



UNIVERSITÄT
DES
SAARLANDES

Saarbrücken, 21.05.2015
Information Systems Group

Vorlesung „Informationssysteme“

Vertiefung Kapitel 4: Von (E)ER ins Relationenmodell

Erik Buchmann (buchmann@cs.uni-saarland.de)



Aus den Videos wissen Sie...

- ...welche Bestandteile das relationale Modell aufweist
 - Und wie sich diese in ER und EER modellieren lassen
- ...dass bei der Überführung von ER ins Relationenmodell
 - Schlüssel besonders beachtet werden müssen
 - manchmal Relationen verschmolzen werden dürfen
 - es manchmal mehrere Optionen gibt, um den gleichen Sachverhalt auszudrücken

- Vertiefung heute:
 - *Mehr zur Umwandlung von EER-Modellen zu UML-Klassendiagrammen*
 - Abbilden von schwierigen Fällen im ER-Modell auf das Relationenmodell
 - Umwandlung von EER-Modellen in das Relationenmodell

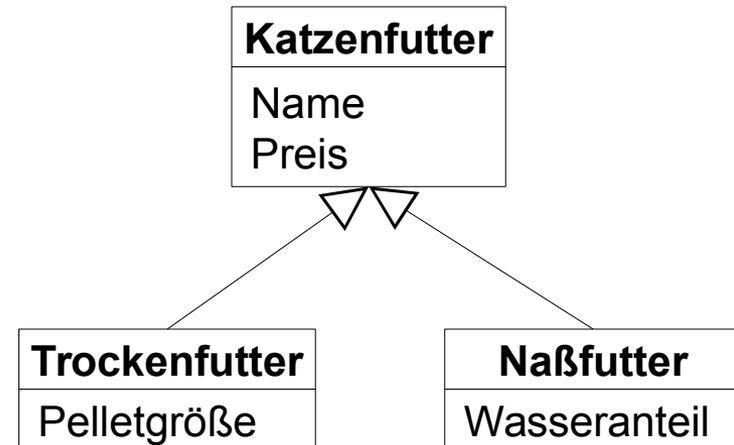
(Anmerkung: Im Folgenden wurden die Datentypen weggelassen)

A nighttime photograph of a university building with a large crowd of people gathered in front. The building has a dark roof with skylights and is illuminated by warm lights. A large crowd of people is standing in front of the building, and there are long light trails from a camera on the road in the foreground. A white banner with black text is overlaid on the image.

EER → UML-Klassendiagramm

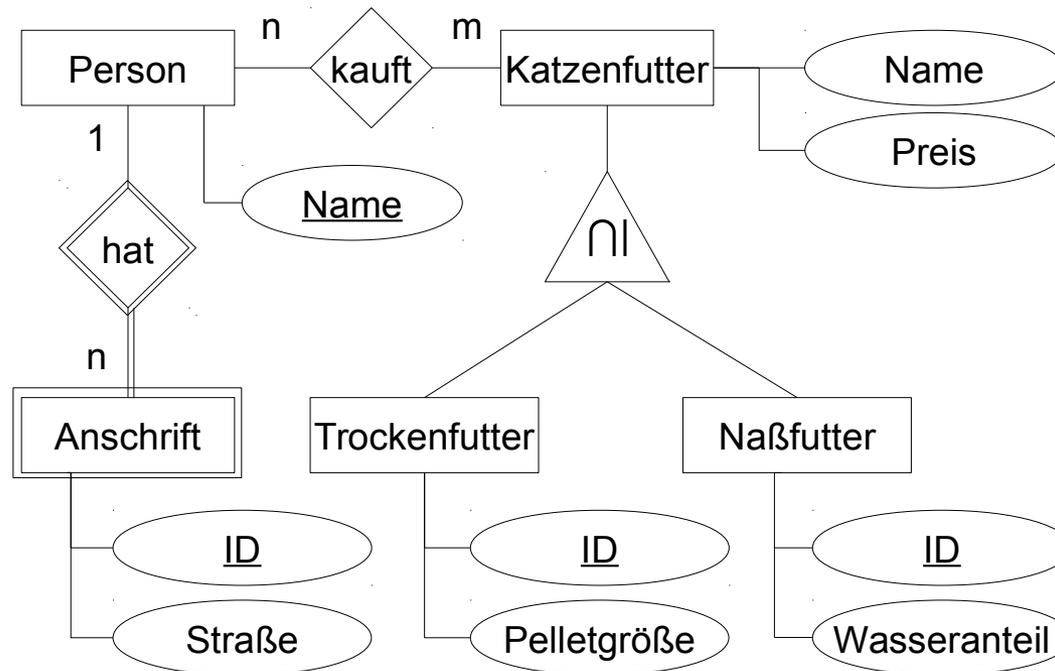
Das Klassendiagramm

- Übersicht über Aufbau und Zusammenspiel von objektorientiert verwalteten Daten
- Im Datenbankkontext wichtig für
 - Mit OO-Entwicklungsumgebung kompatible Schemadefinition



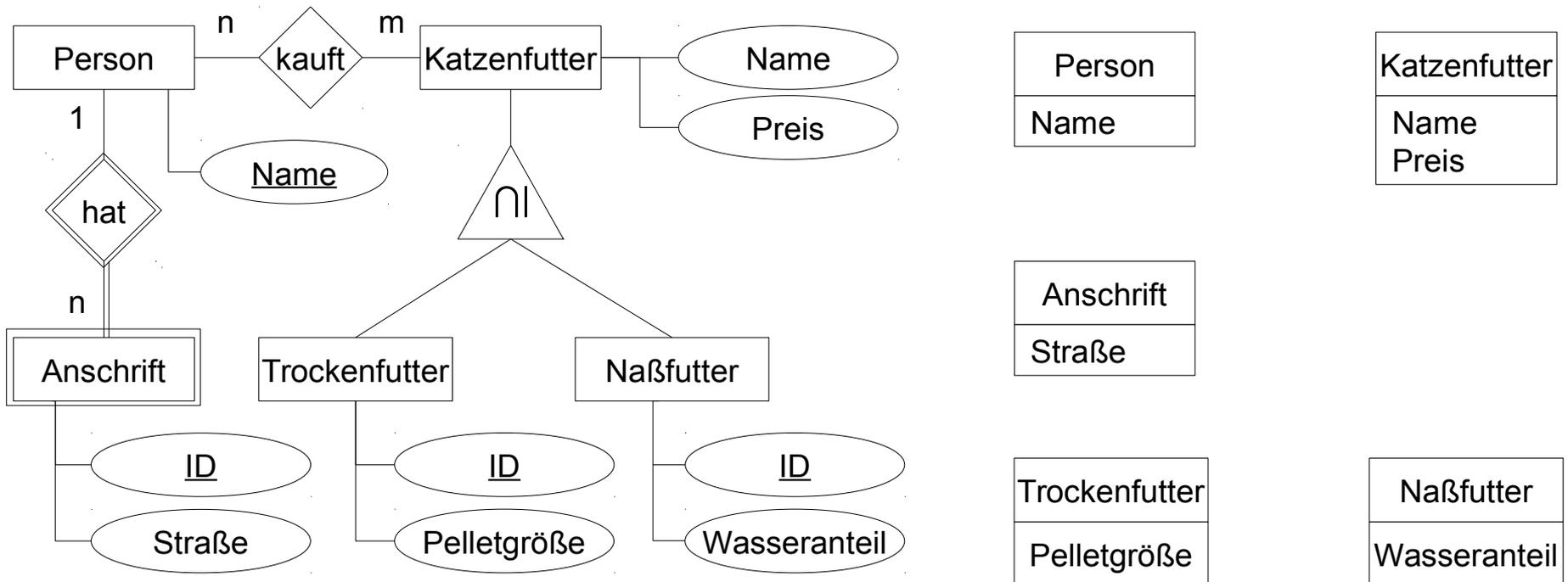
Laufendes Beispiel

- Entwickeln Sie daraus ein UML-Klassendiagramm



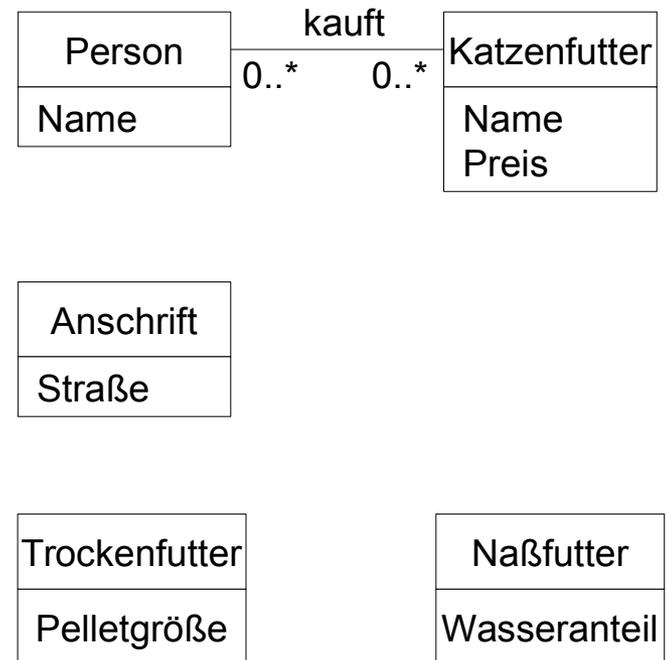
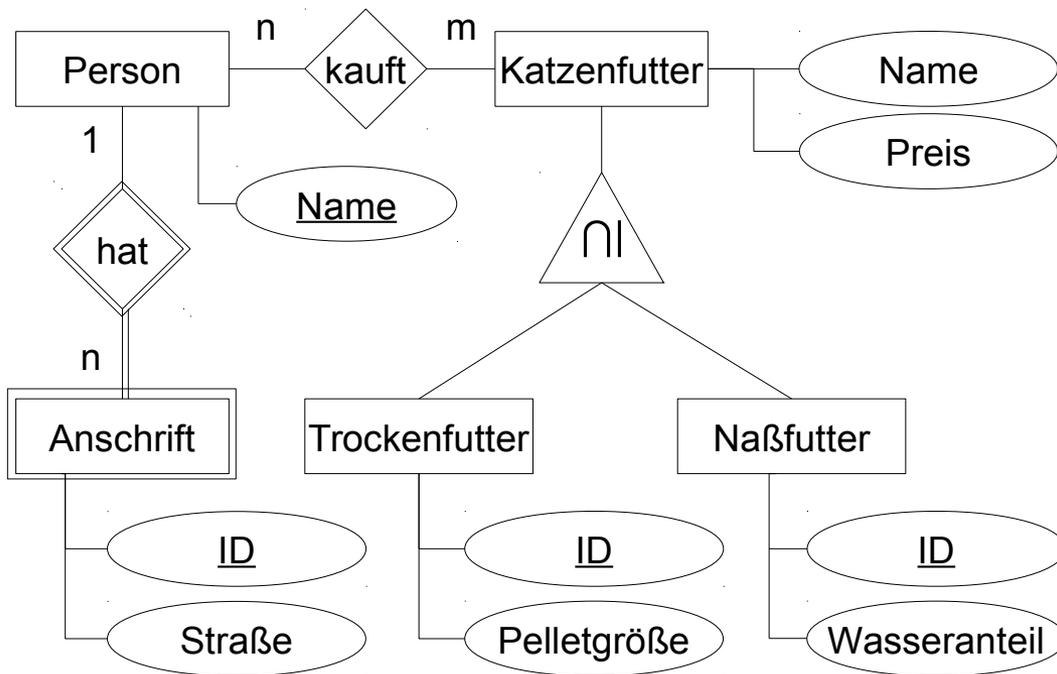
Schritt 1 von 5

- Entitätstypen in Klassen umwandeln
 - Künstliche Schlüssel können weggelassen werden, wenn sie nur eingeführt wurden, weil das EER-Modell sie erzwingt



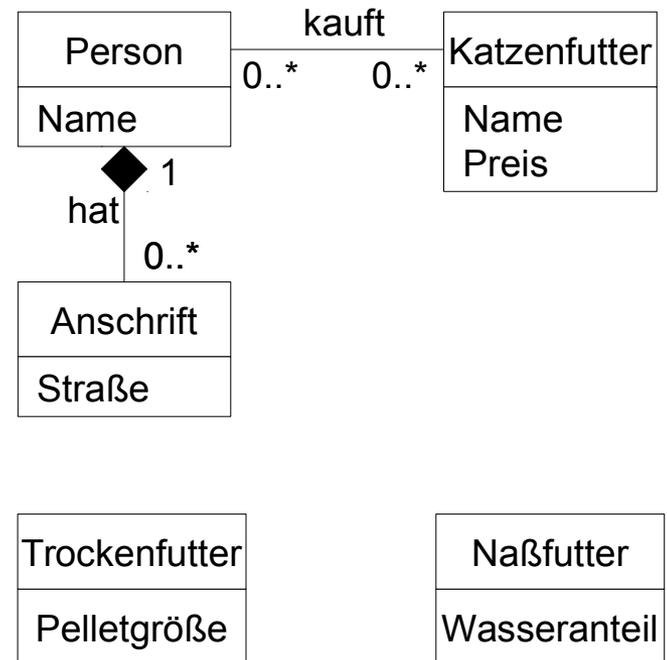
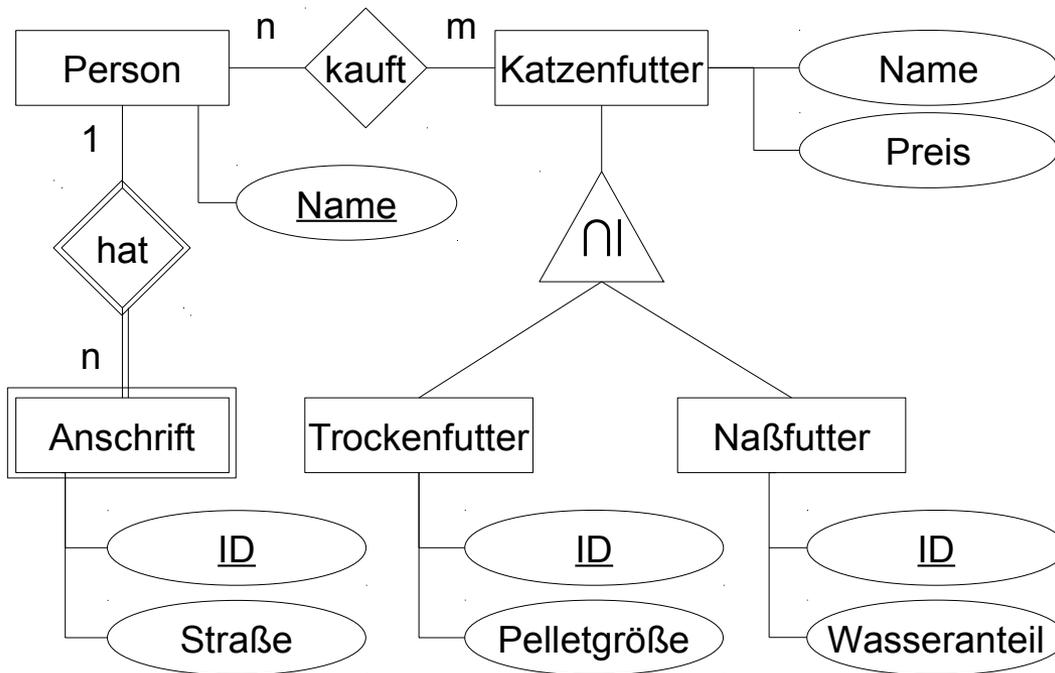
Schritt 2 von 5

- Beziehungstypen in Assoziationen umwandeln
 - Kardinalitäten zu UML-Multiplizitäten



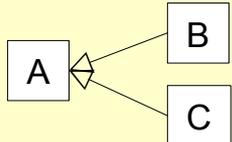
Schritt 3 von 5

- Alle existenzabhängigen Entitytypen werden zu Kompositionen
 - ebenso wie ER immer Kardinalität 1



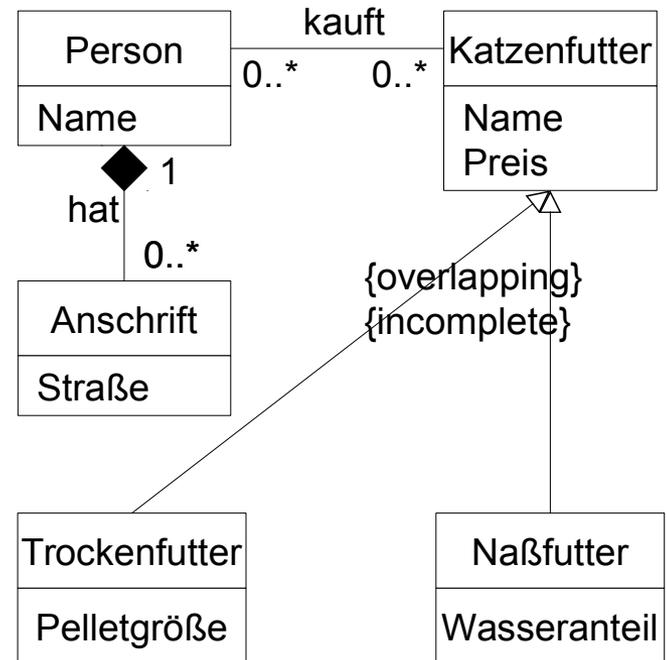
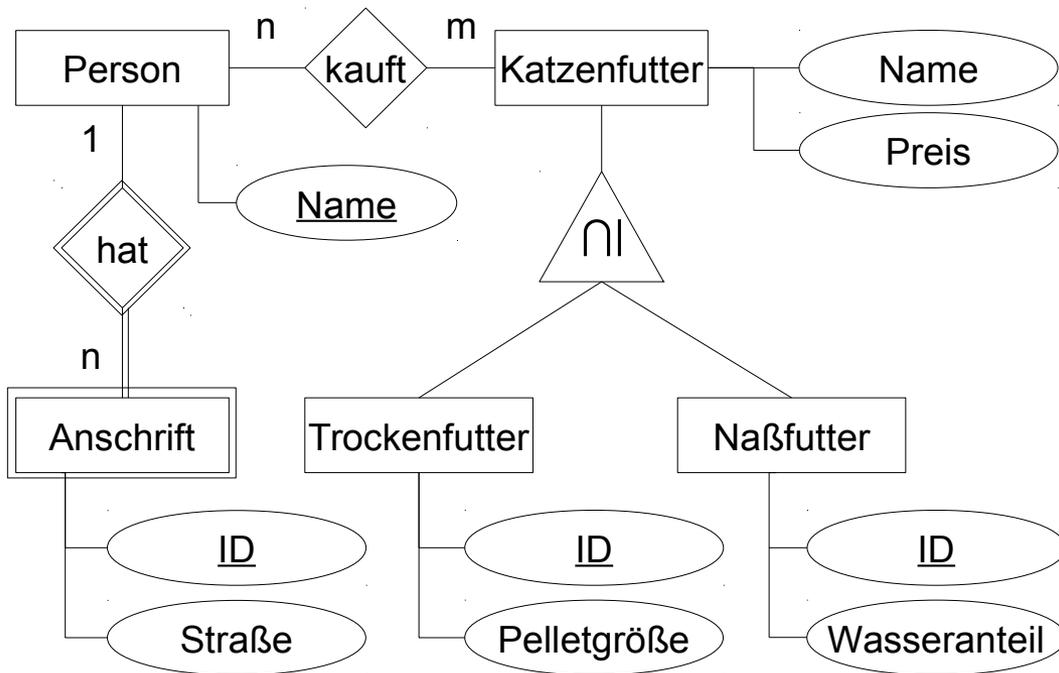
Schritt 4 von 5

- Alle Spezialisierungen, Generalisierungen, Partitionierungen werden zu Vererbungen
 - Prüfen Sie, welche Zusicherung geeignet ist {incomplete}/{complete}, {overlapping}/{disjoint}



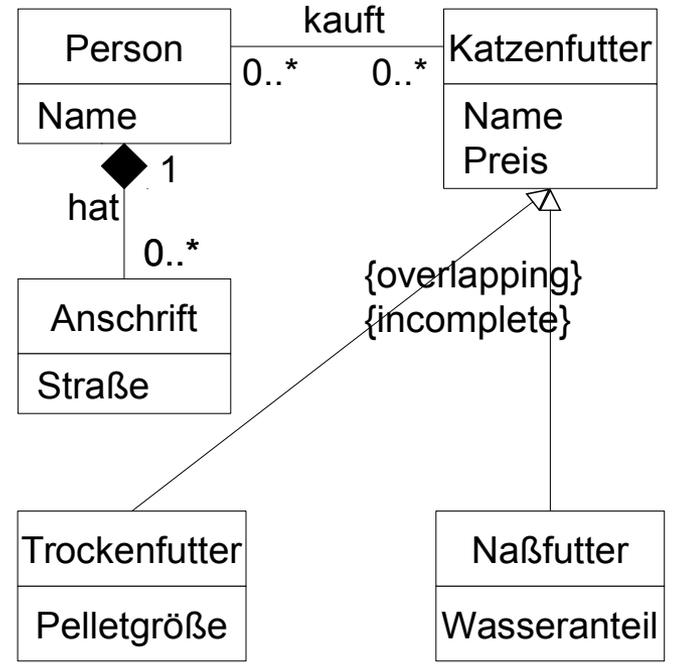
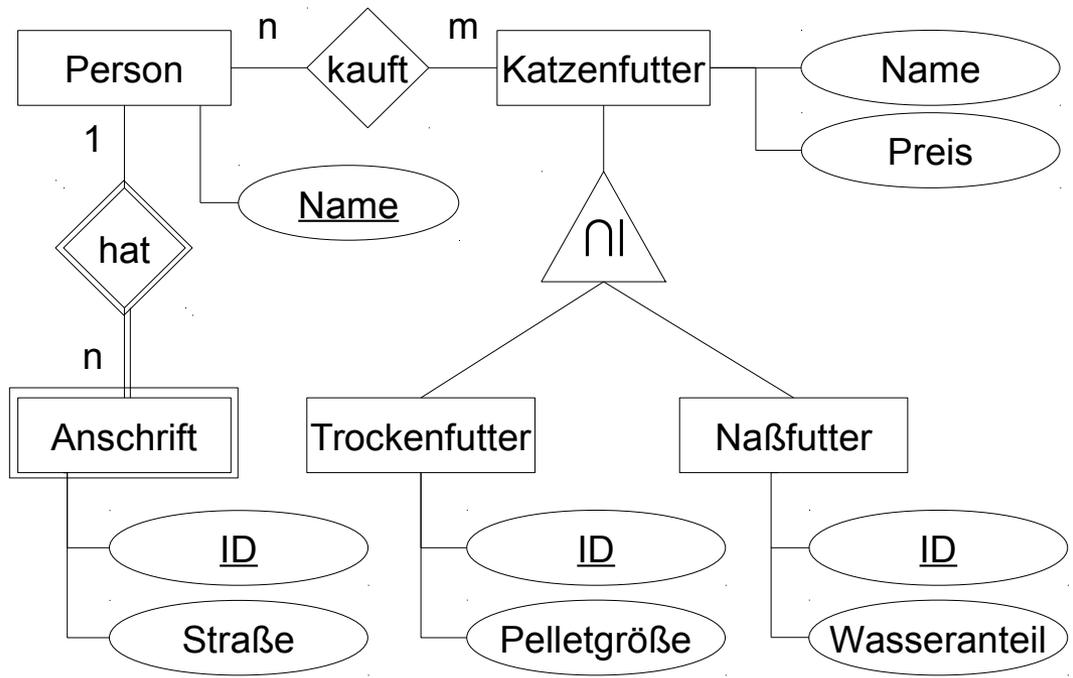
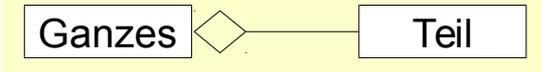
{incomplete}: Außer B und C noch weiter Unterklassen von A möglich
 {complete}: nicht möglich

{overlapping}: Instanzen von A dürfen gleichzeitig in B und C sein
 {disjoint}: dürfen nicht



Schritt 5 von 5

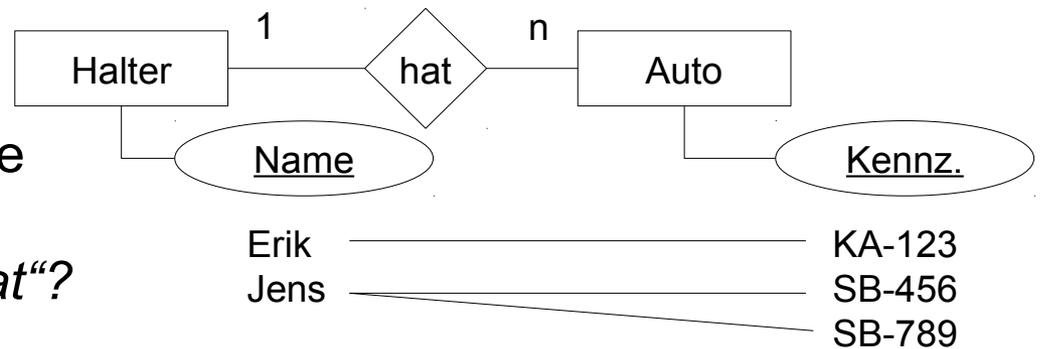
- Prüfen Sie, ob Sie „besteht aus“-Beziehungstypen in eine Aggregation umwandeln können (hier nicht der Fall)
- **Prüfen Sie, ob Ihr UML-Modell dasselbe ausdrückt wie Ihr (E)ER-Modell**



ER → Relationenmodell

Kapazitätserhaltende Abbildung

- Abbildung ins Relationenmodell soll so viele Zustände erlauben wie modelliert!
Was ist der Schlüssel in „hat“?



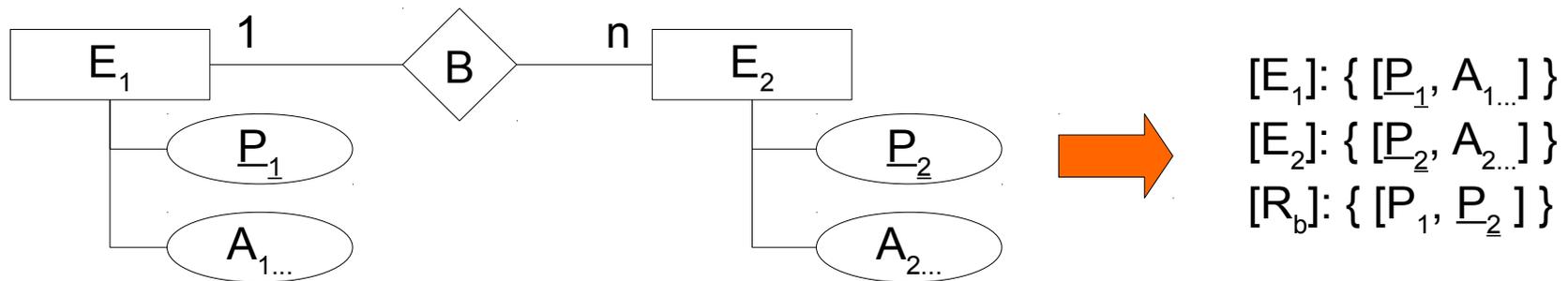
[Halter]: {[Name]}

[Auto]: {[Kennz.]}

- Kapazitätsvermindernd
 - [hat]: {[Name, Kennz.]}
 - Problem: Jens kann kein zweites Auto haben
- Kapazitätserhöhend
 - [hat]: {[Name, Kennz.]}
 - Problem: Ein Auto kann mehrere Halter haben
- Kapazitätserhaltend ← *Richtig*
 - [hat]: {[Name, Kennz.]}

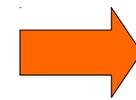
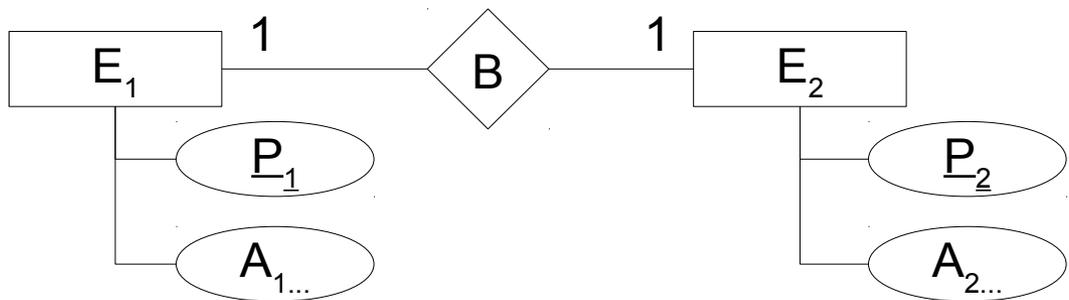
Zusammenfassung der Abbildungsvorschriften

ER-Konzept	abgebildet auf Relationenmodell
Entitytyp E_i Attribute A_i von E_i Primärschlüssel P_i von E_i	Relationenschema R_i Attribute A_i von R_i Primärschlüssel P_i von R_i
Beziehungstyp B zwischen E_1, E_2	Relationenschema R_b mit Attributen P_1, P_2
1:n	P_2 wird Primärschlüssel von R_b
1:1	P_1 oder P_2 wird Primärschlüssel von R_b
m:n	P_1 und P_2 werden Primärschlüssel von R_b



Zusammenfassung der Abbildungsvorschriften

ER-Konzept	abgebildet auf Relationenmodell
Entitytyp E_i Attribute A_i von E_i Primärschlüssel P_i von E_i	Relationenschema R_i Attribute A_i von R_i Primärschlüssel P_i von R_i
Beziehungstyp B zwischen E_1, E_2	Relationenschema R_b mit Attributen P_1, P_2
1:n	P_2 wird Primärschlüssel von R_b
1:1	P_1 oder P_2 wird Primärschlüssel von R_b
m:n	P_1 und P_2 werden Primärschlüssel von R_b

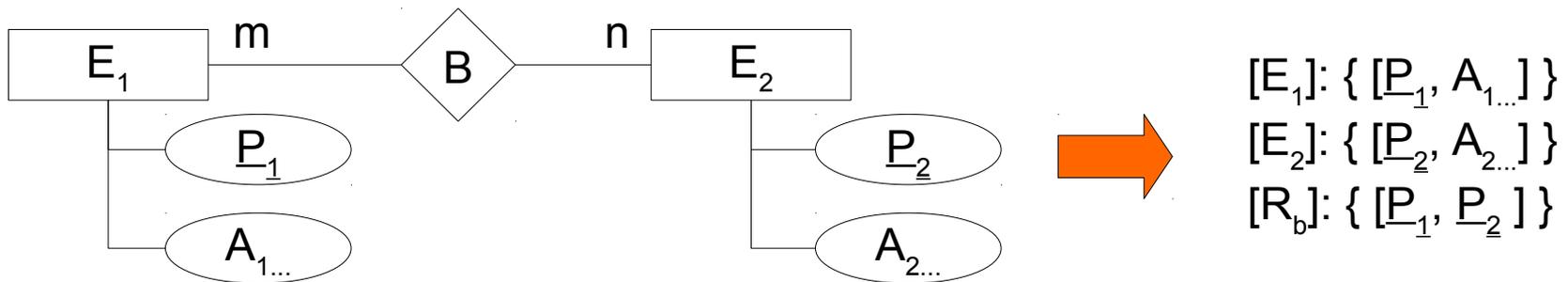


$[E_1]: \{ [P_1, A_{1\dots}] \}$
 $[E_2]: \{ [P_2, A_{2\dots}] \}$

 $[R_b]: \{ [P_1, P_2] \}$
 ODER
 $[R_b]: \{ [P_1, P_2] \}$

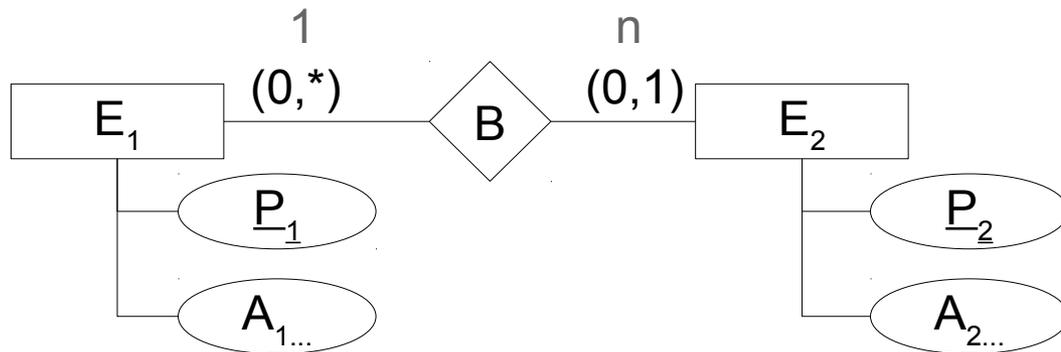
Zusammenfassung der Abbildungsvorschriften

ER-Konzept	abgebildet auf Relationenmodell
Entitytyp E_i Attribute A_i von E_i Primärschlüssel P_i von E_i	Relationenschema R_i Attribute A_i von R_i Primärschlüssel P_i von R_i
Beziehungstyp B zwischen E_1, E_2	Relationenschema R_b mit Attributen P_1, P_2
1:n	P_2 wird Primärschlüssel von R_b
1:1	P_1 oder P_2 wird Primärschlüssel von R_b
m:n	P_1 und P_2 werden Primärschlüssel von R_b



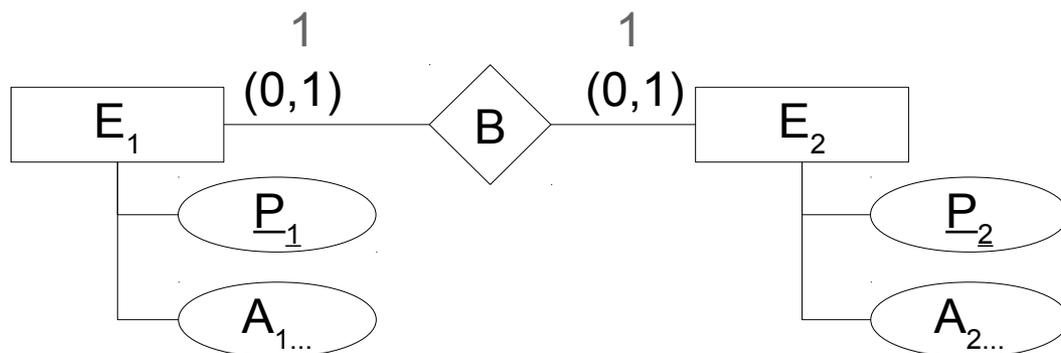
1:n oder 1:1-Beziehungstyp verschmelzen

- Funktioniert nur, wenn Fremdschlüssel *NULL* sein darf, d.h., Datenobjekt muss nicht zwingend existieren



Verschmelzen mit der n-Seite

$$[E_1]: \{ [\underline{P}_1, A_{1\dots}] \}$$

$$[E_2]: \{ [\underline{P}_2, A_{2\dots}, P_1] \}$$


Aussuchen, mit welcher Seite verschmolzen wird

$$[E_1]: \{ [\underline{P}_1, A_{1\dots}] \}$$

$$[E_2]: \{ [\underline{P}_2, A_{2\dots}, P_1] \}$$

oder

$$[E_1]: \{ [\underline{P}_1, A_{1\dots}, P_2] \}$$

$$[E_2]: \{ [\underline{P}_2, A_{2\dots}] \}$$

Schwierige Abbildungen

A nighttime photograph of a large, multi-story building with a dark roof and many windows, some of which are illuminated from within. The building is partially covered in ivy. To the left of the building is a large, dark, abstract sculpture made of several tall, thin, rectangular panels. In the foreground, a large crowd of people is gathered, and there are long, horizontal light trails in red and yellow, suggesting a long exposure. The sky is a deep blue with some clouds. A white rectangular box is overlaid on the image, containing the text 'Schwierige Abbildungen'.

Rekursive Beziehungen

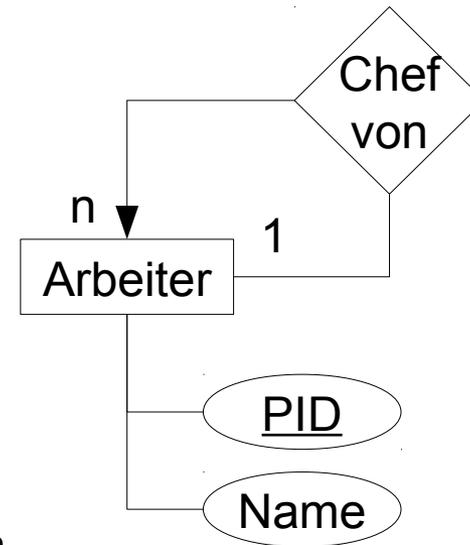
- Beispiel: Arbeiter ist Chef von anderen Arbeitern

[Arbeiter]: { [PID, Name] }

[Chefvon]: { [PID1, PID2] }

Alternativ

[Arbeiter]: { [PID, Name, hatChefPID] }

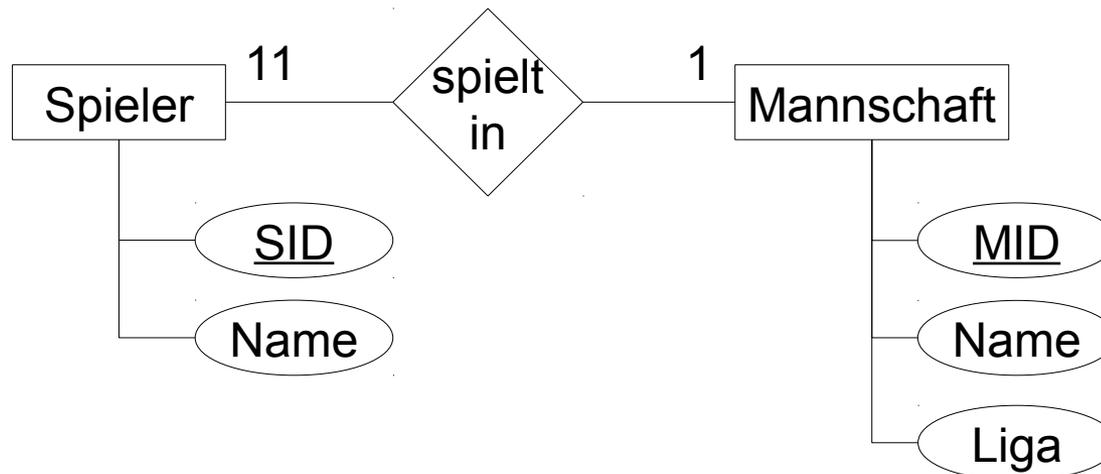


- Was möchte ich ausdrücken?

- Ein Chef kann mehrere Arbeiter unter sich haben
- Ein Arbeiter *kann* genau einen Chef haben
Alternativ: Ein Arbeiter *muss* genau einen Chef haben
- Ein Chef darf nicht sich selber als Chef haben

Kardinalitäten jenseits von 0, 1, n

- In einer Mannschaft spielen genau 11 Spieler
- Umsetzung wie eben besprochen
 - [Spieler]: { [SID, Name] }
 - [Mannschaft]: { [MID, Name, Liga] }
 - [spieltin]: { [SID, MID] }
- genau 11 Spieler so nicht abbildbar



Kardinalitäten jenseits von 0, 1, n

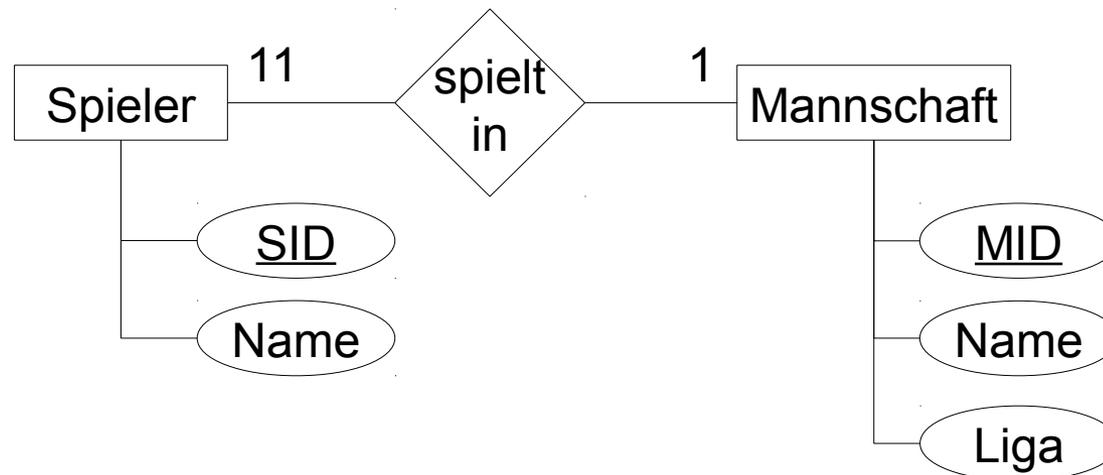
■ Umsetzung mit „Inlining“

- [Spieler]: { [SID, Name] }

[Mannschaft]: { [MID, Name, Liga, SID1, SID2, SID3, SID4, SID5, SID6, SID7, SID8, SID9, SID10, SID11] }

- 11 Spieler abbildbar, aber nicht sichergestellt dass

- unterschiedliche Spieler
- jeder Spieler nur eine Mannschaft
- Reihenfolge der SIDs spielt eine Rolle
→ **bitte nicht nachmachen**



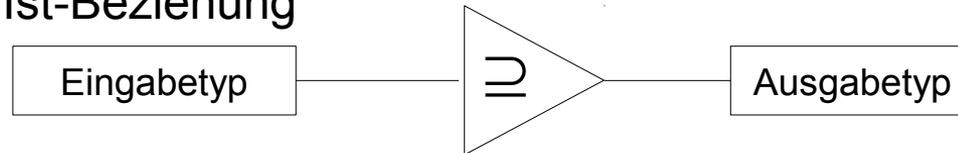
EER → Relationenmodell

A nighttime photograph of a university building with a large crowd of people gathered in front. The building has a dark roof with skylights and is illuminated by warm lights. A large crowd of people is standing in front of the building, and there are long light trails from a camera on the road in the foreground. The sky is dark blue with some clouds. A white text box is overlaid on the image, containing the text 'EER → Relationenmodell'. On the left, there is a large, dark, abstract sculpture. In the background, a church spire is visible against the sky. The overall scene suggests a social event or a gathering at a university.

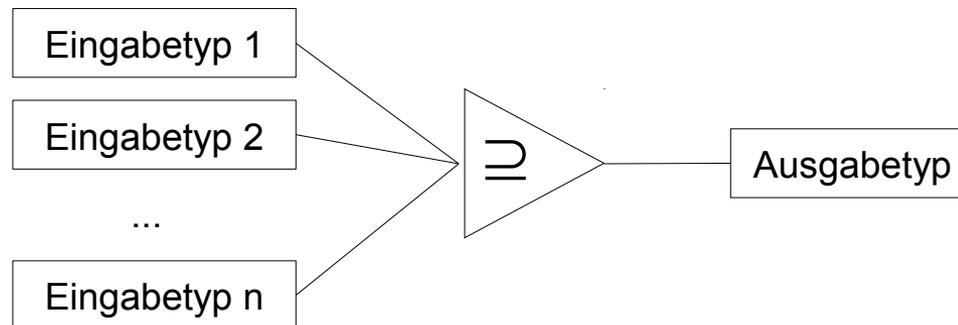
Das EER-Modell: Typkonstruktor

■ Modellierungskonstrukt ergänzt das ER-Modell um

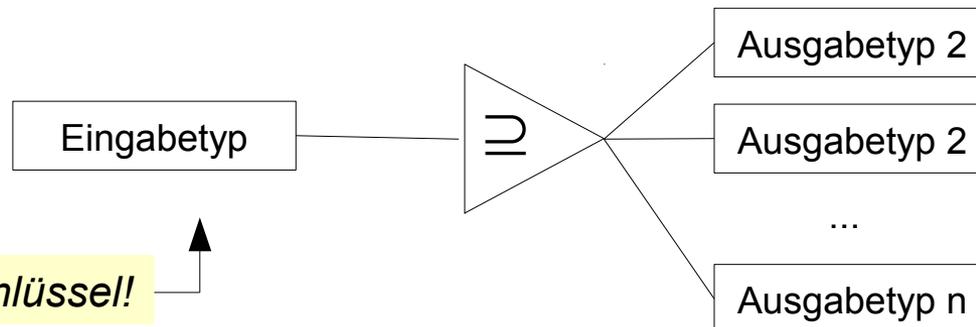
■ Spezialisierung / Ist-Beziehung



■ Generalisierung



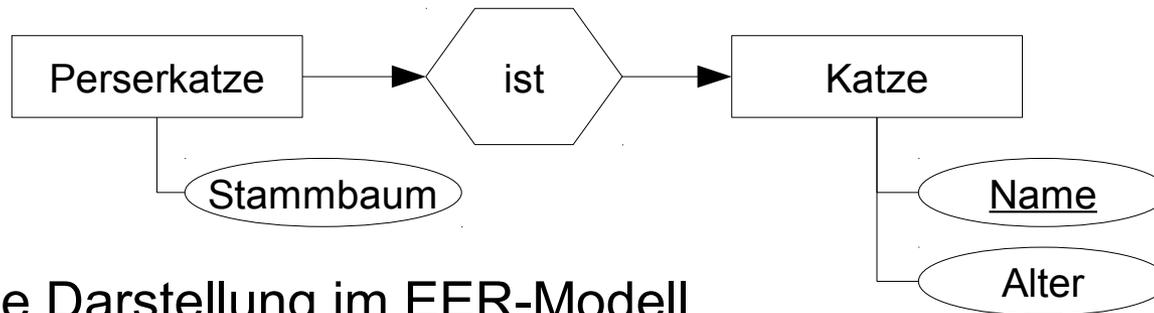
■ Partitionierung



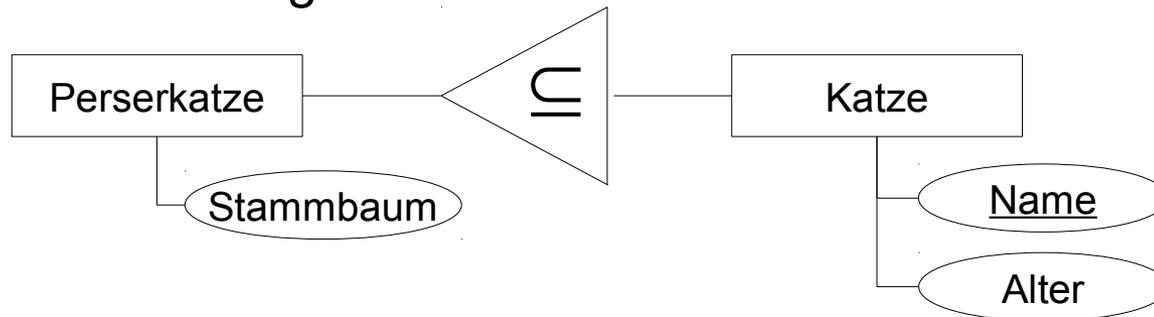
Auf dieser Seite steht der Schlüssel!

Spezialisierung

■ ER-Modell



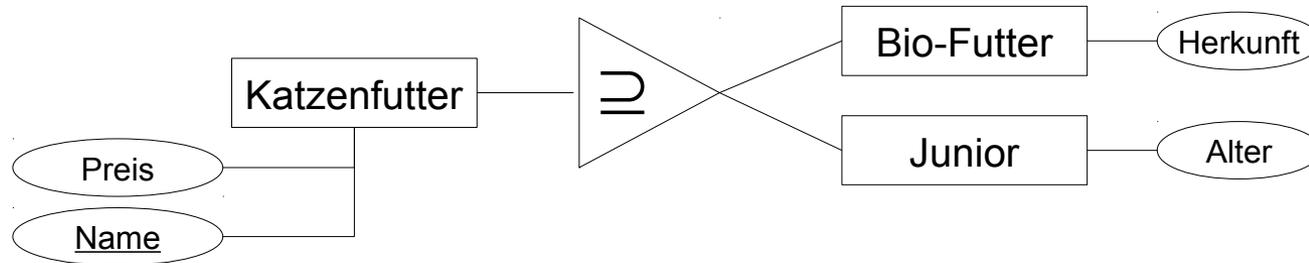
■ Äquivalente Darstellung im EER-Modell



Umwandlung ins Relationenmodell daher ebenso wie ER-Spezialisierung, drei Alternativen:

- 1. [Katze]: { [Name, Alter] }, [Perserkatze]: { [Name, Stammbaum] }
- 2. [Katze]: { [Name, Alter] }, [Perserkatze]: { [Name, Alter, Stammbaum] }
- 3. [Katze]: { [Name, Alter, Stammbaum] } ← Nullwerte, keine gute Idee

EER-Partitionierung



■ Relationenmodell identisch zur ER-Generalisierung

■ Alternative 1

[Katzenfutter]: { [Name, Preis] }

[Bio-Futter]: { [Name, Herkunft] }

[Junior]: { [Name, Alter] }

■ Alternative 2

[Katzenfutter]: { [Name, Preis] }

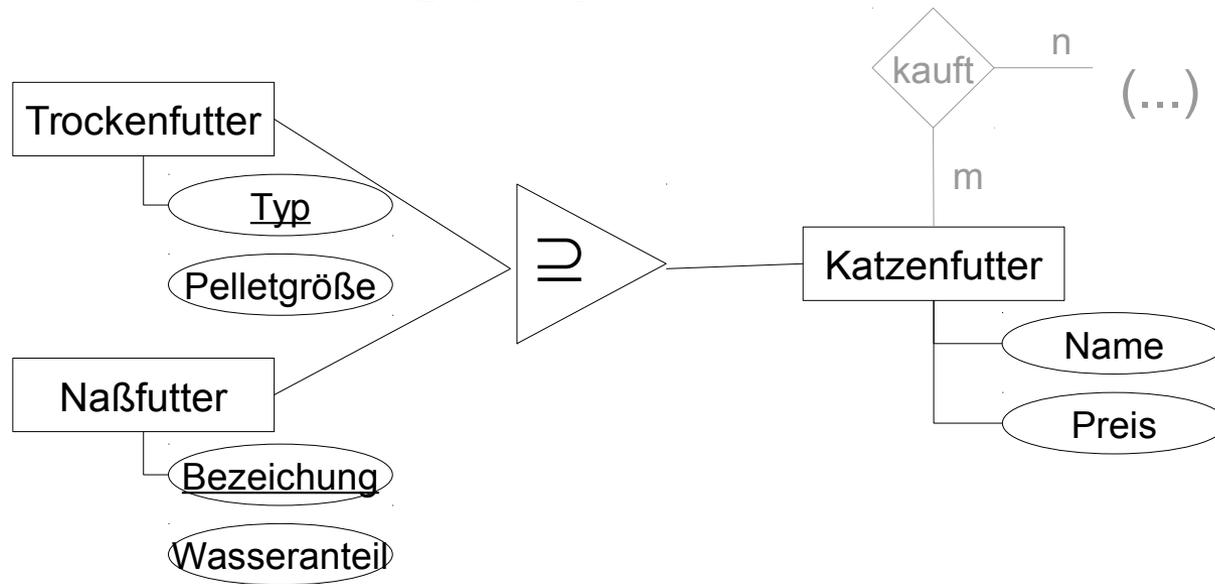
[Bio-Futter]: { [Name, Preis, Herkunft] }

[Junior]: { [Name, Preis, Alter] }

■ Alternative 3 ← Nullwerte, keine gute Idee

[Katzenfutter]: { [Name, Preis, Herkunft, Alter] }

EER-Generalisierung (1/2)

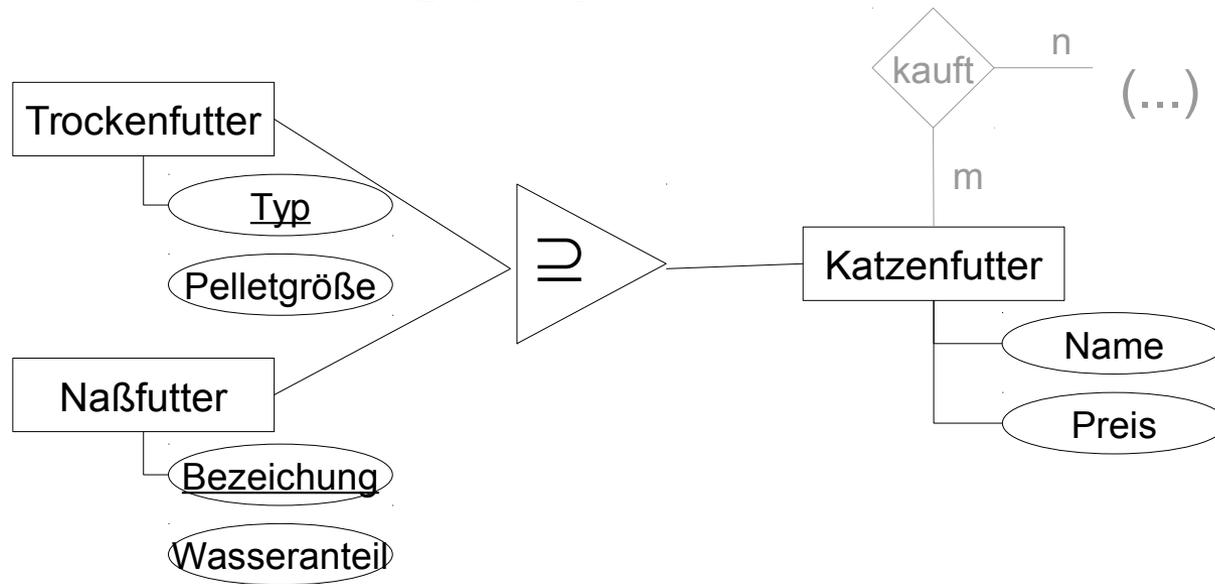


- falsch: Attribute von Katzenfutter in Trockenfutter, Naßfutter übernehmen
- [Trockenfutter]: { [Typ, Pelletgröße, Name, Preis] }
- [Naßfutter]: { [Bezeichnung, Wasseranteil, Name, Preis] }



Problem: es gibt keinen Schlüssel für die „kauft“-Beziehung

EER-Generalisierung (2/2)

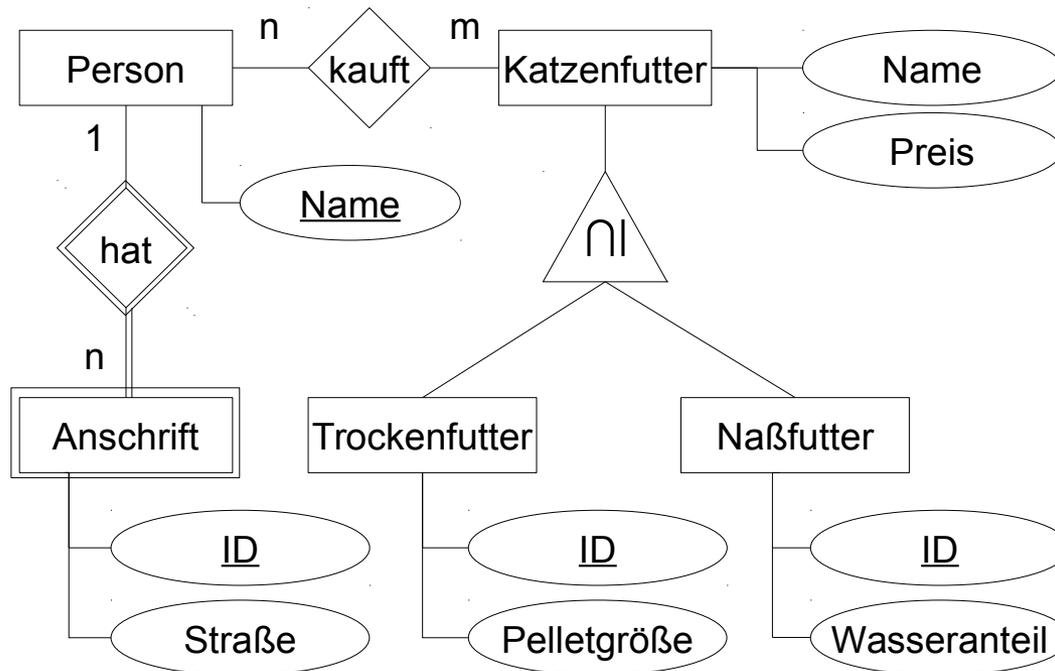


- funktioniert: künstlicher Schlüssel bei Katzenfutter, 1:1-Bez. mit Trocken- und Naßfutter verschmelzen (entspricht mehrfacher Spezialisierung)
- [Katzenfutter]: { [KID, Name, Preis] }
[Trockenfutter]: { [Typ, Pelletgröße, KID] }
[Naßfutter]: { [Bezeichnung, Wasseranteil, KID] }

Nachteil: Fremdschlüssel muss NULL sein dürfen, und dasselbe Katzenfutter darf mehrere Trockenfutter / Naßfutter sein.

Komplexes Beispiel

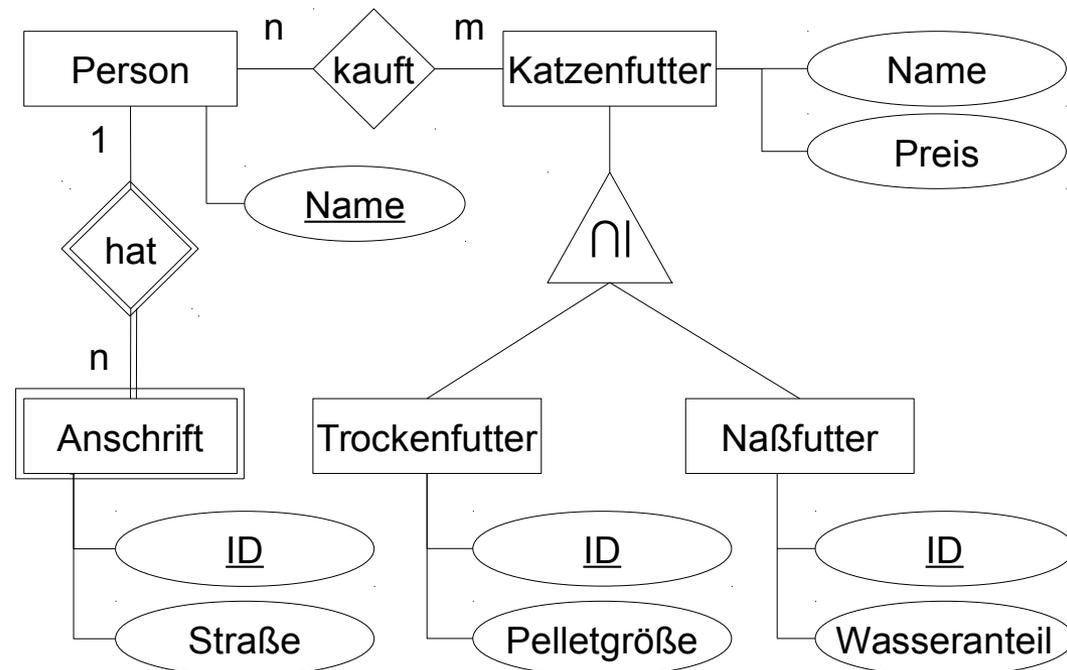
- Entwickeln Sie daraus ein Relationenmodell



Entitytypen in Relationen

[Person]: { [Name] }
[Anschritt]: { [ID, Straße] }
[Katzenfutter]: { [Name, Preis] }
[Trockenfutter]: { [ID, Pelletgröße] }
[Naßfutter]: { [ID, Wasseranteil] }

Als Arbeitsgrundlage
einfach alle Attribute
und Schlüssel aus dem
Modell übernehmen



Spezialisierungen, Generalisierungen, Partitionierungen

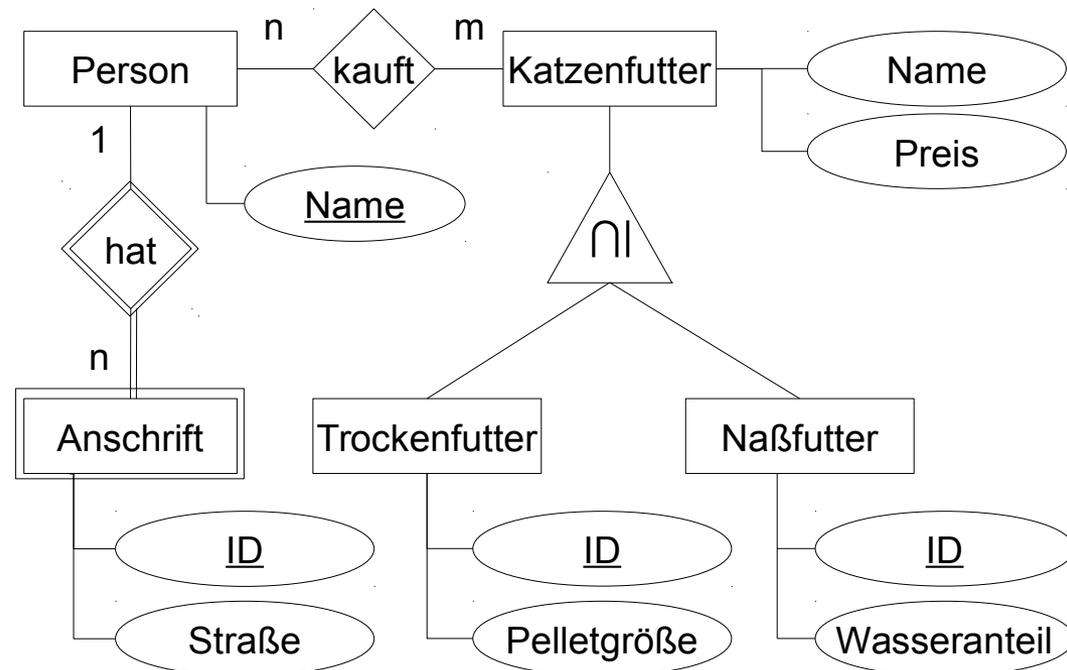
[Person]: { [Name] }

[Anschrift]: { [ID, Straße] }

[Katzenfutter]: { [KID, Name, Preis] }

[Trockenfutter]: { [ID, Pelletgröße, KID] }

[Naßfutter]: { [ID, Wasseranteil, KID] }



Beziehungstypen in Relationen umwandeln

[Person]: { [Name] }

[Anschritt]: { [ID, Straße] }

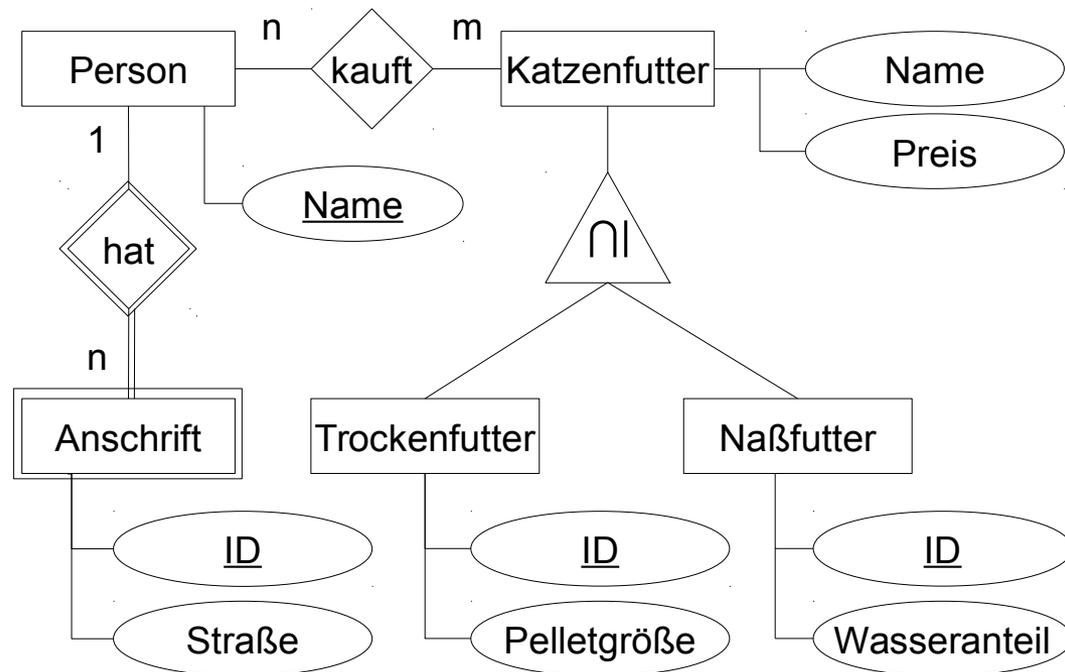
[Katzenfutter]: { [KID, Name, Preis] }

[Trockenfutter]: { [ID, Pelletgröße, KID] }

[Naßfutter]: { [ID, Wasseranteil, KID] }

[kauft]: { KID, Name }

[hat]: { Name, ID }



Beziehungsrelation erhält jeweils Primärschlüssel der beteiligten Relationen

1:1 und 1:n Beziehungen verschmelzen falls möglich

[Person]: { [Name] }

[Anschritt]: { [ID, Name, Straße] }

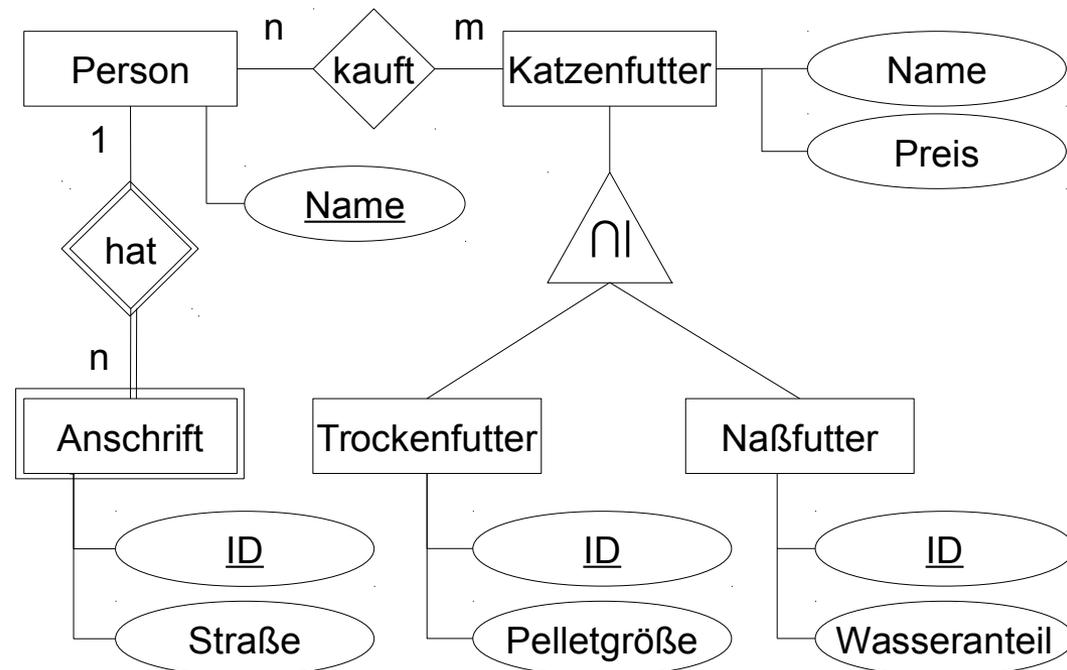
[Katzenfutter]: { [KID, Name, Preis] }

[Trockenfutter]: { [ID, Pelletgröße, KID] }

[Naßfutter]: { [ID, Wasseranteil, KID] }

[kauft]: { KID, Name }

Beim Verschmelzen von „hat“ mit „Anschritt“ muss „Name“ als Primärschlüssel enthalten sein, damit Existenzabhängigkeit erhalten bleibt



A nighttime photograph of a university building with a large crowd of people gathered in front. The building is illuminated by warm lights, and the sky is a deep blue. A large, dark, abstract sculpture is visible on the left. Light trails from a moving vehicle are visible in the foreground. A white text box is overlaid in the center.

Zum Abschluss

Wie geht es weiter?

- bis Sonntag, 24.05., 12 Uhr
 - Abgabe der vierten Gruppenaufgabe als PDF-Datei in Moodle
- bis Montag, 25.05., 12 Uhr
 - Quiz: Relationenalgebra
- Dienstag, 26.05., GHH 12-14 Uhr: Tutoriumstermin
 - Besprechung von Aufgabenblatt 4: UML-Klassendiagramm
 - nächstes Aufgabenblatt: Umwandlung (E)ER ins Relationenmodell
- Donnerstag, 28.05., GHH 10-12 Uhr: Präsenztermin
 - Schwierige Fälle bei der relationalen Algebra