

# Übungen Datenbanken - Buchverlag

Alle Beispiele anhand des Datenbankmanagementsystems MySQL oder PostgreSQL

## Beispiel Buchverlag

Soweit die Skizze einer Datenbankkonzeption. Verarbeitungsanforderungen (z.B. wie häufig werden bestimmte Anfragen gestellt) werden vorerst nicht gestellt. Wir beginnen mit dem konzeptuellen Entwurf mit dem E<sup>3</sup>R – Modell:

### Konzeptuelle Modellierung

Schauen wir uns die Skizze der Datenbankkonzeption aus der globalen Top-Down-Sicht an. Dann ergeben sich im wesentlichen 3 Entity-Typen:

- Buch
- Buchhandlung
- Person (mit den Spezialisierungen Lektor, Herausgeber, Verfasser)

#### Symbole:



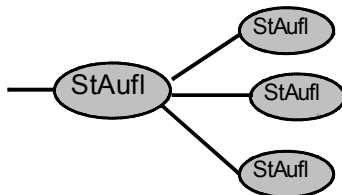
Entity



Attribut



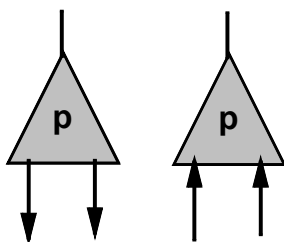
Schlüsselattribut



Zusammengesetztes Attribut

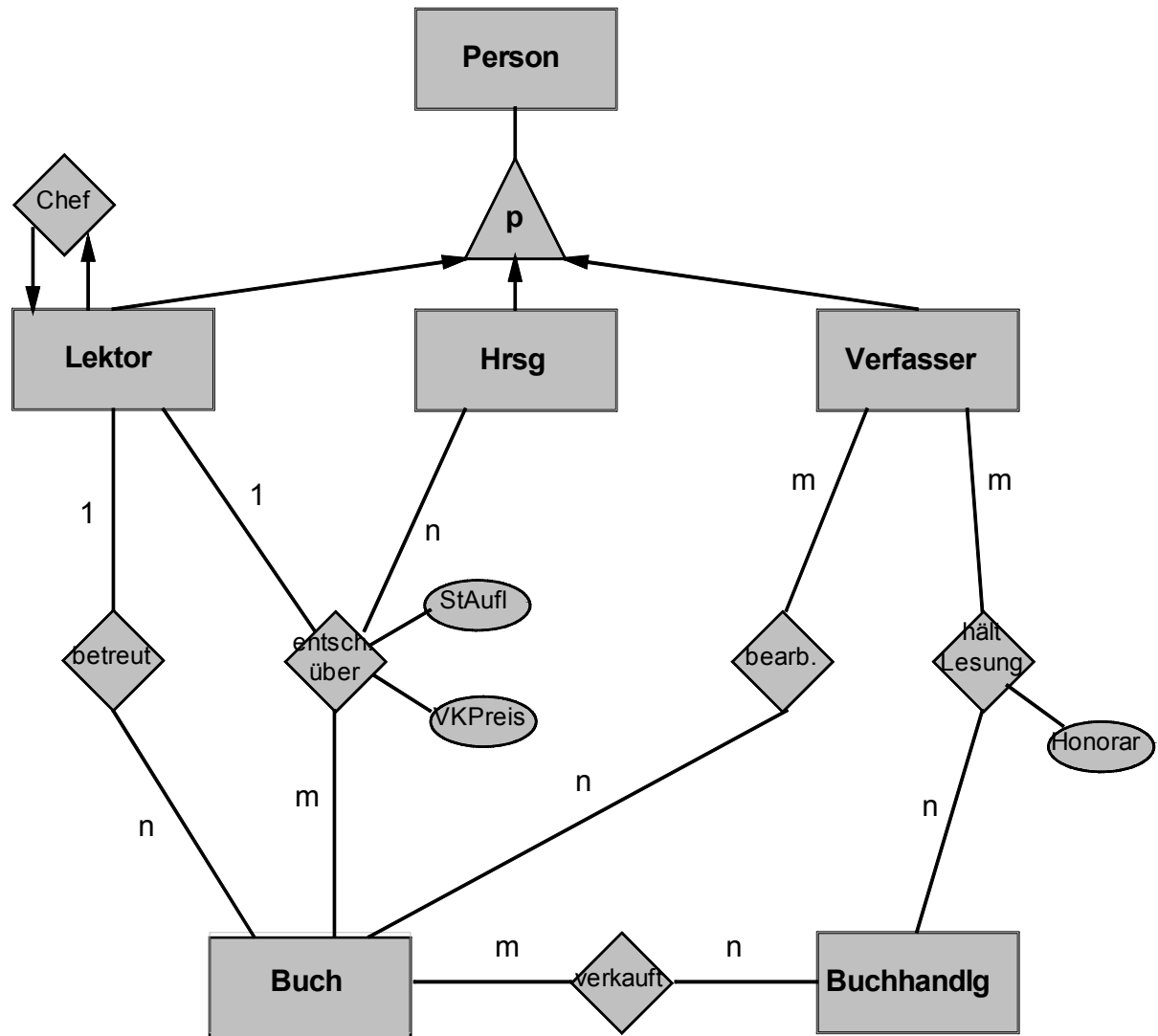


Relationship



IS-A Beziehung, disjunkt / nichtdisjunkt

#### Übersicht Buchverlag:



### Die Entity-Typen und Attribute im Einzelnen:

#### 1. Entity-Typ *Buch*:

Attribute: *ISBN, Titel, Typ, Jahr, Auflage, Fachgebiet*

Schlüssel: *ISBN*

Wertebereichsbedingungen: *Typ {Monographie, Sammelwerk}*  
*1985 ≤ Jahr ≤ 2005*

#### 2. Entity-Typ *Buchhandlung*:

Attribute: *Name, Ort, Besitzer, Einkäufer*

Schlüssel: *Name, Ort*

#### 3. Entity-Typ *Person*:

Attribute: *Name, Adresse (Straße und Ort), Telefonnummer, Kürzel*

Schlüssel: *Kürzel*

### Spezialisierungen des Typs *Person*:

1. Entity-Typ *Lektor*:  
zusätzliche Attribute: *Fachgebiet, Personalnummer, Eintrittsdatum, Gehalt*
2. Entity-Typ *Herausgeber*:  
zusätzliche Attribute: *Arbeitgeber, Fax, Email*
3. Entity-Typ *Verfasser*:  
zusätzliche Attribute: *Akademischer Titel, Anzahl (bereits geschriebener oder editierter) Bücher*

Der Schlüssel in allen drei Spezialisierungen wird von *Person* übernommen. Des weiteren werden folgende Relationships definiert:

1. *betreut* zwischen *Lektor* und *Buch* ( 1:n )
2. *verkauft* zwischen *Buchhandlung* und *Buch* ( m:n)  
Attribute: *Monat, Anzahl*
3. *bearbeitet* zwischen *Verfasser* und *Buch* ( m:n )
4. *entscheidet über* zwischen *Lektor, Herausgeber* und *Buch* ( 1:m:n)  
Attribute: *Startauflage, Verkaufspreis*
5. *ist Chef von* zwischen *Lektor* und *Lektor* (rekursiv)
6. *hält Lesung* zwischen *Verfasser* und *Buchhandlung* (m:n)  
Attribut: *Honorar*

In Abhängigkeit von der Darstellung dieser Relationships auf der Ebene von Attributen kann die Angabe von referentiellen und von Inklusionsbedingungen erforderlich werden. Wird z.B. die Beziehung *verkauft* durch die Schlüsselattribute *Name, Ort* des Typs *Buchhandlung* sowie den Schlüssel *ISBN* des Typs *Buch* dargestellt, so bilden *Name, Ort* einen Fremdschlüssel für *Buchhandlung*, *ISBN* einen solchen in bezug auf *Buch*.

### Logischer Entwurf:

1. Relationenschema *Buch*:  
übernommene Attribute: *ISBN, Titel, Jahr, Auflage, Fachgebiet*  
neues Attribut: *Lektor* (zur Darstellung der Beziehung *betreut* zu *Lektor*; dieses wird Fremdschlüssel bezüglich des Relationenschemas *Lektor*)  
  
Schlüssel: *ISBN*
2. Relationenschema *Buchhandlung*:  
übernommene Attribute: *Name, Ort, Besitzer, Einkäufer*  
Schlüssel: *Name, Ort*
3. Relationenschema *Person*:  
Attribute: *Name, Straße, Ort, Telefon, Kürzel*

Schlüssel: *Kürzel*

4. Relationenschema *Lektor*:

Attribute: *Kürzel, Fachgebiet, Personalnummer, Eintrittsdatum, Gehalt*  
weiteres Attribut: *Chef* (zur Darstellung der rekursiven Beziehung)  
Schlüssel: *Kürzel* (wird Fremdschlüssel bezüglich Schema *Person*)

5. Relationenschema *Herausgeber*:

Attribute: *Kürzel, Arbeitgeber, Fax, Email*  
Schlüssel: *Kürzel* (wird Fremdschlüssel bezüglich Schema *Person*)

6. Relationenschema *Verfasser*:

Attribute: *Kürzel, AkadTitel, AnzBücher*  
Schlüssel: *Kürzel* (wird Fremdschlüssel bezüglich Schema *Person*)

7. Relationenschema *Verkauf*:

Attribute: *Name, Ort, ISBN, Monat; Anzahl*  
Schlüssel: *Name, Ort, ISBN, Monat*  
Fremdschlüssel: *Name, Ort* bezüglich *Buchhandlung*  
Fremdschlüssel: *ISBN* bezüglich *Buch*

8. Relationenschema *Bearbeitet*:

Attribute: *Kürzel, ISBN*  
Schlüssel: *Kürzel, ISBN*  
Fremdschlüssel: *Kürzel* bezüglich Schema *Verfasser*  
Fremdschlüssel: *ISBN* bezüglich Schema *Buch*

9. Relationenschema *Entscheid*:

Attribute: *KürzelL* (eines Lektors), *KürzelH* (eines Herausgebers), *ISBN, Startauflage, Verkaufspreis (VK)*  
Schlüssel: *KürzelH, ISBN*  
Fremdschlüssel: bezüglich der Relationenschemata *Lektor, Herausgeber, Buch*

10. Relationenschema *Lesung*:

Attribute: *Autor, Buchshop* (Name einer Buchhandlung), *Ort, Honorar*  
Schlüssel: *Autor, Buchshop, Ort*  
Fremdschlüssel: bezüglich der Relationenschemata *Verfasser, Buchhandlung*

## SQL-Implementierung

Es folgt eine Deklaration wesentlicher Teile des konzeptionellen Schemas. Dabei werden skalare Datentypen verwendet.

```

create table Buch
  (ISBN smallint not null,
   Titel varchar(40) not null,
   Typ char(1) not null check (Typ in ('M', 'S')),
   Jahr smallint not null
     check (Jahr between 1982 and 2010),
   Auflage smallint,
   Fachgebiet varchar(35),
   Lektor char(3) not null,
   primary key (ISBN),
   foreign key (Lektor) references Lektor);

create table Buchhandlung
  (Name varchar(25) not null,
   Ort varchar(25) not null,
   Besitzer varchar(20) not null,
   Einkaufspreis varchar(2),
   primary key (Name, Ort));

create table Verkauf
  (Name varchar(25) not null,
   Ort varchar(25) not null,
   ISBN smallint not null,
   Monat smallint not null,
   Anzahl integer,
   primary key (Name, Ort, ISBN, Monat),
   foreign key (Name, Ort) references Buchhandlung,
   foreign key (ISBN) references Buch);

```

**Tabelle Person sowie zwei Spezialisierungen dieser Tabelle:**

```

create table Person
  (Name varchar(25) not null,
   Kuerzel char(3) not null,
   Strasse varchar(30),
   Ort varchar(25),
   Telefon integer not null,
   primary key (Kuerzel));

create table Lektor
  (Kuerzel char(3) not null,
   Fachgebiet varchar(35) not null,
   PersNr smallint not null unique,
   Eintritt date not null,
   Gehalt smallint check (Gehalt > 25000),
   Chef char(3)
     check (Chef in (select Kuerzel from Lektor)),
   primary key (Kuerzel),
   foreign key (Kuerzel) references Person);

create table Herausgeber
  (Kuerzel char(3) not null,
   Arbeitgeber varchar(30),
   Email varchar(30) not null,
   Fax integer not null,

```

```
primary key (Kuerzel),  
foreign key (Kuerzel) references Person);
```

Als letztes Beispiel sei die Deklaration des Relationenschemas zur Beschreibung der Beziehung Entscheid angegeben:

```
create table Entscheid  
(Kuerzell char(3) not null,  
KuerzelH char(3) not null,  
ISBN smallint not null,  
Startaufl smallint not null,  
VK dec(3,2) not null,  
primary key (KuerzelH, ISBN),  
foreign key (Kuerzell) references Lektor,  
foreign key (KuerzelH) references Herausgeber,  
foreign key (ISBN) references Buch);
```

### **Aufgabe:**

Vervollständigen Sie die Fallstudie Buchverlag. Insbesondere geben Sie Tabellendeklarationen für folgende Tabellen an:

- *Verfasser*
- *Bearbeitet*
- *Lesung*