

# Tag 2

## Inhaltsverzeichnis

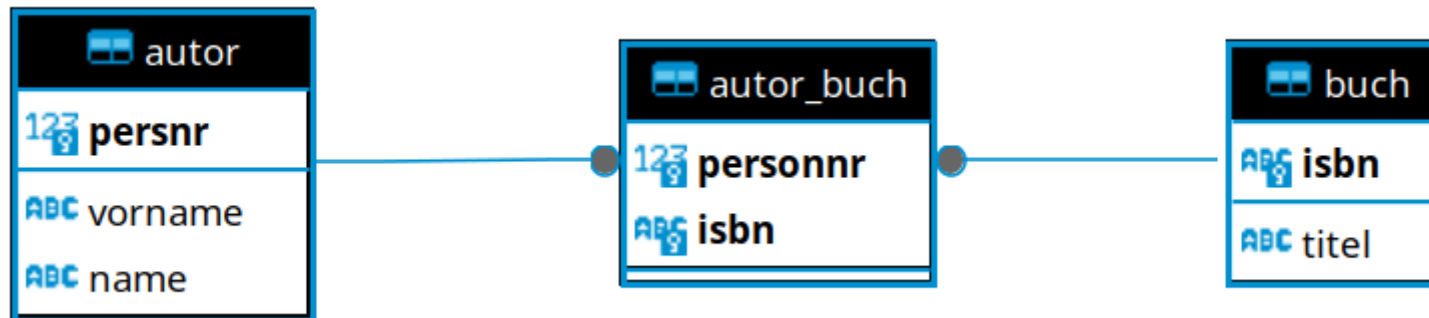
- **Relationales Modell**
  - Beispiel, Pro und Kontra
  - Relation und Schema
  - Umsetzung ER-Modell --> relationales Schema
  - DB Schema: Empfehlungen
- **Typen von Abfragesprachen**
- **SQL Teil 1**
  - Ziel und Geschichte
  - Daten- und Tabellentypen
  - Befehle (Überblick)
  - Administration- und Client-Programme
  - autorbuch Datenbank erstellen und ausfüllen
- **Übungen**

# Das relationale Modell

## Beispiel

125 persnr	ABC vorname	ABC name
12	alfons	kemper
34	michael	kofler
56	gilles	maitre
78	larry	wall

ABC isbn	ABC titel
111	visualbasic 2008
123	datenbanksysteme
222	mathematica
456	mysql 5
789	linux
999	db administration



125 personnr	ABC isbn
34	111
12	123
34	222
34	456
34	789

# Das relationale Modell

## Pro und Kontra

### Pro

- Mathematisch basiert
- Weit verbreitet
- Lange Erfahrung
- Guter programmatischer Support

### Kontra

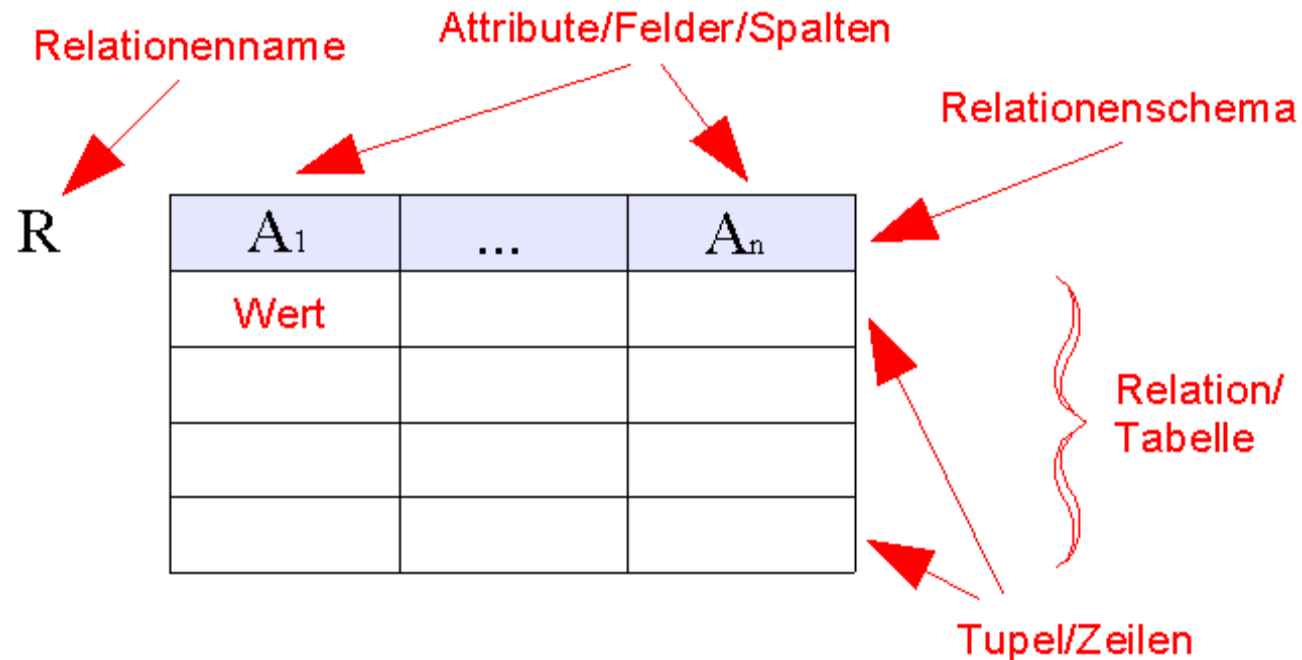
- Künstliche Datensegmentierung
- Rekursion und Vererbung nicht *direkt* modellierbar

Produkte (*Achtung: alle sind verschieden*):  
Oracle, MySQL, PostgreSQL, MS SQL Server,  
Sybase, DB2, (MS Access?), ...

Applikationen: zahlreiche...

# Das relationale Modell

## Relation



- Relationale DB: Sammlung von **Tabellen**
- Jede Tabelle: Sammlung von **Zeilen**
- Jede Zeile: Menge von gleichen unstrukturierten **Attributen**
- Freie Ordnung von Zeilen und Attributen
- Abfragen, um Daten zu manipulieren

# Das relationale Modell

## Primär- und Fremdschlüssel

Primärschlüssel



Personal	PersNr	Vorname	Name	...
...	234	Frieda	Müller	...
...	993	Horst	Winter	...
...	670	Hans	Sauer	...
...	...	...	...	...

IstChefVon	Vorgesetzter	Untergebener
...	...	...
...	234	993
...	234	670
...	...	...

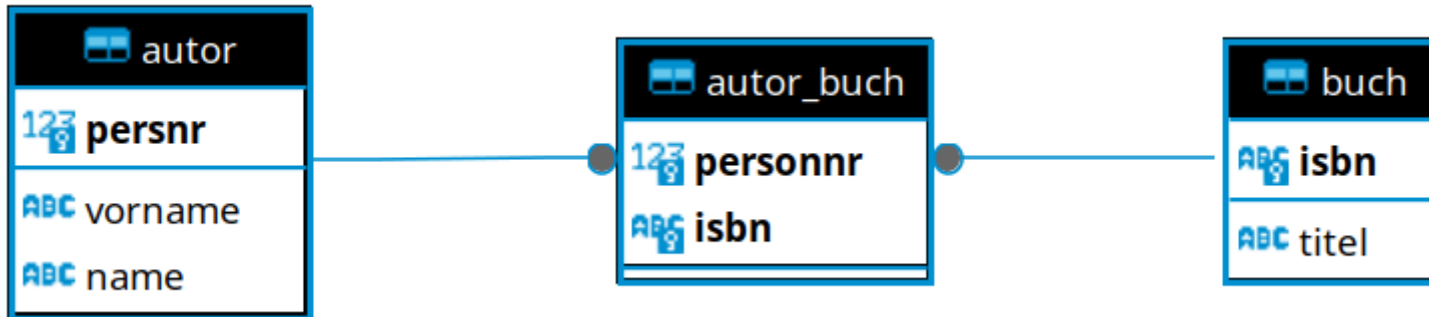
Fremdschlüssel

Primärschlüssel von IstChefVon

- *Primärschlüssel*: Minimale Menge von Attributen, die einen Datensatz in einer Tabelle **eindeutig** identifiziert.  
(Primary-Key, PK)
- *Fremdschlüssel*: Attribut einer Relation, welches den Primärschlüssel einer (anderen) Relation **referenziert**.  
(Foreign-Key, FK)

# Das relationale Modell

## Schlüssel und Schema Definition



Wo sind welche Schlüssel? PK *rot*, FK *blau*

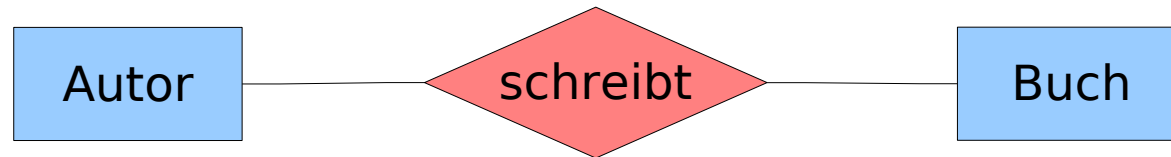
autor : { persnr: int, vorname: String, name: String }

autor\_buch : { personnr: int, isbn: String }

buch : { isbn: String, titel: String }

personnr + isbn bilden zusammen einen PK

# Umsetzung ER-Modell in rel. Schema Entitäten



autor : { persnr: int, vorname: String, name: String }

<u>123</u> persnr	ABC vorname	ABC name
12	alfons	kemper
34	michael	kofler
56	gilles	maitre
78	larry	wall

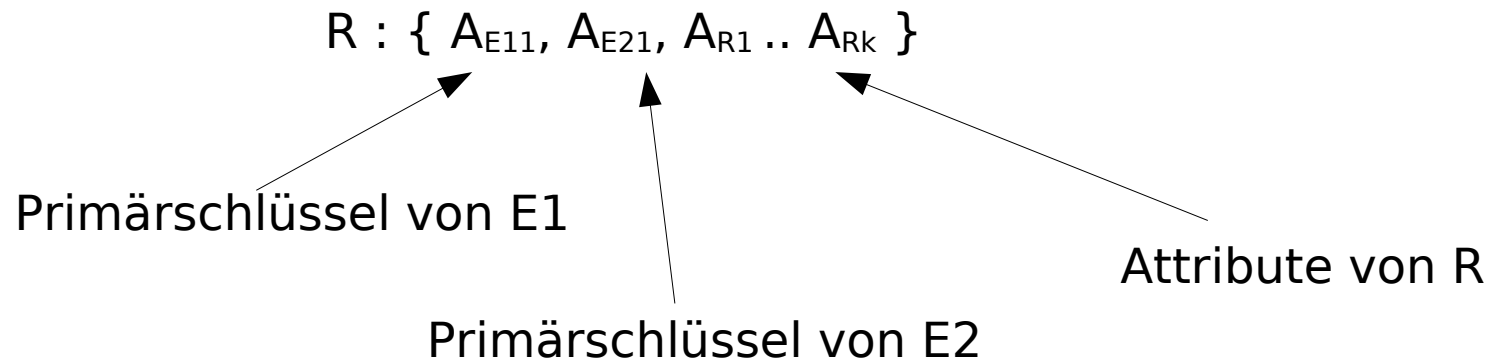
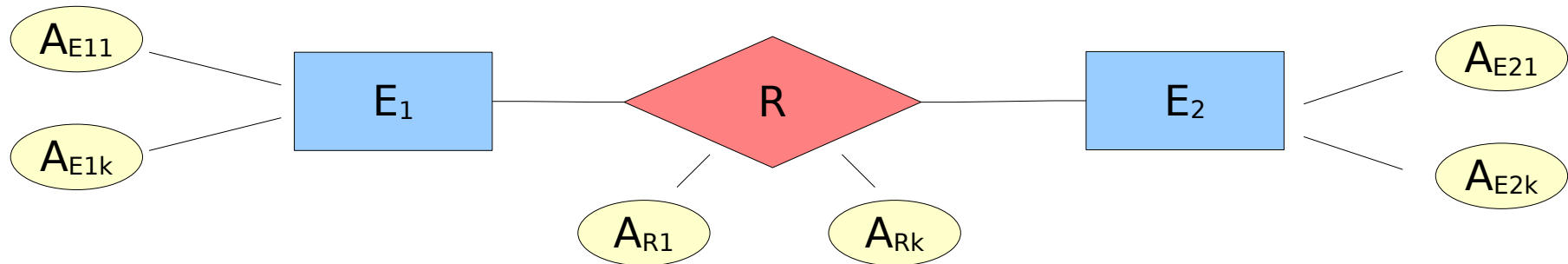
Meine Notation:  
Tabelle: {Feld: Typ, ...}

buch : { isbn: String, titel: String }

ABC <u>isbn</u>	ABC titel
111	visualbasic 2008
123	datenbanksysteme
222	mathematica
456	mysql 5
789	linux
999	db administration

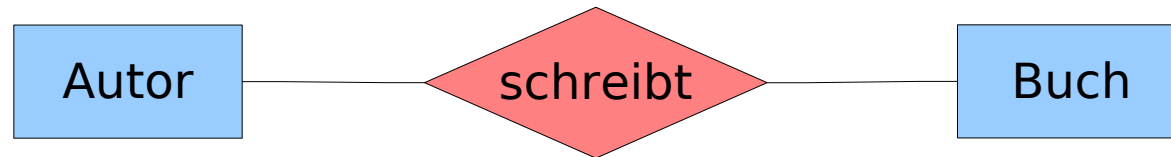
Meine Notation:  
PK wird unterstrichen.

# Umsetzung ER-Modell in rel. Schema Beziehungen (1)





# Umsetzung ER-Modell in rel. Schema Beziehungen (2)



`schreibt : { personnr: int, isbn: String }`

<a href="#">123 personnr</a>	<a href="#">ABC isbn</a>
<a href="#">34</a>	<a href="#">111</a>
<a href="#">12</a>	<a href="#">123</a>
<a href="#">34</a>	<a href="#">222</a>
<a href="#">34</a>	<a href="#">456</a>
<a href="#">34</a>	<a href="#">789</a>

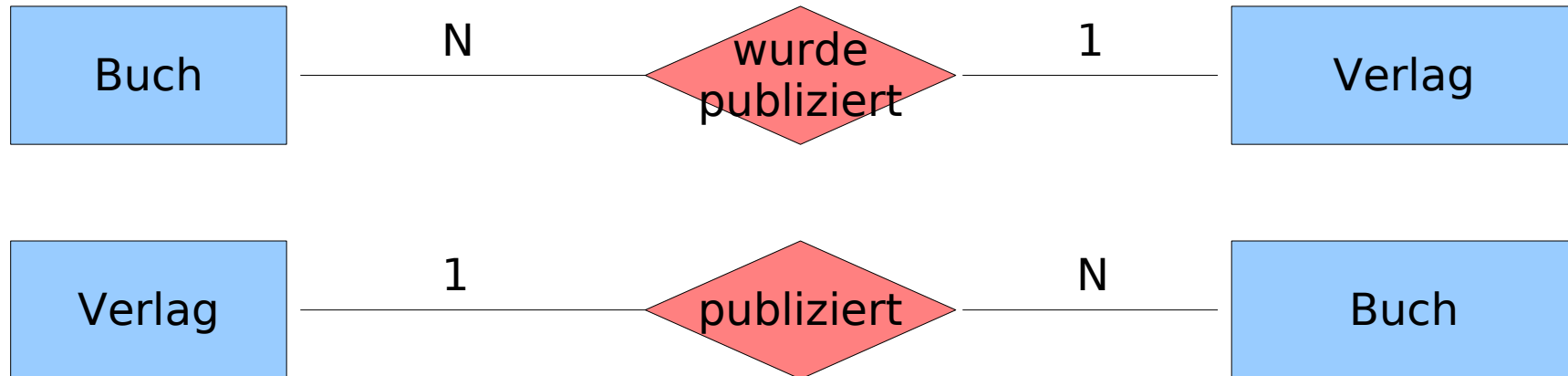
oder eher

`autor_buch : { personnr: int, isbn: String }`

# Umsetzung ER-Modell in rel. Schema Verfeinerung (1)

- Ziel: Zusammenfassung von Relationen
- Möglich nur für 1:1, 1:n oder n:1 Beziehungen
- Geht nicht für n:m Beziehungen

# Umsetzung ER-Modell in rel. Schema Verfeinerung (2)



Nicht ideale Lösung (wäre korrekt für "n:m" Bez.)

buch : { isbn: String, titel: String }  
verlag : { verlagid: Integer, kategorie: String }  
buch\_verlag : { isbn: String, **verlagid**: Integer }

## Vereinfachte Lösung

buch : { isbn: String, titel: String; verlagid: Integer }  
verlag : { **verlagid**: Integer, kategorie: String }

# Abfragesprachen

## Welche Autoren haben "Michael" als Vorname?

autor : { persnr: int, vorname: String, name: String }

persnr	vorname	name
12	alfons	kemper
34	michael	kofler
56	gilles	maitre
78	larry	wall

Relationale Algebra  $\prod_{Name} (\sigma_{Vorname=Michael}(Autor))$

Rel. Tupelkalkül  $\{a.n \mid a.n \in Autor \wedge a.v = Michael\}$

Rel. Domänenkalkül  $\{n \mid \exists [n, v] \in Autor \wedge v = Michael\}$

**SQL**

```
select name from autor where  
vorname='Michael';
```

# SQL

## Das Ziel

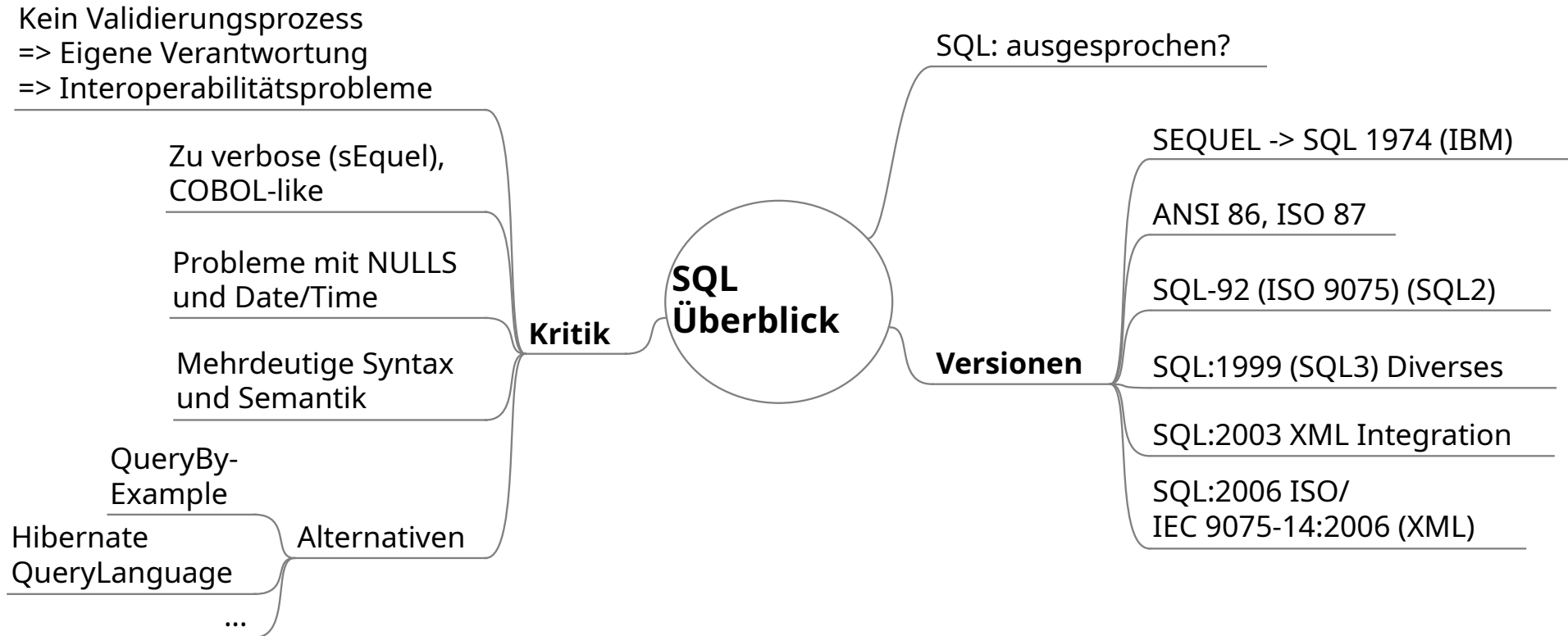
- Standardisierte
- Abfragesprache
  - Definition (DDL)
  - Manipulation (DML)
  - Control (DCL)
- Für **relationale** Datenbanken



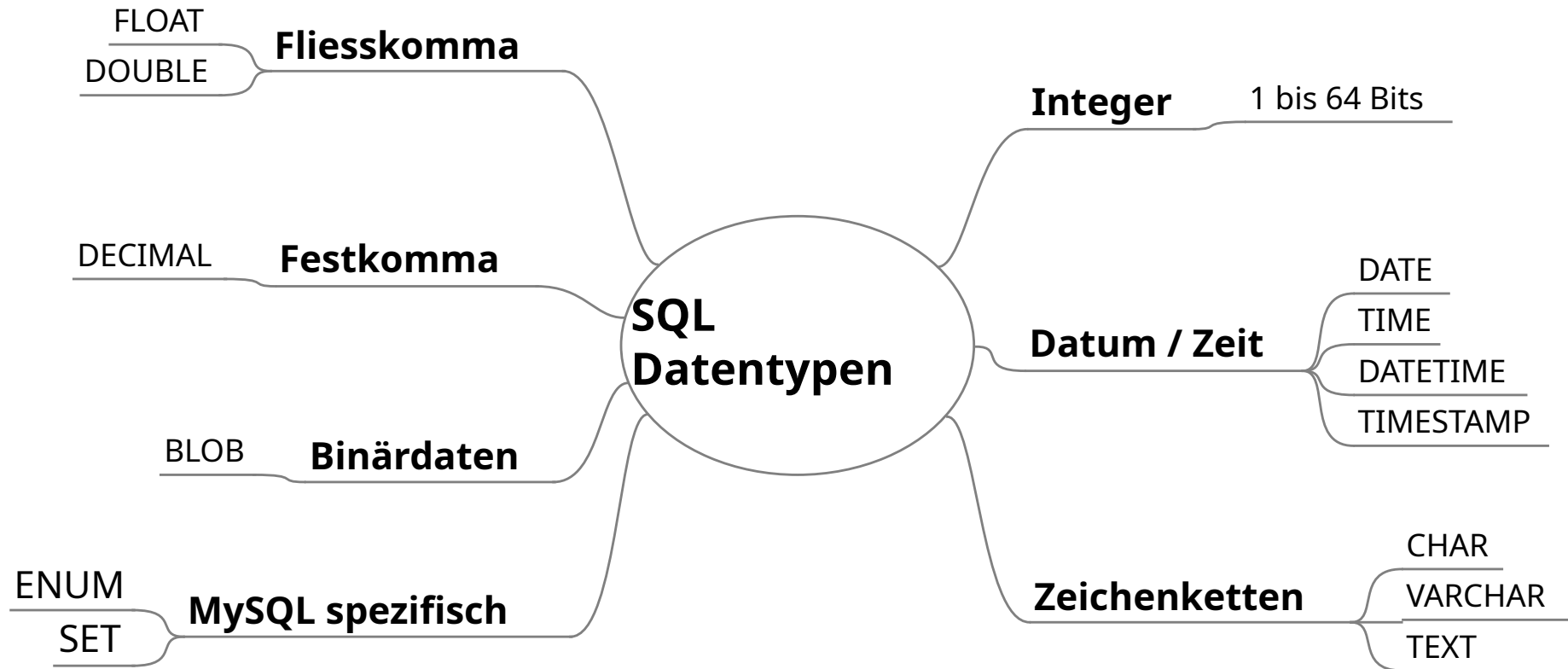
Structured Query Language (SQL)

# SQL

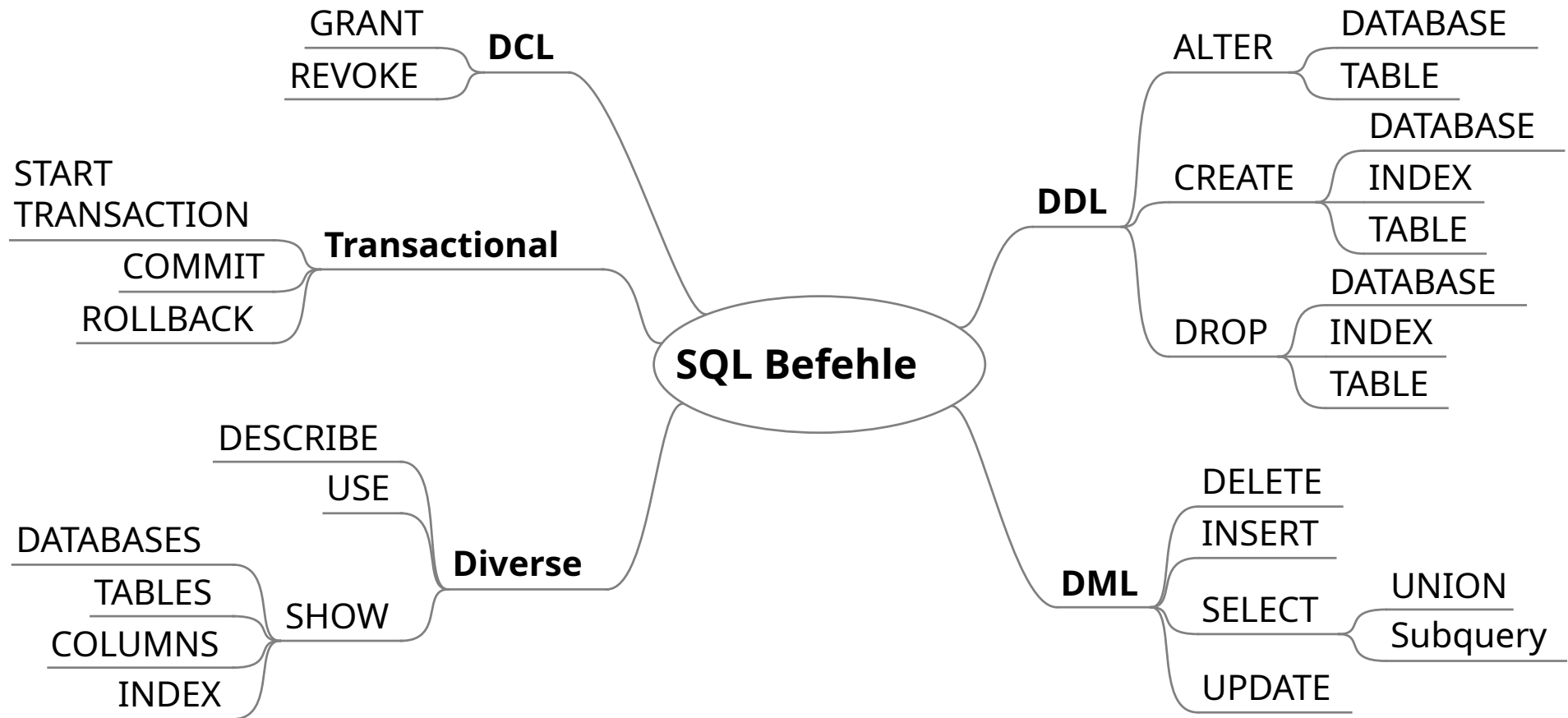
## Die Geschichte



# (My)SQL Datentypen



# SQL Alle Befehle





# Eine DB auswählen

## MySQL vs. MariaDB

### MySQL

- An SUN für eine Milliarde \$ 2009 verkauft. 2010 kauft Oracle SUN...
- Seitdem Proprietäre DB, mit Community-Version
- GUI Admin Tool: MySQL Workbench (free)
- Command Line Tool: MyCLI, MySQL

### MariaDB

- Fork von MySQL in 2010
- OpenSource Projekt
- Mehr Tabellentypen unterstützt
- GUI Tool: SQLyog (nicht frei)
- Command Line Tool: MyCLI

# MySQL Tabellentypen

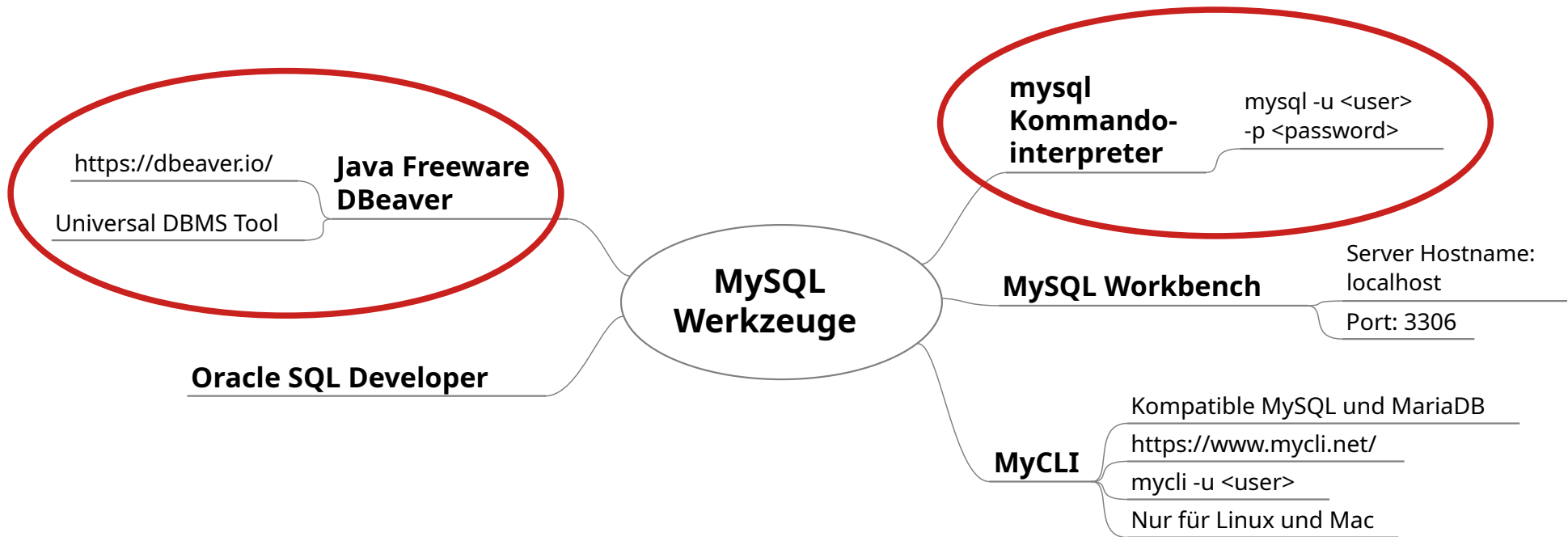
## InnoDB

- Transaktionsfähig
- Foreign Key Support
- Crash Recovery
- Braucht viel Ressourcen
- **Default** für MySQL Tabellen

## MyISAM

- *Nicht* transaktionsfähig
- *Kein* Foreign Key Support
- *Kein* Crash Recovery
- Schlank und effizient

# (My)SQL Admin Client-Programme



# autorbuch: DB erstellen und ausfüllen

- DB kreieren
- DB Schema anlegen
- DB mit Daten ausfüllen / ändern

- Wir arbeiten mit beiden Werkzeugen:
  - MySQL Kommando-Interpreter.
  - DBeaver.
- Gibt es ein SQL Coding Standard?
  - leider gibt nichts Offizielles.
  - Für MySQL: [Siehe z.B. hier](#) (die akzeptierte Antwort).
  - Wir schreiben alles klein im Unterricht.

# autorbuch und DBeaver Neue Verbindung erstellen

Verbindung "localhost" Konfiguration

Verbindungseinstellungen  
MySQL Verbindungseinstellungen

MySQL

Verbindungseinstellungen

Initialisierung  
Shell-Befehle  
Clientidentifizierung  
Transaktionen  
Allgemein  
Metadaten  
Fehlerhandler  
Datenübertragung  
> Ergebnismengen  
> SQL-Editor

Allgemein Treibereigenschaften SSH SSL + Network configurations...

Server

Connect by:  Host  URL

URL: jdbc:mysql://localhost:3306/?useSSL=false&allowPublicKeyRetrieval=true

Server-Host: localhost Port: 3306

Datenbank:

Authentifizierung (Database Native)

Benutzername: root

Passwort:   Passwort speichern

Erweitert

Zeitzone des Servers: Automatische Erkennung

Lokaler Client: /usr

[Connection variables information](#) [Database documentation](#)

Treibername: MySQL

# autorbuch DB kreieren (1)

```
mysql> help create database
```

```
Name: 'CREATE DATABASE'
```

```
Description:
```

```
Syntax:
```

```
CREATE {DATABASE | SCHEMA} [IF NOT EXISTS] db_name  
      [create_specification ...]
```

```
create_specification:
```

```
      [DEFAULT] CHARACTER SET charset_name
```

```
      | [DEFAULT] COLLATE collation_name
```

```
      ...
```

Wie gespeichert?

String-Vergleich  
z.B: ä == ae,  
ß == ss

```
Default character set: utf8mb4
```

```
Default collation: utf8mb4_0900_ai_ci
```

```
Überprüfung: select @@character_set_database, @@collation_database;
```

# autorbuch DB kreieren (2)

```
mysql> create database autorbuch;  
Query OK, 1 row affected (0.00 sec)
```

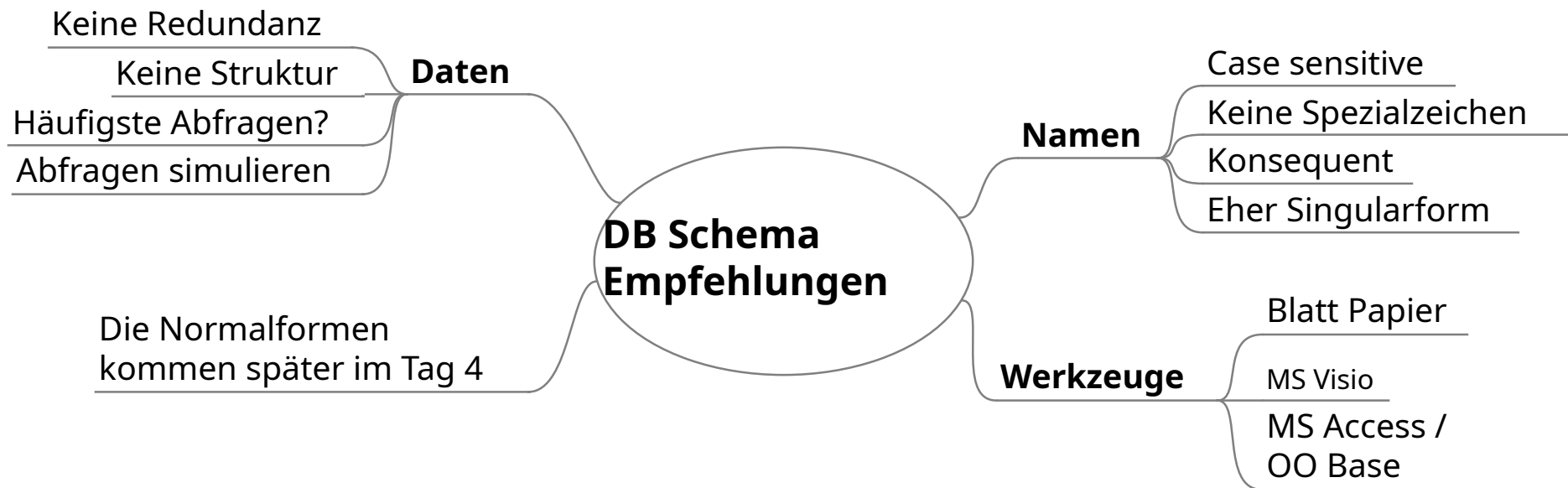
```
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| ... |  
| autorbuch |  
+-----+  
2 rows in set (0.00 sec)
```

```
mysql> use autorbuch;  
Database changed
```

```
mysql> select @@character_set_database, @@collation_database;  
+-----+-----+  
| @@character_set_database | @@collation_database |  
+-----+-----+  
| utf8mb4 | utf8mb4_0900_ai_ci |  
+-----+-----+
```

# autorbuch DB Schema anlegen (1)

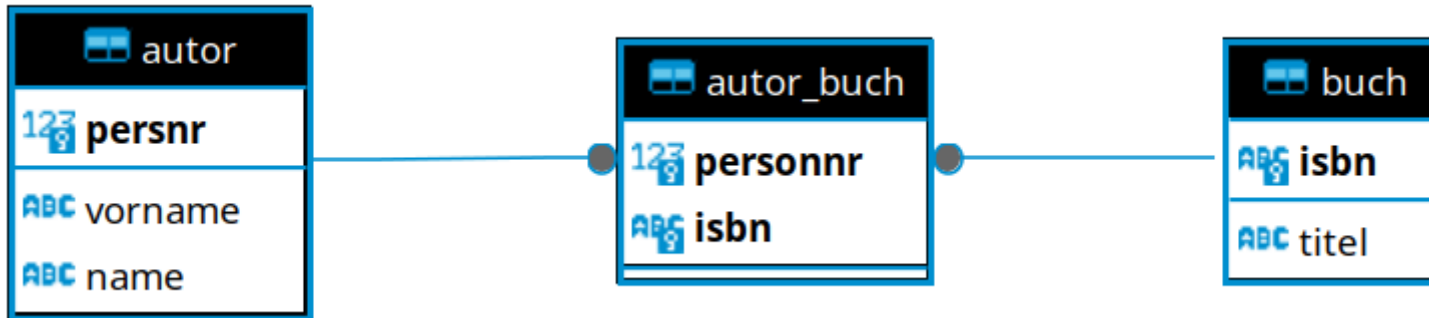
## Empfehlungen





# autorbuch DB Schema anlegen (2)

## Tabelle Autor kreieren



```
mysql> create table autor (  
    persnr int not null auto_increment,  
    vorname varchar(20) not null,  
    name varchar(20) not null,  
    primary key (persnr)  
    ) engine=innodb;
```

Query OK, 0 rows affected (0.01 sec)

```
mysql>
```

## autorbuch DB Schema anlegen (3)

```
mysql> show tables;
```

```
+-----+
| Tables_in_AutorBuch |
+-----+
| autor                |
+-----+
```

```
1 row in set (0.00 sec)
```

```
mysql> desc autor;
```

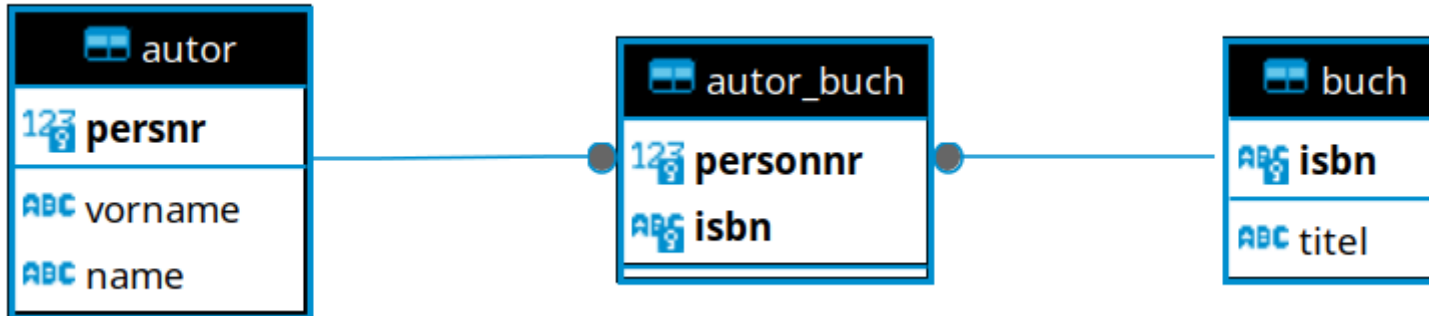
```
+-----+-----+-----+-----+-----+-----+
| Field      | Type           | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| persnr     | int            | NO   | PRI | NULL    |      |
| vorname    | varchar(20)    | NO   |     | NULL    |      |
| name       | varchar(20)    | NO   |     | NULL    |      |
+-----+-----+-----+-----+-----+-----+
```

```
3 rows in set (0.00 sec)
```

```
mysql>
```

# autorbuch DB Schema anlegen (4)

## Tabelle "autor\_buch" kreieren



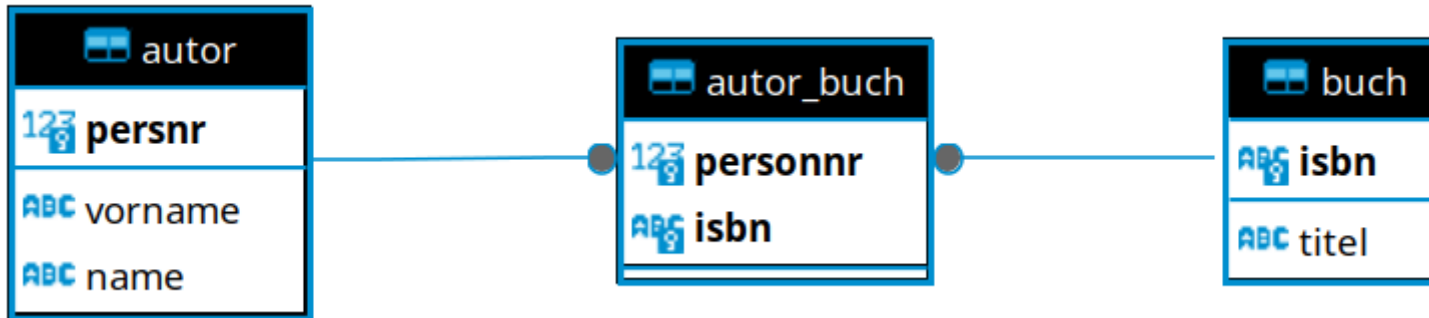
```
mysql> create table autor_buch (  
    personnr int not null,  
    isbn varchar(10) not null,  
    primary key (personnr,isbn),  
    ) engine=innodb;
```

Query OK, 0 rows affected (0.01 sec)

```
mysql>
```

# autorbuch DB mit Daten ausfüllen (1)

## Tabelle autor ausfüllen



```
mysql> insert into autor set  
vorname="Thomas",  
name="Kudrass";
```

```
Query OK, 1 row affected, 1 warning (0.01 sec)
```

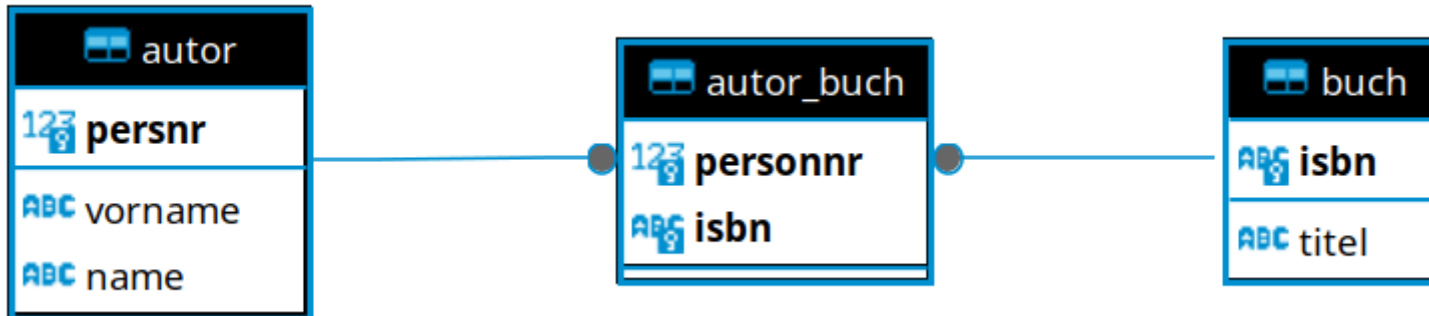
```
mysql> insert into autor values  
(null, "Michael", "Kofler");
```

```
Query OK, 1 row affected, 1 warning (0.01 sec)
```

```
mysql>
```

# autorbuch DB mit Daten ausfüllen (2)

## Tabelle autor anzeigen

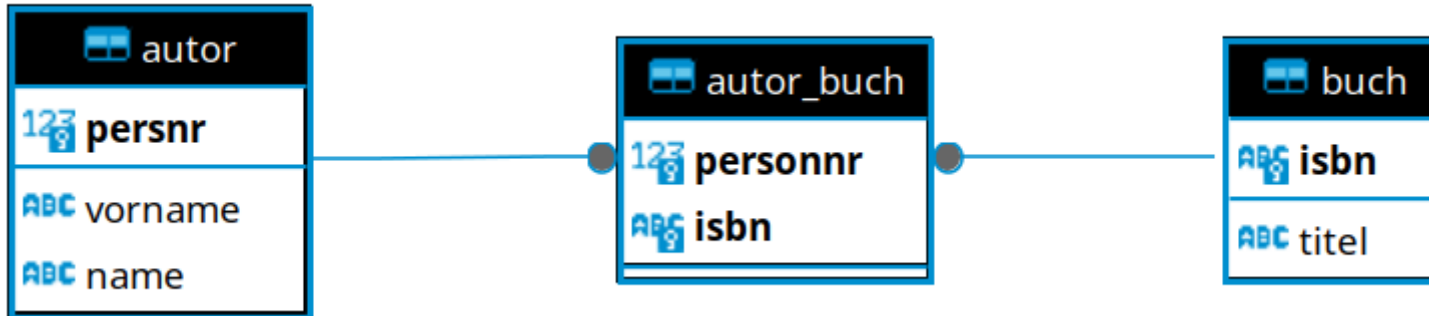


```
mysql> select * from autor;
```

```
+-----+-----+-----+
| persnr | vorname | name   |
+-----+-----+-----+
|      1 | Thomas  | Kudrass |
|      2 | Michael | Kofler  |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

```
mysql>
```

# autorbuch DB Tabelle autor ändern (1)



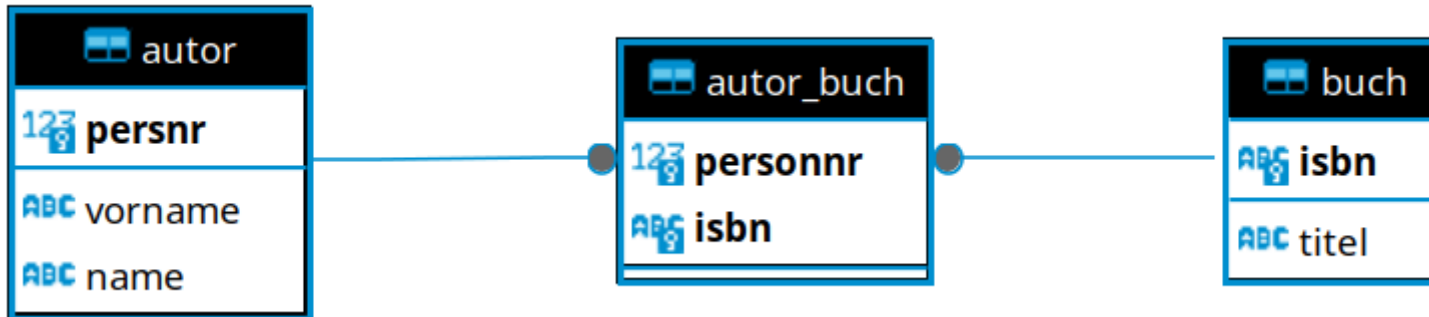
```
mysql> update autor
set vorname="Mike" where PersNr=2;
```

```
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

```
mysql> select * from autor;
+-----+-----+-----+
| persnr | vorname | name   |
+-----+-----+-----+
|      1 | Thomas  | Kudrass |
|      2 | Mike    | Kofler  |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

```
mysql>
```

# autorbuch DB Tabelle autor ändern und löschen



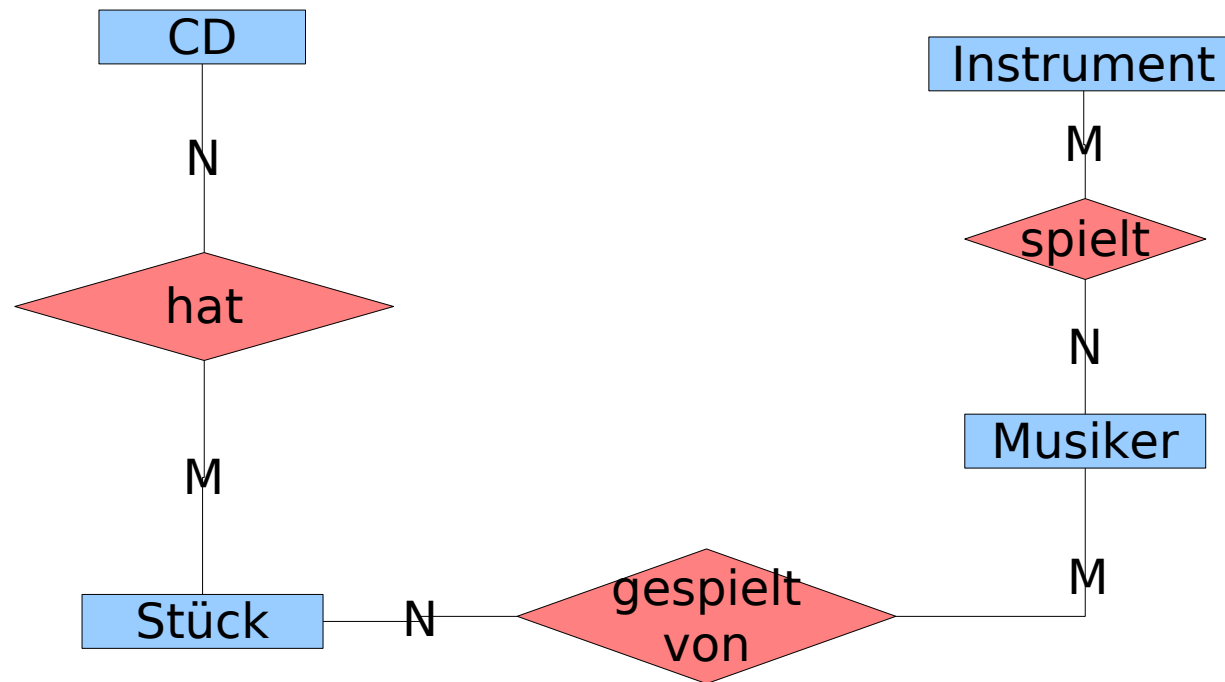
```
mysql> delete from autor where persnr=2;
Query OK, 1 row affected (0.00 sec)
```

```
mysql> select * from autor;
+-----+-----+-----+
| persnr | vorname | name   |
+-----+-----+-----+
|      1 | Thomas  | Kudrass |
+-----+-----+-----+
1 row in set (0.00 sec)
```

```
mysql> drop table autor;
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> drop database autorbuch;
Query OK, 0 rows affected (0.00 sec)
```

# Übungen (basierend auf CD Sammlung)



- 1) Dieses ER-Modell ins relationale Modell umwandeln (Siehe RDB2 S. 4)
- 2) Die CD-Sammlung Datenbank kreieren, Schema anlegen und Daten hinzufügen und abändern.

Arbeiten Sie mit dem MySQL Kommando-Interpreter und mit DBeaver.