

# Tag 2


## Inhaltsverzeichnis

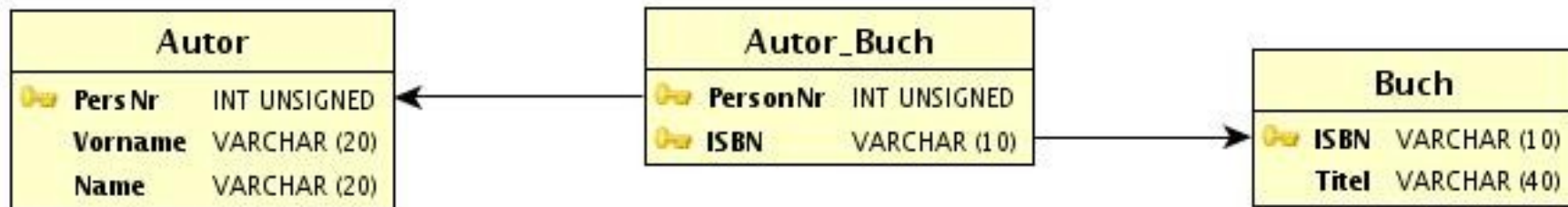
- **Relationales Modell**
  - Beispiel, Pro und Kontra
  - Relation und Schema
  - Umsetzung ER-Modell --> relationales Schema
  - DB Schema: Empfehlungen
- **Typen von Abfragesprachen**
- **SQL Teil 1**
  - Ziel und Geschichte
  - Daten- und Tabellentypen
  - Befehle (Überblick)
  - Administration-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



# Das relationale Modell

## Beispiel

	 PersNr	Vorname	Name
1	12	Alfons	Kemper
2	34	Michael	Kofler
3	56	Gilles	Maitre
4	78	Larry	Wall

	 ISBN	Titel
1	111	VisualBasic 2008
2	123	Datenbanksysteme
3	222	Mathematica
4	456	MySQL 5
5	789	Linux
6	999	DB Administration



	 PersonNr	 ISBN
1	34	111
2	12	123
3	34	222
4	34	456
5	34	789

# Das relationale Modell

## Pro und Kontra

### Pro

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

### Kontra

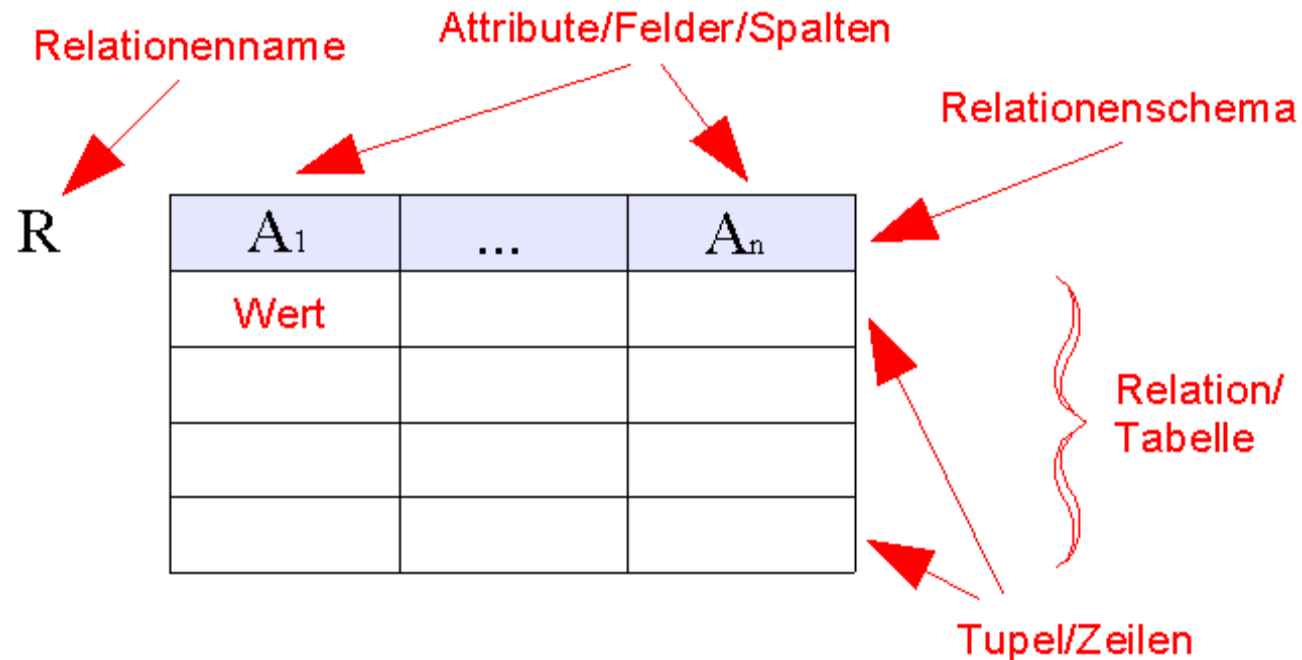
- Künstliche Datensegmentierung
- Rekursion und Vererbung nicht *direkt* modellierbar

Produkte (*Achtung: alle sind verschieden*):  
Oracle, MySQL, PostgreSQL, Ingres, 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
...	...	...



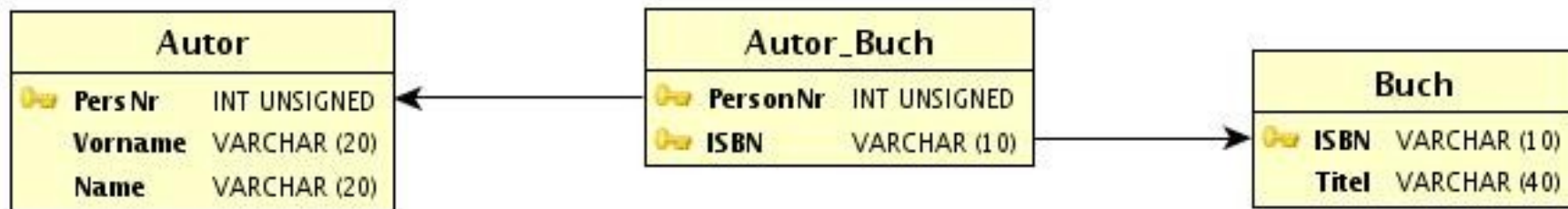
Fremdschlüssel

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**

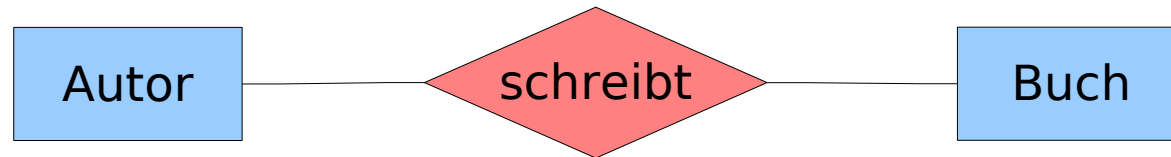
Autor : { **PersNr: int**, Vorname: String, Name: String }

Autor\_Buch : { **PersonNr: int, ISBN: String** }

Buch : { **ISBN: String**, Titel: String }

# Umsetzung ER-Modell in rel. Schema

## Entitäten



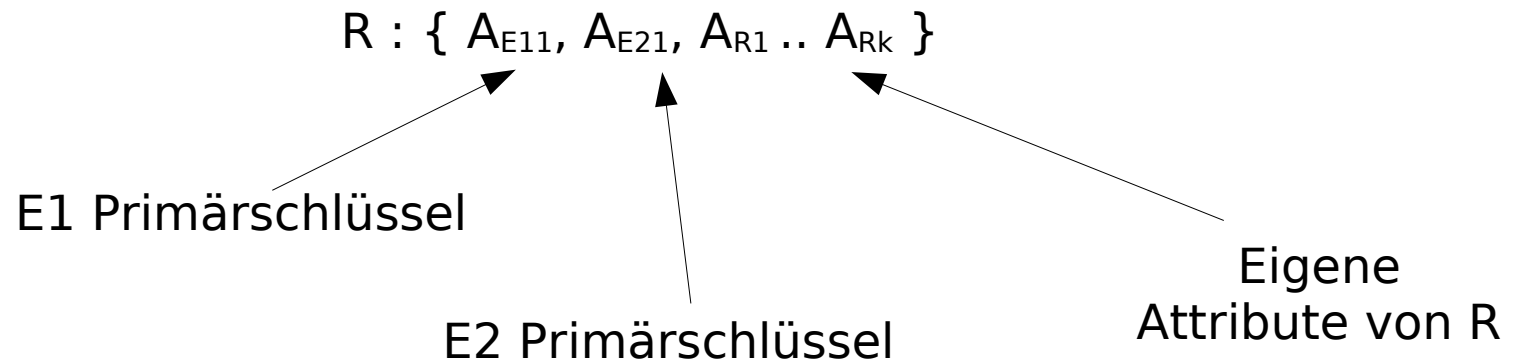
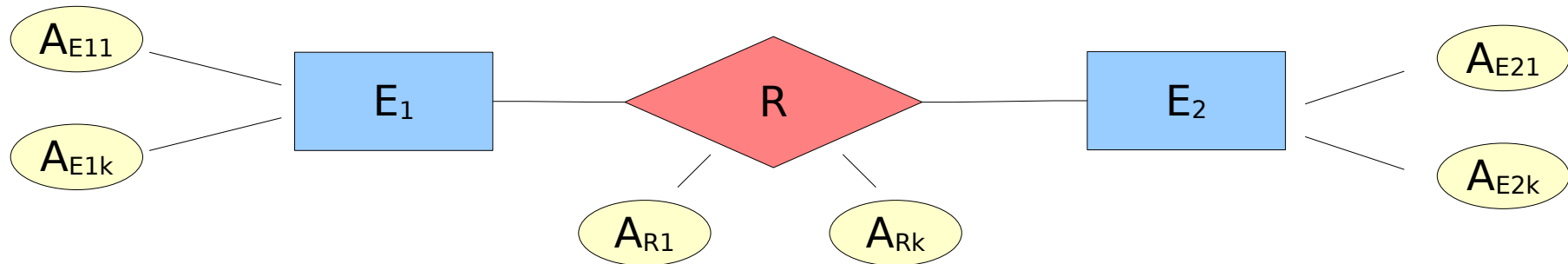
Autor : { PersNr: int, Vorname: String, Name: String }

	PersNr	Vorname	Name
1	12	Alfons	Kemper
2	34	Michael	Kofler
3	56	Gilles	Maitre
4	78	Larry	Wall

Buch : { ISBN: String, Titel: String }

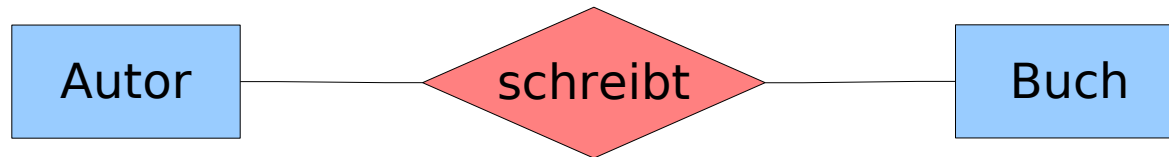
	ISBN	Titel
1	111	VisualBasic 2008
2	123	Datenbanksysteme
3	222	Mathematica
4	456	MySQL 5
5	789	Linux
6	999	DB Administration

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





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



Schreibt : { PersonNr: int, ISBN: String }

	PersonNr	ISBN
1	34	111
2	12	123
3	34	222
4	34	456
5	34	789

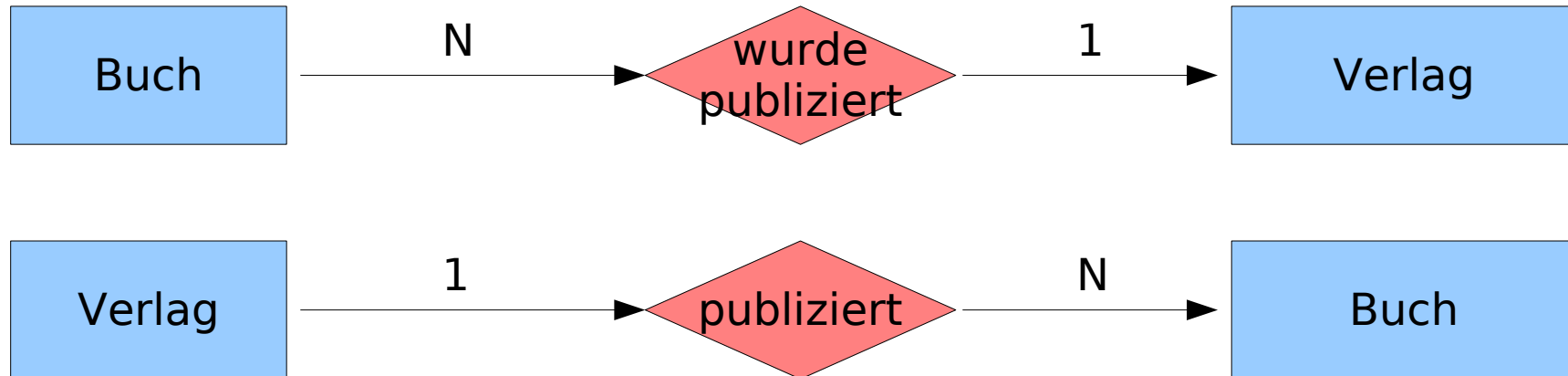
oder eher

*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 }

## Bessere 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
1	12	Alfons	Kemper
2	34	Michael	Kofler
3	56	Gilles	Maitre
4	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  $\{n \mid \exists [n, v] \in Autor \wedge v = Michael\}$

```
SELECT Name FROM Autor WHERE  
Vorname='Michael' ;
```

# SQL

## Das Ziel

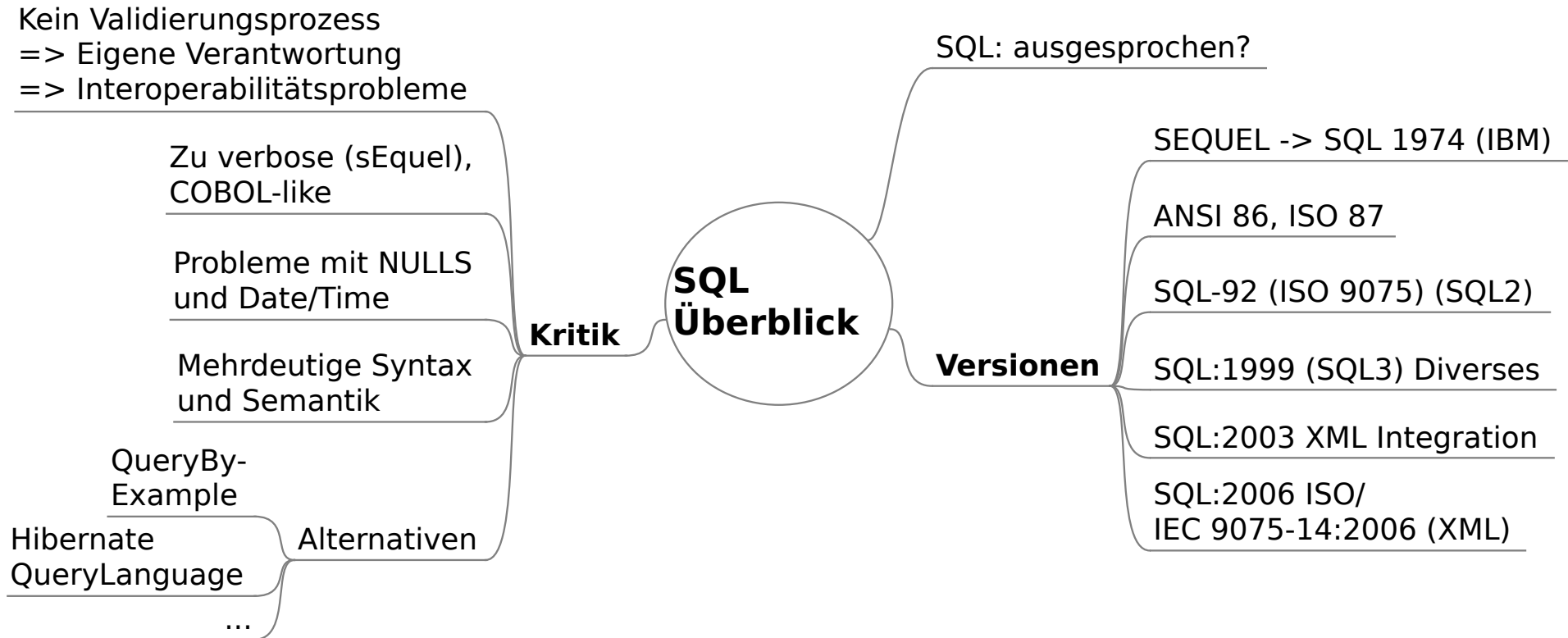
- Standardisierte
- Abfragesprache
  - Definition (DDL)
  - Manipulation (DML)
  - Control (DCL)
- Für **relationale** Datenbanken



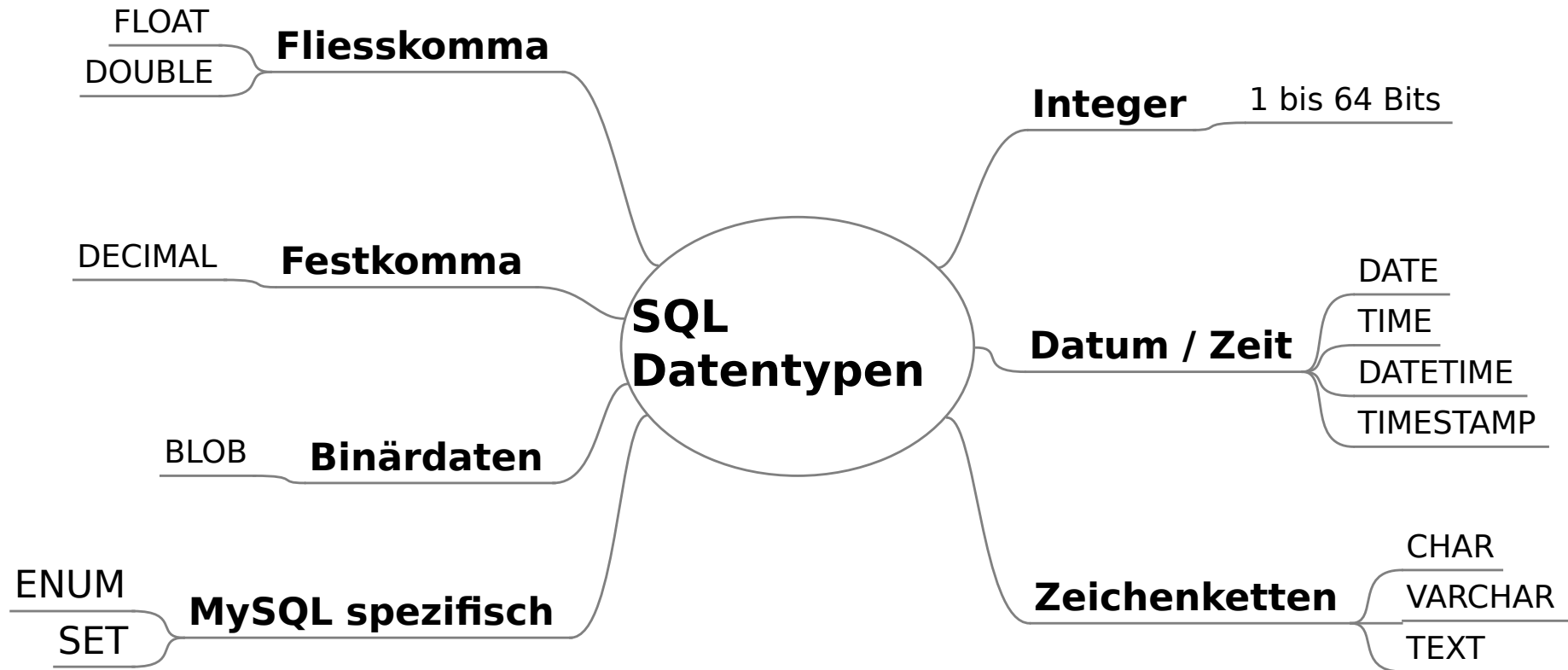
Structured Query Language (SQL)

# SQL

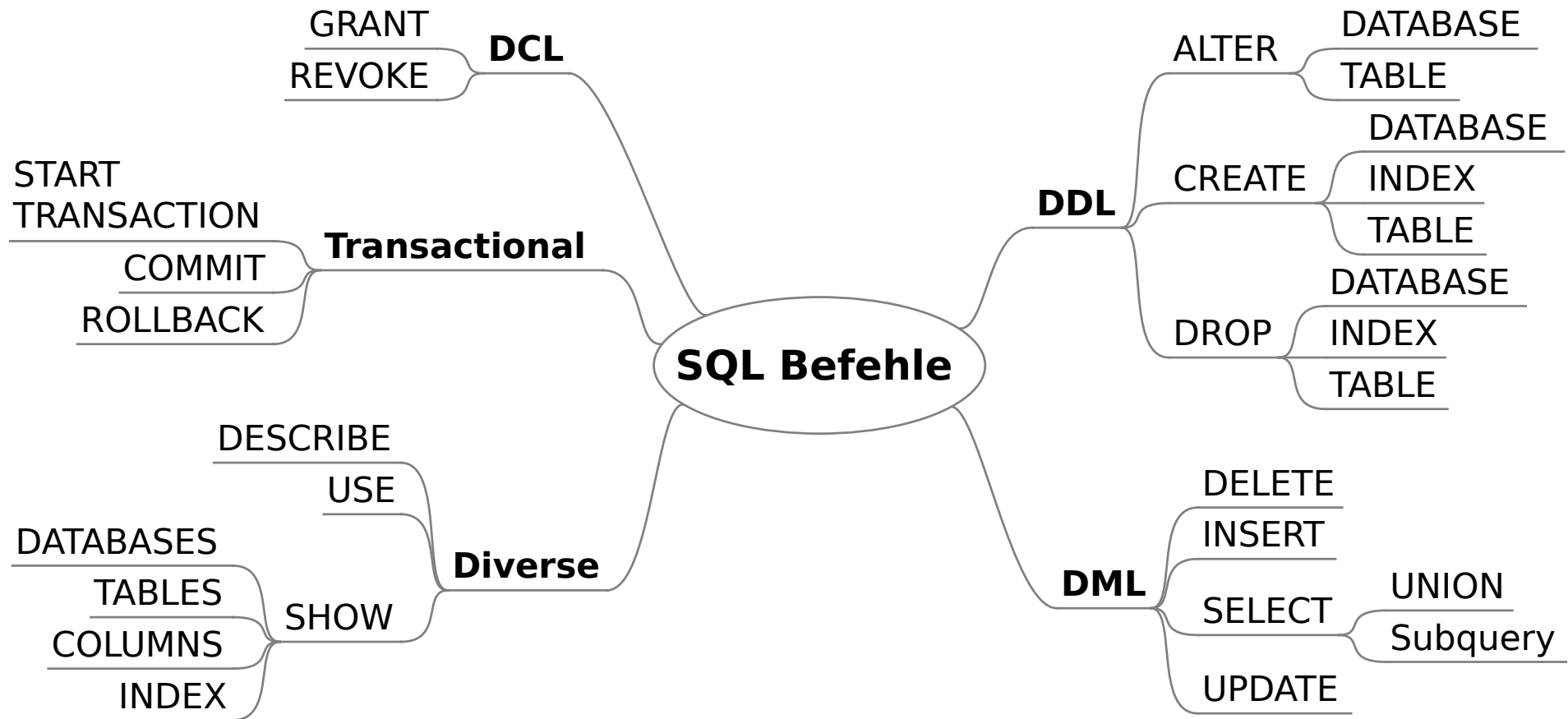
## Die Geschichte



# (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

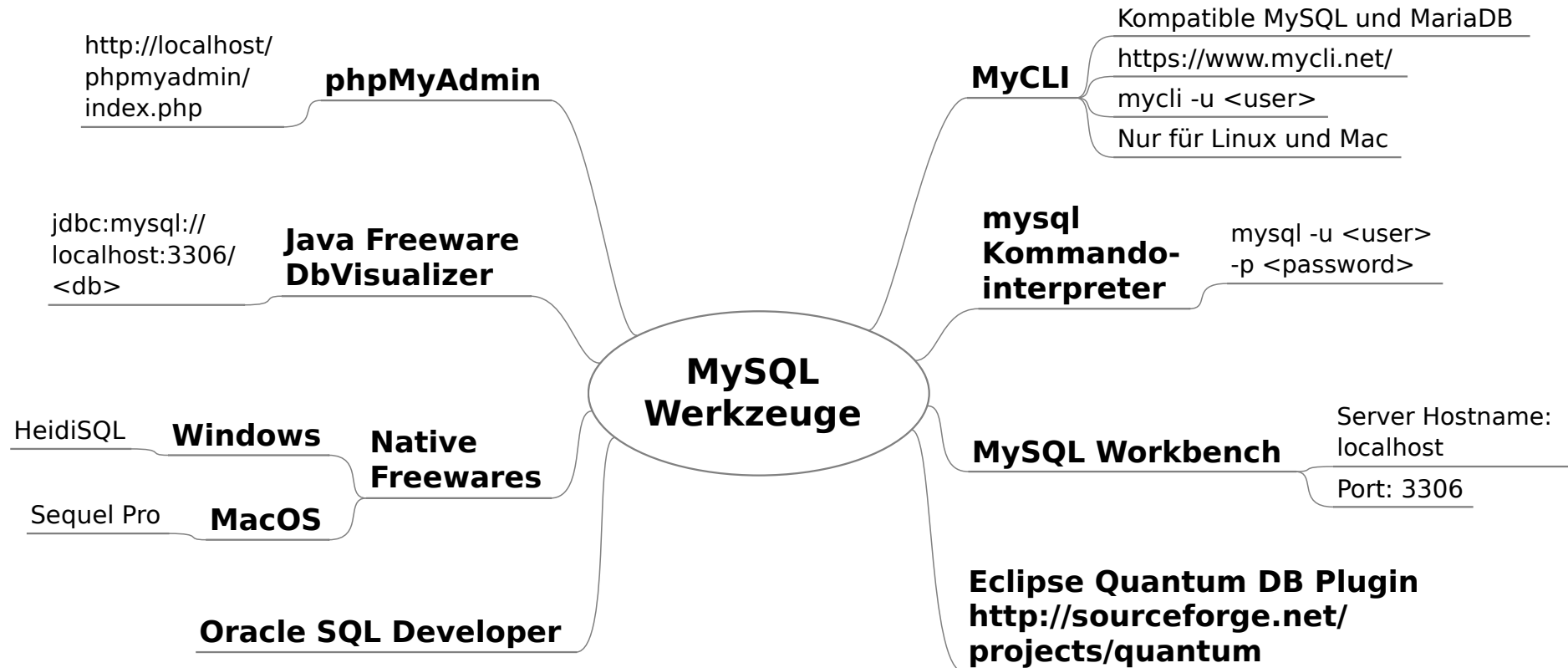
## InnoDB

- 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 dem MySQL Kommando-Interpreter*

# 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
```

```
      ...
```

Default character set: latin1

Default collation: latin1\_swedish\_ci;

Empfohlener character set: utf8

Empfohlener collation: utf8\_unicode\_ci

Wie  
gespeichert?

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

## AutorBuch DB kreieren (2)

```
mysql> create database AutorBuch charset utf8 collate utf8_unicode_ci;  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> show databases;
```

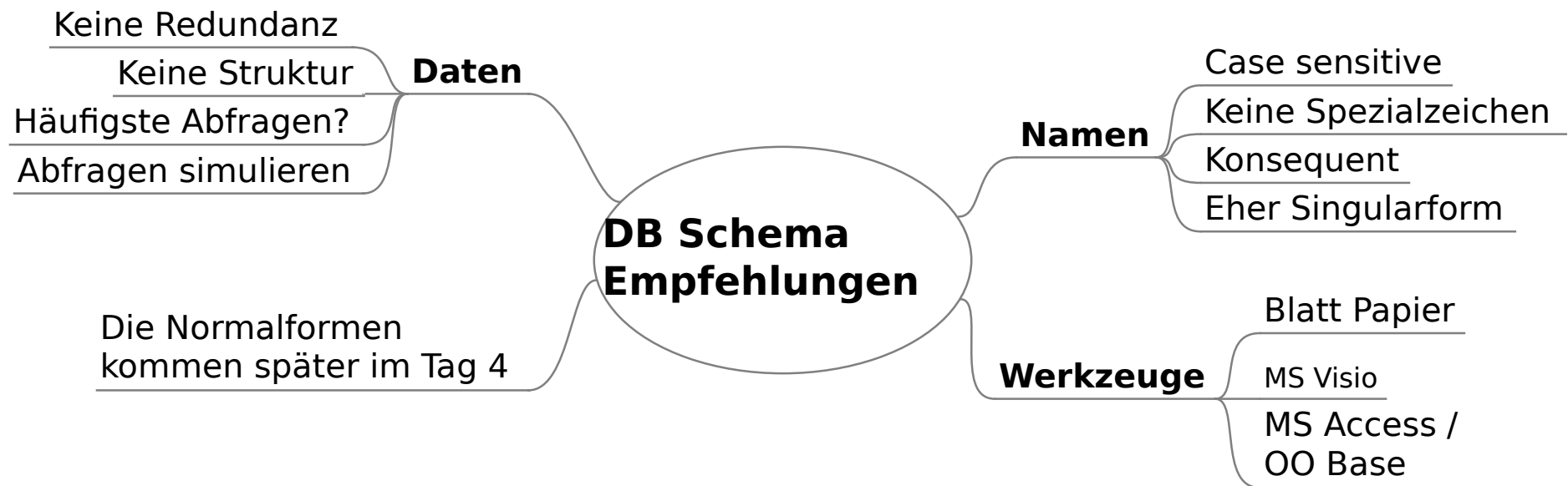
```
+-----+  
| Database |  
+-----+  
| ... |  
| AutorBuch |  
+-----+  
2 rows in set (0.00 sec)
```

```
mysql> use AutorBuch;  
Database changed
```

```
mysql>
```

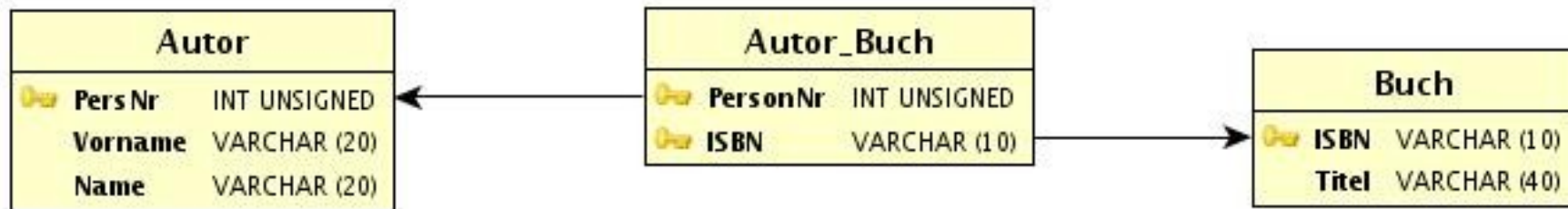
# AutorBuch DB Schema anlegen (1)

## Empfehlungen



# AutorBuch DB Schema anlegen (2)

## Tabelle Autor kreieren



```
mysql> create table `Autor` (  
  `PersNr` int(10) unsigned 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)
```

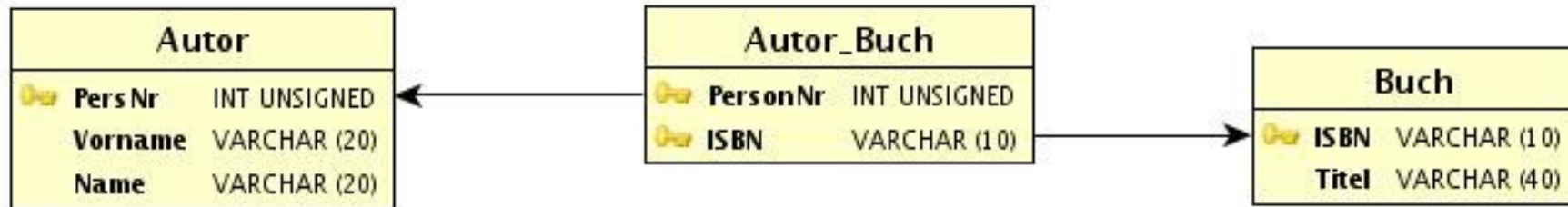
```
mysql> desc Autor;
```

```
+-----+-----+-----+-----+-----+-----+  
| Field      | Type                | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| PersNr     | int(10) unsigned   | 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



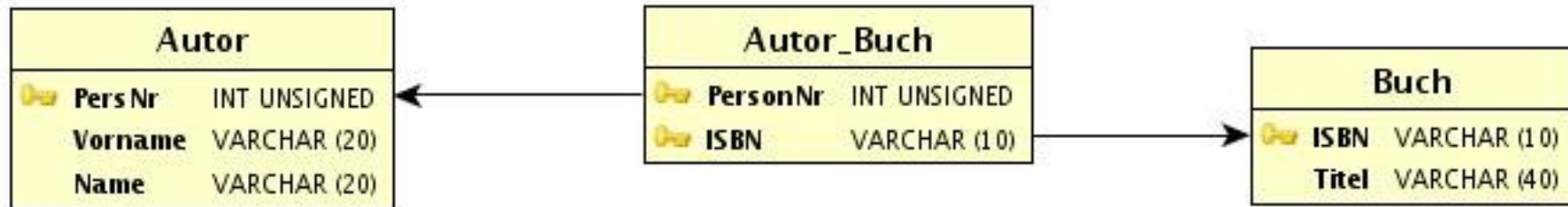
```
mysql> create table `Autor_Buch` (  
  `PersonNr` int(10) unsigned not null,  
  `ISBN` varchar(10) not null,  
  primary key (`PersonNr`, `ISBN`)  
) engine=InnoDB;
```

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)
```

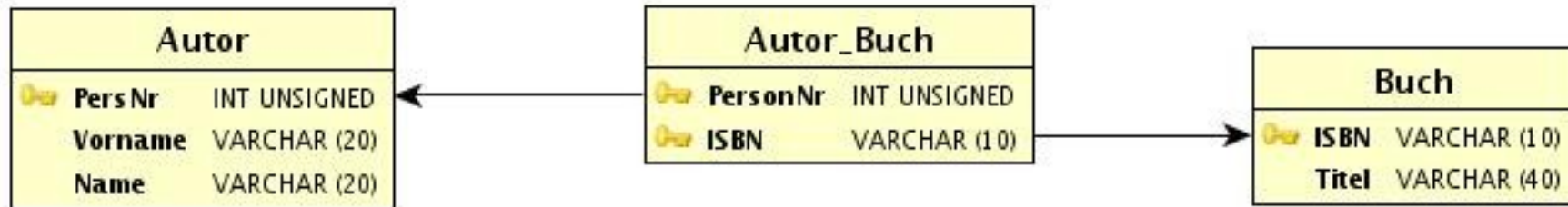
```
mysql> insert into Autor values  
(null, "Michael", "Kofler");
```

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

```
mysql>
```

# AutorBuch DB mit Daten ausfüllen (2)

## Tabelle Autor anzeigen

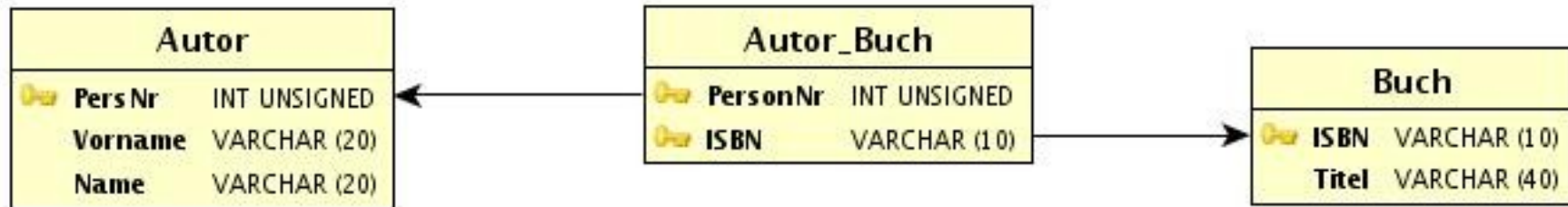


```
mysql> select * from Autor;
```

```
+-----+-----+-----+
| 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
```

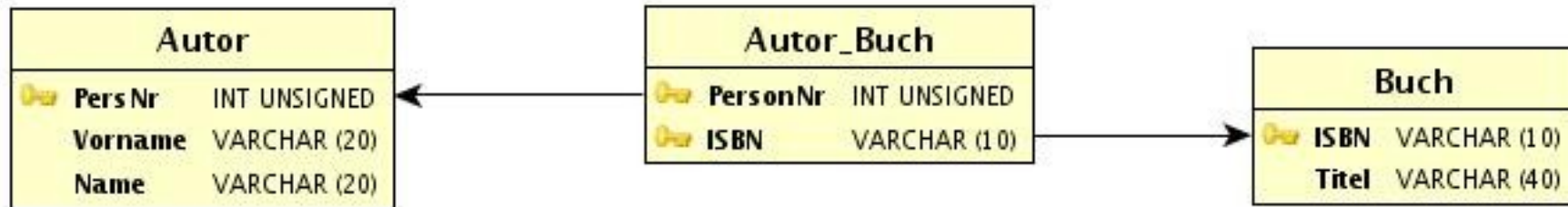
```
mysql> select * from Autor;
```

```
+-----+-----+-----+  
| PersNr | Vorname | Name   |  
+-----+-----+-----+  
|      1 | Thomas  | Kudrass |  
|      2 | Mike    | Kofler  |  
+-----+-----+-----+
```

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

```
mysql>
```

# AutorBuch DB Tabelle Autor ändern und löschen



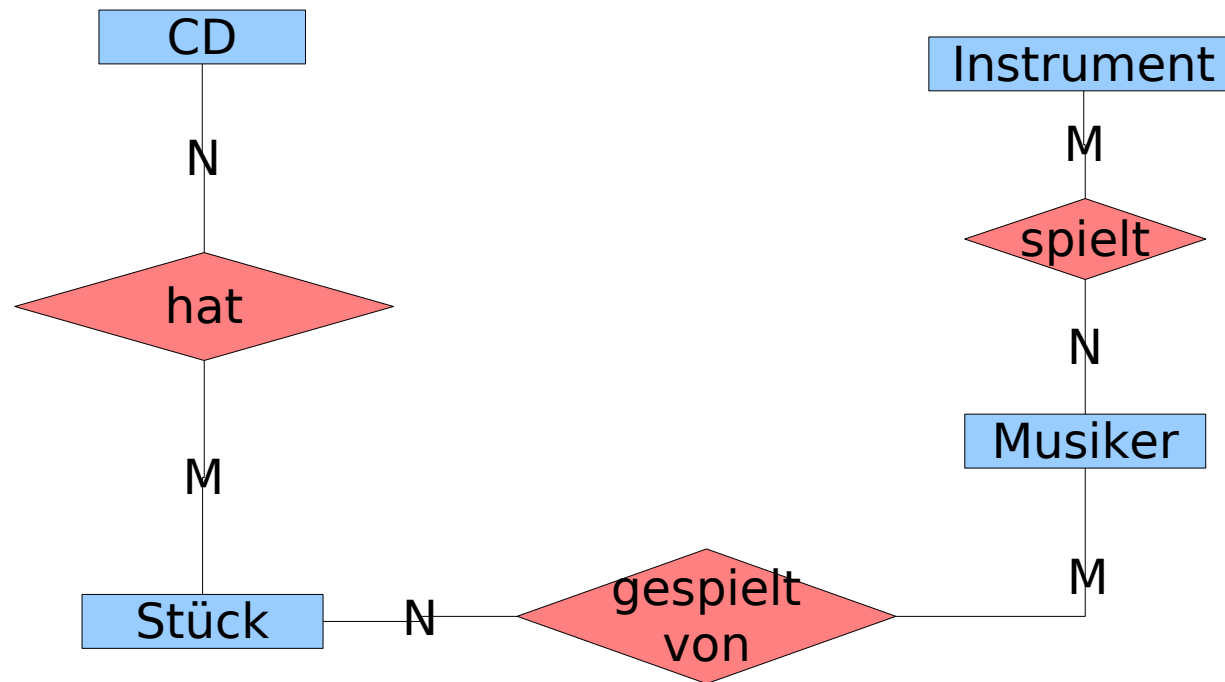
```
mysql> delete from Autor where PersNr=2;  
Query OK, 1 row affected (0.00 sec)
```

```
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 MySQL Workbench.