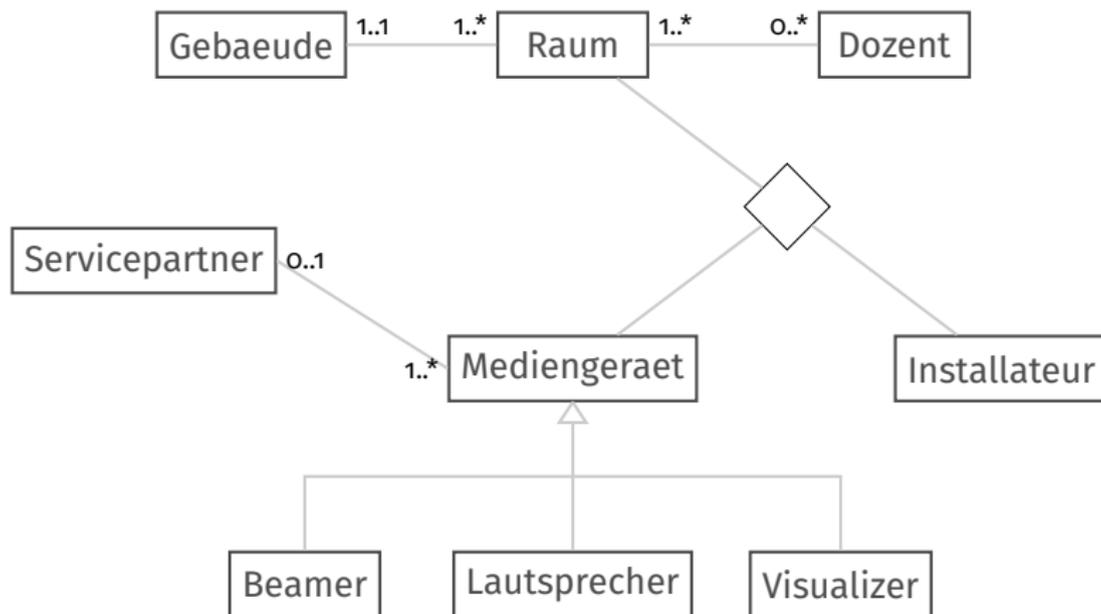
Visualizer











## Tabellen das ER-Modell

Tabellenname	Primärschlüssel	weitere Attribute
Gebaeude	GebaeudeNr	Bezeichnung, Strasse, PLZ, Ort
Raum	RaumNr	Name, Groesse, <u>GebaeudeNr</u>
Dozent	DozentNr	Name
Mediengeräet	MediengeräetNr	Kaufdatum, Preis, <u>ServicepartnerNr</u>
Beamer	MediengeräetNr	Aufloesung, Lichtstrom
Lautsprecher	MediengeräetNr	Leistung
Visualizer	MediengeräetNr	Kameratyp, Lichtstärke
Servicepartner	ServicepartnerNr	Ansprechpartner, Firmenname
Installateur	InstallateurNr	Name, Vorname, Strasse, PLZ, Ort
DozentRaum	<u>DozentNr, RaumNr</u>	
Installation	<u>DozentNr, RaumNr, MediengeräetNr</u>	Datum, Arbeitszeit

**Frage:** Welche Dozenten unterrichten in den Räumen, die mit AW 01 beginnen?

```
SELECT Dozent.name
FROM Dozent, DozentRaum, Raum
WHERE
    Dozent.DozentNr = DozentRaum.DozentNr
AND Raum.RaumNr = DozentRaum.RaumNr
AND Raum.name LIKE "AW 01%"
```

**Frage:** Welche Dozenten unterrichten in den Räumen, die mit AW 01 beginnen?

Alternative:

```
SELECT Dozent.name
FROM Dozent
LEFT JOIN DozentRaum
  ON Dozent.DozentNr = DozentRaum.DozentNr
LEFT JOIN Raum
  ON Raum.RaumNr = DozentRaum.RaumNr
WHERE Raum.name LIKE "AW 01%"
```