

Programmierung von Datenbanken (PDB)

Vorlesung 2

Teil:PDB-1



Letzte Vorlesung

- Grundbegriffe
- SQL
 - create table
 - insert
 - select



Heute Übersicht

- Modellierung (ER-Diagramme)
 - Entitäten und Entity-Typen
 - Beziehungen und Beziehungstypen
 - Attribute
 - für Entitäten(Typen) und Beziehungen(Typen)
 - einfacher oder zusammengesetzter Schlüssel
 - Kardinalitäten
 - Generalisierung
 - Business Rules
- ... mehr zu SQL



Erstellung einer Datenbank: Erster Schritt ...

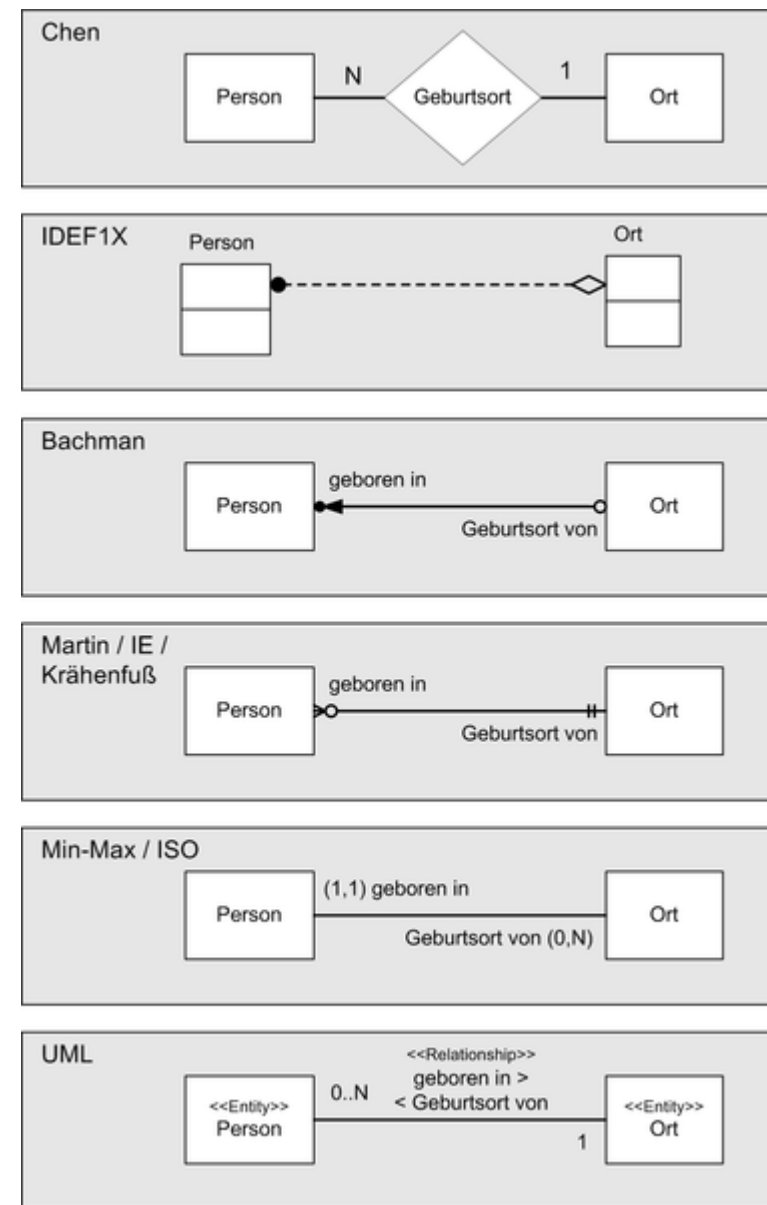
- Was sind die Anforderungen?
- Was sind die Ziele?
- Ist bekannt was gespeichert werden soll?

→ **Design** des *Datenmodells*



Entity-Relationship-Modell (ER-Modell)

- Siehe auch: C. Maria Keet: *A formal comparison of conceptual datamodeling languages*
(<http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-337/paper3.pdf>)



Picture by: Frank Roeing



ER-Modell

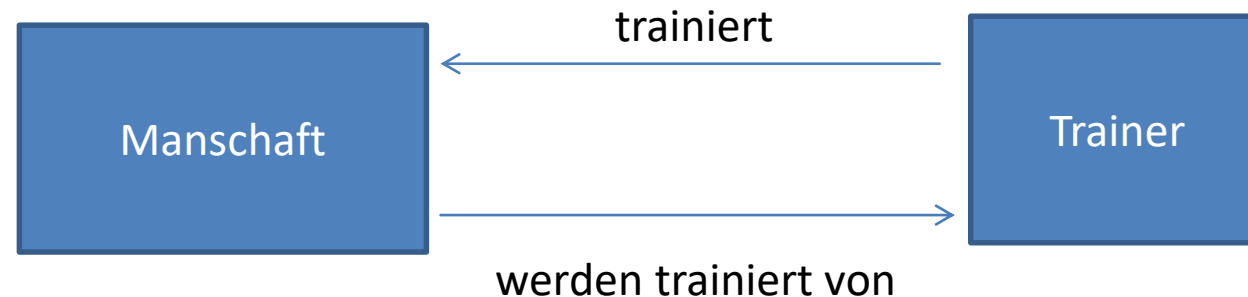
Ziel: Abbildung der Wirklichkeit auf Strukturebene!

Beispiel:

– Instanzebene



– Strukturebene

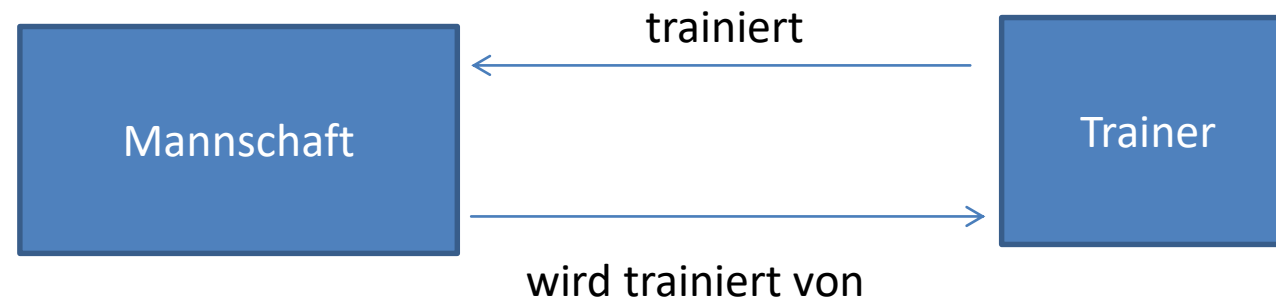


Toni Söderholm

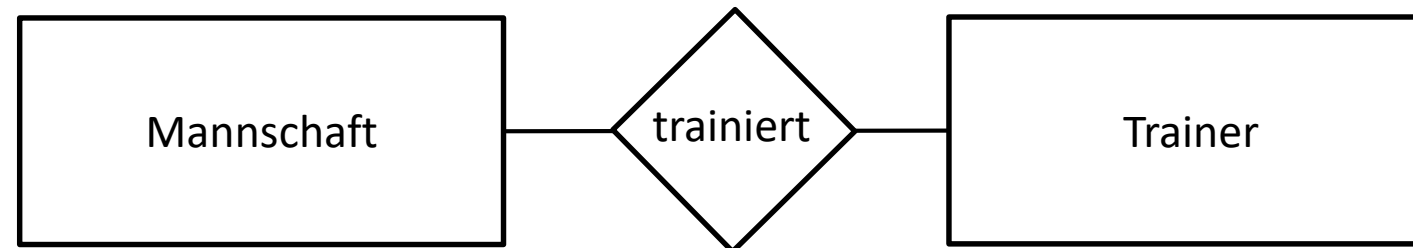


ER-Modell

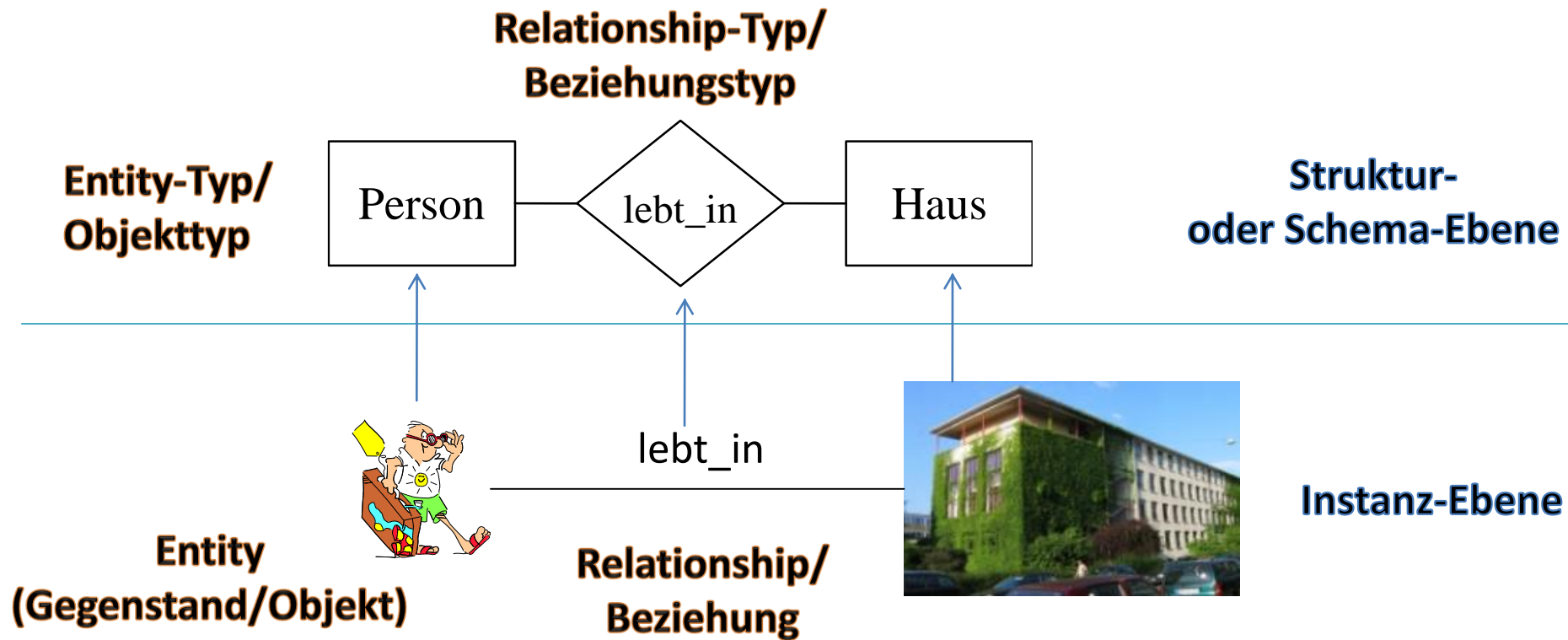
– Strukturebene



– bei uns (PDB)

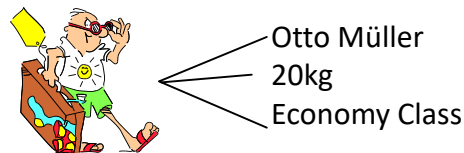
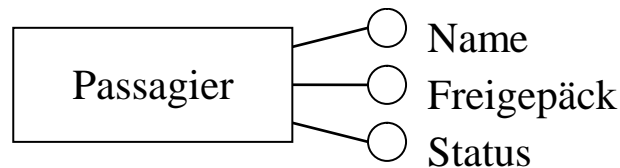


- Otto Müller lebt in Frankfurt am Main, in der Robert-Mayer-Str. 11.



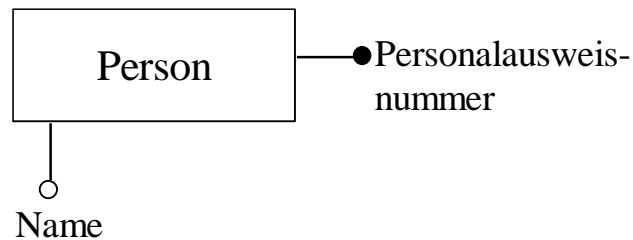
(Objekttyp) Attribute

- Ein Objekttyp ist durch einen bestimmten Satz von Merkmalen (Attributen) gekennzeichnet.
- Jedes Merkmal kann Werte (values), das sind in der Umwelt beobachtbare oder messbare Größen, aus einem bestimmten Wertebereich (value set) annehmen.
- Beispiel:

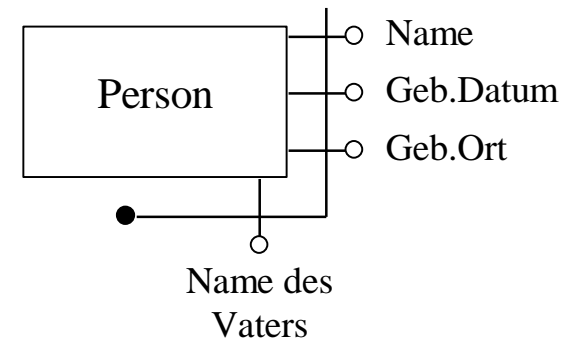


Schlüssel

Ein ***Schlüssel*** besteht aus einer Menge von Attributen, deren Werte eine Instanz (Entity) eines Objekttyps eindeutig bestimmt.



einfacher Schlüssel



zusammengesetzter Schlüssel



Prinzipien des digitalen Speicherns

Wenn möglich sollten vorhandene Identifizierungsmerkmale als Schlüssel wiederverwendet werden.

Key

<u>ISBN</u>	TITLE	
3-12-517154-7		
...		



ISBN Nummer

Man muss aber auch mögliche Änderungen sinnvoll handhaben:
ISBN 10 → ISBN 13



... Stifte mit gleichem Barcode ...



Die Universitätsbibliothek Johann Christian Senckenberg ist g
Bitte beachten Sie aktuelle Informationen unter <https://www.ub.u>

kemper, alfons

SUCHEN

[Erweiterte Suche](#)

Ihre Suchbegriffe : Einfache Suche: (Alle Felder: kemper, alfons)

BÜCHER & MEHR (43)

ARTIKEL & MEHR (1820)

IHR KONTO

[« zurück zum Suchergebnis](#)

[« zurück 4 von 43 weiter »](#)

Datenbanksysteme

Titel: [Datenbanksysteme](#) : eine Einführung / Alfons Kemper ; André Eickler

Verfasser: [Kemper, Alfons W](#); [Eickler, André](#)

Ausgabe: 10., aktualisierte und erw. Aufl.

Veröffentlicht: Berlin [u.a.] : De Gruyter Oldenbourg, 2015

Umfang: 879 S. : Ill., graph. Darst. ; 24 cm x 17 cm

Format:  Buch

Sprache: Deutsch

Schriftenreihe/
mehrbändiges
Werk: De Gruyter Studium

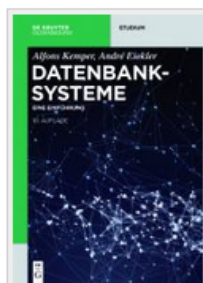
RVK-Notation: [ST 270](#) [INFO](#)

Schlagworte: [Datenbanksystem](#) [W](#)

ISBN: 3110443759 ; 9783110443752

Hinweise zum
Inhalt: [Hinweise zum Inhalt](#)

[Inhaltsverzeichnis](#)



EXEMPLARE

MEHR INFORMATIONEN

INHALTSVERZEICHNIS

REZENSIONEN


AUSSCHNITT

ZUSAMMENFASSUNG

Universitätsbibliothek J. C. Senckenberg, Zentralbibliothek (ZB)

Signatur: 90.787.58

Kommentar: verfuegbar - UB geschlossen: derzeit keine Ausleihe


Status:  [ausleihbar](#)

Universitätsbibliothek J. C. Senckenberg, Informatikbibliothek

Signatur: H.0-3 ex 15

Klassifikation: [H.0](#)


Kommentar: verfuegbar - UB geschlossen: derzeit keine Ausleihe

Status:  [ausleihbar](#)

Signatur: H.0-3 ex 16

Klassifikation: [H.0](#)


Kommentar: Der Band ist ausgeliehen. Vormerken ist derzeit leider nicht moeglich.

Status:  [ausleihbar](#)

Signatur: H.0-3 ex 17

Klassifikation: [H.0](#)

Kommentar: verfuegbar - UB geschlossen: derzeit keine Ausleihe

Status:  [ausleihbar](#)



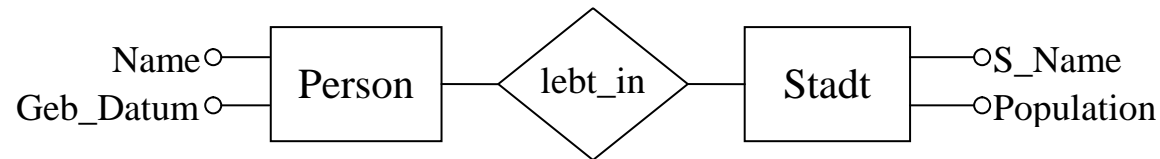
Signatur:

LB: Xd 244 <10> (1.-5. Ex.) **Standort:** Lehrbuchsammlung Informatik



Mathematische Betrachtung

Ein Beziehungstyp zwischen zwei Objekttypen kann als eine **mathematische Relation** aufgefasst werden.



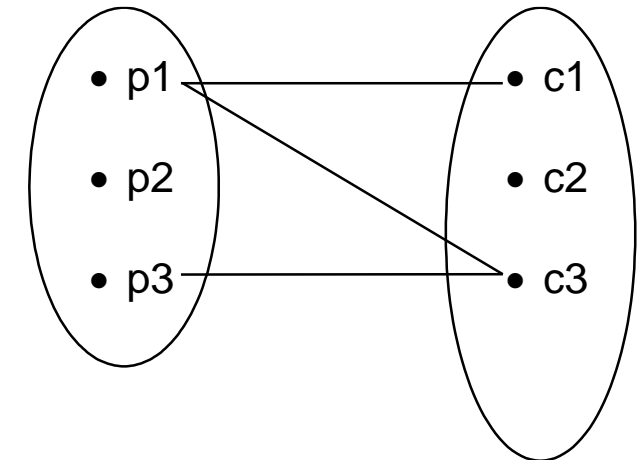
Beispiel:

Person = { p1, p2, p3 }

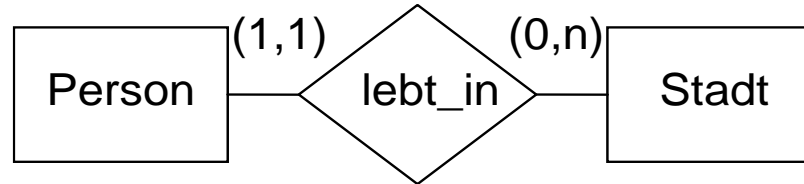
Stadt = { c1, c2, c3 }

lebt_in = { <p1,c1>, <p1,c3>, <p3,c3>, ~~<p2,c3>~~ }

lebt_in = { <p1,c1>, <p1,c3>, <p3,c3> }
als **Mengendiagramm**

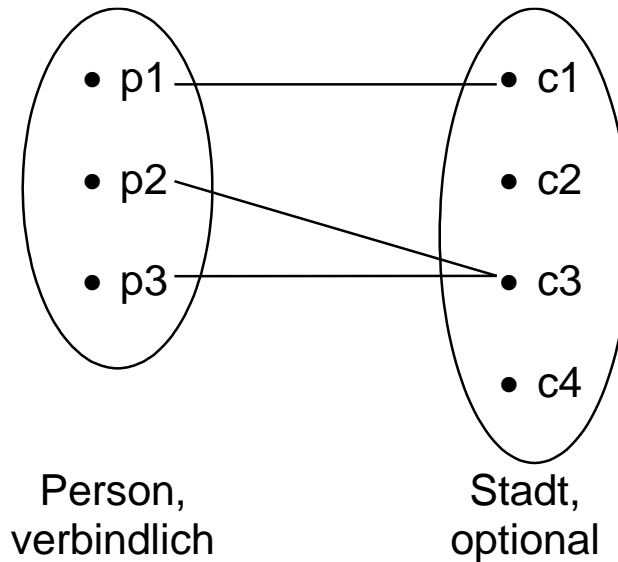


Min/Max Kardinalitäten



- $\text{min_card}(\text{Person}, \text{Lebt_in}) = 1$
- $\text{max_card}(\text{Person}, \text{Lebt_in}) = 1$
- $\text{min_card}(\text{Stadt}, \text{Lebt_in}) = 0$
- $\text{max_card}(\text{Stadt}, \text{Lebt_in}) = n$

$\text{lebt_in} = \{ \langle p1, c1 \rangle, \langle p2, c3 \rangle, \langle p3, c3 \rangle \}$
als **Mengendiagramm**



Es gilt immer: $\text{min_card} \leq \text{max_card}$!

Bem.: Es gibt andere Notationen, z.B. wird manchmal nur max_card angegeben.



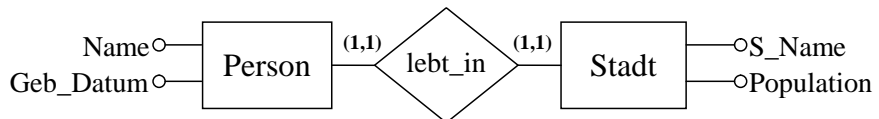
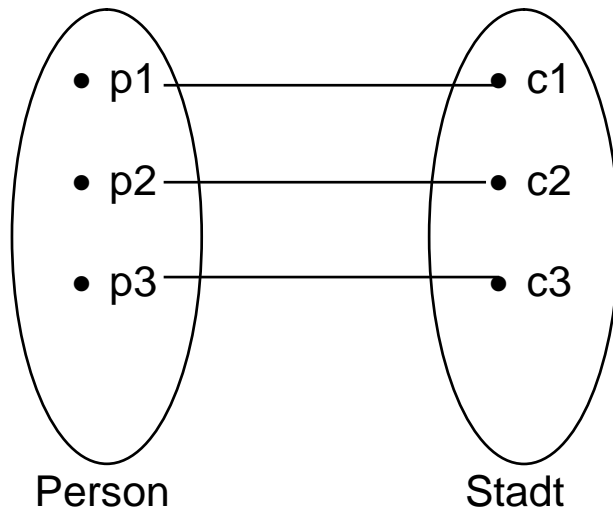
Kardinalitäten

Instanz:

Person = { p1, p2, p3 }

Stadt = { c1, c2, c3 }

lebt_in = { <p1,c1>, <p2,c2>, <p3,c3> }

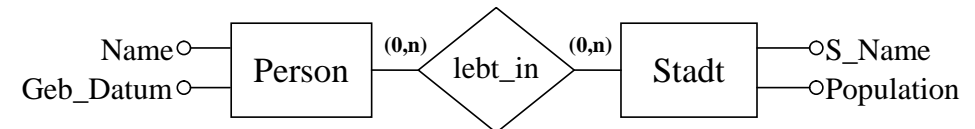
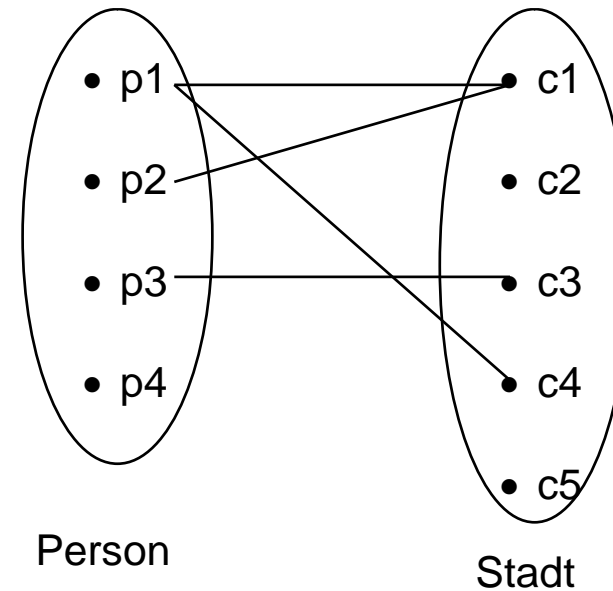


Instanz:

Person = { p1, p2, p3, p4 }

Stadt = { c1, c2, c3, c4, c5 }

lebt_in = { <p1,c1>, <p2,c1>, <p3,c3>, <p1, c4> }

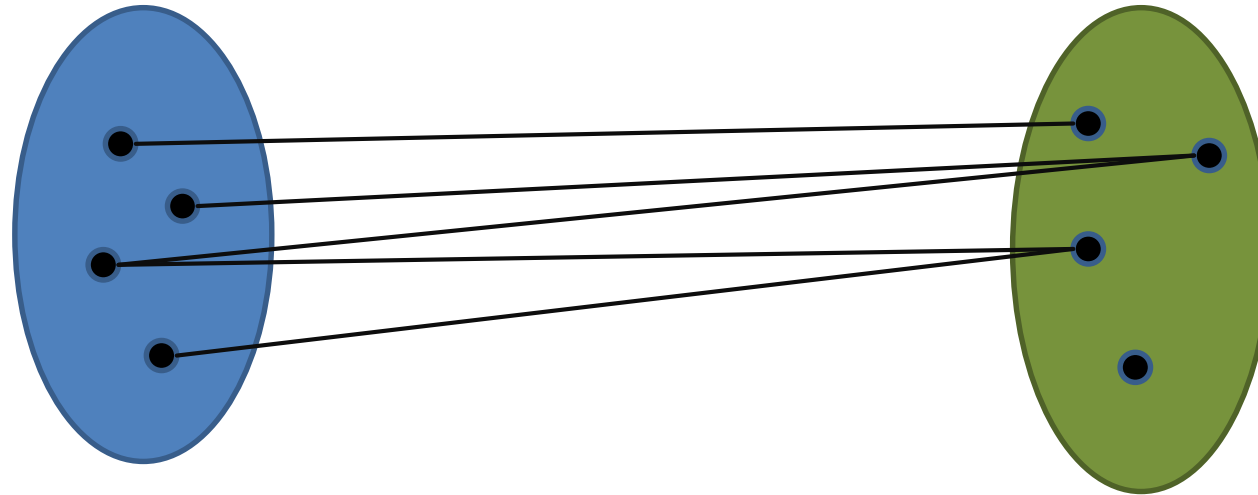
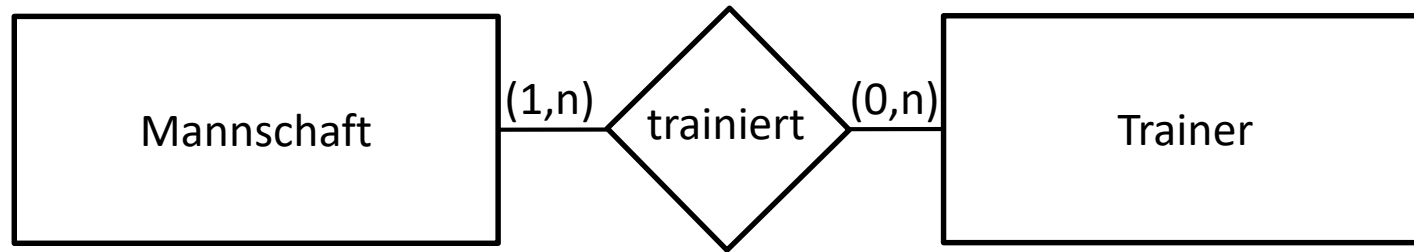


Übung 1

- Aussage:
 - Mannschaften werden von mindestens einem Trainer trainiert.



Mannschaften werden von mindestens einem Trainer trainiert.



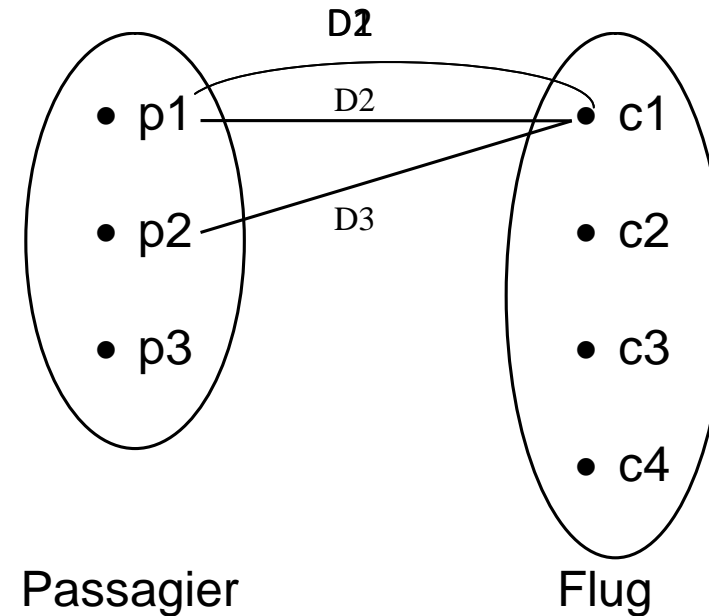
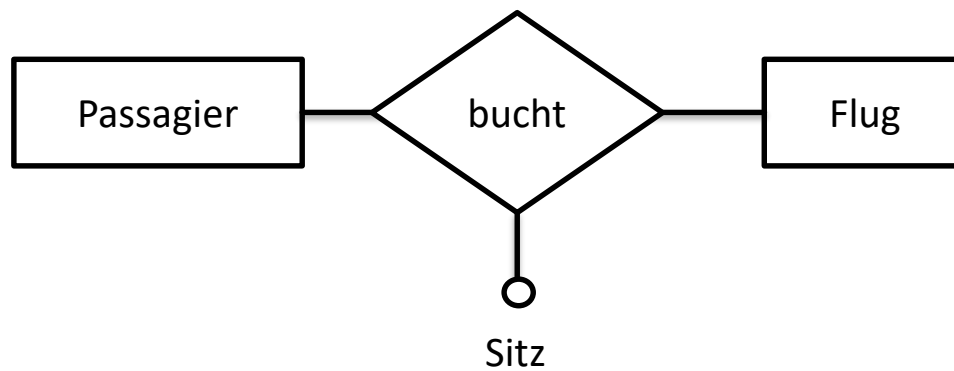
(Beziehungs) Attribute

Instanz:

Passagier = { p1, p2, p3 }

Flug = { c1, c2, c3 }

bucht = { <p1,c1, „D2“>, <p2,c1, „D3“> }

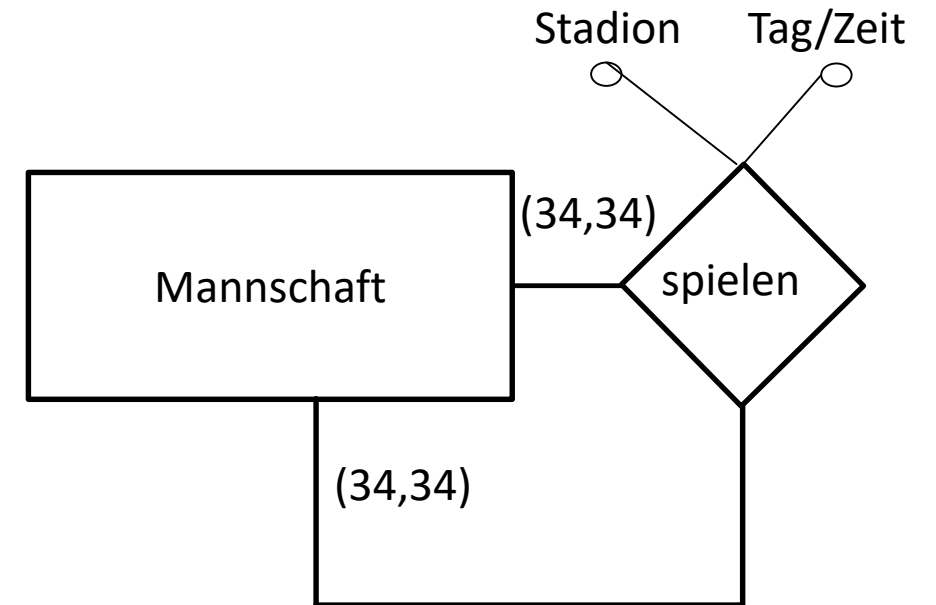
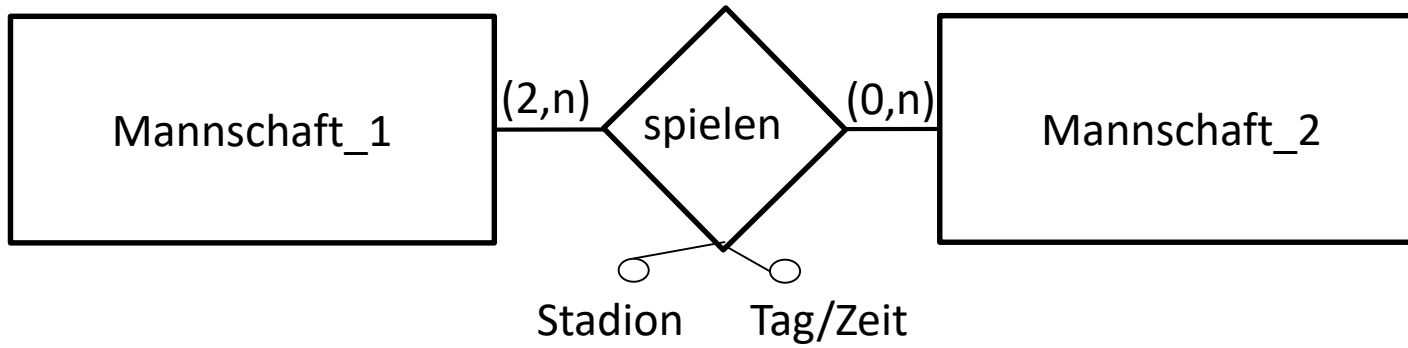


Übung 2 – ER-Diagr. für folgende Aussage erstellen

- Aussage:
 - Bei einem Spiel spielen zwei Mannschaften an einem bestimmten Tag und Uhrzeit in einem Stadion gegeneinander.



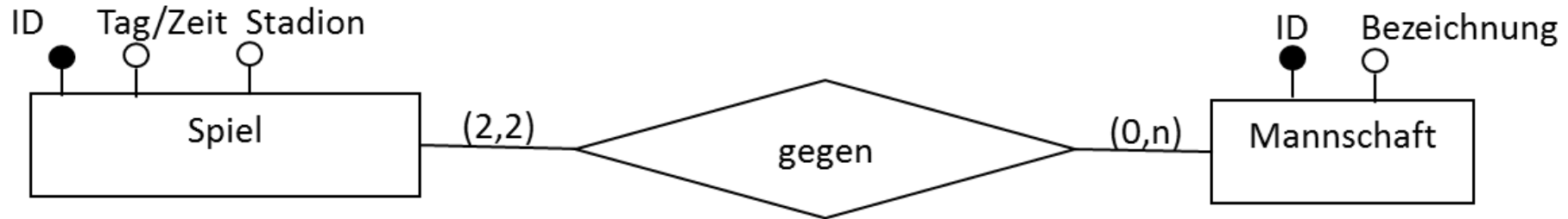
Häufige Lösungsansätze ...



Übung 2 – ER-Diagr. für folgende Aussage erstellen

- Aussage:
 - Bei einem Spiel spielen zwei Mannschaften an einem bestimmten Tag und Uhrzeit in einem Stadion gegeneinander.

... eine weitere Möglichkeit!



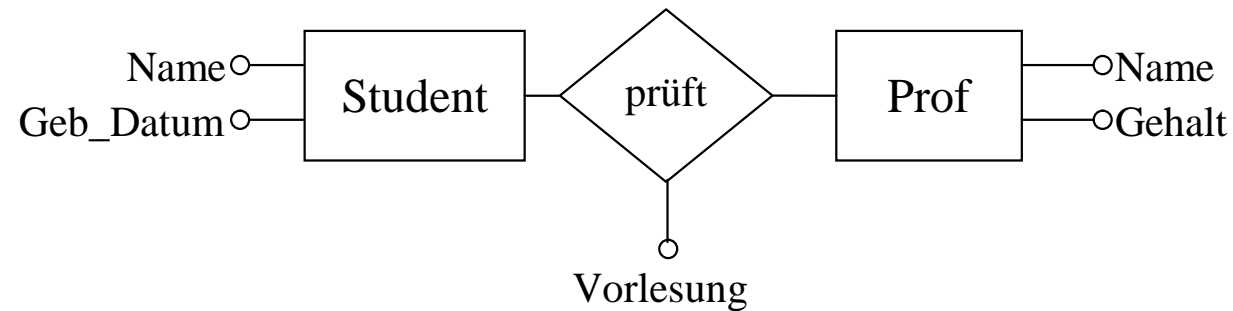
Bem.: So wird verhindert, dass eine Mannschaft gegen sich selbst spielt.
Wenn man bei Spiel einen zusammengesetzten Schlüssel wählt (Tag/Zeit/Stadion)
kann man auch modellieren, dass nicht zwei Spiele gleichzeitig im gleichen Stadion stattfinden.



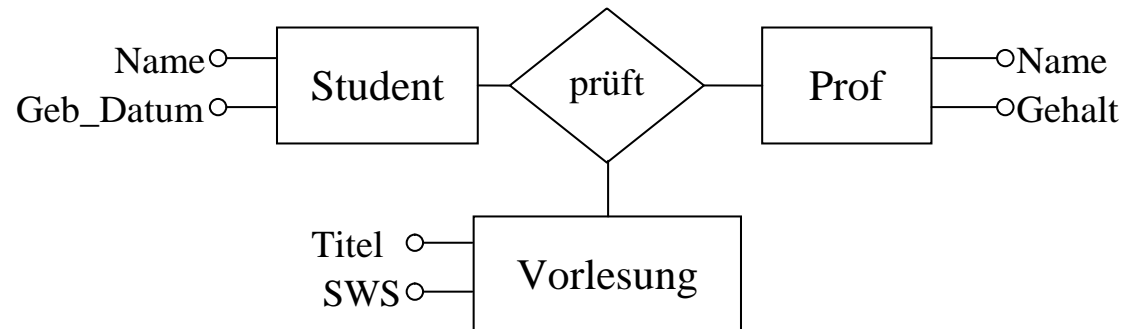
Die Uni ...

Studenten können sich von Professoren über eine Vorlesung mündlich prüfen lassen.

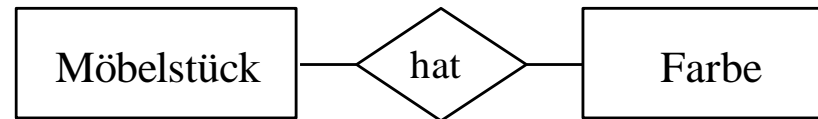
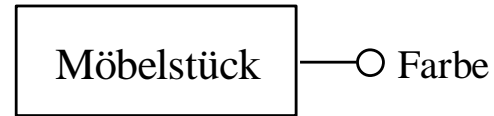
Alt. 1:



Alt. 2:
(N-näre Beziehung)

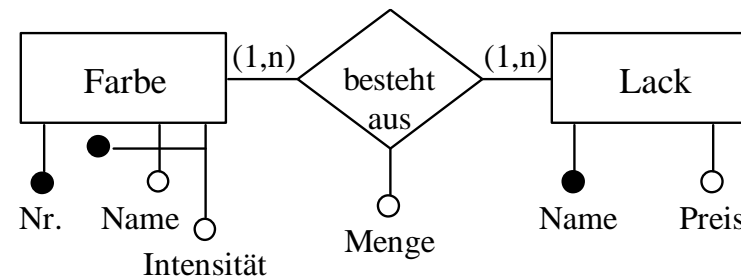


Entity-Typ oder Attribut???



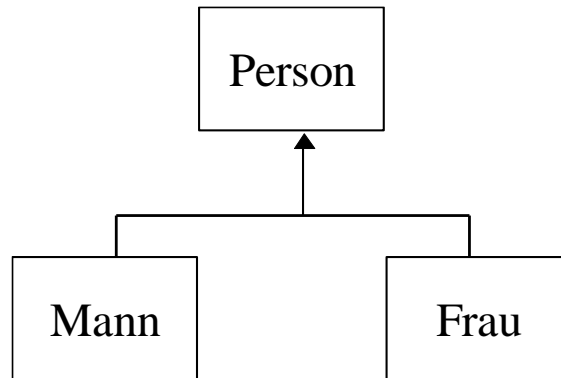
- Entities sind Instanzen der Entity-Typen und nehmen *keine Werte* an.
- ... nur deren Attribute stellen die beschreibenden Eigenschaften dar und nehmen Werte an.

Die Entscheidung ist abhängig vom *Kontext (Situation/Anwendungsfall)*.

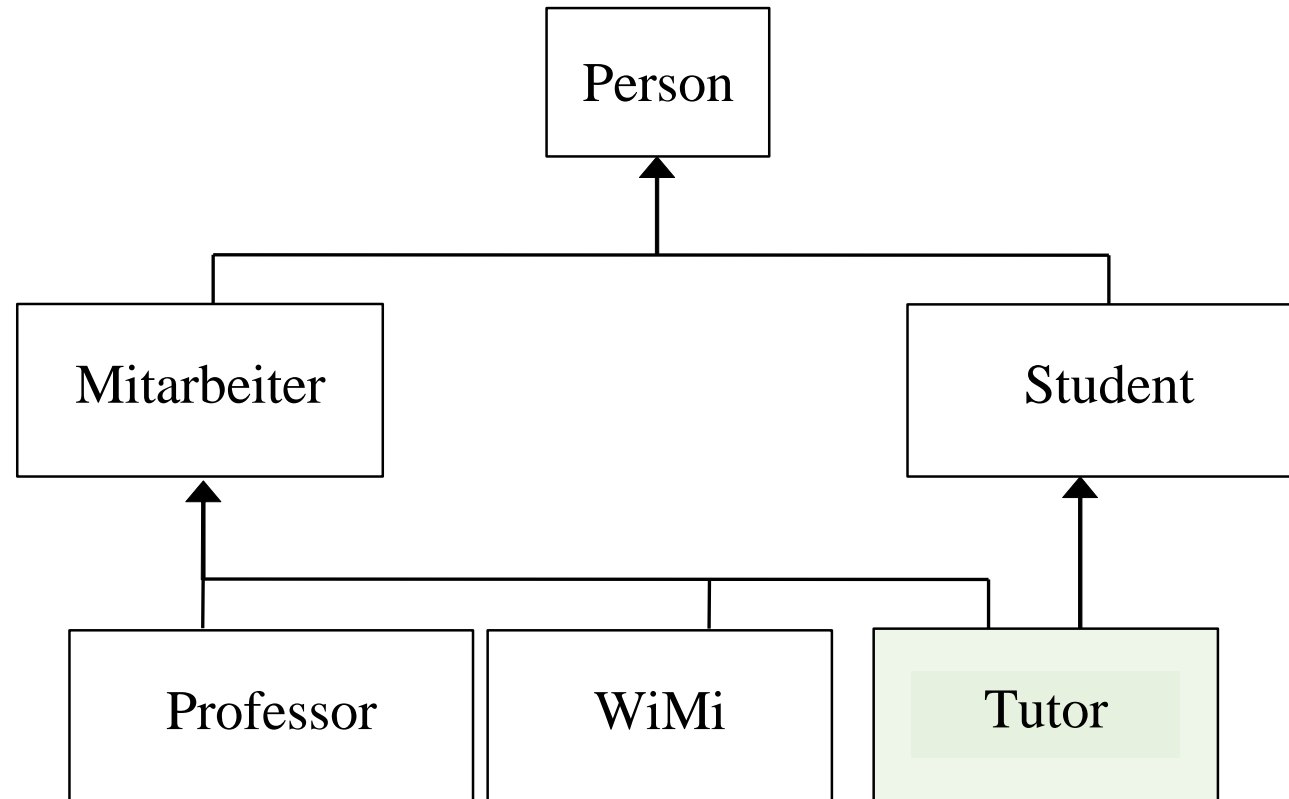


Generalisierung

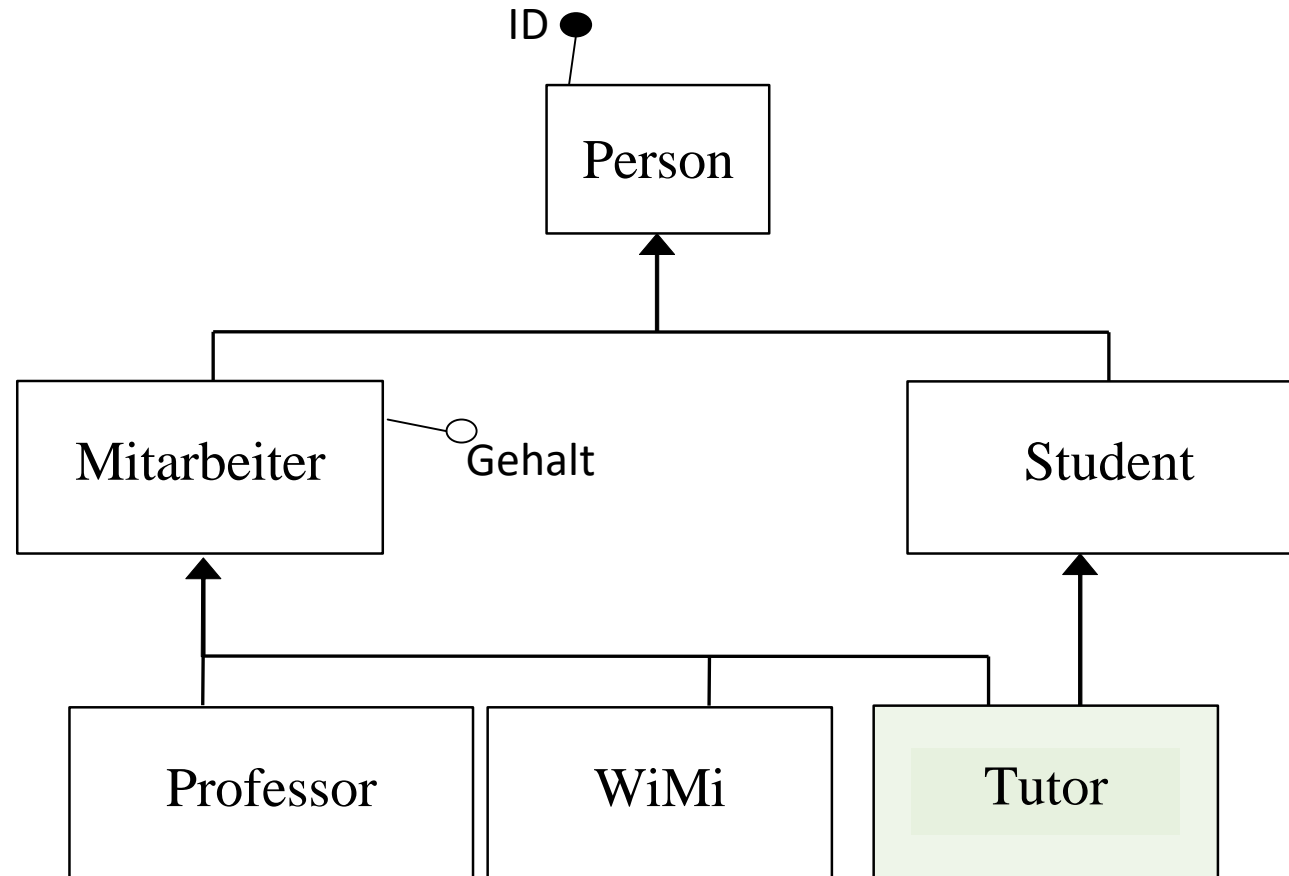
- Hierarchien für Objekttypen (entspricht Klassenhierarchie in OO)



... mit Mehrfachvererbung



... mit Mehrfachvererbung



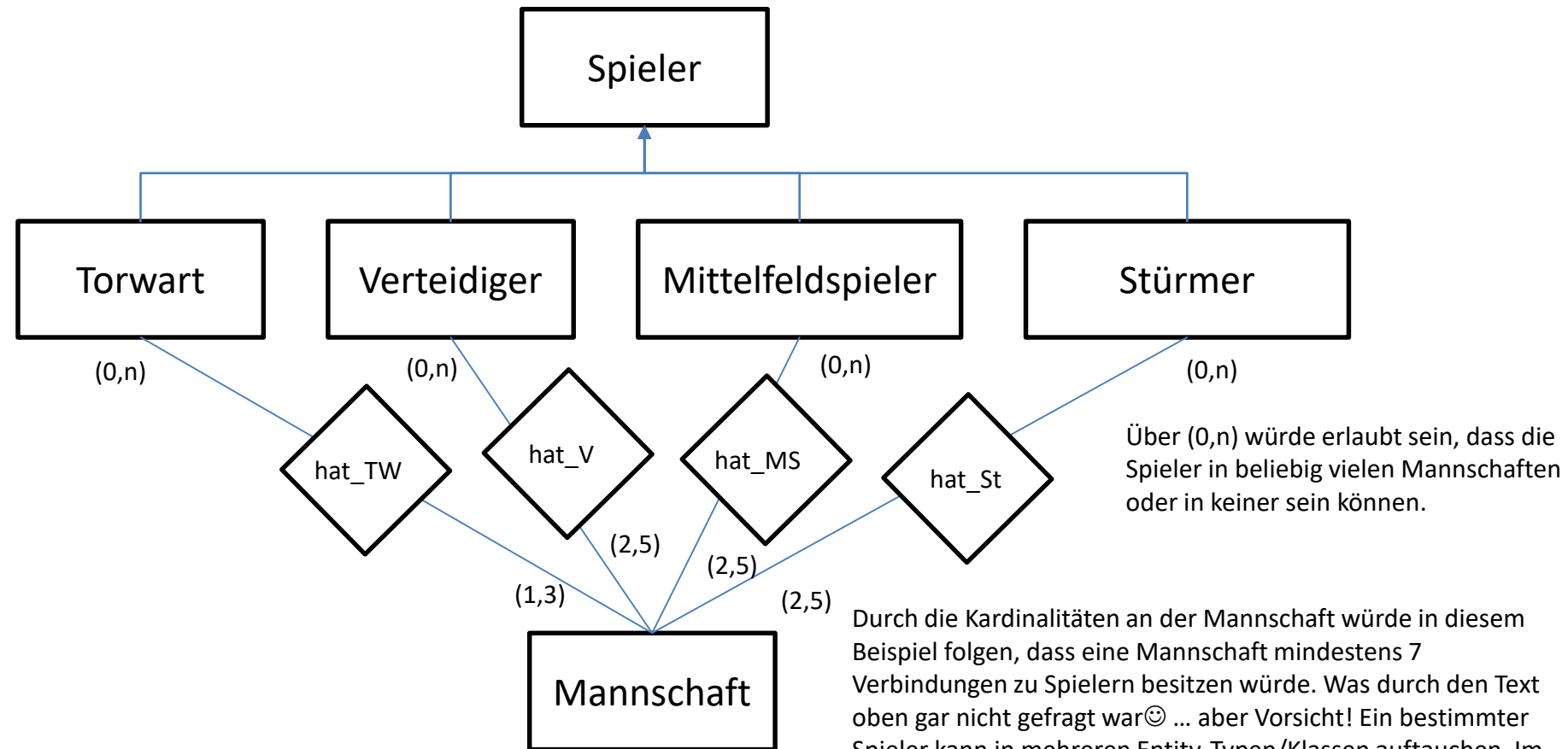
Übung 3

- Aussage:
 - Eine Mannschaft besteht aus Spieler, die sich aufteilen in Verteidigung, Mittelfeld, Angriff oder Torhüter.



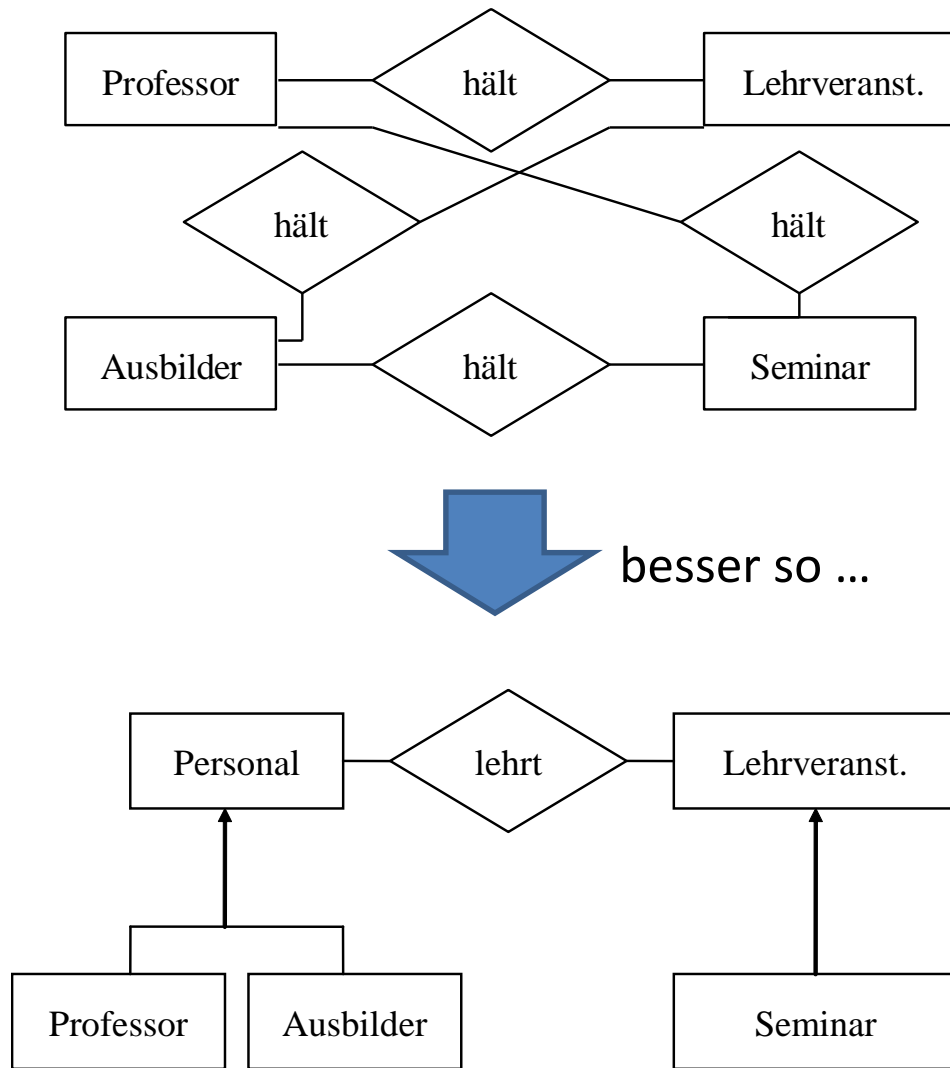
Übung 3

- Aussage:
 - Eine Mannschaft besteht aus Spieler, die sich aufteilen in Verteidigung, Mittelfeld, Angriff oder Torhüter.

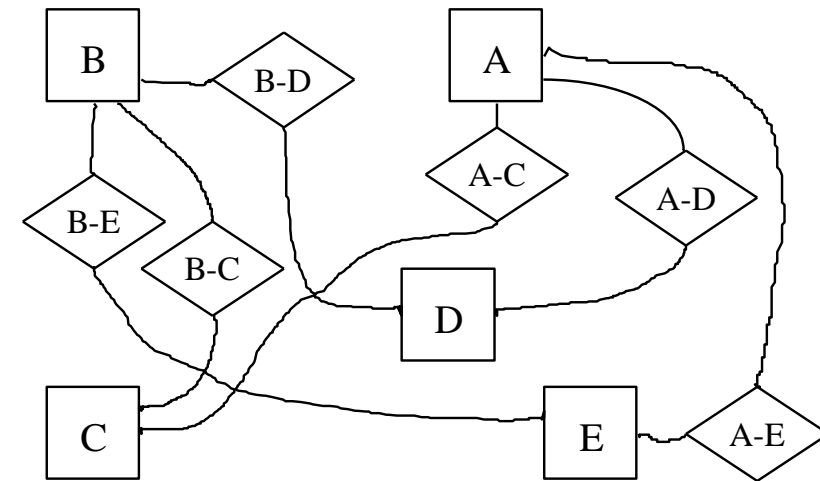


Durch die Kardinalitäten an der Mannschaft würde in diesem Beispiel folgen, dass eine Mannschaft mindestens 7 Verbindungen zu Spielern besitzen würde. Was durch den Text oben gar nicht gefragt war ☺ ... aber Vorsicht! Ein bestimmter Spieler kann in mehreren Entity-Typen/Klassen auftauchen. Im Extrem-Fall kann dies hier dazu führen, dass eine Mannschaft nur mit zwei Spielern verbunden ist!

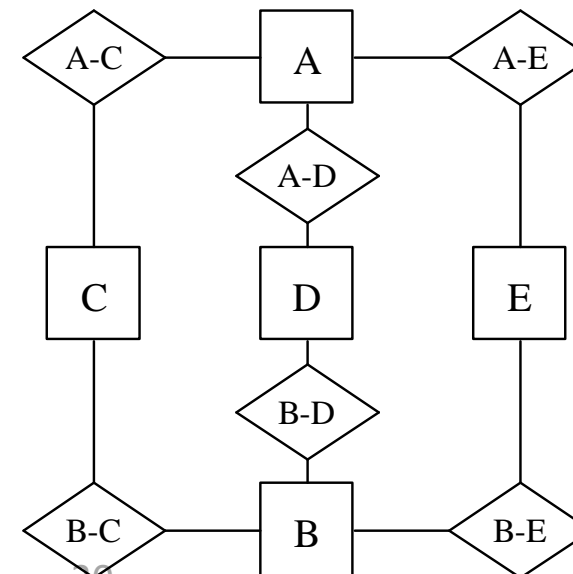




besser so ...



besser so ...



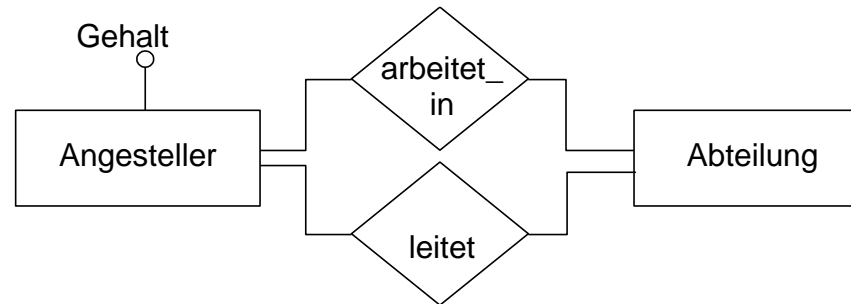
ER-Modell

- Vorteile
 - Unabhängig von Implementierungsdetails
 - Grafische Darstellung (leicht zu lesen)
 - ...
- Nachteile
 - Abbildung des ER-Modells in das relationale Modell ist nicht eindeutig → nicht automatisch
 - ER-Modell ist „statisch“
 - ...



Ausdruckskraft

- Ein Angestellter einer Abteilung soll nicht mehr verdienen, als der entsprechende Abteilungsleiter.



Benötigt zusätzliche Beschreibung,
sogenannte ***Business Rules***.

Ein Angestellter **darf nicht** mehr Gehalt bekommen als der Abteilungsleiter, zu dessen Abteilung der Angestellte gehört.

Ein Abteilungsleiter **muss** zu der Abteilung gehören, die er leitet.



Business Rules (im weitesten Sinne) können angesehen werden als:

1. Die **semantische Definition** eines für Anwendungen relevanten Konzeptes, genauer, die semantische Definition
 - eines Objektes,
 - eines Attributes,
 - einer Relation

des ER-Modells.

Für diesen Fall werden natürlich sprachliche Sätze verwendet, da es unmöglich ist hierfür eine präzise Syntax zu definieren.

2. **Integritätsbedingungen** für die Daten einer Anwendung (als zusätzliche Beschreibung der im ER-Modell enthaltenen Bedingungen oder zusätzliche Bedingungen).
3. **Abgeleitete Bedingungen** bzw. Folgerungen aus anderen Bedingungen (z.B. Brutto ist Summe aus Netto plus Steuer).



ER Zusammenfassung

- Entitäten und Entity-Typen
- Beziehungen und Beziehungstypen
- Attribute
 - für Entitäten(Typen) und Beziehungen(Typen)
 - einfacher oder zusammengesetzter Schlüssel
- Kardinalitäten
- Generalisierung
- Business Rules



Tools zum Zeichnen von ER-Diagrammen

... einfach PowerPoint oder andere Zeichentools, eine genaue 1-1 wie in der Vorlesung ist mir nicht bekannt.

... wer mag, kann sich auch gerne folgende Tools ansehen (nur eine Auswahl):

- Edraw - <https://www.edrawsoft.com>
- yEd - <https://www.yworks.com/products/yed>
- Lucidchart - <https://www.lucidchart.com>
- Draw.io - <https://drawio-app.com/entity-relationship-diagrams-with-draw-io/>



SQL - Primärschlüssel

Mittels der Klausel **primary key** kann ein Attribut einer Relation als Schlüssel ausgezeichnet werden. Hierdurch kann ein Datensatz identifiziert werden und das DBS sorgt dafür, dass keine Duplikate eingetragen werden.

```
create table kunde (  
    name varchar(30) primary key not null,  
    vorname varchar(20),  
    strasse varchar(50),  
    stadt varchar(25),  
    kinder int not null default 0,  
    gebdatum date  
);
```

abhängig vom DBMS, ob „not null“
nötig ist – MariaDB nicht nötig.



Primärschlüssel

Wenn mehr als ein Attribut als Primärschlüssel definiert werden sollen, wird die Klausel in der Form **primary key** (*Attributnamen-Liste*) verwendet.

```
create table kunde2 (  
    name varchar(30) not null,  
    vorname varchar(20) not null,  
    strasse varchar(50),  
    stadt varchar(25),  
    kinder int not null default 0,  
    gebDatum date,  
    primary key (name, vorname) ) ;
```



Primärschlüssel

- Es kann für eine Tabelle nur maximal einen Primärschlüssel (PK) geben.
- Eine Tabelle kann neben dem PK weitere Schlüssel (Schlüsselkandidaten) besitzen. Diese können über „Unique Index“ im DBS modelliert werden.



Aggregatfunktionen

Die sog. Aggregatfunktionen können in der **select**-Klausel anstelle von einzelnen Attributen angegeben werden.

Ergebnis einer Aggregatfunktion ist ein Wert, kein Tupel.

```
select count(*) as AnzahlKunden  
from kunde;
```



Aggregatfunktionen

- **min(A)** zur Berechnung des Minimalwerts aller Tupel unter dem Attribut A.
- **max(A)** zur Berechnung des Maximalwerts aller Tupel unter dem Attribut A.
- **avg([distinct] A)** zur Berechnung des Durchschnittswerts aller Tupel unter dem Attribut A, wobei unter Angabe von distinct mehrfach gleiche Werte nur einmal in die Berechnung eingehen.
- **sum([distinct] A)** zur Berechnung der Summe aller Tupel unter dem Attribut A, wobei unter Angabe von distinct mehrfach gleiche Werte nur einmal in die Berechnung eingehen.
- **count(*)** zum Zählen der Tupel der betrachteten Relation.
- **count([distinct] A)** zum Zählen der Tupel der betrachteten Relation, wobei zunächst eine Duplikateneliminierung bezogen auf Werte unter dem Attribut A stattfindet.

Behandlung von NULL-Values beachten!



Anfrage über mehrere Relationen

Werden in der **from**-Klausel mehrere Relationen spezifiziert, so erfolgt die Berechnung des **kartesischen Produktes**.

konto (konto_nr, kunden_nr, name, saldo)

filiale (name, leiter)

select * from filiale, konto;

... im Ergebnis wird jeder DS auf Filiale
mit jedem DS aus Konto verbunden!



Anfrage über mehrere Relationen

konto (konto_nr, kunden_nr, name, saldo)

filiale (name, leiter)

select * from filiale, konto;

select * from filiale, konto
where name = ' test1';

→ Fehlermeldung: Name nicht eindeutig!

→ select * from filiale f, konto k
where f.name = ' test1';



Anfrage über mehrere Relationen

... hier mit where-Bedingung, was einem sogenannten „Join“ entspricht.

... Verwendung von Fremdschlüsseln eventuell sinnvoll (siehe Video)!

konto (konto_nr, kunden_nr, name, saldo)

filiale (name, leiter)

```
select * from filiale f, konto k  
      where f.name = k.name;
```

