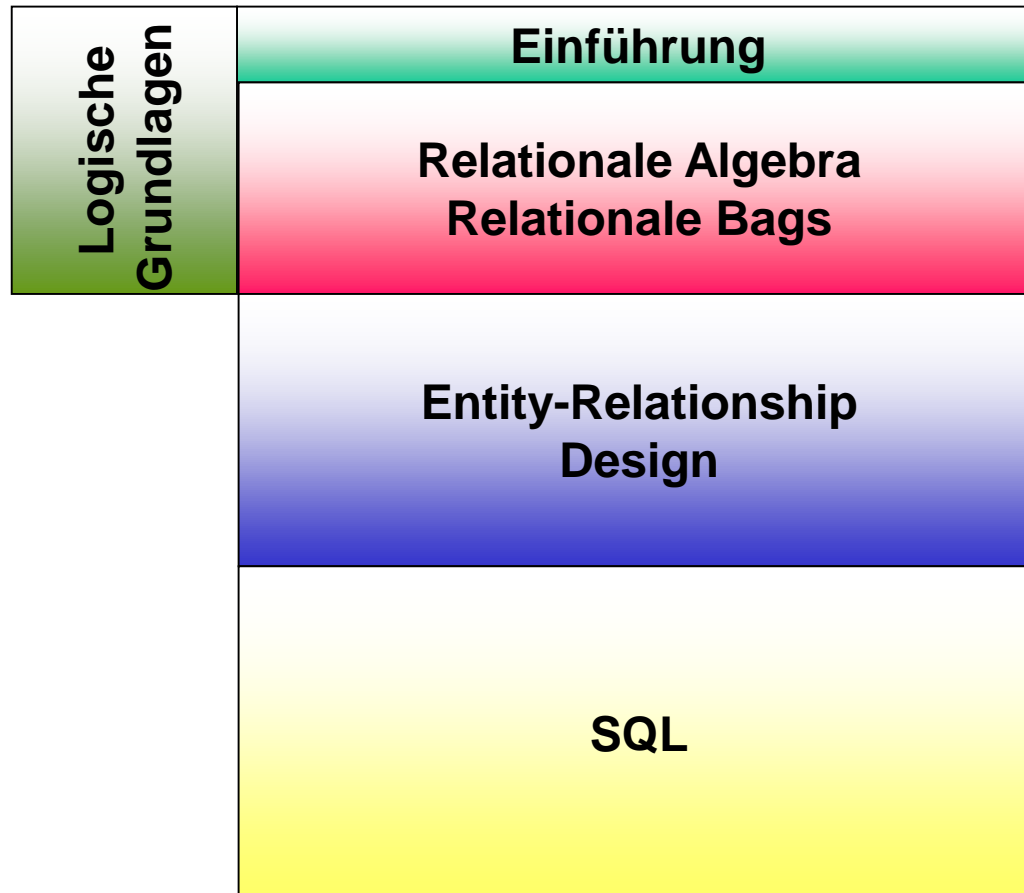


DAB1 – Datenbanken 1

Dr. Daniel Aebi (aebd@zhaw.ch)

Lektion 6: Design mit dem Entity-Relationship Modell (ERM)

Wo stehen wir?



← "You are here"

Rückblick

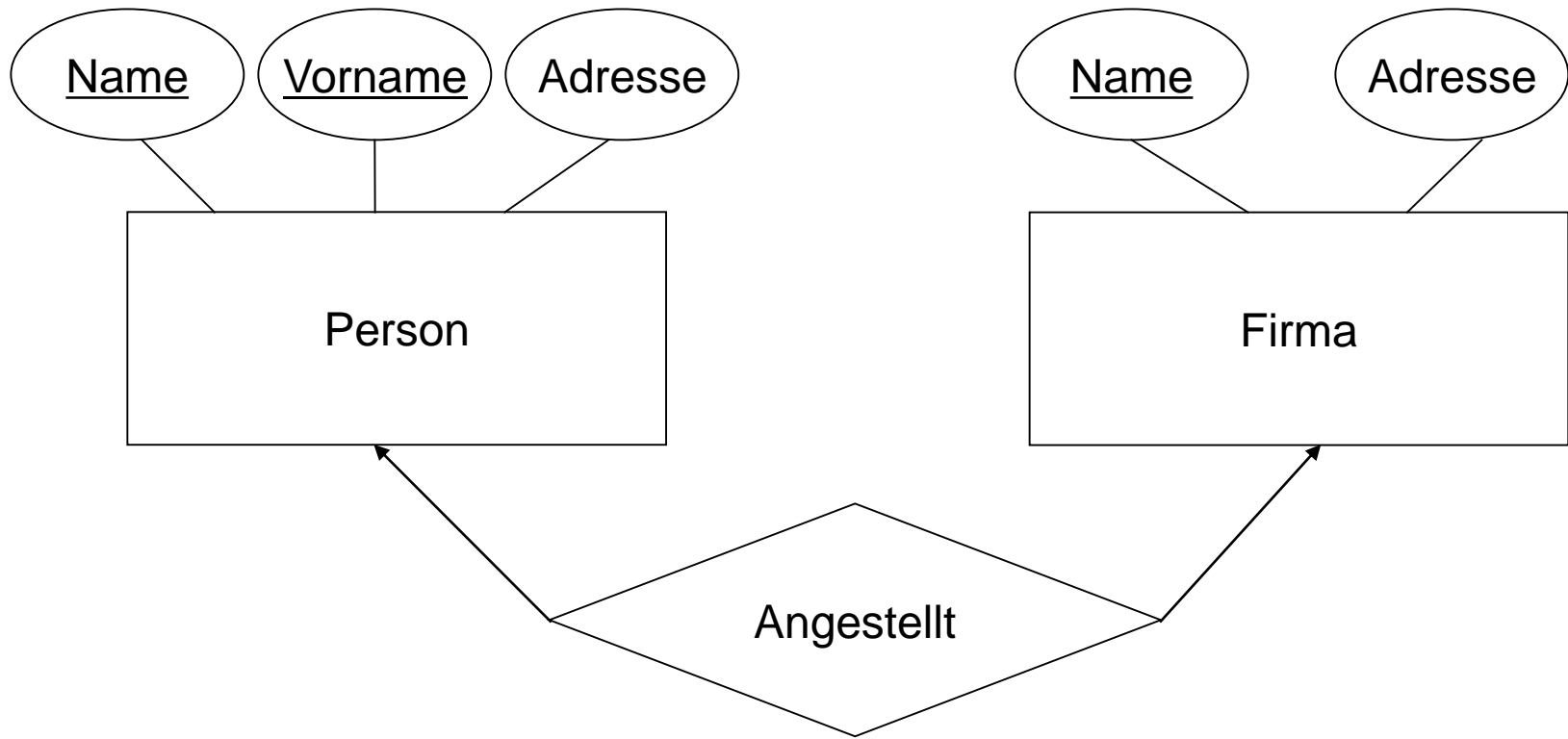
- Schlüsselbegriffe:
 - Schlüssel, Superschlüssel
 - Schlüsselkandidat
 - Primärschlüssel
 - Fremdschlüssel
- Grundsätzliches Vorgehen beim Datenbankentwurf
 - Konzeptioneller Entwurf
 - Logischer Entwurf
 - Physischer Entwurf
- ER-Modell:
 - Entitätstypen
 - Attribute
 - Beziehungstypen
 - Kardinalitäten, Schlüssel

Lernziele Lektion 6

- Kardinalitäten \leftrightarrow Schlüssel verstehen
 - 1: entspricht keiner oder genau einer
 - m: keiner, einer oder mehrere
- Weitere Elemente von ERM kennen:
 - ISA-abhängiger Entitätstyp
 - ID-abhängiger Entitätstyp
 - Zusammengesetzter Entitätstyp

Beziehungstyp

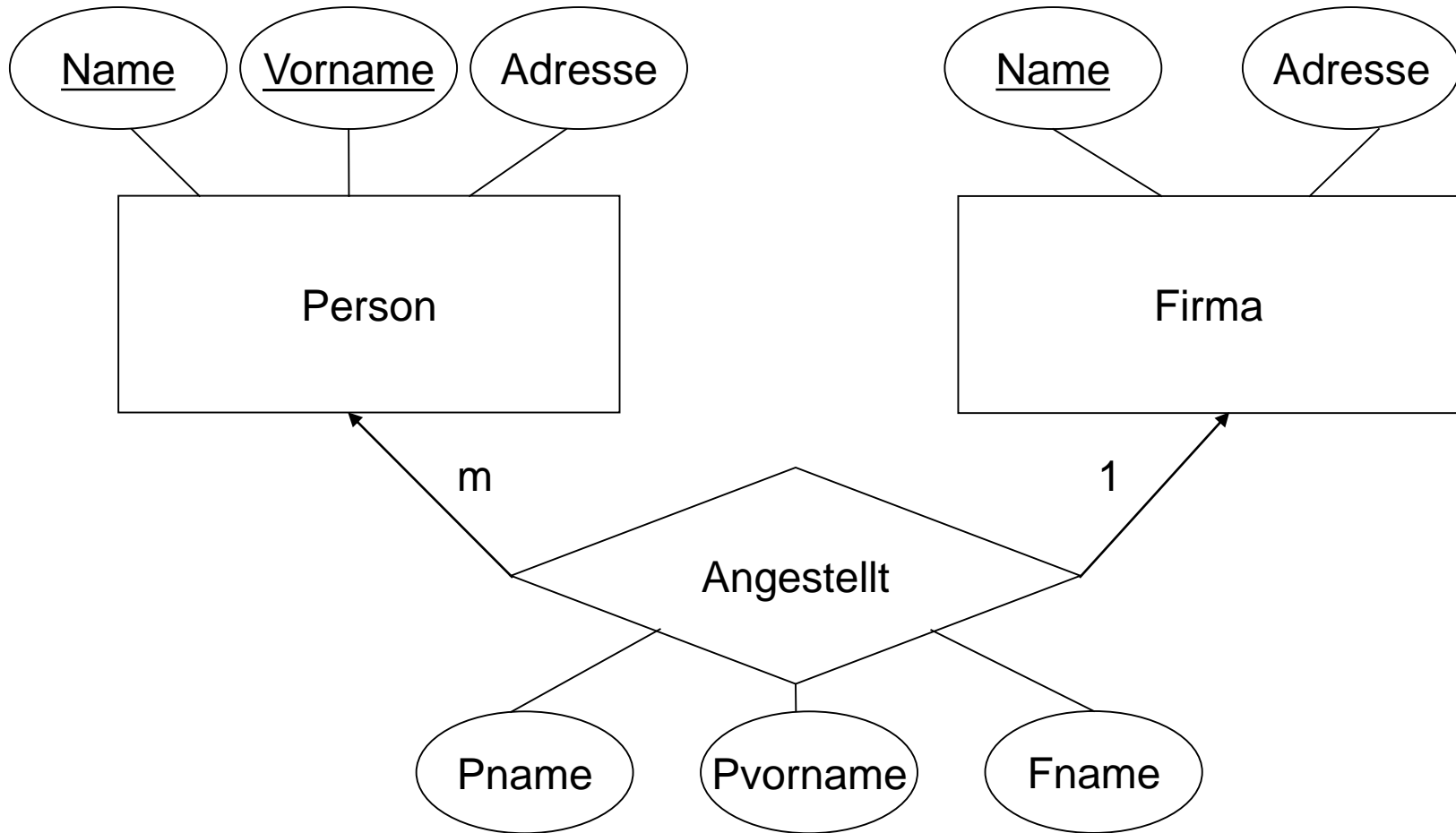
- Wir stellen einen Beziehungstyp (hier: «Angestellt») durch einen Rhombus dar.



Beziehungstyp, Attribute

- Welche Attribute werden mindestens benötigt, damit «Angestellt» die Verbindung zwischen «Person» und «Firma» bilden kann?
- Was müssen wir sonst noch wissen, damit wir die Beziehung zwischen «Person» und «Firma» spezifizieren können?
- Inwiefern ist «Angestellt» andersartig als «Person» oder «Firma»? Warum führen wir einen neuen Begriff «Beziehungstyp» ein?

Beziehungstyp

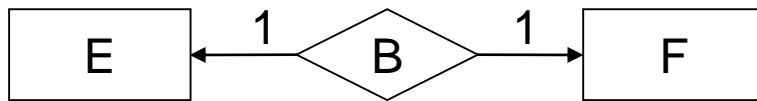


Beziehungstyp

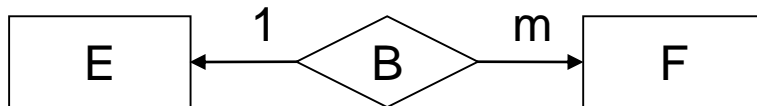
- Der Beziehungstyp «Angestellt» **erbt** die Primärschlüsselattribute der Entitätstypen «Person» und «Firma», von denen er abhängig ist.
- Er kann auch noch weitere, «eigene» Attribute haben (z.B. «seit» – um auszudrücken, seit wann die Anstellung besteht).
- Die **Fremdschlüssel** wählt man wenn möglich mit gleichem Namen, ausser bei Namenskonflikten.
- Die Pfeilmarkierungen 1,m (**Kardinalitäten**) drücken aus, dass pro Person **höchstens eine** Firma als Arbeitgeber existiert, während eine Firma **beliebig viele** (auch 0!) Angestellte haben kann.
- «m» bedeutet also beliebig viele, «unbestimmt» o.ä.
- Will man «Angestellt» als Relation abbilden, muss {Pname,Pvorname} als Schlüssel gewählt werden.

Beziehungstyp

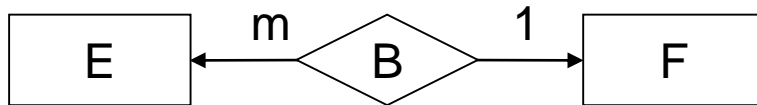
- Beziehungstypen haben **Schlüssel**, keine **Primärschlüssel** (Es sei denn, sie werden referenziert; siehe hierzu später)
- Mögliche Kombinationen: (inkl. passendem Schlüssel in B)



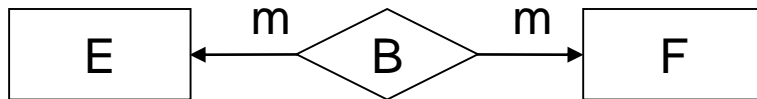
Schlüssel von E **und** F



Schlüssel von F



Schlüssel von E



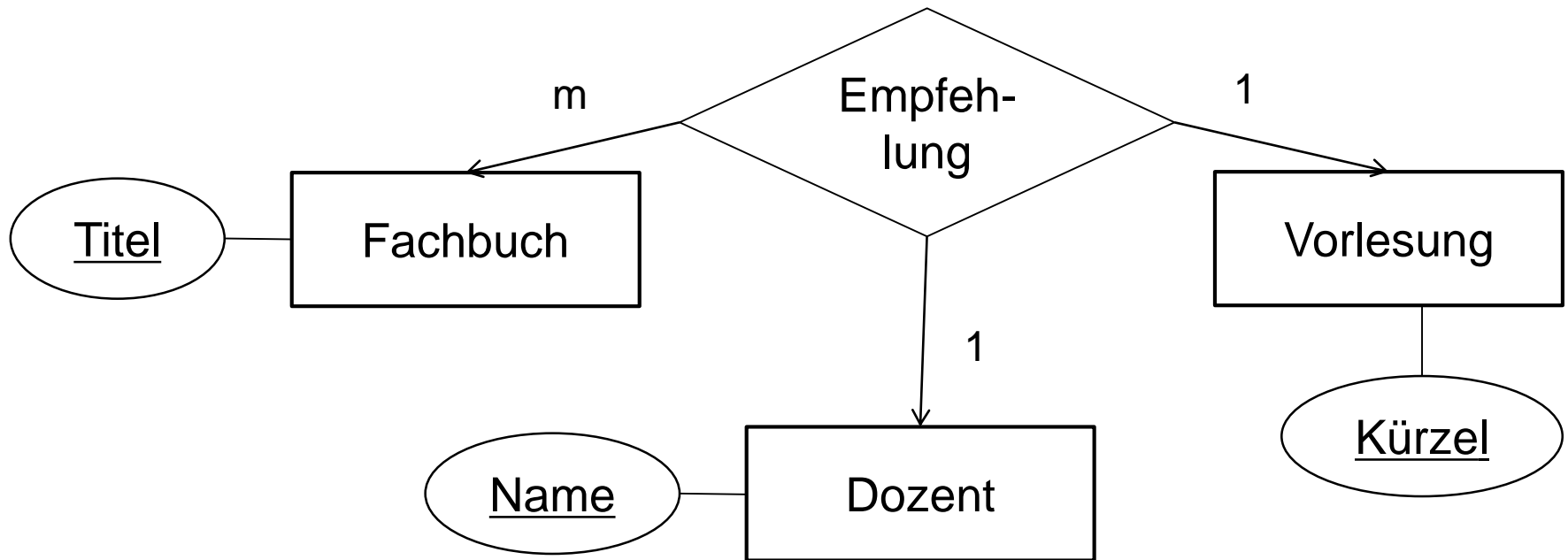
Schlüssel von E **komb.** F

- Der Beziehungstyp ist **existentiell abhängig** von den Entitätstypen, welche er referenziert.

Beispiel mit dreiwertiger Beziehung

Beispiel zum Thema Schlüssel:

- An der ZHAW sollen Dozenten Fachbücher empfehlen zu Vorlesungen. Diese Empfehlungen sollen in einer Datenbank festgehalten werden. Die Datenbank ist wie folgt modelliert:



Beispiel mit dreiwertiger Beziehung

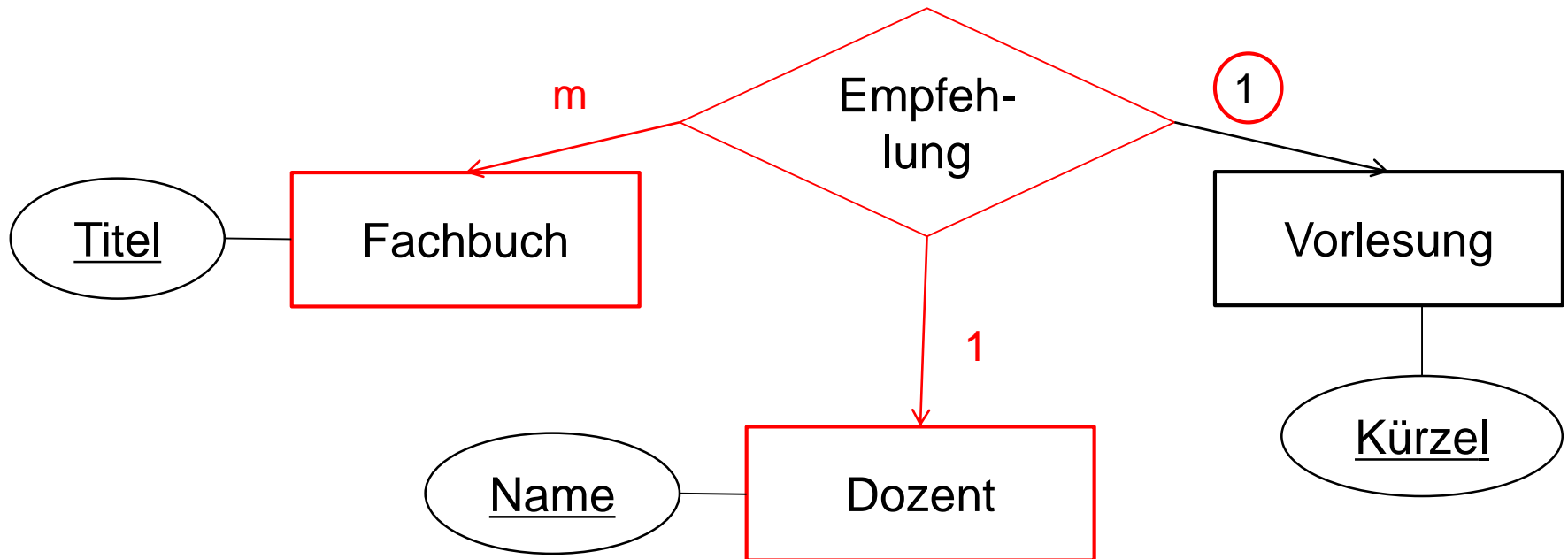
Fragen:

- Kann ein Fachbuch nicht empfohlen werden?
 - Kann ein Dozent keine Empfehlung abgeben?
 - Kann für eine Vorlesung keine Empfehlung abgegeben werden?
 - Können mehrere Bücher für eine Vorlesung empfohlen werden?
 - Kann ein Buch von mehreren Dozenten empfohlen werden?
 - Kann ein Buch in mehreren Vorlesungen empfohlen werden?
 - Kann ein Buch von einem Dozenten mehrfach empfohlen werden?
 - ...?
-
- Allgemein: Was heissen die m- und 1-Kardinalitäten?



Beispiel mit dreiwertiger Beziehung

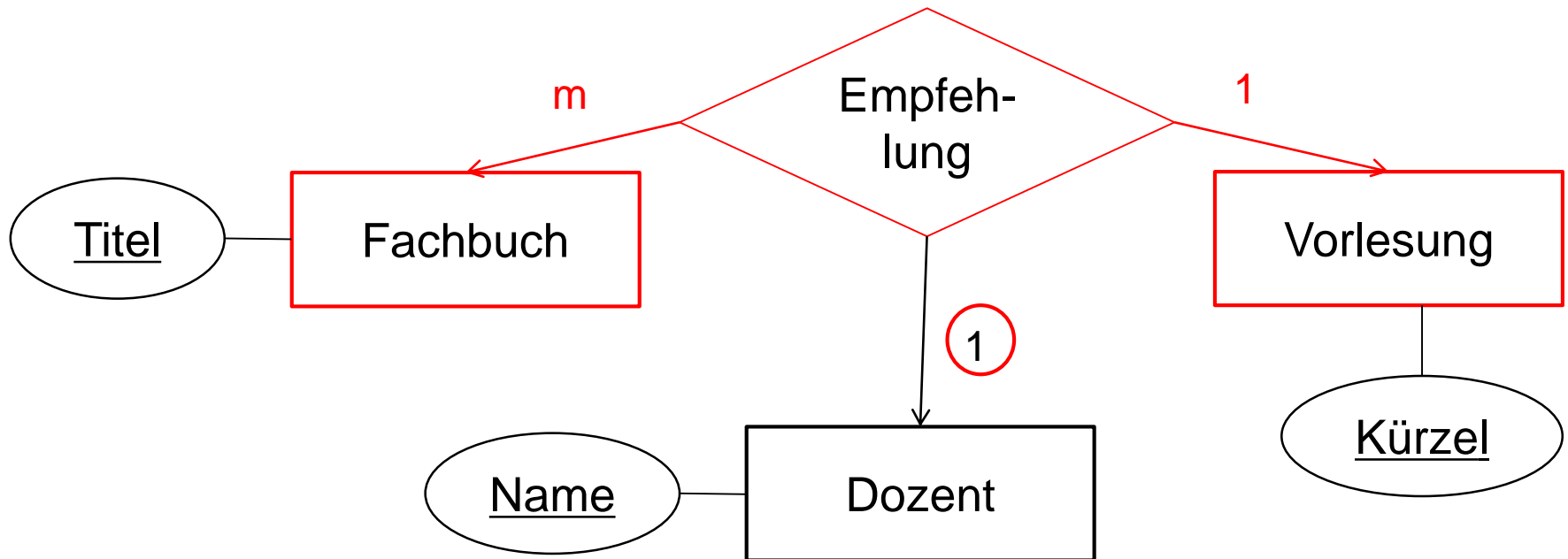
- Allgemein: Was heissen die m- und 1-Kardinalitäten?



- Ein Buch kann von einem Dozenten für höchstens eine Vorlesung empfohlen werden!

Beispiel mit dreiwertiger Beziehung

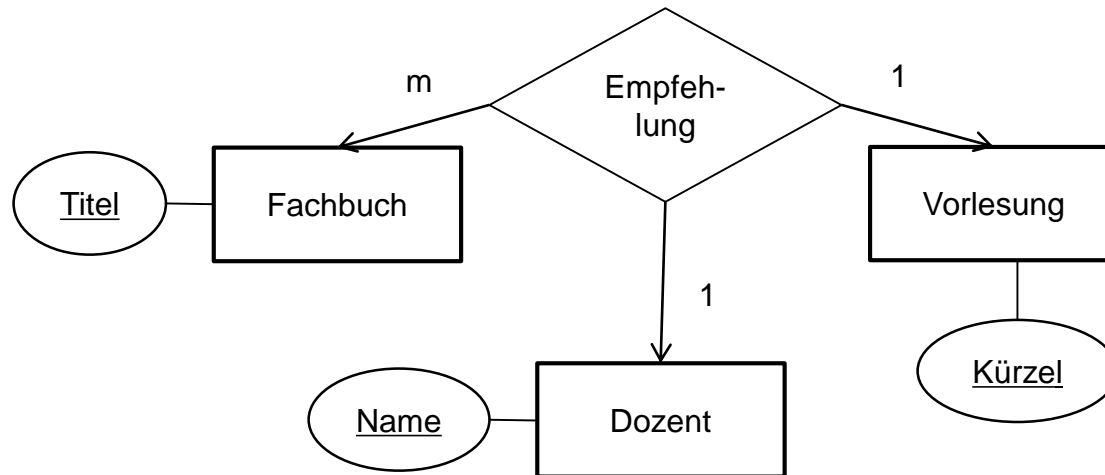
- Allgemein: Was heissen die m- und 1-Kardinalitäten?



- Ein Buch kann für eine Vorlesung **von höchstens einem** Dozenten empfohlen werden!

Beispiel mit dreiwertiger Beziehung

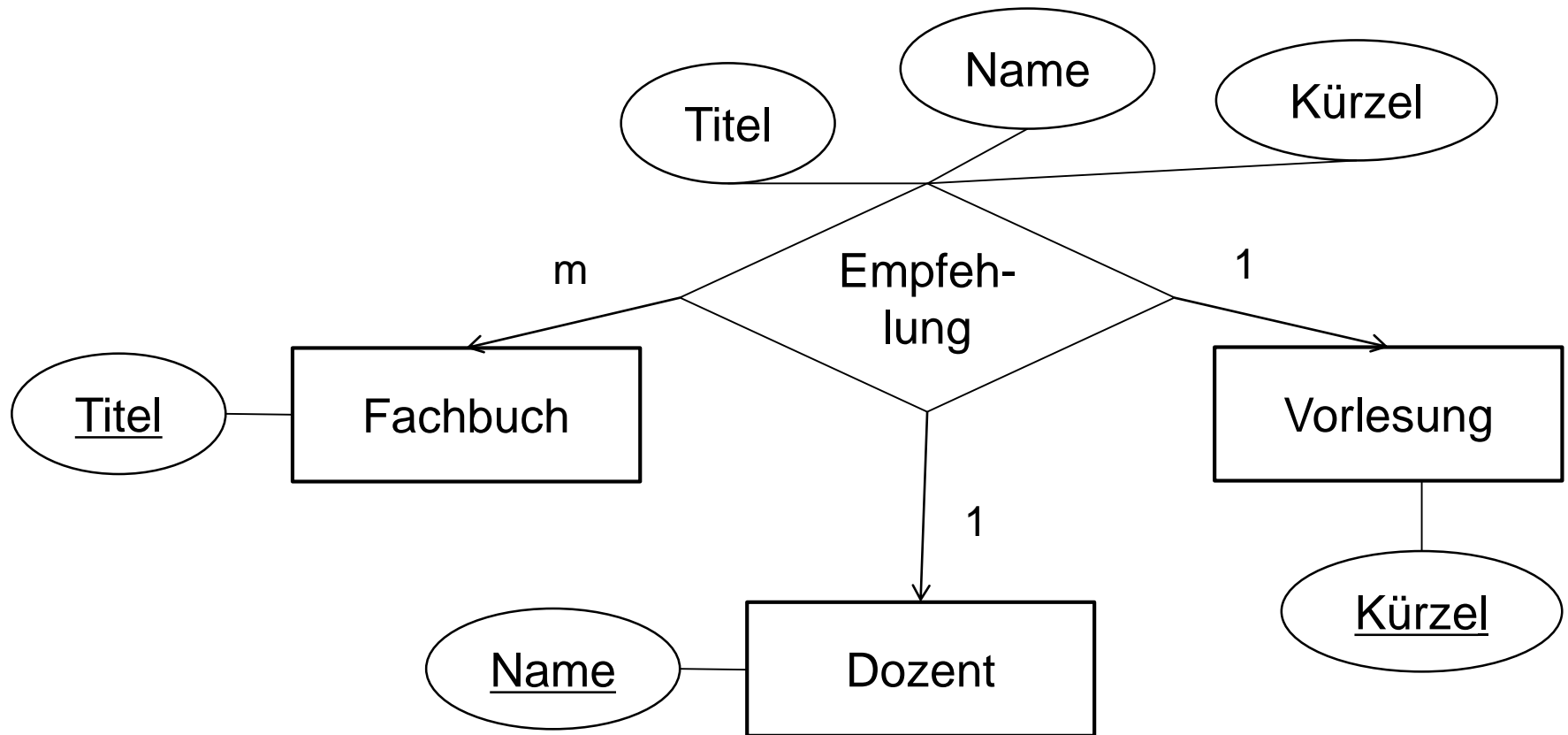
- Die m- und 1-Kardinalitäten bedeuten nicht:



- Ein Buch kann nur von einem Dozenten empfohlen werden.
- Ein Buch kann nur in einer Vorlesung empfohlen werden.
- Ein Dozent kann nur ein Buch empfehlen.
- ...

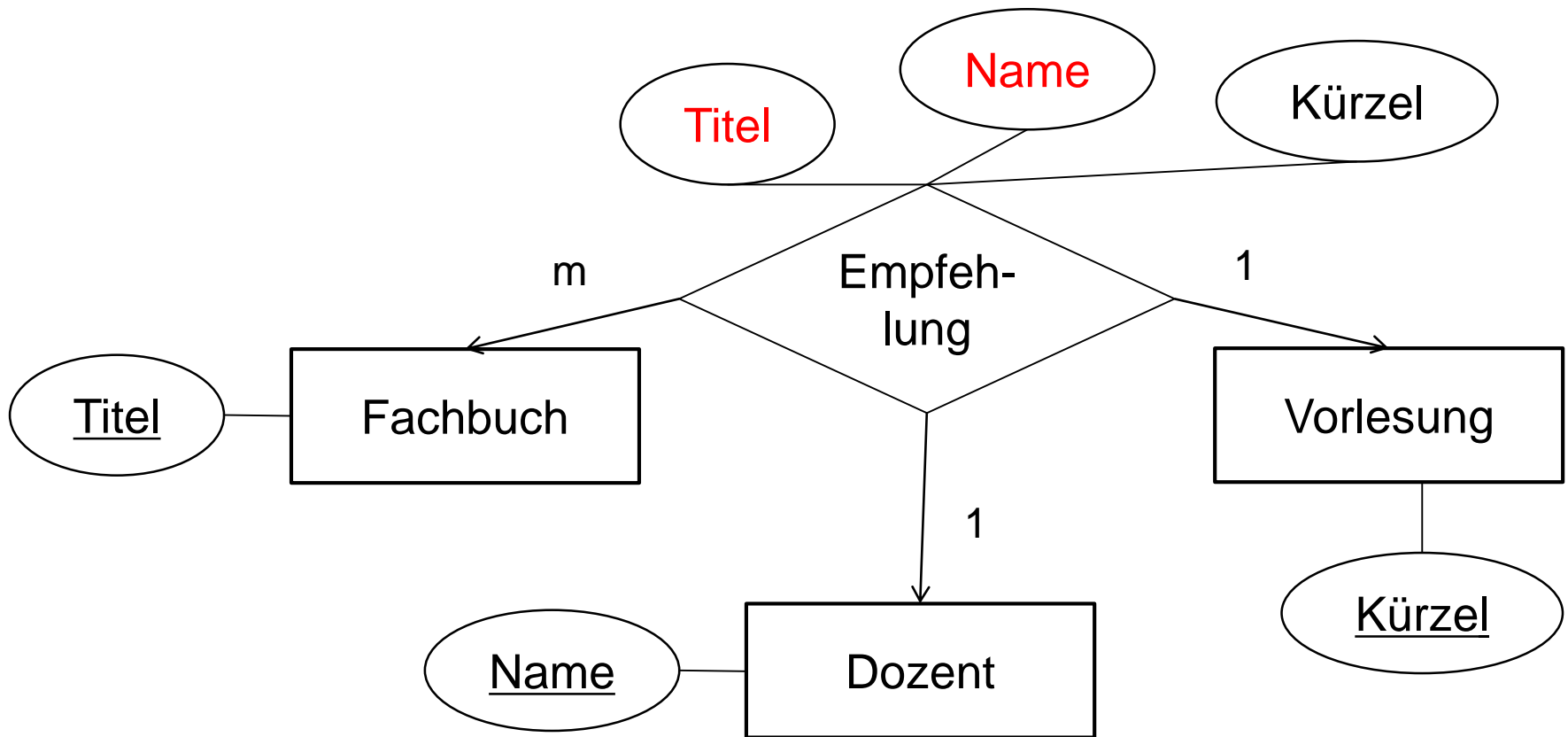
Beispiel mit dreiwertiger Beziehung

- Wie stellen wir das sicher? Mit Schlüsseln!



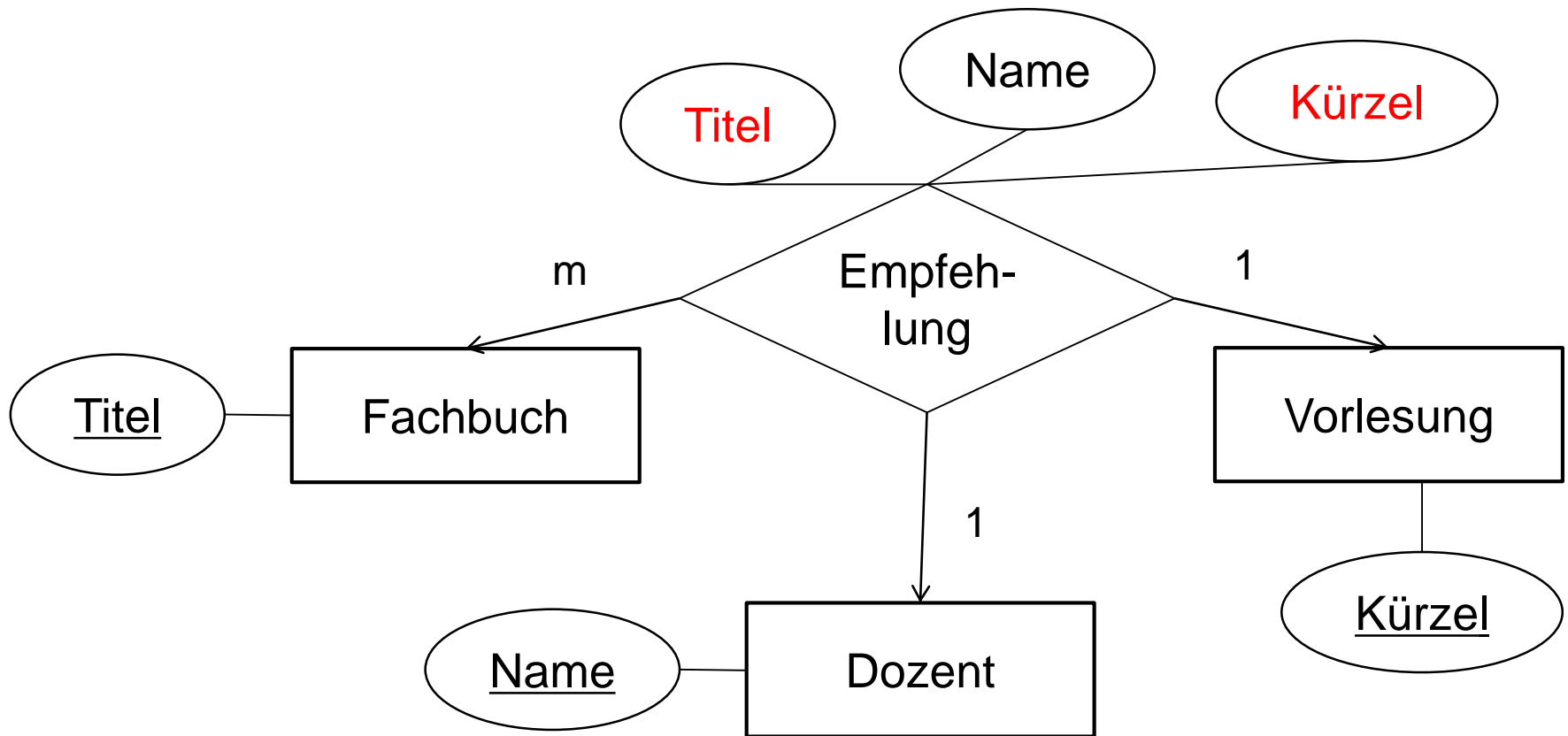
Beispiel mit dreiwertiger Beziehung

- Ein Buch kann von einem Dozenten höchstens einmal empfohlen werden (für irgendeine Vorlesung):



Beispiel mit dreiwertiger Beziehung

- Ein Buch kann in einer Vorlesung höchstens einmal (von irgendeinem Dozenten) empfohlen werden!



Beispiel mit dreiwertiger Beziehung

- Visualisierung mit Beispieldaten:

Fachbuch (Titel):

SQL

JAVA

Programmieren in C

Entwurf mit ERM

...

Dozent (Name):

Aebi

Bitto

Meier

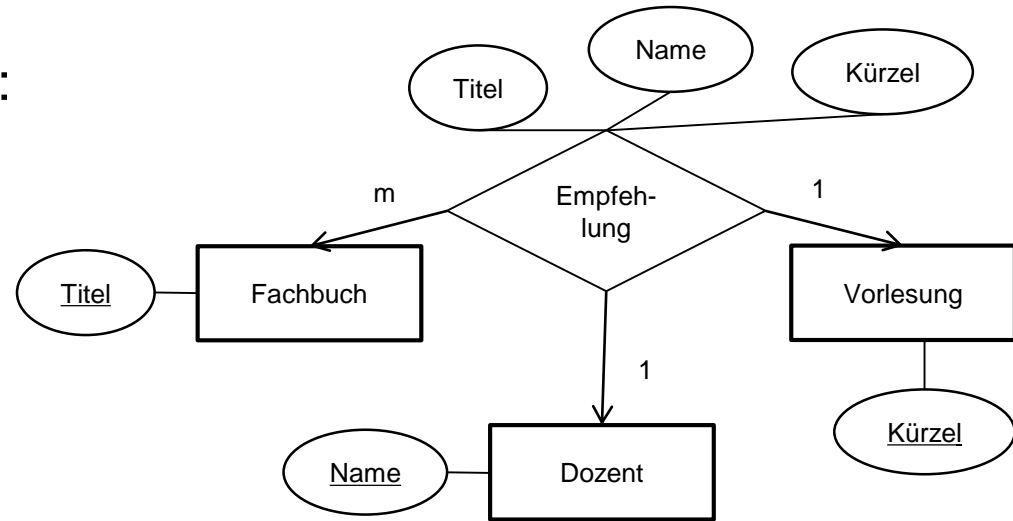
...

Vorlesung (Kürzel):

DAB1

MANIT

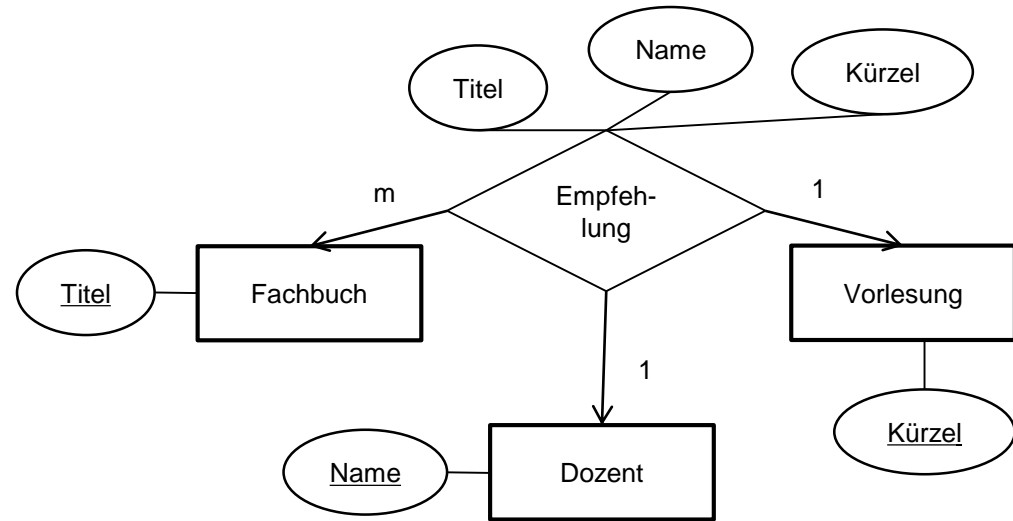
...



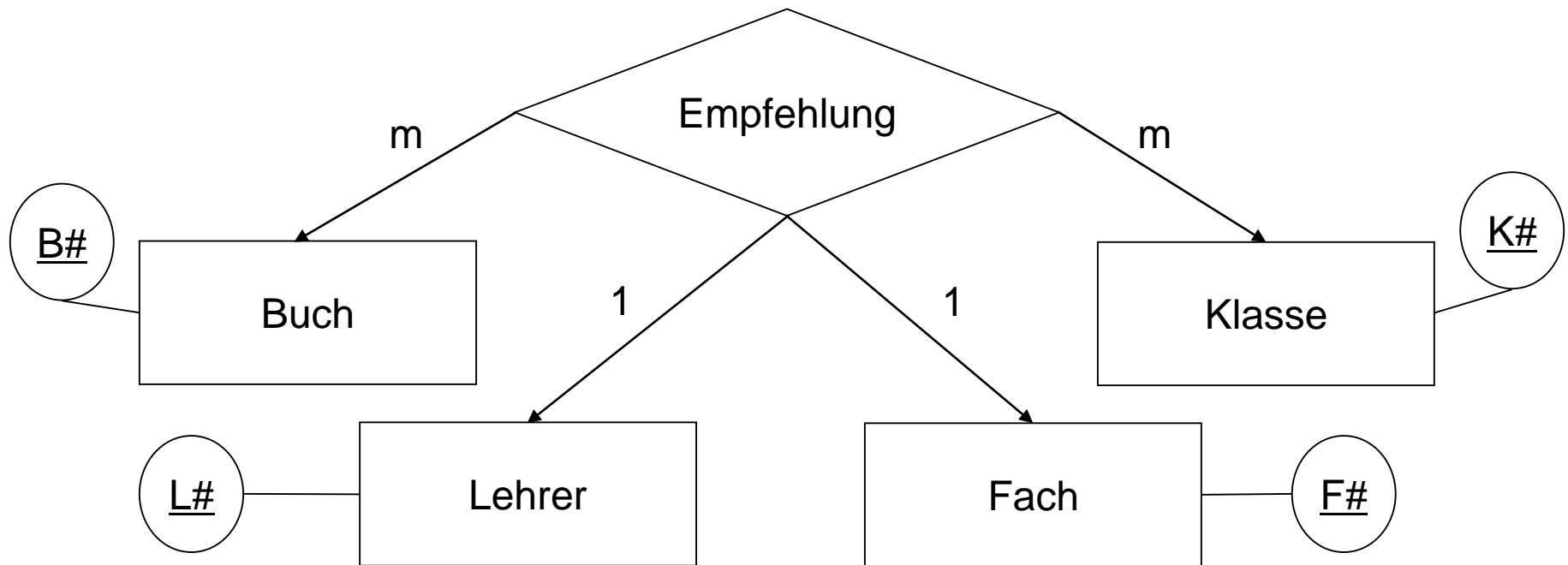
Titel	Name	Kürzel
Titel	Name	Kürzel
SQL	Aebi	DAB1
JAVA	Aebi	DAB1
SQL	Bitto	MANIT
SQL	Bitto	DAB1
SQL	Aebi	MANIT

Beispiel mit dreiwertiger Beziehung

- Konsequenzen:
- Wir brauchen **ZWEI** Schlüssel.
- In der Datenbank müssen **BEIDE** Schlüssel durchgesetzt werden.
- Der Begriff Primärschlüssel ist unglücklich. In unserer ER-Sprache bezeichnet er einen Schlüssel, auf den über einen Fremdschlüssel Bezug genommen wird. In SQL bezeichnet er einfach einen "wichtigen", "ersten", ... Schlüssel.



Weiteres Beispiel



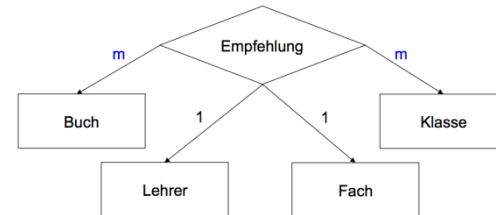
- Dieser Beziehungstyp ist **existentiell abhängig** von vier Entitätstypen (siehe Pfeile).
- Zwei Schlüsselbedingungen (aus Lehrer und Fach).

Weiteres Beispiel

- Die Markierung "1" beim Pfeil zu "Lehrer" bedeutet:
 - ein Buch wird einer Klasse in einem Fach von **höchstens einem** Lehrer empfohlen
- Die Markierung "1" beim Pfeil zu "Fach" bedeutet:
 - ein Buch wird einer Klasse von einem Lehrer in **höchstens einem** Fach empfohlen

- Zwei Schlüssel (aus Lehrer und Fach):

- $\{B\#, K\#, F\# \}$, sowie $\{B\#, K\#, L\# \}$



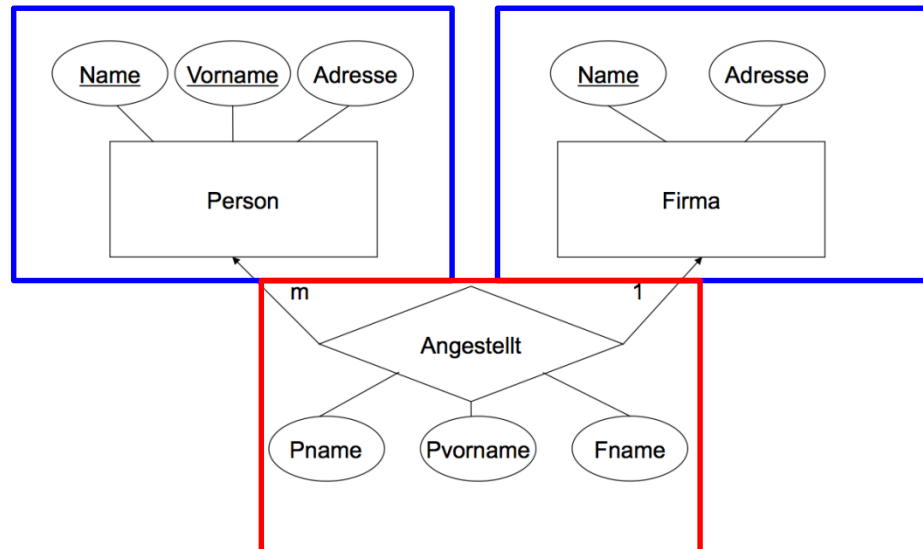
- Mehr ist aus den **Kardinalitätsbedingungen** NICHT herauszulesen!
- Man läuft leicht Gefahr, weitere erwünschte Bedingungen in solche Beziehungen zwischen mehr als 2 Entitätstypen «hinein zu interpretieren» (z.B. «pro Buch höchstens ein Fach» – dies ist NICHT der Fall!)
- Vorsicht bei solchen höherwertigen Beziehungstypen!

Schlüsselbedingungen für Beziehungstypen

- Allgemein können wir sagen:
- Es sei $R(E_1, L_1, E_2, L_2, E_3, L_3, \dots, E_n, L_n)$ ein Beziehungstyp:
 - Dieser hängt von den Entitätstypen $E_1 \dots E_n$, ab
 - wobei die Pfeile je mit $L_1 \dots L_n$ markiert seien.
- Mit M_j bezeichnen wir die Menge der Fremdschlüsselattribute von R , welche dem Primärschlüssel des Entitätstypen E_j entspricht:
 - M_j nicht leer,
 - sowie M_i und M_j sind paarweise elementfremd
 - $M = M_1 \cup M_2 \cup M_3 \cup \dots \cup M_n$ z.B. $\{B\#, K\# F\#\}$, sowie $\{B\#, K\#, L\#\}$
- Ist für mindestens ein j $L_j = 1$ (Kardinalität), so gilt für jedes dieser j mit $L_j = 1$:
 - Die Menge $M \setminus M_j$ ist ein Schlüssel von R
- Sind alle $L_j = m$ (Kardinalität), so ist M ein Schlüssel von R (Achtung: wir wollen Relationen, nicht Bags).

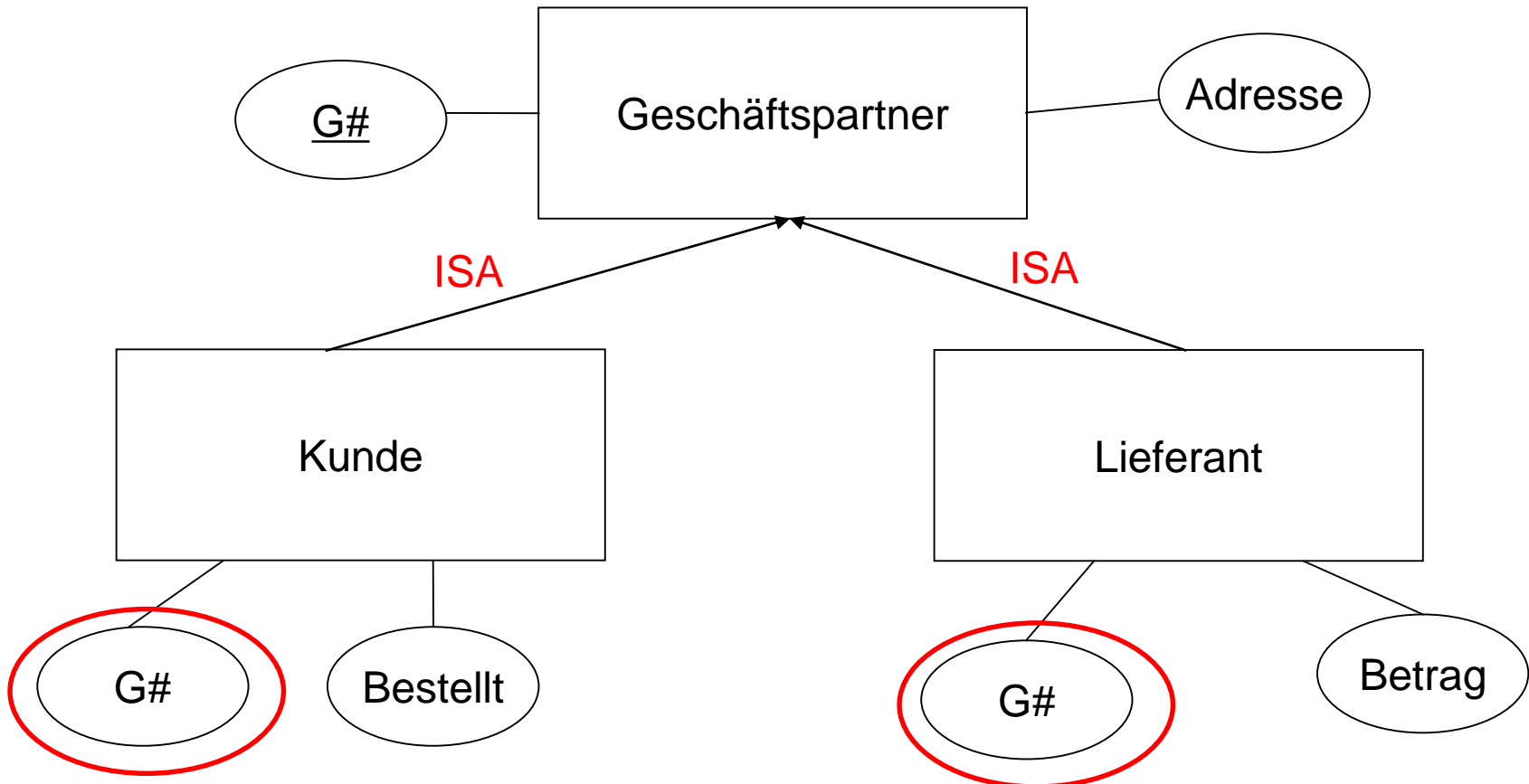
Unabhängiger Entitätstyp

- Wir sprechen bei den Entitätstypen wie bisher kennengelernt von **unabhängigen Entitätstypen** (z.B. Person, Firma).
- Diese Entitätstypen repräsentieren Entitäten, die **für sich selbst «leben»** können.
- Einträge im **Beziehungstyp «Angestellt»** können nicht für sich selbst **existieren**, sie beziehen sich immer auf Personen und Firmen.



ISA-abhängiger Entitätstyp

- Jeder Kunde resp. jeder Lieferant ist auch Geschäftspartner



ISA-abhängiger Entitätstyp

- Die **Pfeile** zeigen eine **existentielle Abhängigkeit** («is a» = «ist ein»)
- Die Pfeile führen zu **Schlüsselbedingungen**
- **{G#}** ist **Schlüssel in Kunde und in Lieferant**:
 - Es wird erzwungen, dass jedem Kunden ein Geschäftspartner entspricht – der Kunde selbst!
- Ein Geschäftspartner kann **Kunde UND/ODER Lieferant** sein
(disjunkte Subtypen können nicht gezeichnet werden → textuell erfassen!)
- «ISA» ist ein **Generalisierungs-/Spezialisierungsmuster**.
 - Kunde und Lieferant werden zu Geschäftspartnern verallgemeinert
 - Kunde und Lieferant sind Spezialisierungen von Geschäftspartner

ISA-abhängiger Entitätstyp

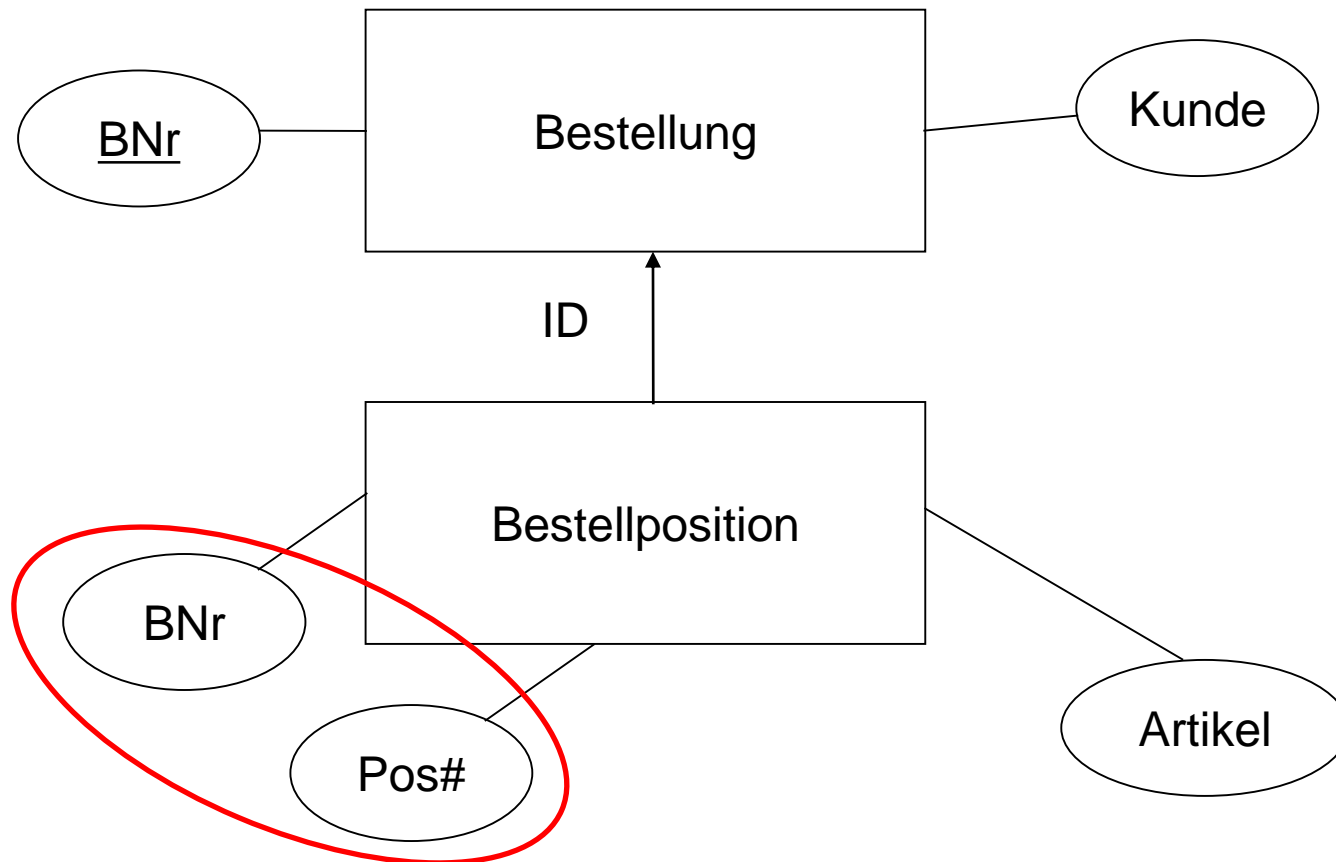
? Wann setzen wir eine solche Generalisierung gewinnbringend ein?

ISA-abhängiger Entitätstyp

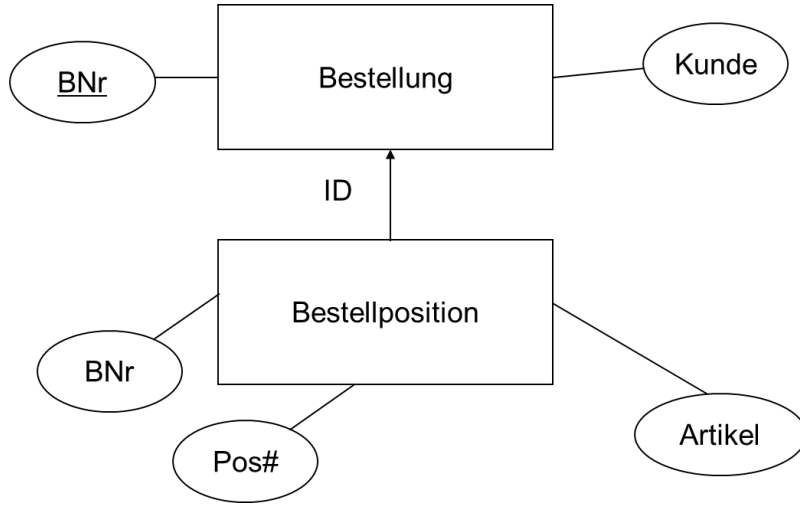
- Eine Generalisierung ist interessant wenn:
 - Die einzelnen Spezialisierungen sich deutlich voneinander **unterscheiden** (Kunde hat andere Attribute als Lieferant, oder hängt mit anderen Beziehungstypen zusammen).
 - Die Generalisierung «Geschäftspartner» kann gemeinsame Strukturen auffangen (in unserem Fall ein Attribut, «Adresse»).

ID-abhängiger Entitätstyp

- Bestellungen umfassen verschiedene Bestellpositionen:

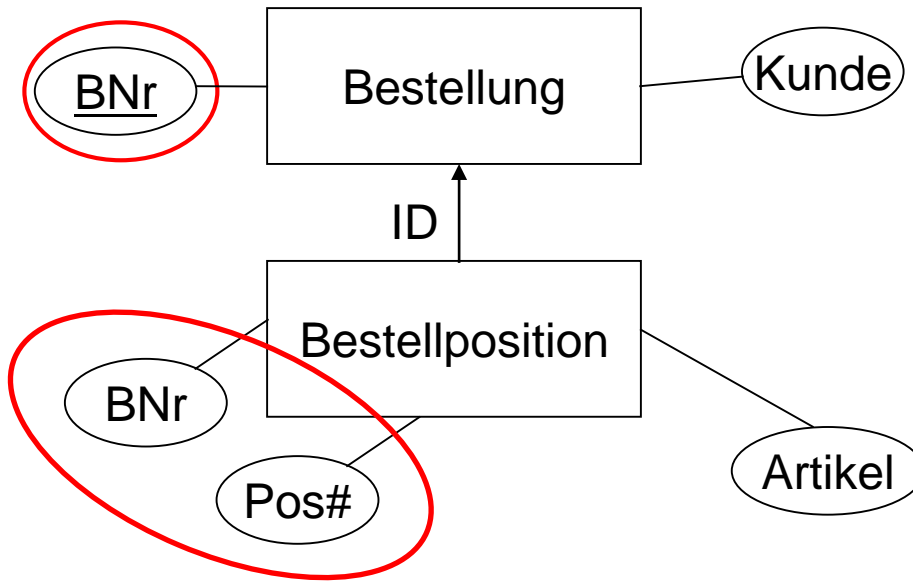


ID-abhängiger Entitätstyp



- Der Entitätstyp «Bestellposition» ist **ID-abhängig** vom Entitätstyp «Bestellung».
- Zu einer Bestellung (Primärschlüssel {BNr}) gehört eine **Menge** von Bestellpositionen.

ID-abhängiger Entitätstyp



- «Innerhalb» einer Bestellung brauchen wir ein **weiteres Attribut**, z.B. «Pos#», um die einzelnen Bestellpositionen zu **unterscheiden**.
- Es bildet also {BNr, Pos#} einen Schlüssel.

ID-abhängiger Entitätstyp

- Der Entitätstyp «Bestellposition» ist **ID-abhängig** vom Entitätstyp «Bestellung».
- **«1:M Relation»** zwischen Bestellung und Bestellposition(en).
- «Innerhalb» einer Bestellposition brauchen wir ein weiteres Attribut, z.B. «Pos#», um die einzelnen Bestellpositionen zu **unterscheiden**
 - Es bilden also **{BNr, Pos#}** einen **Schlüssel**
- Eine Entität des ID-abhängigen Typs kann «auf natürliche Weise» nur **innerhalb der Hierarchie** identifiziert werden:
 - Bsp. Kind durch Vornamen innerhalb der Familie

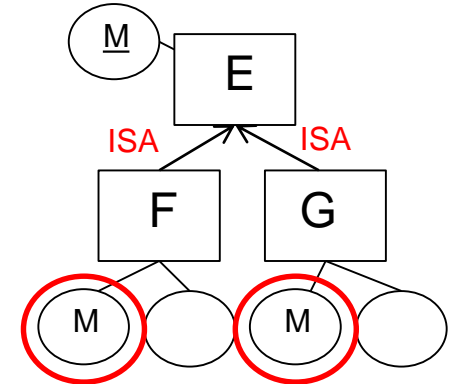
ID vs. ISA-abhängiger Entitätstyp

? Was ist der Unterschied von ID-abhängigem und ISA-abhängigem Entitätstyp?

ID vs. ISA-abhängiger Entitätstyp

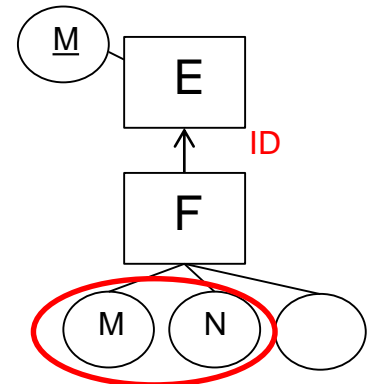
ISA-abhängig:

- Genauere Spezifikation (**Spezialisierung**)
- Ist Entitätstyp F von Entitätstyp E ISA-abhängig, so gilt:
 - Ist M die Menge der Primärschlüsselattribute in E,
 - so muss **M in F ein Schlüssel** sein.



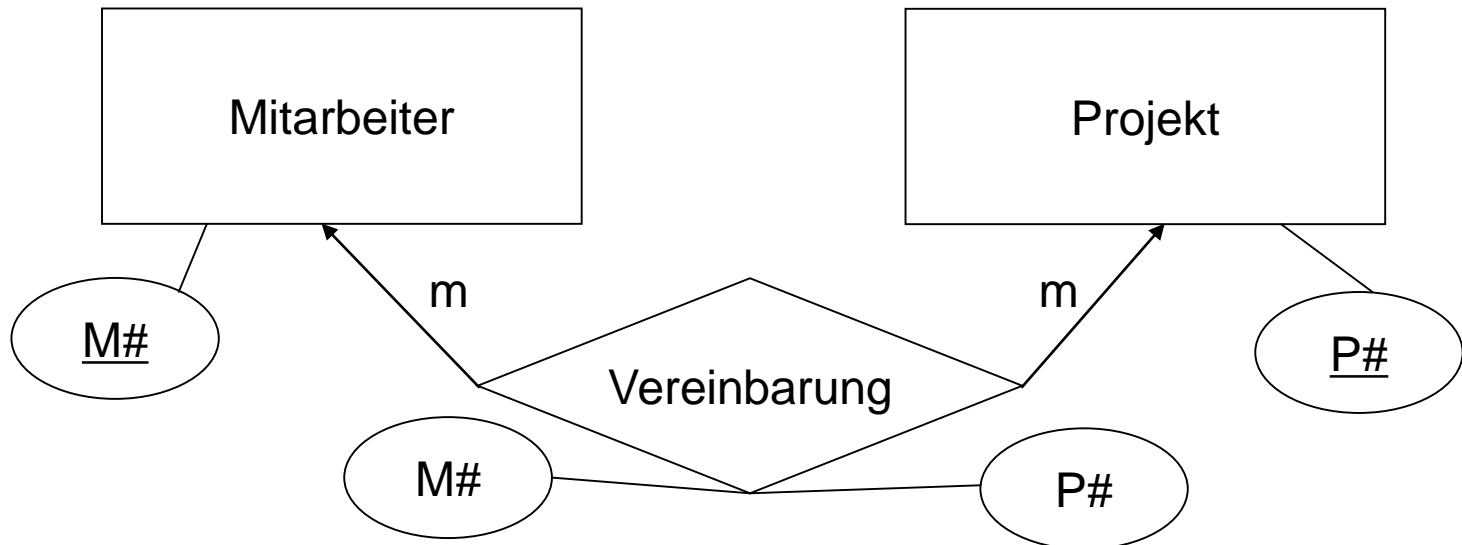
ID-abhängig:

- Hierarchie, hängt an "Oberklasse". Ist **nur innerhalb der Hierarchie definiert**
- Ist Entitätstyp F von Entitätstyp E ID-abhängig, so gilt:
 - Ist M die Menge der Primärschlüsselattribute in E,
 - so muss **$M \cup N$ ein Schlüssel in F** sein,
 - wobei **N eine zu M elementfremde Menge von Attributen von F ist** (min. 1 Element)



Zusammengesetzter Entitätstyp

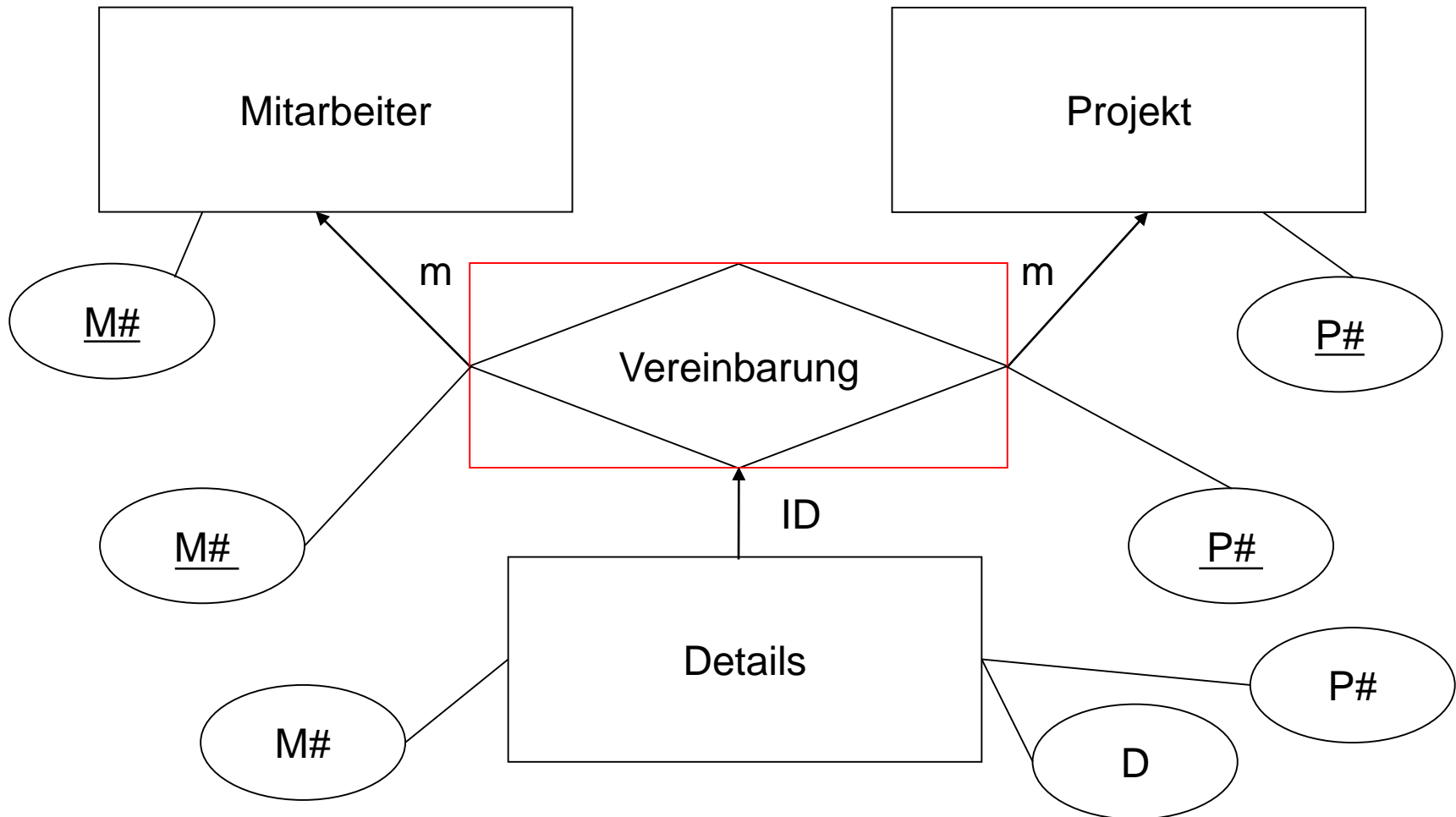
- **Zusammengesetzter** Entitätstyp (composite entity type).
- Wir behandeln noch einen letzten Sonderfall, der auftritt, wenn wir an Beziehungstypen Entitätstypen **anhängen** wollen.
- Ausgangslage:



- Wir wollen unterschiedlich viele (auch keine) Details zu jeder Vereinbarung festhalten können.

Zusammengesetzter Entitätstyp

- Zusammengesetzter Entitätstyp (composite entity type), Beispiel:



Zusammengesetzter Entitätstyp

- Wir hängen einen ID-abhängigen Entitätstyp «Details» an, der von Vereinbarung zu Vereinbarung **unterschiedlich viele** Details aufnehmen soll.
- Der Beziehungstyp wird «umgewandelt» in einen **zusammengesetzten Entitätstyp** (zeichnerisch: wir zeichnen ein Rechteck um den Beziehungstyp).

Es sei $\{M\#,P\#$ **Primärschlüssel** in Vereinbarung.

? Was können wir über **Schlüssel** in «Details» sagen?

Zusammengesetzter Entitätstyp

- Es ist mindestens ein weiteres Attribut nötig. So kann z.B. {M#,P#,D#} ein **Schlüssel** sein (wegen ID-Abhängigkeit).

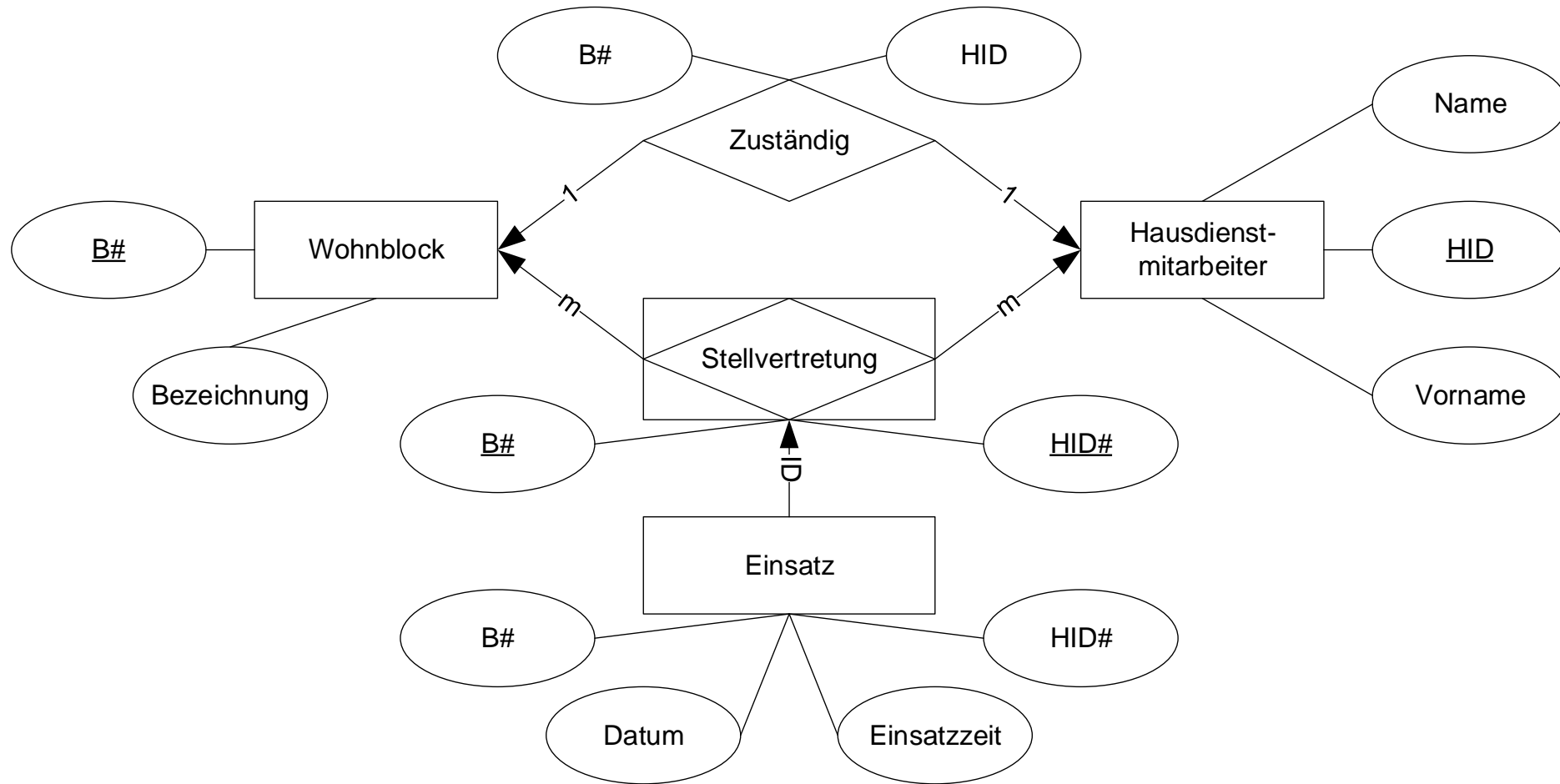
? Verständnisfrage: Warum ist «Details» ID-abhängig, und nicht ISA-abhängig?

Zusammengesetzter Entitätstyp

- ACHTUNG: An einem zusammengesetzten Entitätstypen **muss nicht zwingend eine ID-Abhängigkeit «hängen»**. Es kann durchaus auch ein anderer Beziehungstyp sein.

Hörsaalübung

- In einer grösseren Wohnsiedlung sind mehrere Hausdienstmitarbeiter für einzelne Wohnblocks zuständig. Für einen Wohnblock soll eine eindeutige Blocknummer (B#) sowie eine Bezeichnung festgehalten werden. Für Hausdienstmitarbeiter soll eine eindeutige Identifikationsnummer (HID) sowie ein Name und ein Vorname festgehalten werden. Jeder Hausdienstmitarbeiter ist für genau einen Wohnblock zuständig. Für jeden Wohnblock gibt es genau einen zugeteilten Hausdienstmitarbeiter. Einzelne Hausdienstmitarbeiter können aber für mehrere andere als Stellvertreter zugeteilt werden. Bei Stellvertretungen sollen zudem für jeden Einsatz als Stellvertreter das Einsatzdatum und die geleisteten Stunden (Einsatzzeit genannt) festgehalten werden (ein Hausdienstmitarbeiter kann für denselben Wohnblock mehrfach als Stellvertreter eingesetzt werden). Zeichnen Sie das dazu passende vollständige ER-Schema.



Und weiter...

Das nächste Mal: ERM, komplexeres Beispiel

