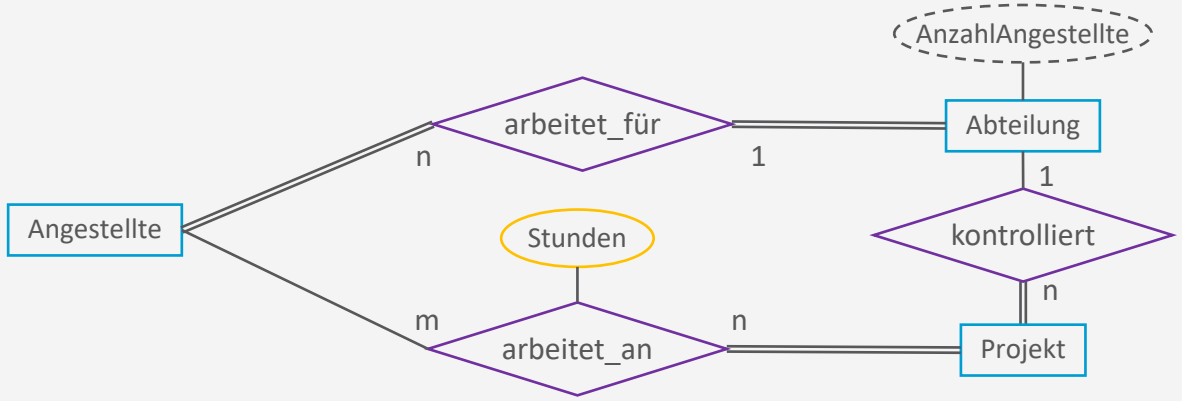
Universität Münster

Datenbank-Modellierung

Datenbanken



wissen.leben

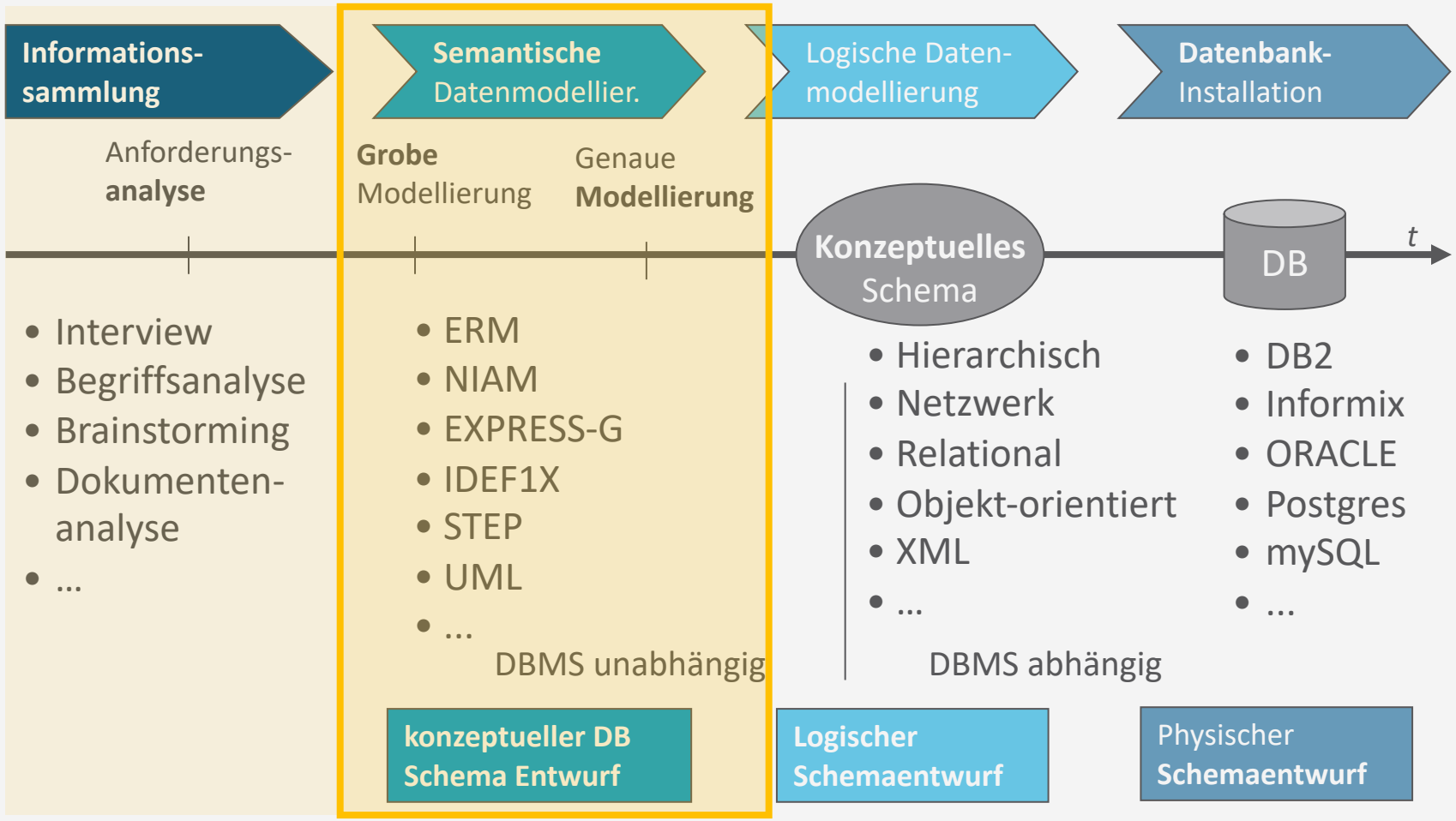
Tanya Braun
Arbeitsgruppe Date Science, Institut für Informatik

Inhalte: Datenbanken (DBs)

1. **Einführung**
   - Anwendungen
   - Datenbankmanagementsysteme
2. **Datenbank-Modellierung**
   - Entity-Relationship-Modell (ER-Modell)
   - Beziehung zwischen ER und UML
3. **Das relationale Modell**
   - Relationales Datenmodell (RM)
   - Vom ER-Modell zum RM
   - Relationale Algebra als Anfragesprache
4. **Datenbank-Entwurf**
   - Funktionale Abhängigkeiten
   - Normalformen
5. **Structured Query Language (SQL)**
   - Datendefinition
   - Datenmanipulation
6. **Anfrageverarbeitung**
   - Architektur
   - Indexierung
   - Anfragepläne, Optimierung
7. **Transaktionen**
   - Transaktionsverarbeitung, Schedules, Sperren
   - Wiederherstellung
8. **Verteilte Datenbanken**
   - Fragmentierung, Replikation, Allokation; CAP
   - Anfragebeantwortung, föderierte Systeme

Phasen des DB-Entwurfs

- Ausblick: Von der Anwendung her
  - Teil von 2. DB-Modellierung
    - Methode: ERM
  - Teil von 3. Das relationale Datenmodell
    - Methode: relationale Modellierung
  - Teil von 4. DB-Entwurf
  - Teil von 5. SQL & Übergang zu „Hinter den Kulissen“



Überblick: 2. Datenbank-Modellierung

A. Motivation
- Modelle und Modellierung

B. Entity-Relationship-Modell (ER)
- Komponenten
- Von Anforderungen zum ER-Modell
- Interpretation von ER-Modellen

C. Enhanced ER-Modell (EER)
- Spezialisierung, Generalisierung
- Modellierungsvorgehen
- Beziehung zu UML

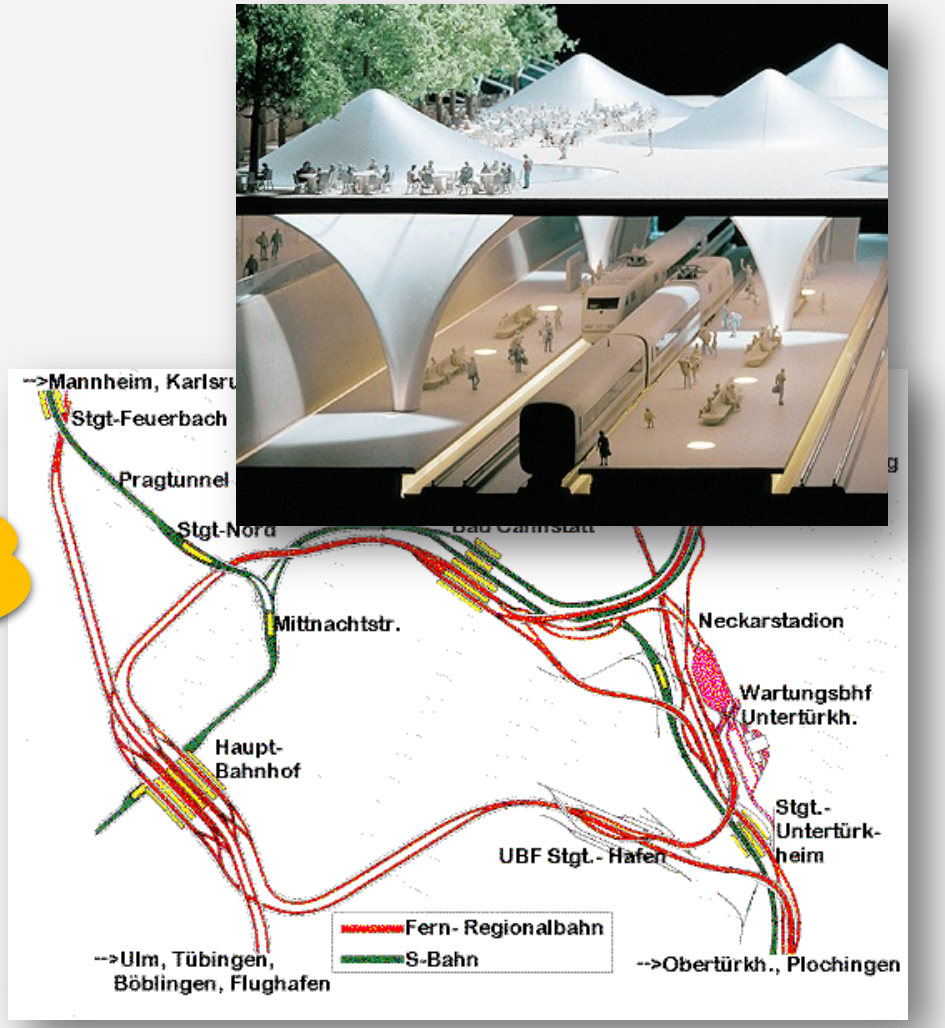
D. Dokumentation & Bewertung
- Dokumentation
- Qualitätskriterien EER

Modelle

- Modell: Abbild, Vorbild
- Typen von Modellen
  - Konkrete / abstrakte Modelle
    - Prototyp eines neuen Laptop / Erlösmodell
  - Konkrete / abstrakte Originale
    - Laptop / Personalentwicklung
- Was wird modelliert?
  - Struktur: Elemente eines Originals
  - Eigenschaften: Attribute der Elemente
  - Beziehungen zwischen Elementen
  - Verhalten: Dynamik der Elemente

Welche Modelle gibt es in der Informatik?

Was davon bildet ein DB-Modell (Schema) ab? Und was nicht?



Warum das Modell und nicht das Original?

- Motivation für Modellierungen
  - Operationen, die am Original nicht oder nur mit größerem Aufwand durchführbar sind
    - Ermöglichen Simulation
  - Untersuchen und Verstehen komplexer Zusammenhänge
  - Kommunikationsgrundlage
  - Fixieren von Anforderungen
- Verwendung bestimmt, was modelliert wird und was nicht:
  - Gebäudemodell: optischer Eindruck
  - Grundriss: Grundstücks- und Raumeinteilung
  - Gewerkeplan: Bauabwicklung
  - Kostenplan: Finanzierung
- Verwendung sollte auch die Modellierungssprache bestimmen

Überblick: 2. Datenbank-Modellierung

A. Motivation
- Modelle und Modellierung

B. Entity-Relationship-Modell (ER)
- Komponenten
- Von Anforderungen zum ER-Modell
- Interpretation von ER-Modellen

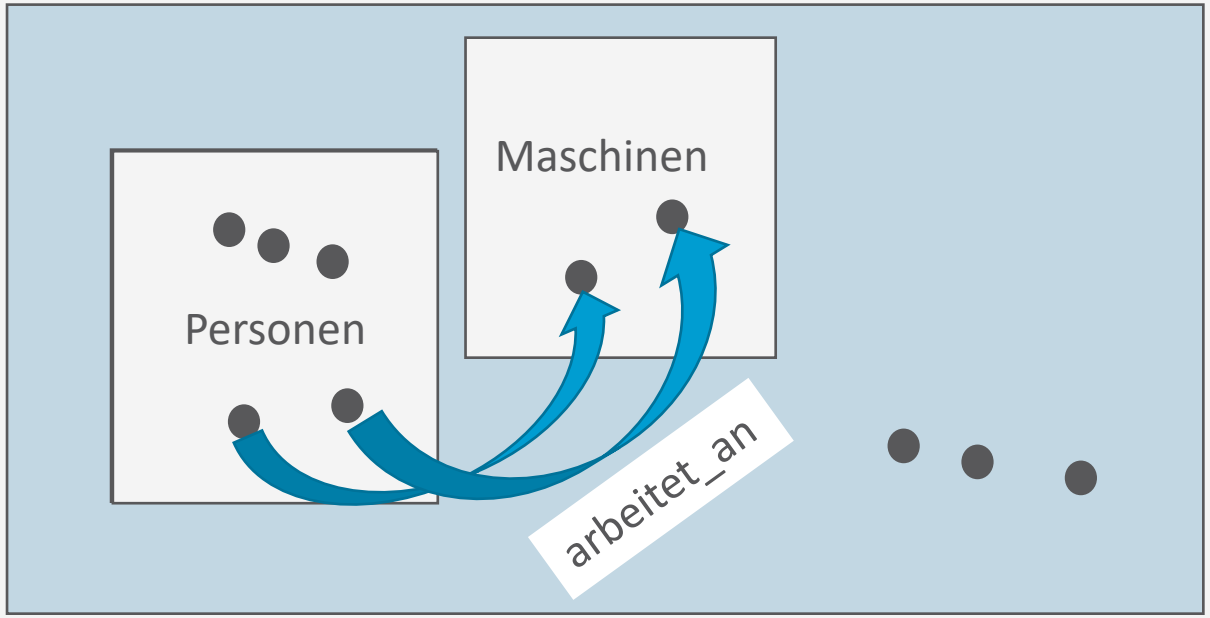
C. Enhanced ER-Modell (EER)
- Spezialisierung, Generalisierung
- Modellierungsvorgehen
- Beziehung zu UML

D. Dokumentation & Bewertung
- Dokumentation
- Qualitätskriterien EER

ER-Modellierung

- Objekte (Entitäten) mit ähnlichen Eigenschaften zu Mengen (Entitätstypen, Klassen) zusammenfassen
  - Jedes Objekt ist *Instanz* einer oder mehrerer Klassen
- Extension (Menge aller Instanzen einer Klasse)
- Objekte können in Beziehung gesetzt werden (Beziehungstyp, Relationship)
- ER-Modell wird durch einen Graphen dargestellt

Was für Komponenten brauchen wir?



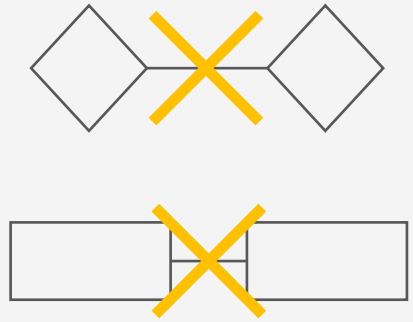
Alle Objekte (Universum)

● steht für Objekt oder Entität

Grundlegende Elemente von ER-Diagrammen

- Objekttyp 
  - Graphische Darstellung: Rechteck
  - Auch Entitätstyp oder Klasse genannt
  - Menge von Objekten
- Werttyp 
  - Graphische Darstellung: Ellipse
  - Für Basisdatentypen, Attribute
  - Menge von Werten bzw. Literalen
- Beziehungstyp 
  - Graphische Darstellung: Raute
  - Relationen zwischen Objekttypen
  - Menge von Tupeln von Objekten

- Elemente von ER-Diagrammen bilden einen bipartiten Graphen:
  - Verbindungen zwischen Symbolen der gleichen Typen sind nicht erlaubt

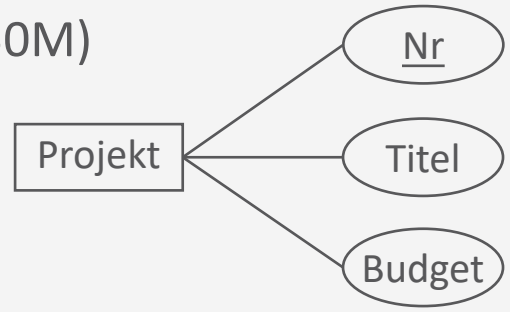


P. Chen: The Entity-Relationship Model - Toward a Unified View of Data. In *ACM Transactions on Database Systems*, vol. 1(1), pp. 9–36, 1976.

Objekte/Entitäten und Attribute

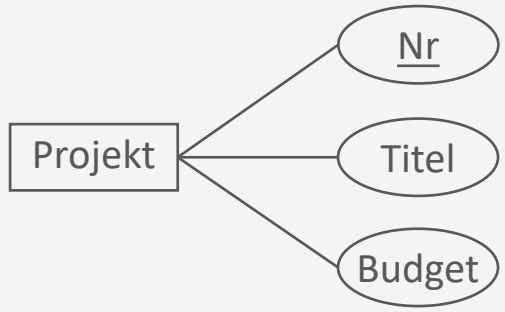
- Entitäten
  - Wesentliche Konzepte („abgrenzbare Dinge“) realer oder gedachter Miniwelten
  - Haben eine eigenständige Existenz
- Attribute beschreiben Eigenschaften von Entitäten
- Mathematische Bedeutung:
  Menge von Tupeln von Werten
  - Tupel = Aggregat von Basiswerten
- Schlüssel: Attribute, die Tupel eindeutig kennzeichnen
  - In der Graphik sind diese Attribute unterstrichen
  - Entitäten mit eigenem Schlüssel = starke Entität

- Beispiel:
  - Entität: Projekt
  - Attribute: Projekt beschrieben durch
    - Eine Nummer
    - Einen Titel
    - Das Budget
  - Als Menge von Tupeln:
    - (<u>42</u>, Flughafen, 100M)
    - (<u>21</u>, Bahnhof, 50M)



Identifikation und Schlüssel

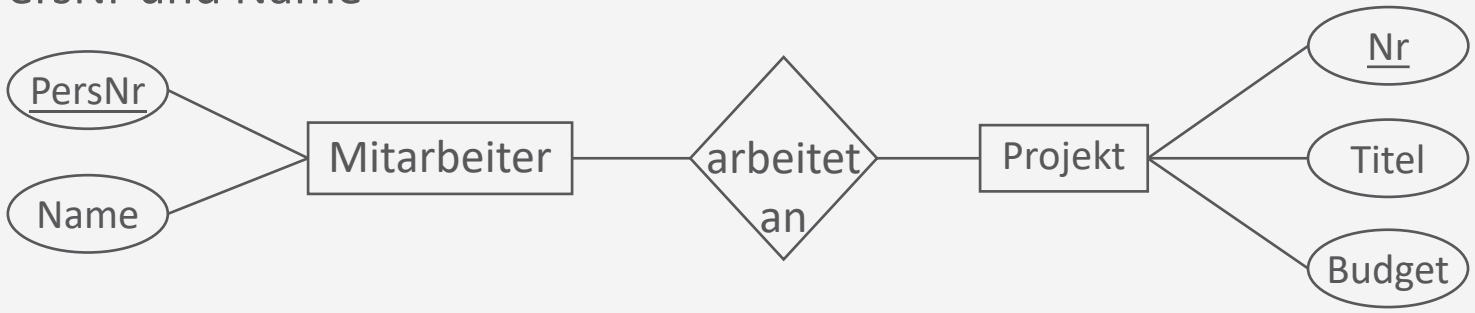
- Identifikation: Zwei grundlegende Ansätze in DB-Modellen
  - Referentielle Identifikation
    - Direkte Verweise auf Objekte (Zeiger in Programmiersprachen)
  - Assoziative Identifikation
    - Werte von Attributen oder Attributkombinationen, um sich eindeutig auf Objekte zu beziehen (Schlüssel: Ausweisnummer, Fahrgestellnummer, ...)
    - Schlüssel
      - Attribute oder Attributkombinationen mit innerhalb einer Klasse eindeutigen Werten
      - Mehrere Schlüsselkandidaten möglich (dazu mehr später)



Assoziation/Relationship

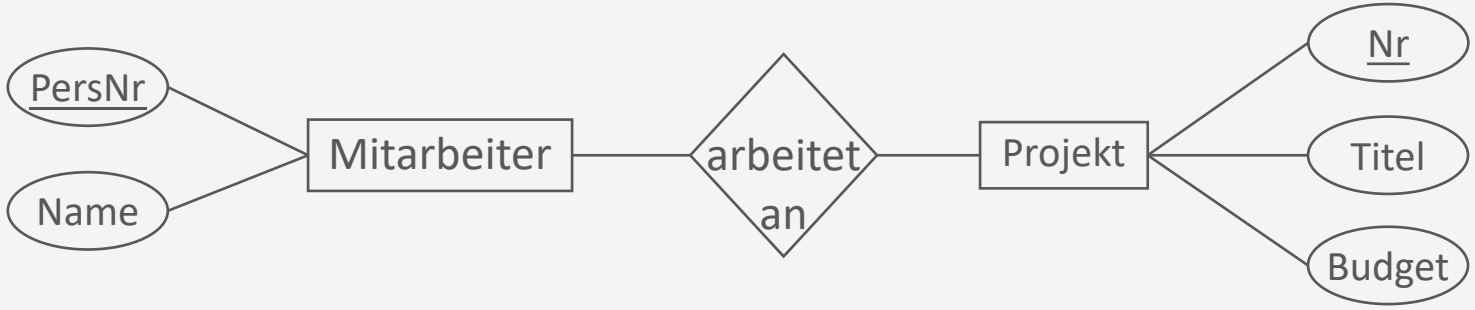
- Objekte können miteinander in Beziehung gesetzt (assoziiert) werden:
  - Binäre (ternäre, ...) Beziehungen assoziieren zwei (drei, ...) Klassen oder Objekte
  - Allgemein: n-äre Beziehungen zwischen n Klassen oder Objekten
    - n Grad der Beziehung
  - Mathematische Bedeutung: Menge an verknüpften Tupeln

- Beispiel
  - An Projekten arbeiten Mitarbeiter
    - Mitarbeiter: Entität mit Attributen PersNr und Name
    - Verknüpftes Tupel:
      ((4711, Mustermann),
      (42, Flughafen, 100M))



Assoziation/Relationship

- Schlüssel um Objekte in Assoziationen zu identifizieren
  - Schlüssel stellen als Attributwerte Beziehungen zu anderen Objekten her
  - Referenzierte Objekte müssen existieren (→ referentielle Integrität)
  - Erlauben vereinfachter Referenzierung in Tupeln
- Beispiel
  - Attribut „Nr“ Schlüssel für Projekte
  - Attribut „PersNr“ Schlüssel für Mitarbeiter
  - Ursprüngliches Tupel
    - ((4711, Mustermann),
      (42, Flughafen, 100M))
  - Unter Zuhilfenahme der Schlüssel:
    - (4711, 42)



Identifikation und referentielle Integrität

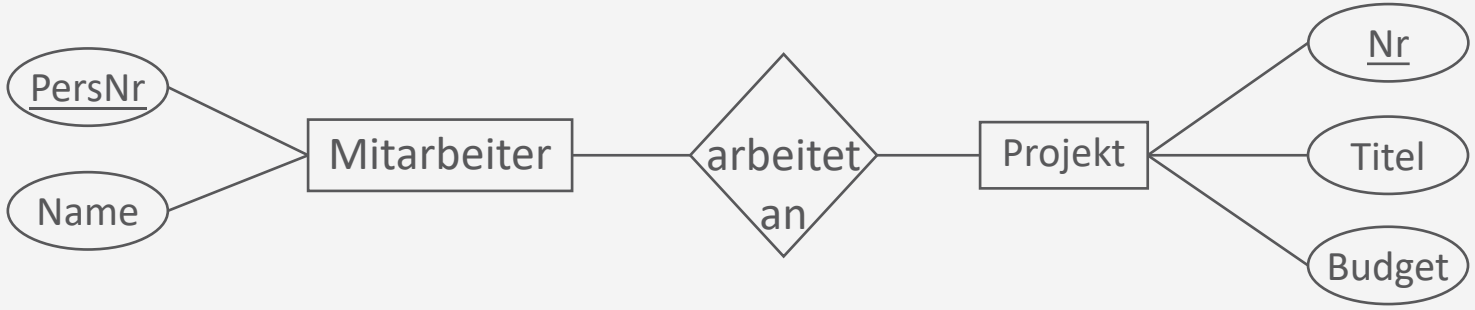
- Beispiel: Projekt
  - Projekte werden durch eine Nummer eindeutig identifiziert
  - Zwei Möglichkeiten zur Identifikation von Projekten innerhalb der Assoziation „arbeitet an“
  - Annahme: Projekt 54 existiert nicht → referentielle Integrität verletzt!

Referentielle Identifikation

	Projekt	Mitarbeiter
		...
		...
NULL		...
	...	...

Assoziative Identifikation

Projekt	Mitarbeiter
42	...
21	...
54	...
....	...

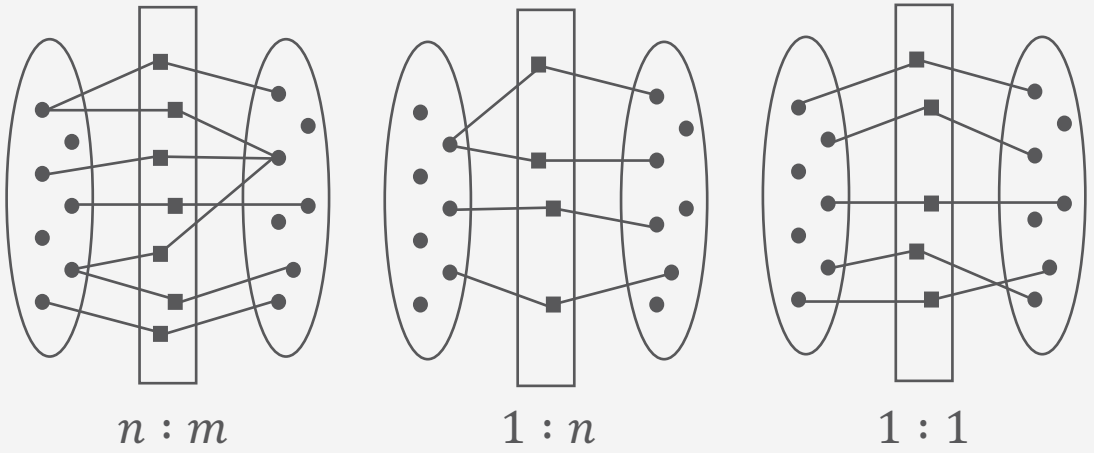


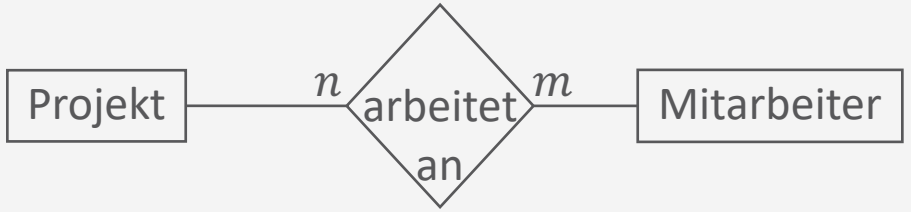
Funktionalitäten

Achtung: n, m gelten lokal pro Assoziation (und nicht global über verschiedene Assoziationen hinweg)

Beispiele für $1:n, 1:1$?

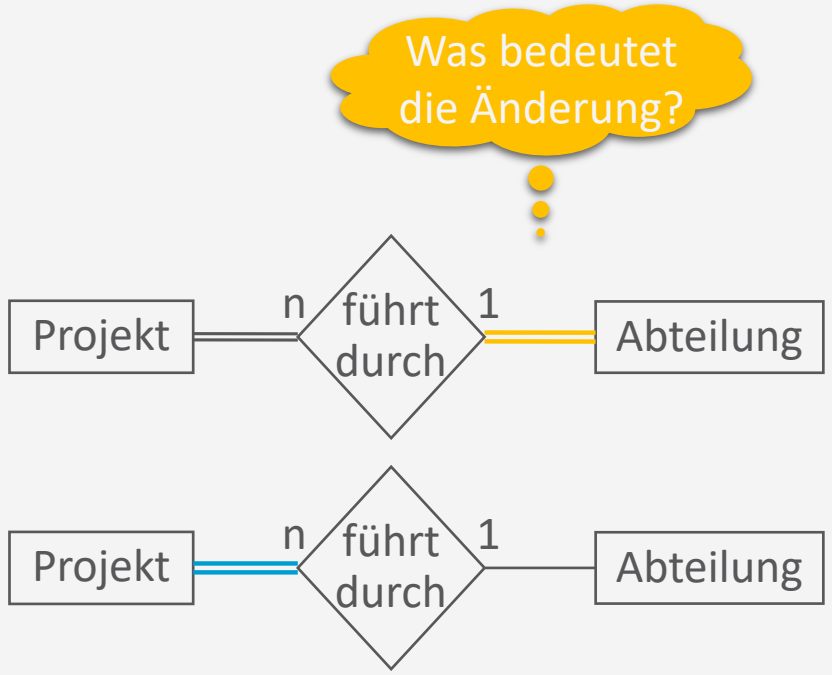
- Funktionalitätsangaben definieren Einschränkungen:
  - Varianten sind $n:m, 1:n, 1:1$
  - Gegeben $x:y$,
    - Jede Instanz der ersten Klasse kann mit bis zu y Instanzen der zweiten Klasse assoziiert werden
    - Jede Instanz der zweiten Klasse mit bis zu x Instanzen der ersten Klasse assoziiert werden
- Beispiel
  - An Projekten arbeiten Mitarbeiter
    - Ein Mitarbeiter kann an mehreren (n) Projekten arbeiten
    - Jedes Projekt wird von beliebig vielen (m) Mitarbeitern bearbeitet





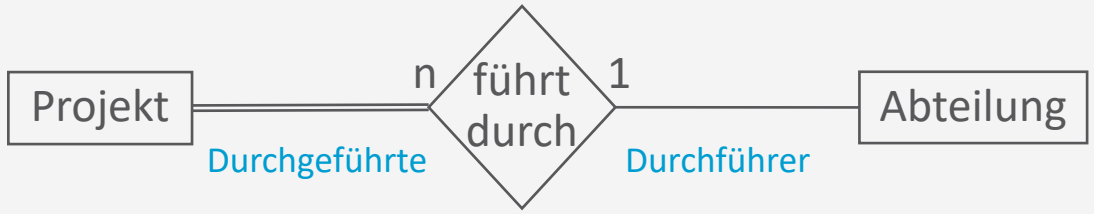
Partizipation

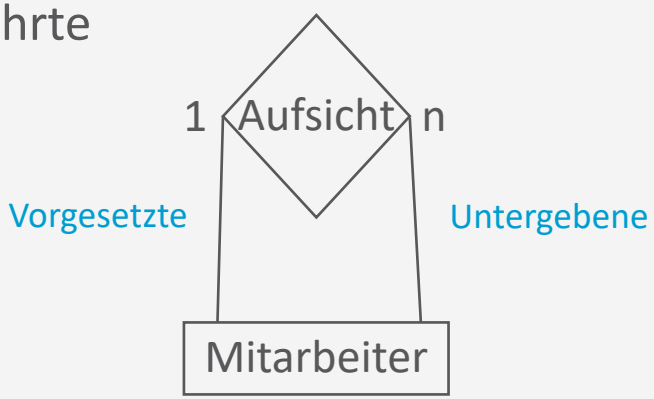
- Beteiligung an einer Beziehung
- Formen:
  - Totale Partizipation: Jede Instanz einer Klasse muss mit einer Instanz der zweiten Klasse in Beziehung stehen (═══)
  - Partielle Partizipation: Eine Instanz einer Klasse kann in Beziehung zu einer Instanz der zweiten Klasse stehen (———)
- Beispiel:
  - Projekte werden von Abteilungen durchgeführt
  - Jedes Projekt muss einer Abteilung zugeordnet sein
  - Eine Abteilung kann mehrere Projekte ausführen.



Partizipation: Rollennamen

- Rollennamen (Namen für die Argumente der Relation) identifizieren die Menge der Instanzen, die mit einer anderen Instanz in Beziehung stehen.
  - Rollen können als abgeleitete Attribute verstanden werden, die die Menge der Instanzen als Attributwerte besitzen
  - Besonders bei rekursiven Assoziationen eingesetzt
  - Beispiele:
    - Abteilungen als Durchführer von Projekten als Durchgeführte
    - Rekursiv: Vorgesetzte und Untergebene

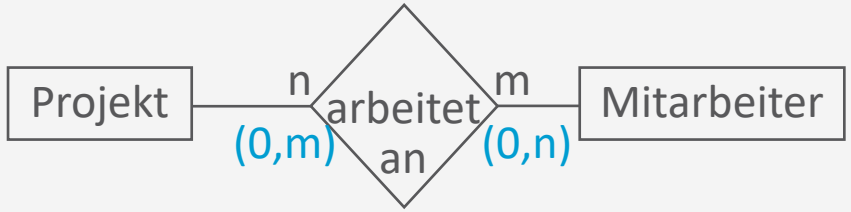


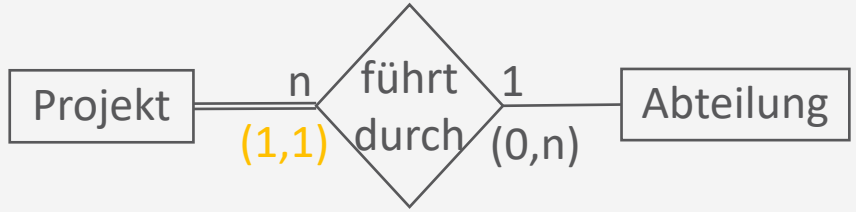


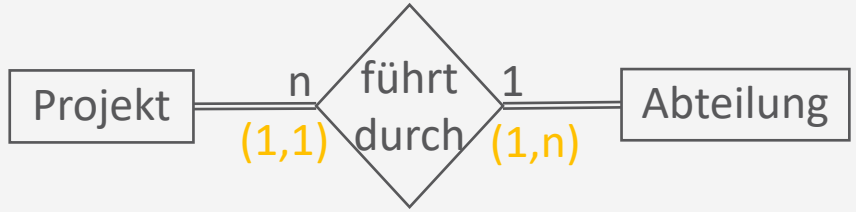
Funktionalitäten (Reprise)

- Explizite Angabe der Kardinalitäten über (min, max)-Kardinalitäten
  - Instanz ist mit min bis max Instanzen assoziiert
  - Leider invers zu $n : m$ Funktionalitäten an den Kanten
  - Erlaubt totale Partizipation direkt anzugeben
    - Durch min=1
    - Doppelte Striche dann nicht mehr nötig
    - Beispiele
      - Totale Partizipation von Projekt
      - Totale Partizipation beider Entitäten

Bemerkung: In der Literatur findet man weitere Beschriftungsregeln.

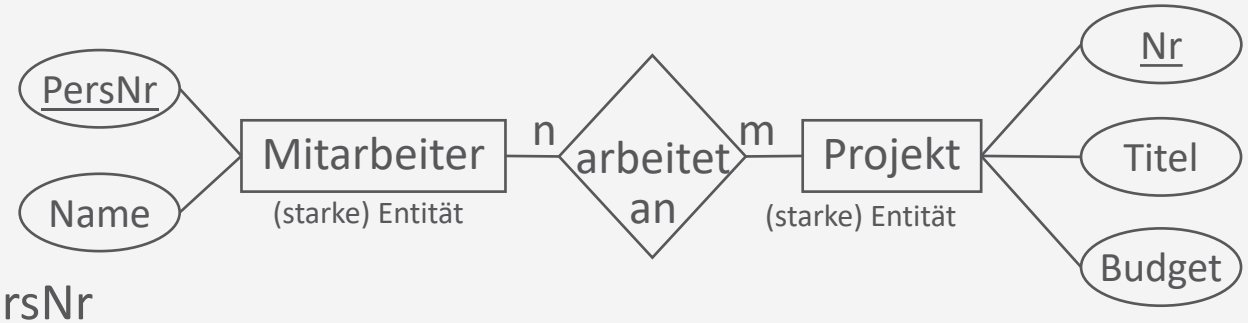






Schwache, existenzabhängige Entitäten

- Starke Entitäten
  - Besitzen ausgezeichnete Attribute zur eindeutigen Identifikation (Schlüssel)
  - Beispiel:
    - Projekt Schlüssel: Nr, Mitarbeiter Schlüssel: PersNr

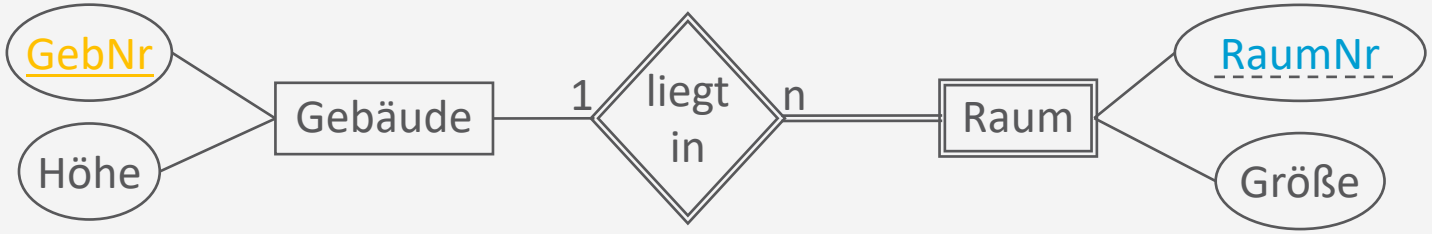


- Schwache Entitäten
  - Benötigen identifizierende Einheit zur Identifikation
  - Starke Entität mit schwacher Entität in Beziehung
    - Totale Partizipation der schwachen Entität nötig
  - Beispiel:
    - Beziehung „Angehörige von“ stellt die Verbindung zur identifizierenden Einheit (Angestellter) her

Schwache Entitäten und Schlüssel

- Beziehung zwischen starkem und schwachem Typ ist immer 1:n (oder 1:1 in seltenen Fällen)
- Schwache Entitäten haben partiellen Schlüssel
  - Wird eindeutig mit Schlüssel der identifizierenden Entität
  - Beispiel:
    - RaumNr ist nur innerhalb eines Gebäudes eindeutig
    - Eindeutiger Schlüssel ist: GebNr und RaumNr

Warum kann das keine n:m-Beziehung sein?



Von der Anforderungsdefinition

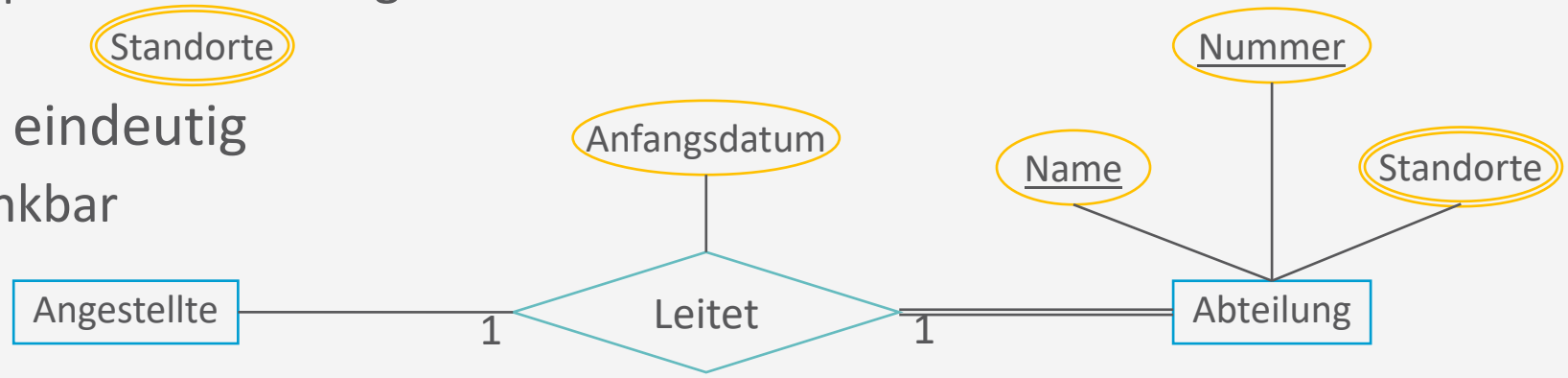
Zum ER-Diagramm

Anwendungsdefinition „Firma“ …

1. Die Firma ist in Abteilungen organisiert. Jede Abteilung hat eine eindeutige Bezeichnung, eine eindeutige Nummer und einen bestimmten Angestellten, der die Abteilung leitet. Es wird das Anfangsdatum verfolgt, ab dem dieser Angestellte die Leitung der Abteilung übernommen hat. Eine Abteilung verfügt über mehrere Standorte.
2. Eine Abteilung kontrolliert eine Reihe von Projekten, die jeweils einen Namen, eine eindeutige Nummer und einen einzigen Standort haben.
3. Zu jedem Angestellten sollen Name, Sozialversicherungsnummer, Adresse, Gehalt, Geschlecht und Geburtsdatum gespeichert werden. Ein Angestellter wird einer Abteilung zugewiesen, kann aber an mehreren Projekten arbeiten, die nicht unbedingt alle von der gleichen Abteilung kontrolliert werden. Dabei wird die Stundenzahl pro Woche verfolgt, die ein Angestellter an jedem Projekt arbeitet. Es wird auch der unmittelbare Vorgesetzte eines Angestellten gespeichert.
4. Zu Versicherungszwecken sollen die Familienangehörigen jedes Mitarbeiters gespeichert werden. Hierzu werden für jeden Angehörigen Vorname, Geschlecht, Geburtsdatum und Verwandtschaftsgrad zum jeweiligen Angestellten erfasst.

Anwendungsdefinition „Firma“ …

1. Die Firma ist in Abteilungen organisiert. Jede Abteilung hat eine eindeutige Bezeichnung, eine eindeutige Nummer und einen bestimmten Angestellten, der die Abteilung leitet. Es wird das Anfangsdatum verfolgt, ab dem dieser Angestellte die Leitung der Abteilung übernommen hat. Eine Abteilung verfügt über mehrere Standorte.
   - Interpretation: totale Partizipation der Abteilung
- Mehrwertige Attribute:
  - Graphische Darstellung: doppelte Umrandung
  - Mehrere Orte pro Abteilung
- Anforderungen nicht immer eindeutig
  - Standort auch als Entität denkbar

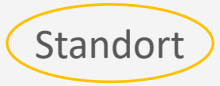


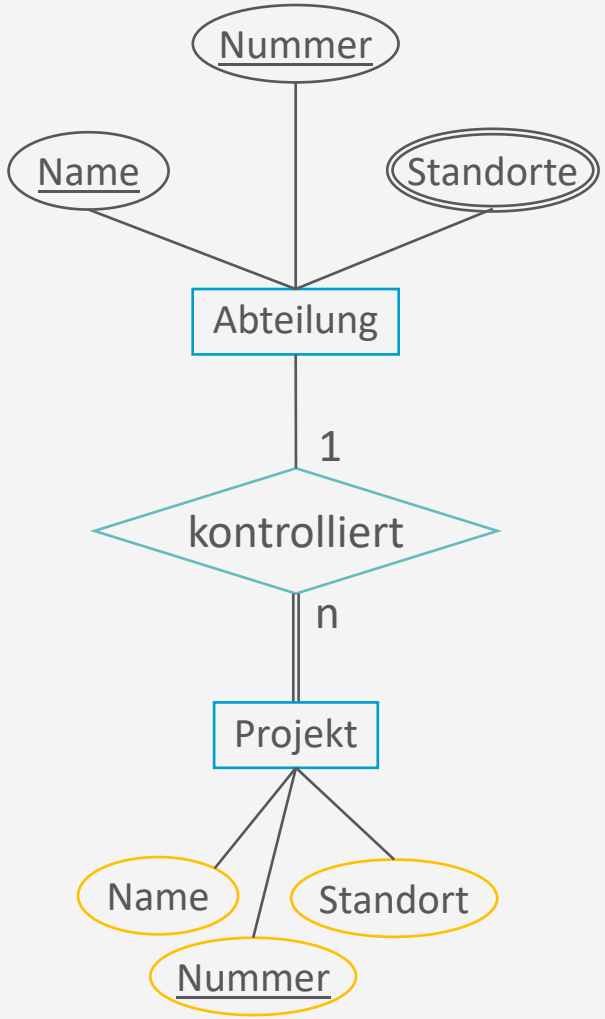
Anwendungsdefinition „Firma“ …

2. Eine Abteilung kontrolliert eine Reihe von Projekten, die jeweils einen Namen, eine eindeutige Nummer und einen einzigen Standort haben.
   - Interpretation: totale Partizipation der Projekte

- Mehrwertige vs. einwertige Attribute
  - Standorte: mehrere Orte pro Abteilung

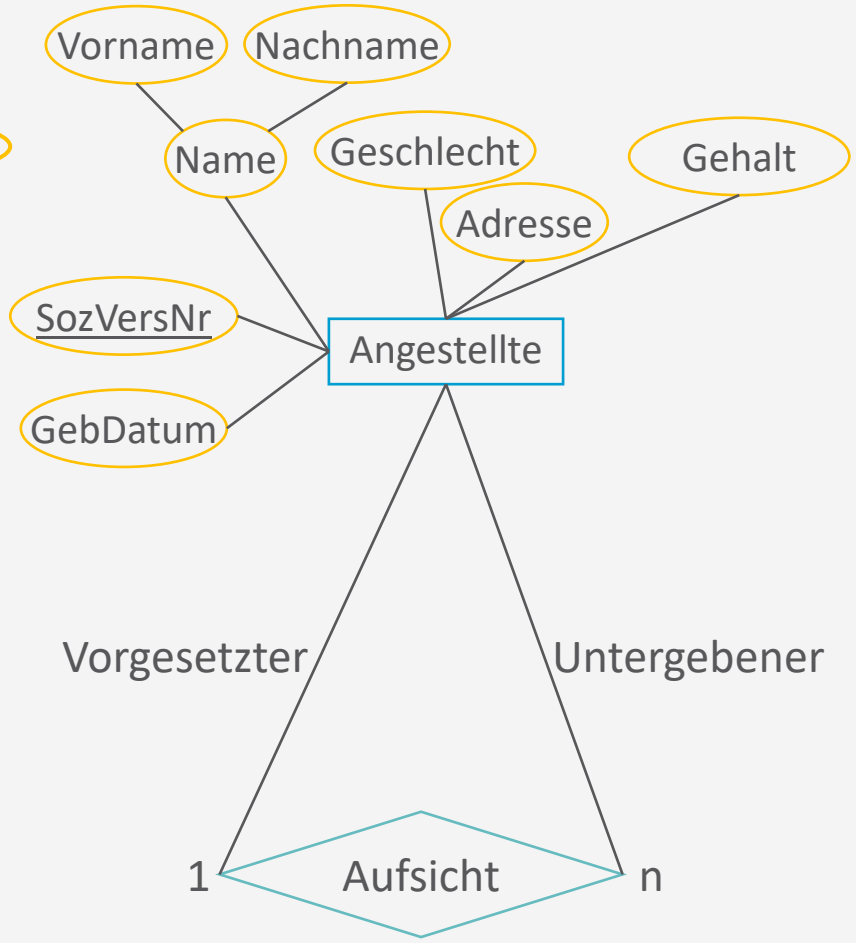
  - Standort: ein Ort pro Projekt





Anwendungsdefinition „Firma“ …

3. Zu jedem Angestellten sollen Name, Sozialversicherungsnummer, Adresse, Gehalt, Geschlecht und Geburtsdatum gespeichert werden. … Es wird auch der unmittelbare Vorgesetzte eines Angestellten gespeichert.
   - Interpretation: Name als mehrwertiges Attribut, auch bei anderen Attributen denkbar

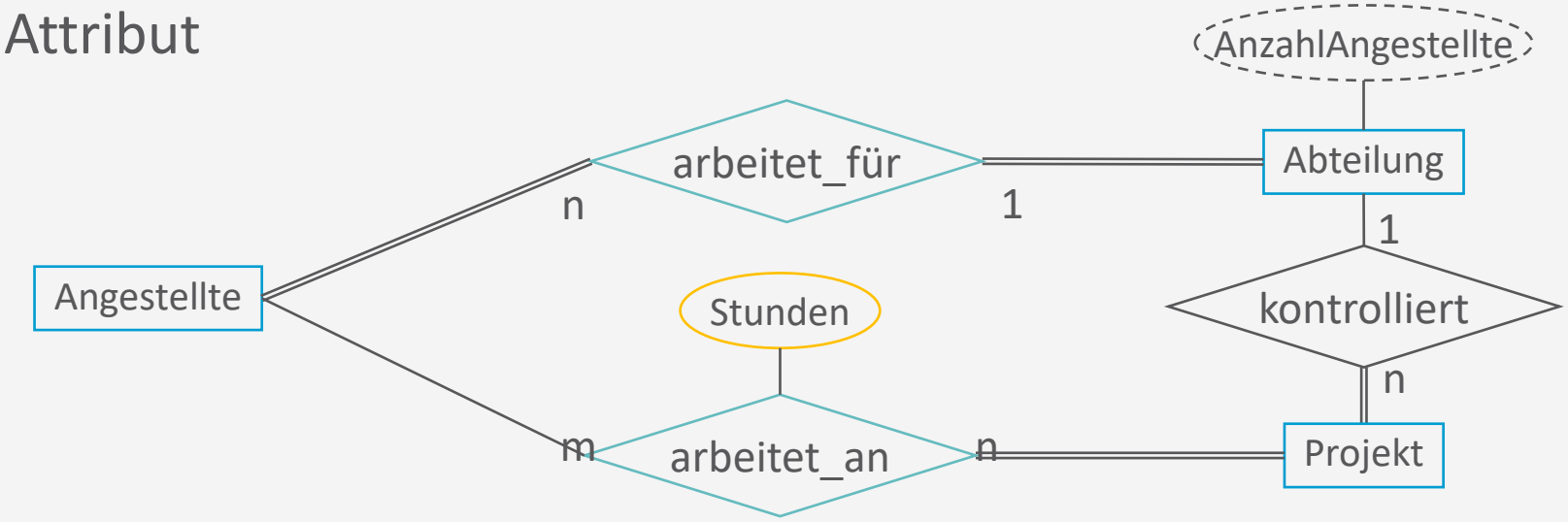


- *Neues Konzept*: Atomare vs. zusammengesetzte Attribute
  - Atomar: unteilbar (nicht sinnvoll zerlegbar); z.B. SozVersNr
  - Zusammengesetzt: aus mehreren Attributen bestehend; z.B. Name besteht aus Vorname und Nachname

Anwendungsdefinition „Firma“ …

3. … Ein Angestellter wird einer Abteilung zugewiesen, kann aber an mehreren Projekten arbeiten, die nicht unbedingt alle von der gleichen Abteilung kontrolliert werden. Dabei wird die Stundenzahl pro Woche verfolgt, die ein Angestellter an jedem Projekt arbeitet…
   - Interpretation: totale Partizipation der Abteilung und der Projekte in den Beziehungen zu Angestellten
- Neues Konzept: abgeleitetes Attribut
  - Kann berechnet werden
  - Beispiel:

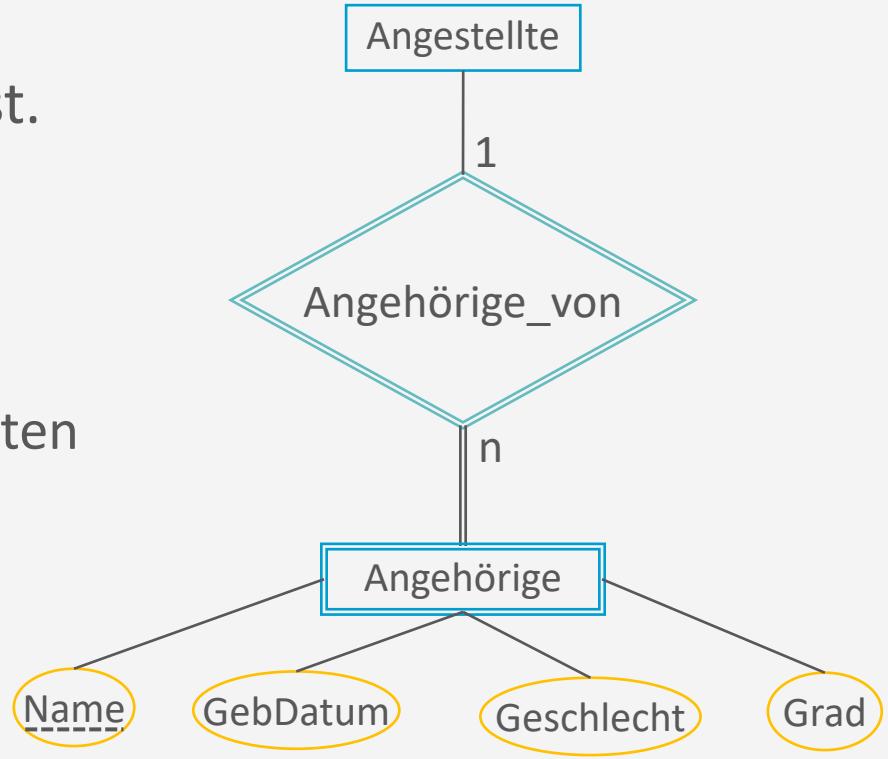
AnzahlAngestellte

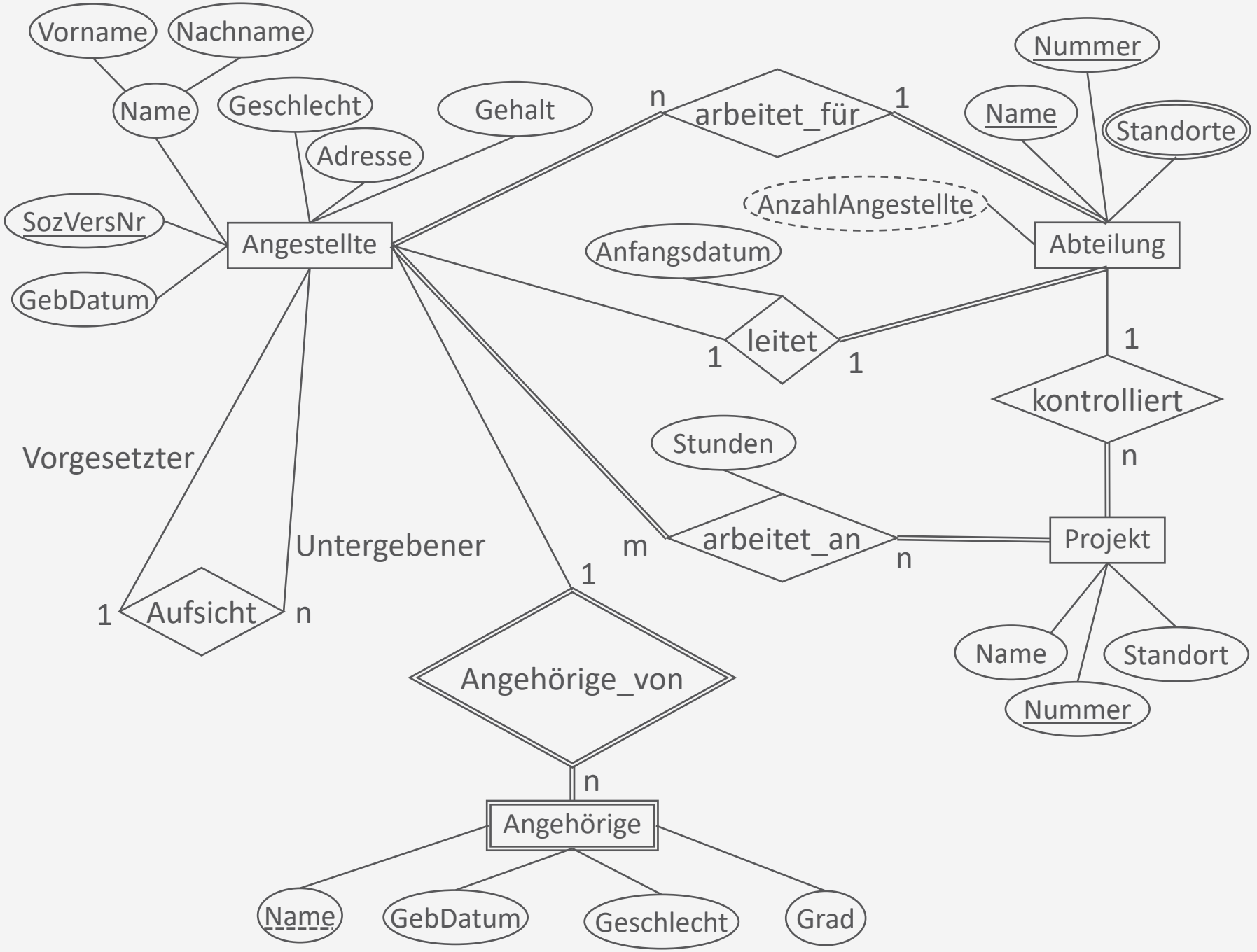


Anwendungsdefinition „Firma“ …

4. Zu Versicherungszwecken sollen die Familienangehörigen jedes Mitarbeiters gespeichert werden. Hierzu werden für jeden Angehörigen Vorname, Geschlecht, Geburtsdatum und Verwandtschaftsgrad zum jeweiligen Angestellten erfasst.

- Partielle Schlüsselattribute Name
  - Name ist nicht eindeutig ohne den dazugehörigen Angestellten

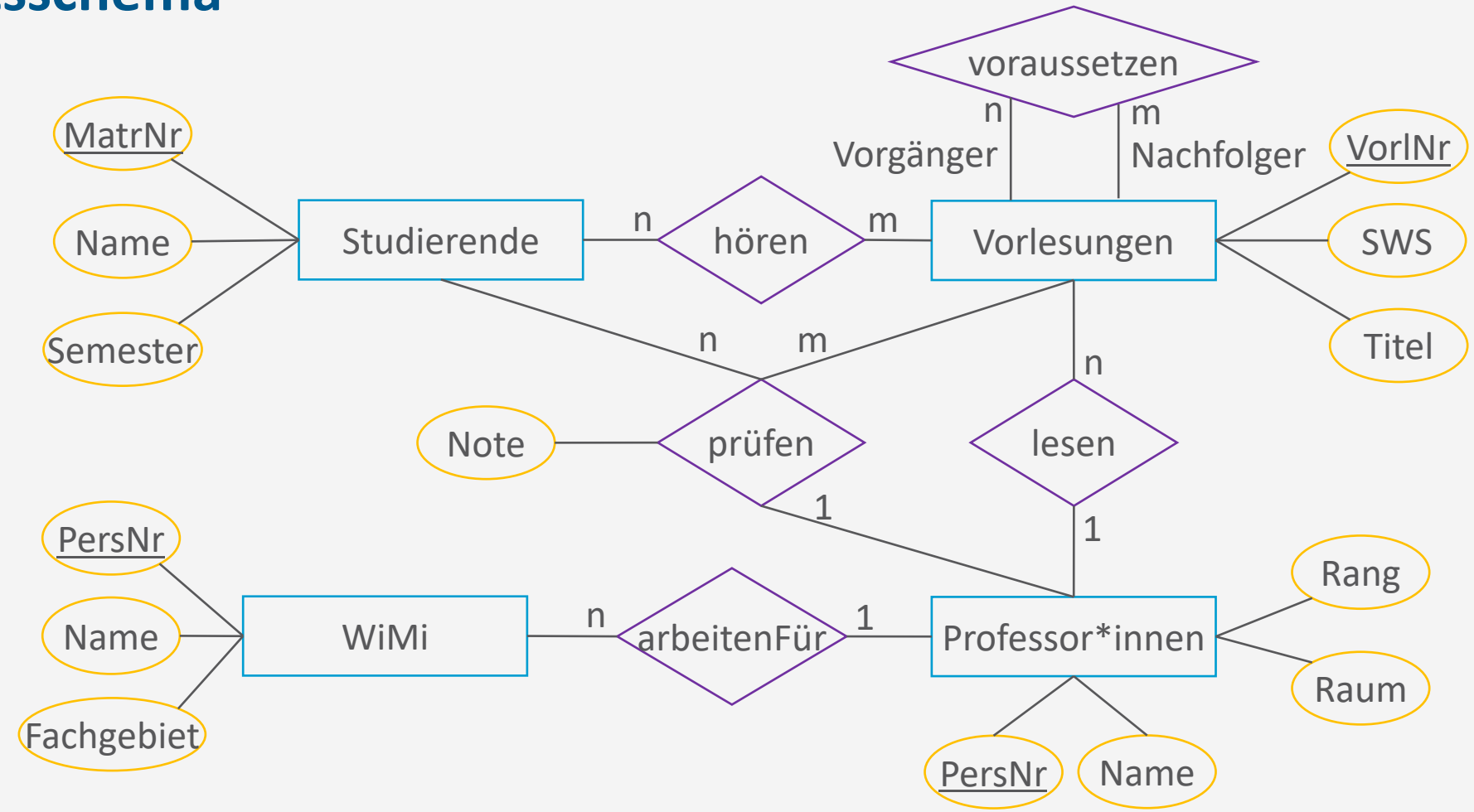



Vorname
Nachname
Name
Geschlecht
Adresse
Gehalt
SozVersNr
GebDatum
Angestellte
n
arbeitet_für
1
Nummer
Name
Standorte
AnzahlAngestellte
Abteilung
Anfangsdatum
leitet
1
1
1
kontrolliert
n
Vorgesetzter
Untergebener
1
Aufsicht
n
Stunden
m
arbeitet_an
n
Projekt
1
Angehörige_von
n
Angehörige
Name
GebDatum
Geschlecht
Grad
Name
Standort
Nummer

Lesen von ER-Diagrammen

Interpretation eines gegebenen ER-Diagramms

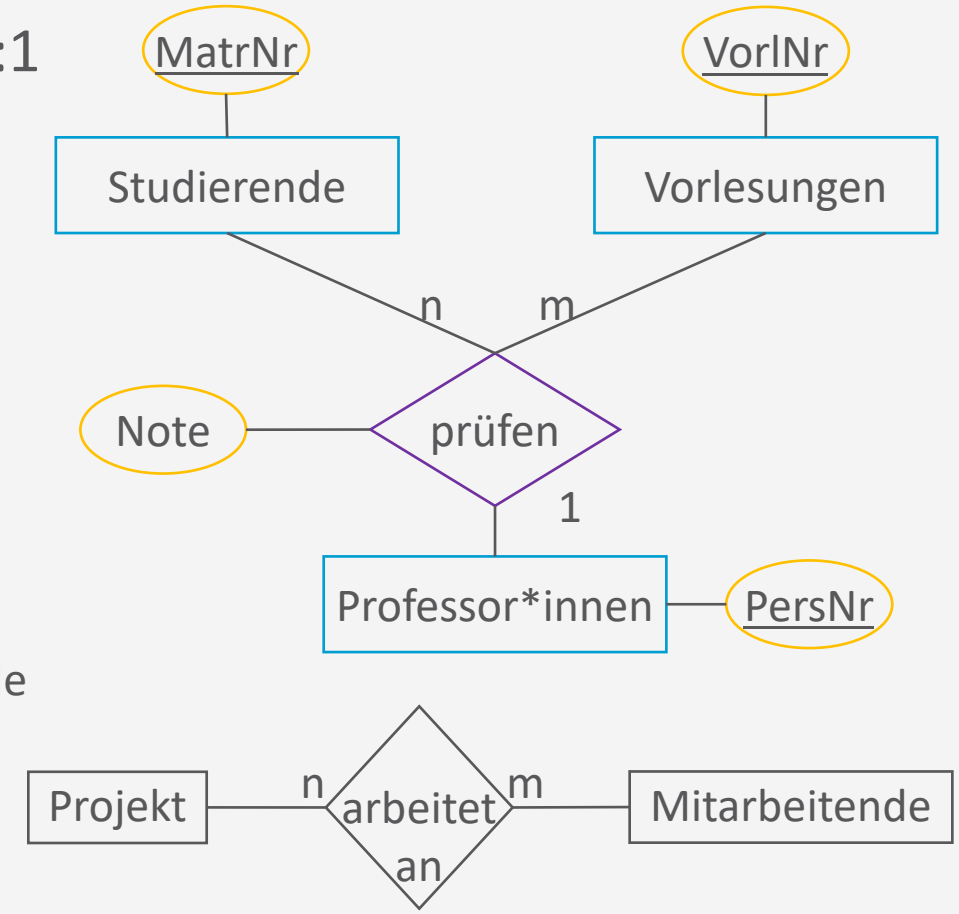
Universitätsschema



Beziehungen mit Grad > 2 werden auch n-äre Beziehungen genannt (wie auch hier auf Folie 12). Um Verwechslung mit dem n aus n:m zu vermeiden, heißt es hier k-äre Beziehung.

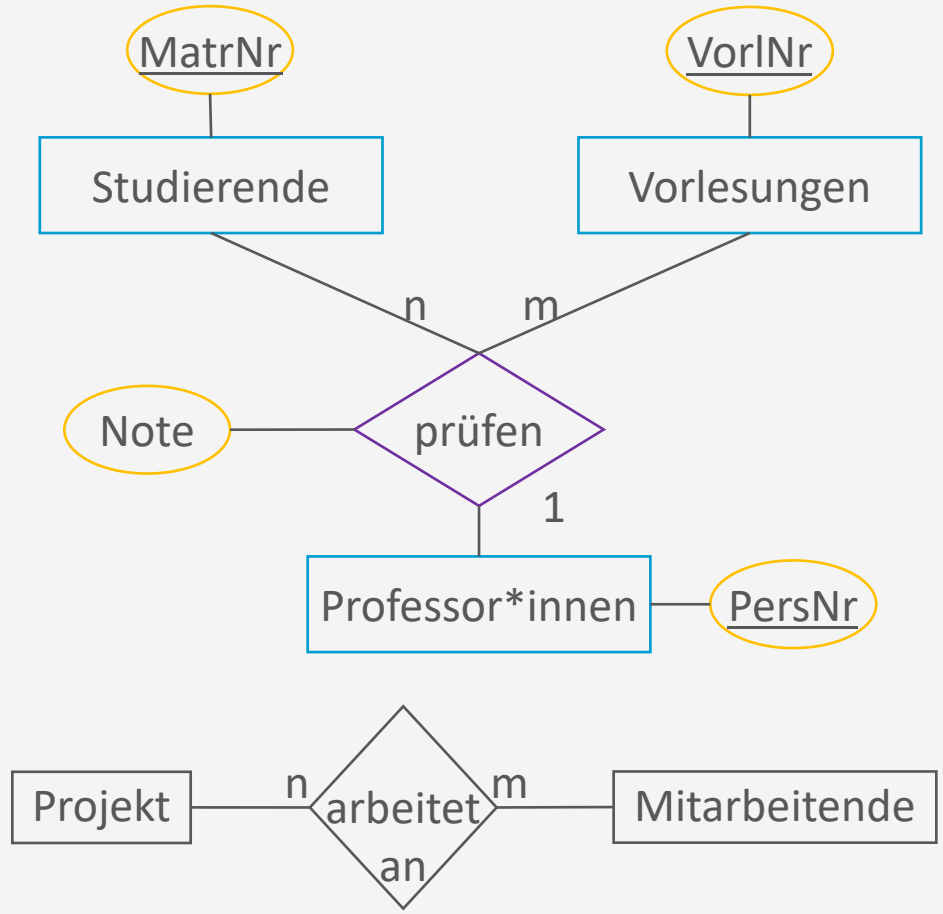
Funktionalitäten (Reprise)

- Funktionalitäten bei binären Beziehungen: n:m, 1:n, 1:1
- Generalisierung:
  Funktionalitätsangaben bei k-ären Beziehungen
  - Funktionalität zwischen einem Tupel von $k-1$ Entitäten und des verbliebenen Entitätstyps
  - Beispiel
    - Ternäre Beziehung „prüfen“
      - Ein*e Studierende*r in einer Vorlesung kann von einem*r Professor*in geprüft werden → 1
      - Ein*e Professor*in in einer Vorlesung kann mehrere Studierende prüfen → n
      - Ein*e Professor*in kann eine*n Studierende*n in mehreren Vorlesungen prüfen → m (≠ n)

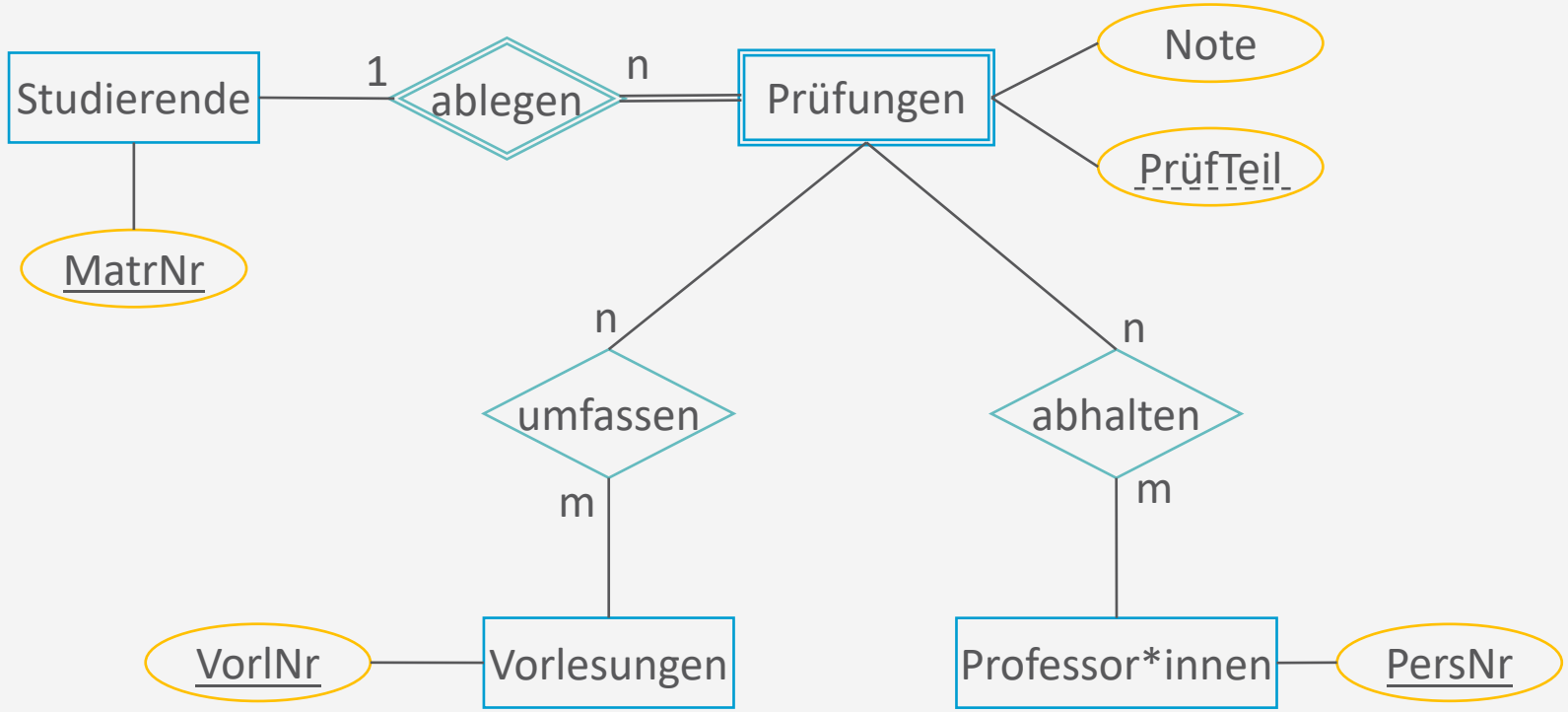


Funktionalitäten (Reprise)

- Bestimmung von Funktionalitäten / Partizipation
  - *Kann* ein Tupel von $k-1$ Entitäten mit *mehreren* Instanzen des verbliebenen Entitätstyps in Beziehung stehen? $\rightarrow n$
    - Mit genau einer Instanz? $\rightarrow 1$
  - *Muss* ein Tupel von $k-1$ Entitäten mit *mindestens* einer Instanz des verbliebenen Entitätstyps in Beziehung stehen? $\rightarrow$ total
    - Sonst $\rightarrow$ partiell



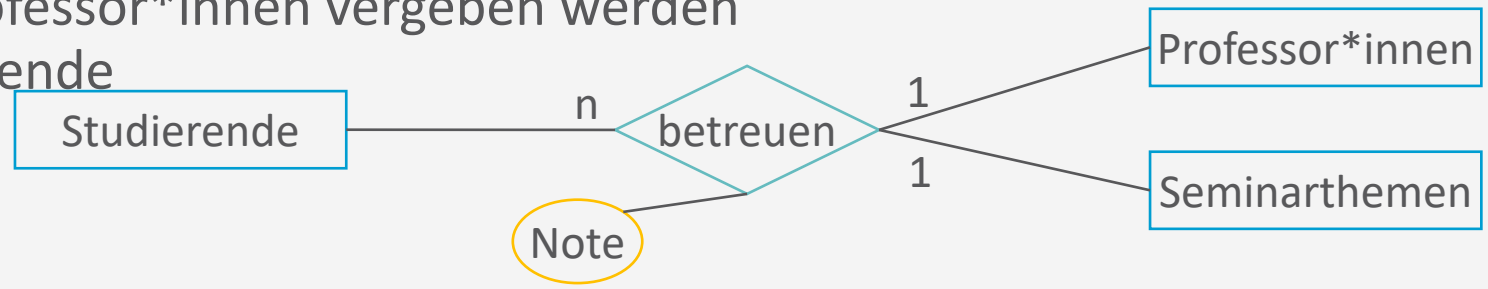
Prüfungen



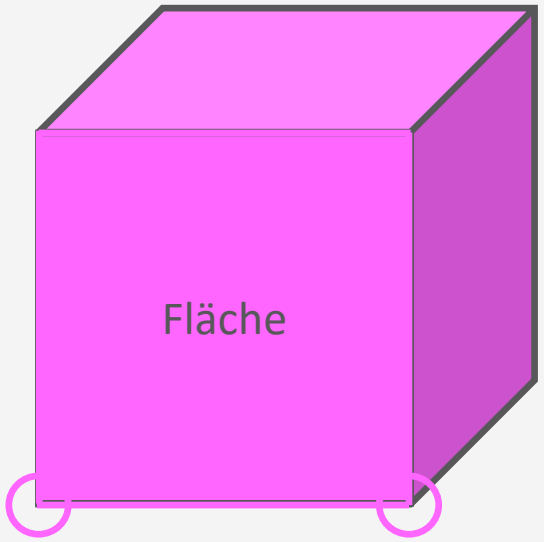
- Mehrere Prüfende in einer Prüfung
- Mehrere Vorlesungen werden in einer Prüfung abgefragt

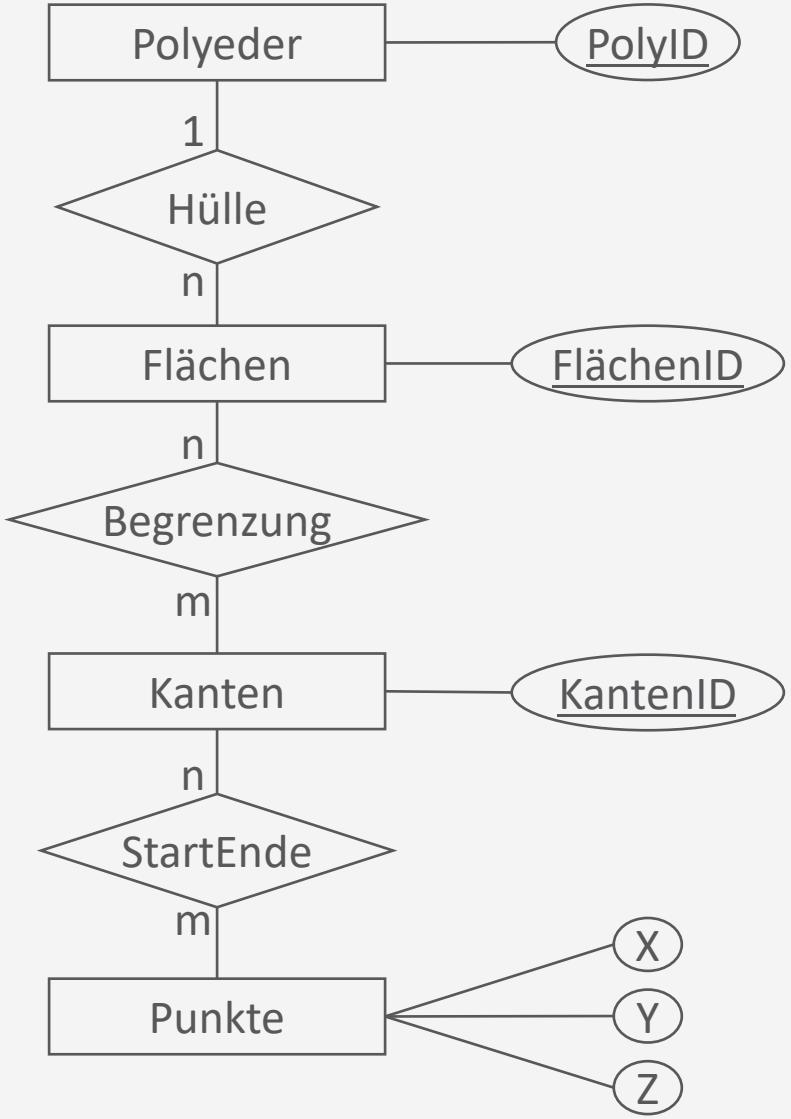
N-stellige Beziehung: *betreuen*

- Erzwungene Konsistenzbedingungen
  - Studierende dürfen bei demselben Professor bzw. derselben Professorin nur ein Seminarthema „ableisten“ (damit ein breites Spektrum abgedeckt wird)
  - Studierende dürfen dasselbe Seminarthema nur einmal bearbeiten – sie dürfen also nicht bei anderen Professor*innen ein schon einmal erteiltes Seminarthema nochmals bearbeiten
- Mögliche Zustände
  - Professor*innen können dasselbe Seminarthema „wiederverwenden“ – also dasselbe Thema auch mehreren Studierenden erteilen
  - Ein Thema kann von mehreren Professor*innen vergeben werden – aber an unterschiedliche Studierende

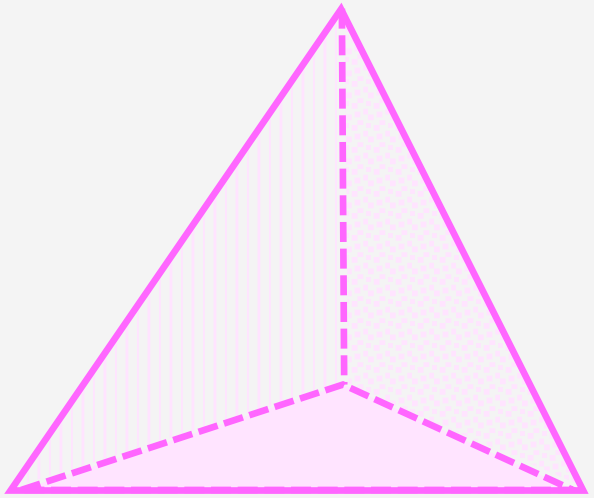


Komplex-strukturierte Entitäten

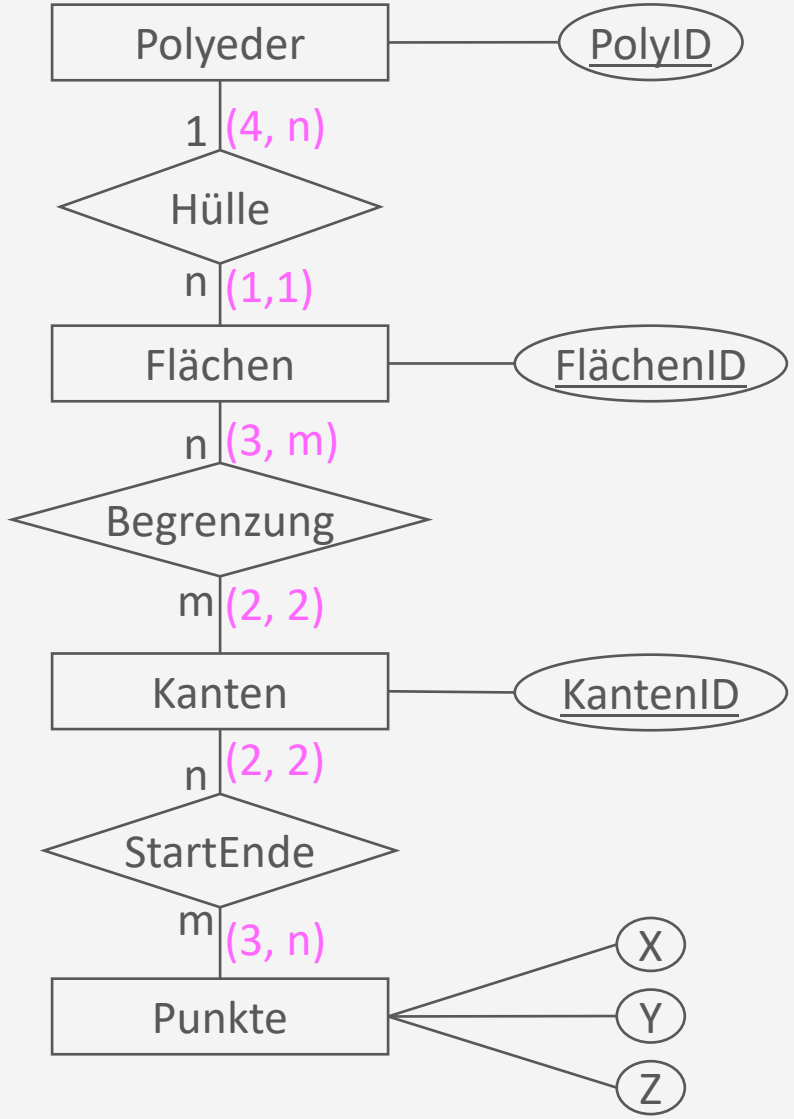




Komplex-strukturierte Entitäten

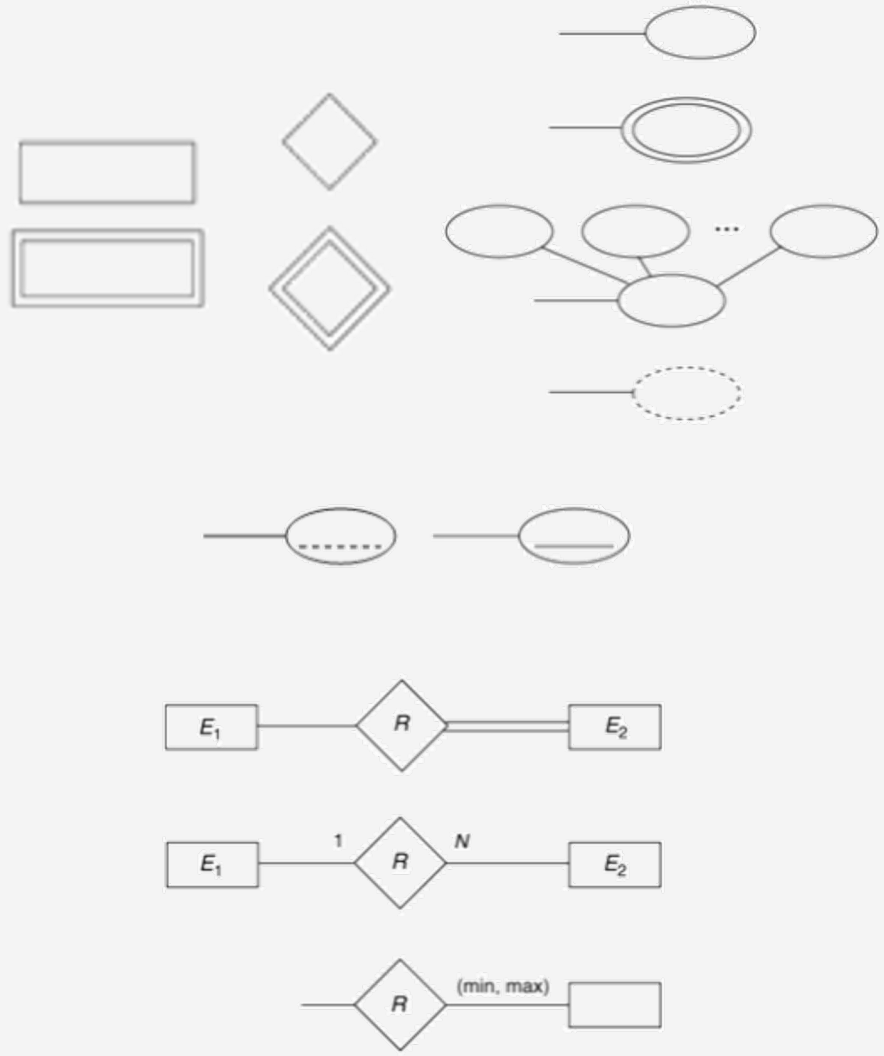


Kleinster Polyeder



Zwischenzusammenfassung

- Entitäten
  - Starke / schwache Entitäten
- Assoziationen / Beziehungen
- Attribute
  - Mehrwertig / zusammengesetzt / abgeleitet
- Schlüssel
  - Referentielle Integrität
  - Partielle Schlüssel
- Funktionalitäten / Kardinalitäten
- Partizipation: total, partiell



Überblick: 2. Datenbank-Modellierung

A. Motivation
- Modelle und Modellierung

B. Entity-Relationship-Modell (ER)
- Komponenten
- Von Anforderungen zum ER-Modell
- Interpretation von ER-Modellen

C. Enhanced ER-Modell (EER)
- Spezialisierung, Generalisierung
- Modellierungsvorgehen
- Beziehung zu UML

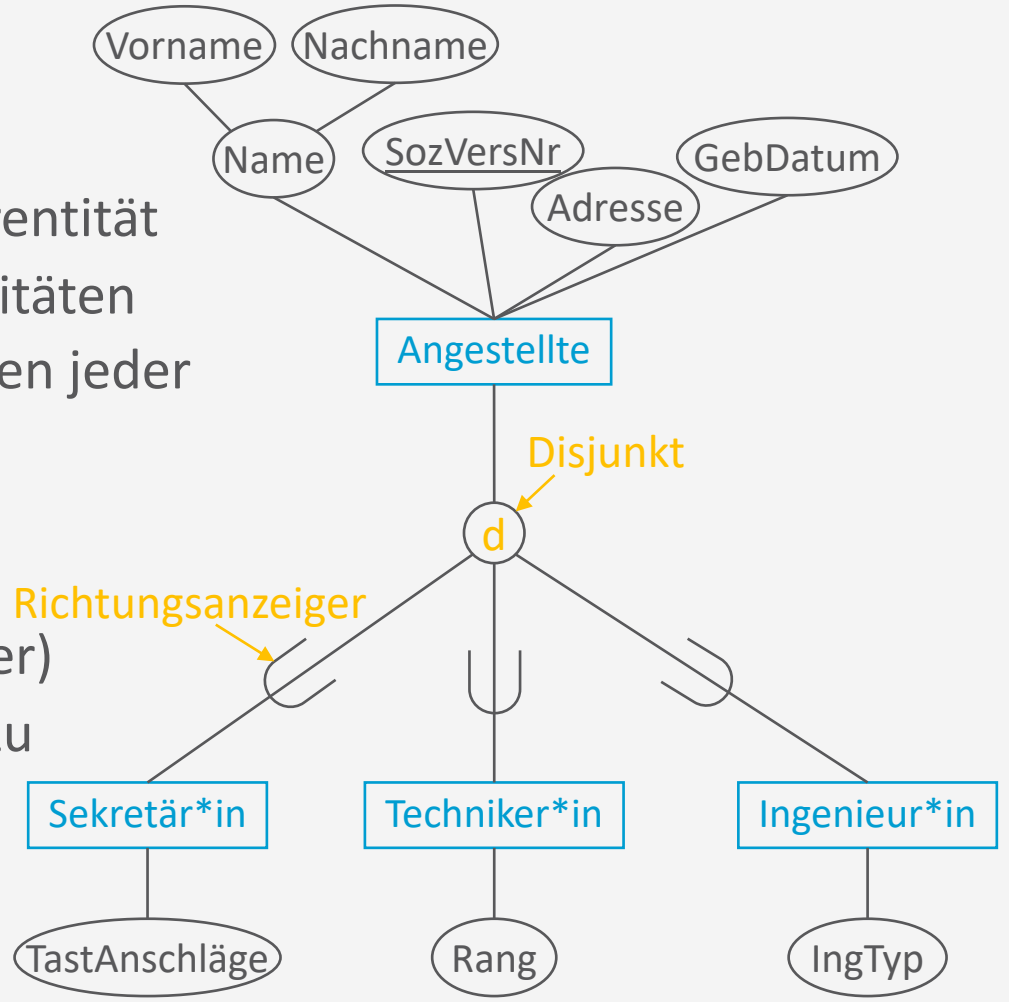
D. Dokumentation & Bewertung
- Dokumentation
- Qualitätskriterien EER

Enhanced ER-Modell (EER-Modell)

- Erweitert das ER-Modell um Spezialisierung und Generalisierung
  - Spezialisierung: *top-down*
    - Definition einer Menge von Subentitäten einer (Super-) Entität
  - Generalisierung: *bottom-up*
    - Bildung einer generalisierenden (Super-) Entität
  - Entitäten, Superentitäten und Subentitäten werden häufig auch als Klassen, Superklassen und Subklassen bezeichnet
    - Bilden eine Klassenhierarchie/Typhierarchie
    - Objekte einer Entität sind auch Objekte der Superklasse
    - Subklassen erben Eigenschaften der Superklasse

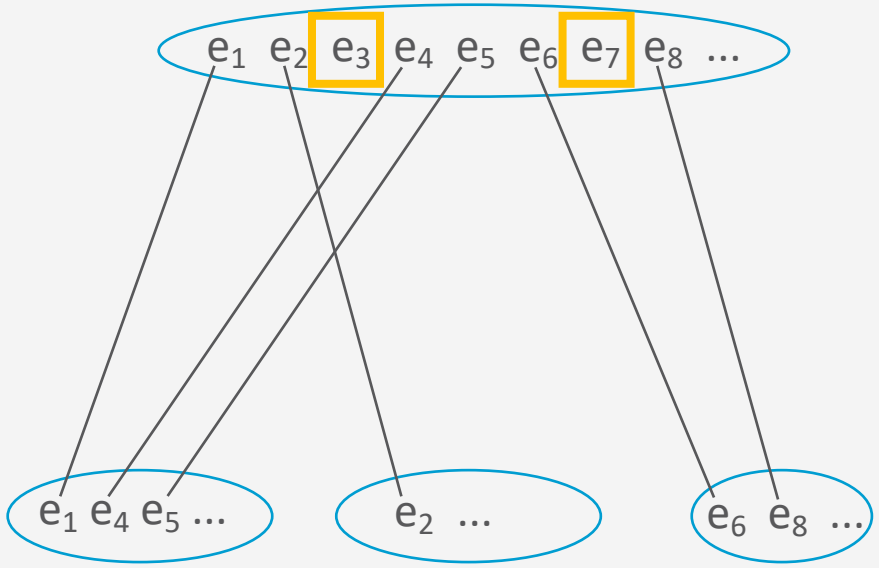
Spezialisierung

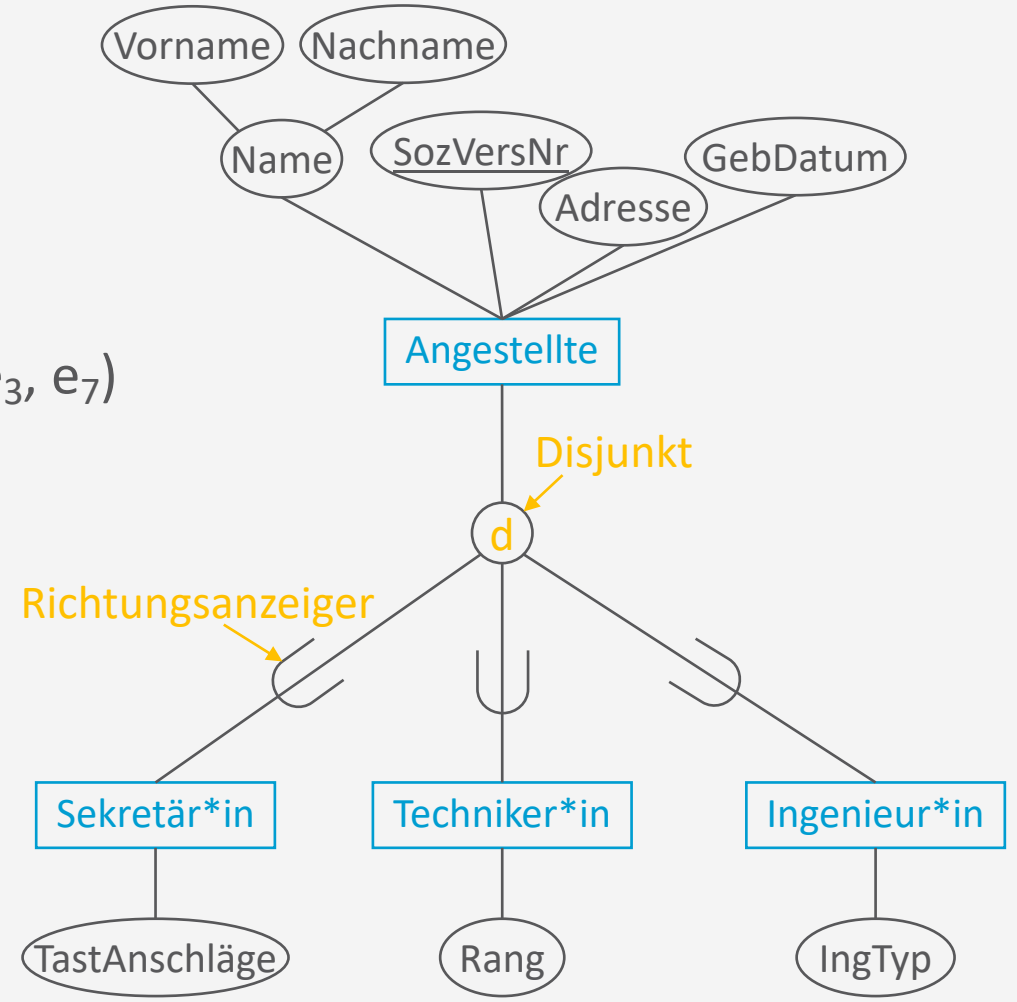
- Ermöglicht
  - Definition einer Menge von Subentitäten zu einer Superentität
  - Erzeugung zusätzlicher spezifischer Attribute für Subentitäten
  - Erzeugung zusätzlicher spezifischer Beziehungen zwischen jeder Subentität und anderen Entitäten oder Subentitäten
- Darstellung:
  - Verzweigungsknoten mit abgehenden Entitäten („Untermengensymbol“ auf Kanten als Richtungsanzeiger)
  - Partizipation: partiell (einfache Kante von Superentität zu Verzweigungsknoten), total (doppelte Kante)
  - Zugehörigkeit: disjunkt (d in Verzweigungsknoten), überlappend (o in Verzweigungsknoten)



Instanzen in Spezialisierungen

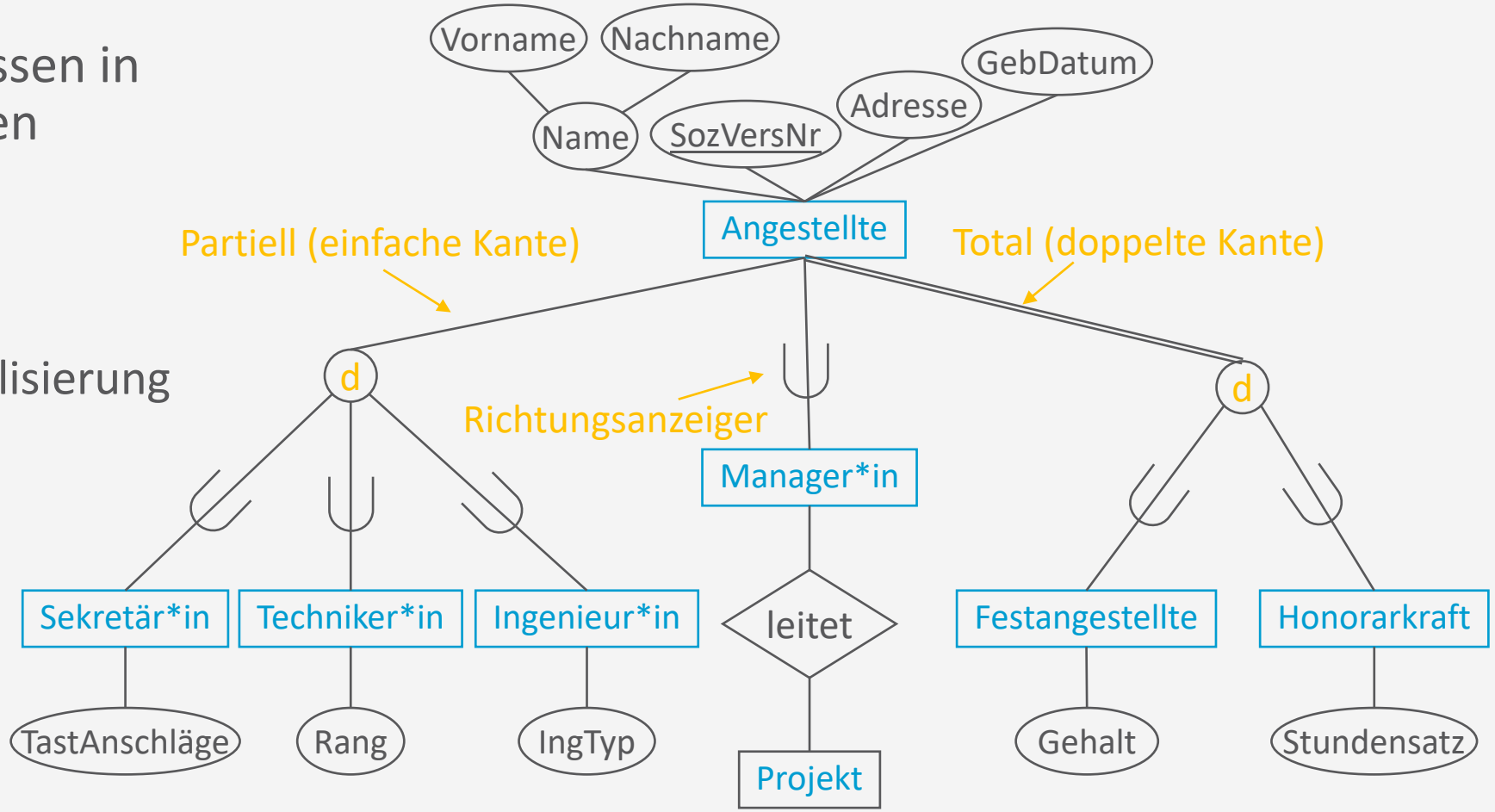
- Disjunkt:
  - Keine Instanz fällt in mehrere Spezial-Klassen
- Partiell:
  - Es gibt Instanzen, die in keine Spezialklasse fallen (z.B. e_3, e_7)





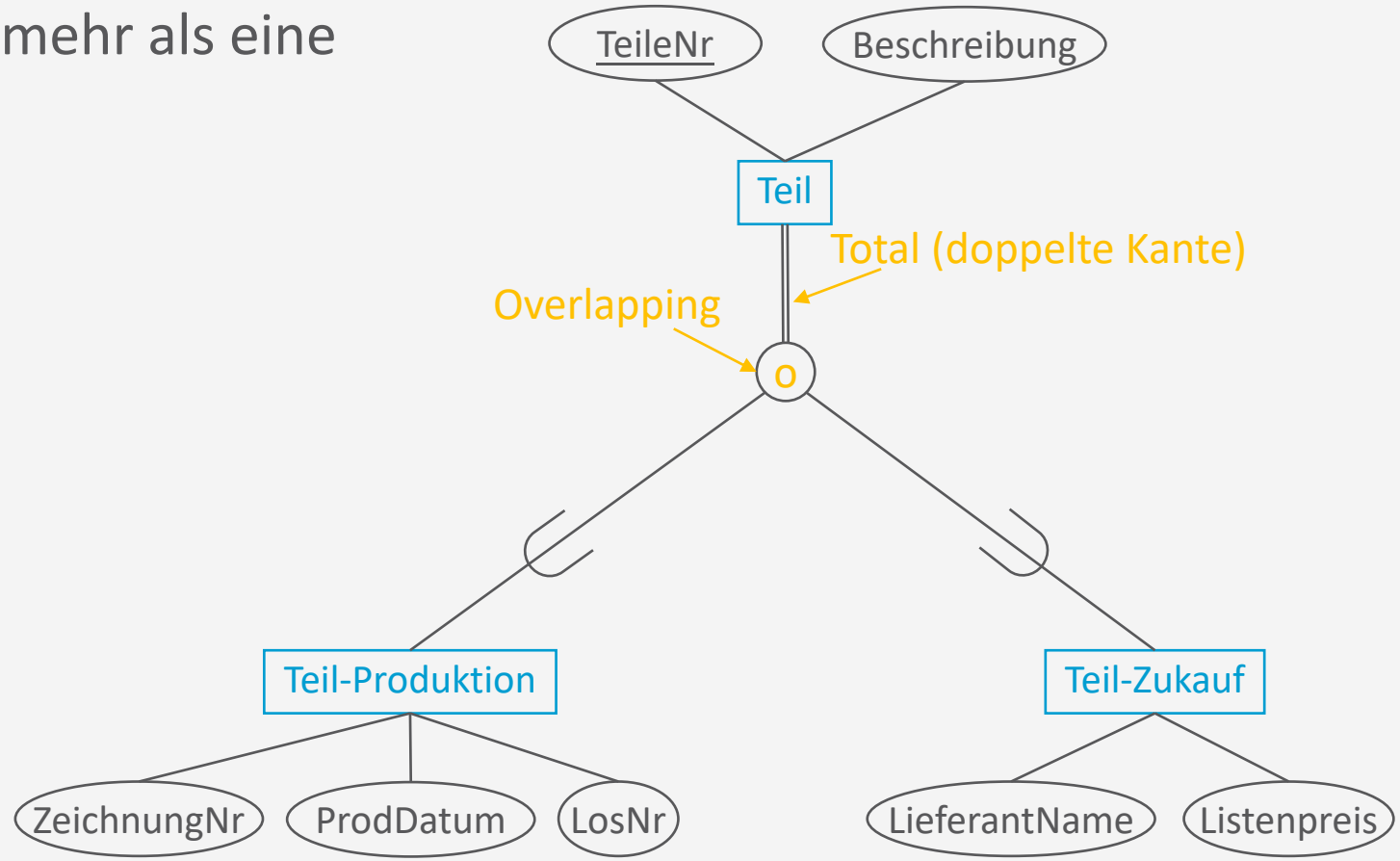
Ausprägungen der Spezialisierung

- Total: alle Instanzen müssen in eine Spezialisierung fallen
  - Doppelte Kante zum Verzweigungsknoten
  - Gegenstück zu partiell
  - Alle müssen eine Spezialisierung haben: total + disjunkt



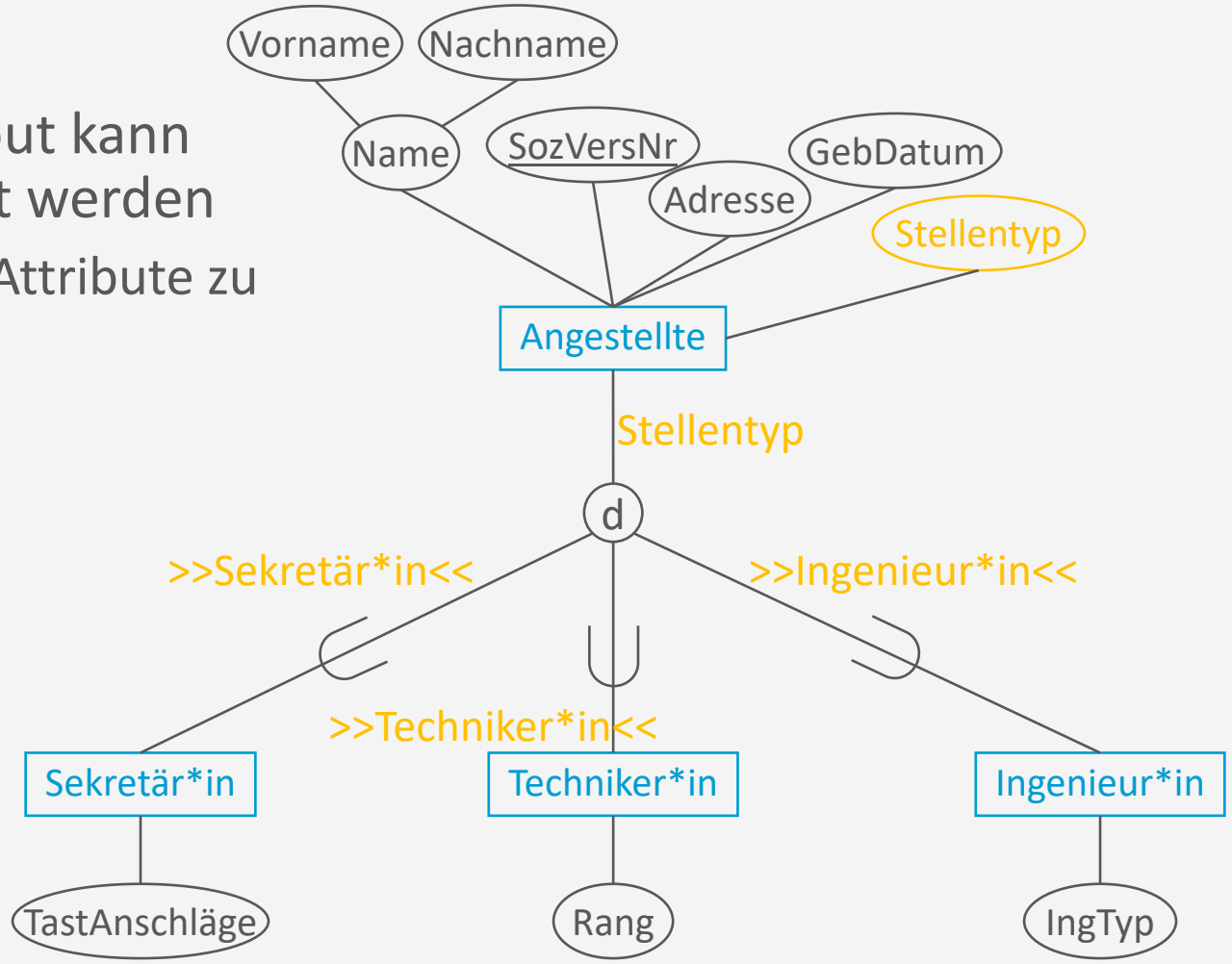
Ausprägungen der Spezialisierung

- Überlappend: Instanzen können in mehr als eine Spezialisierung fallen

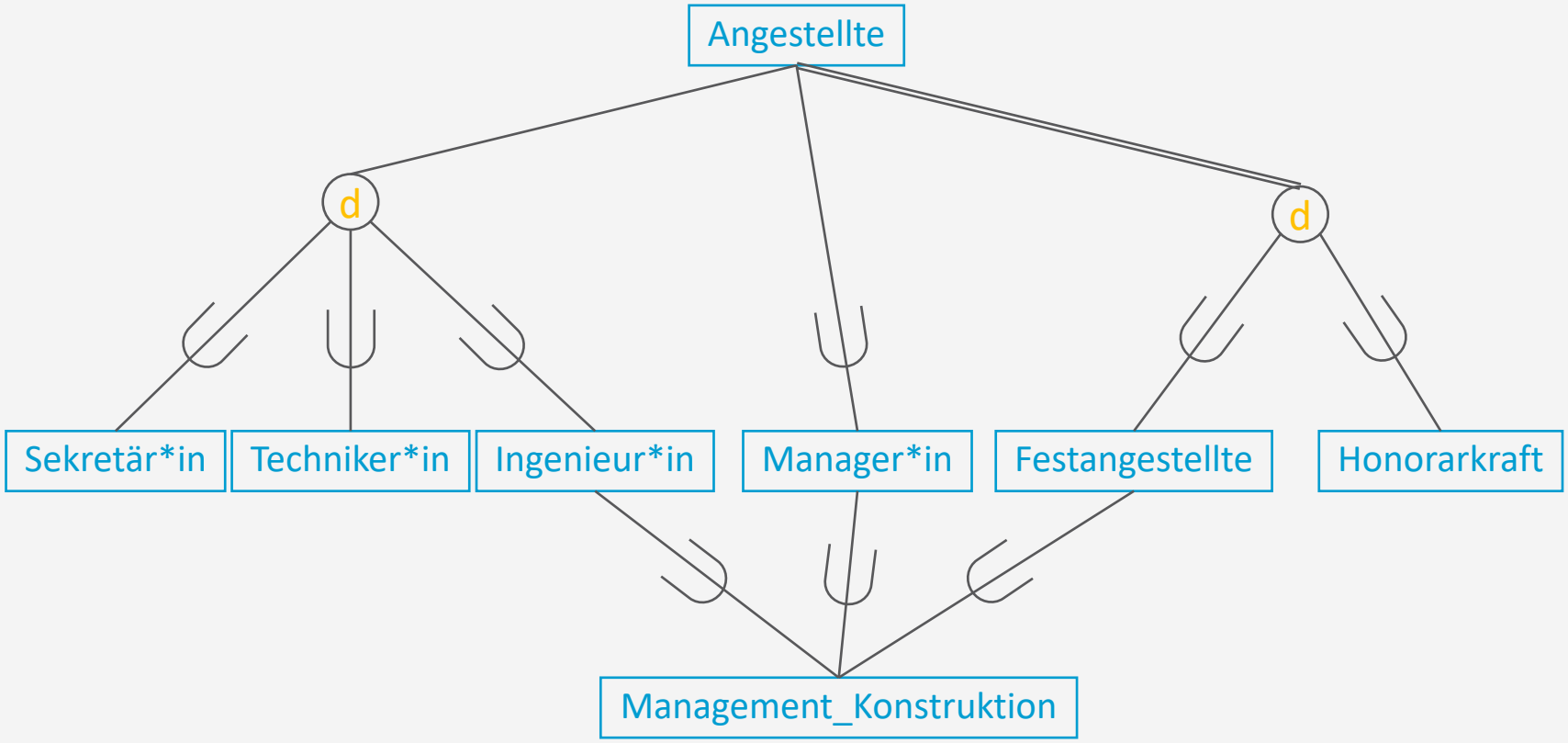


Aufzählungstypen

- Zu einem bestimmten definierenden Attribut kann eine Menge von Spezialisierungen definiert werden
  - Erlaubt Attributwert-spezifische zusätzliche Attribute zu definieren
  - Beispiel: Stellentyp mit Spezialisierungen
    - Sekretär*in
      - Attribut: TastAnschläge
    - Techniker*in
      - Attribut: Rang
    - Ingenieur*in
      - Attribut: IngTyp



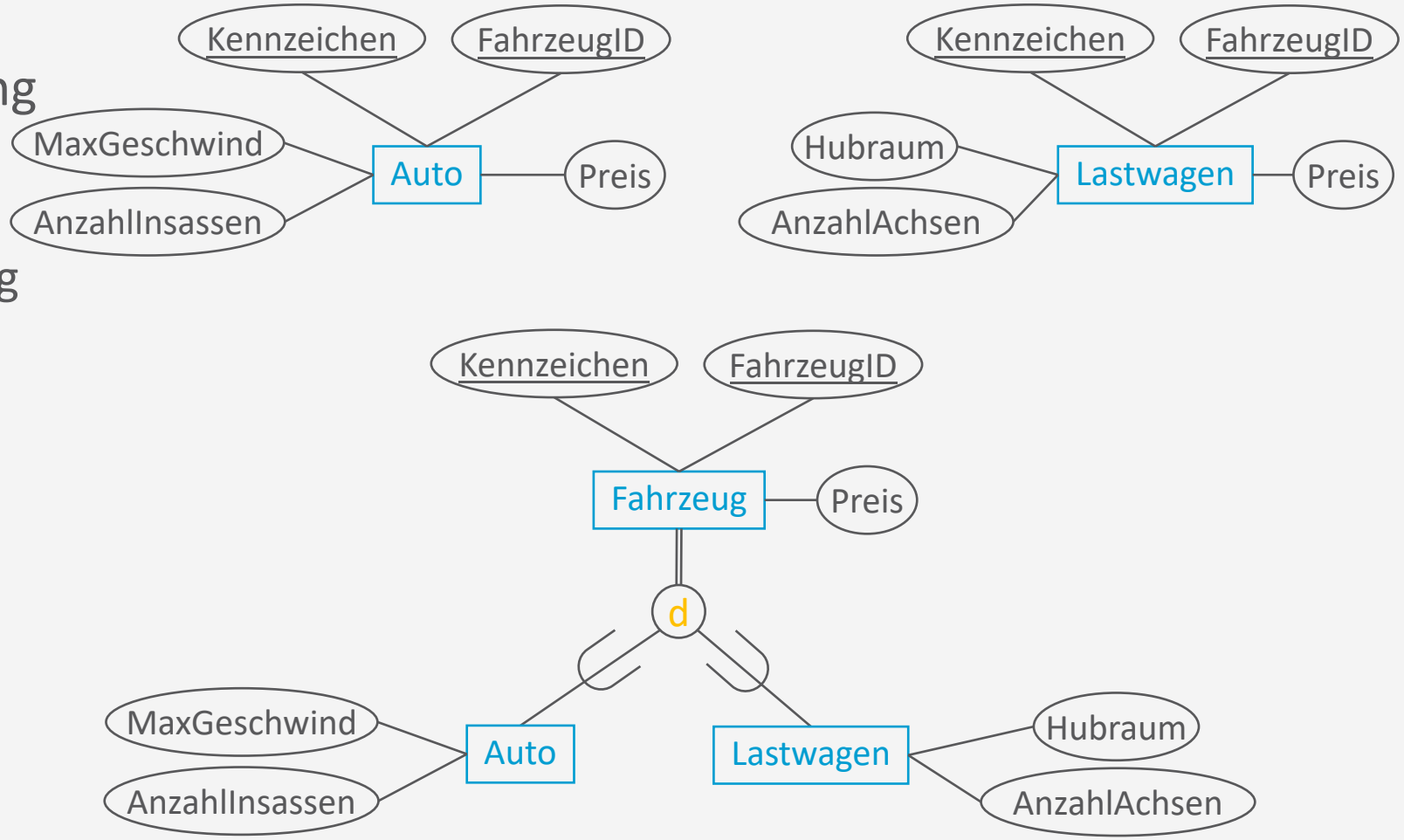
Spezialisierungsnetzwerk



Generalisierung

- Umkehrung der Spezialisierung
  - Zusammenfassung von spezielleren Entitäten
  - Ausprägungen und Darstellung gleich zu Spezialisierung

Manchmal auch als **is-a Relation** dargestellt



EER-Modellierung

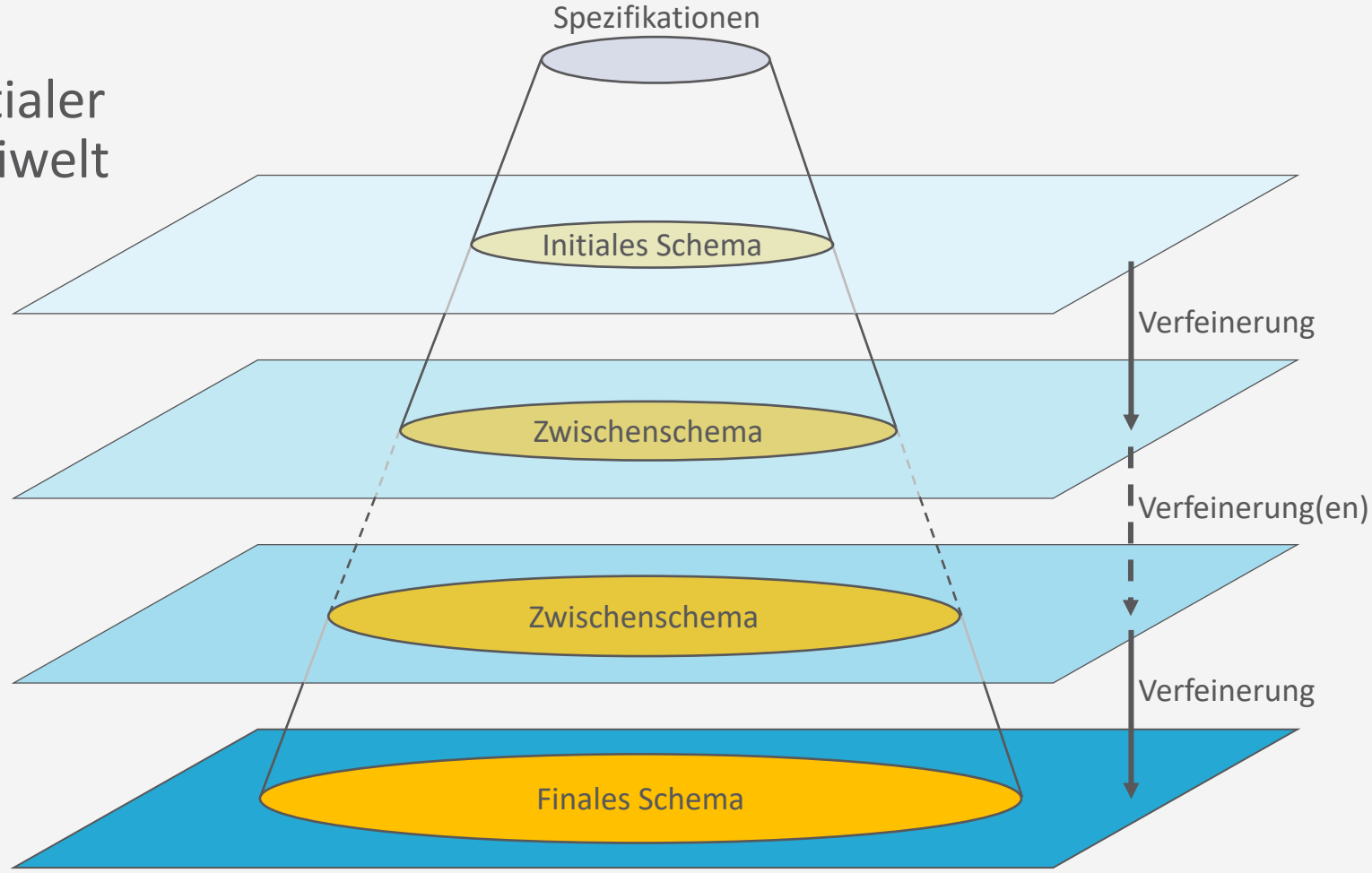
Vorgehen

Vorgehen

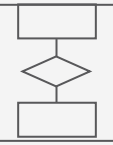
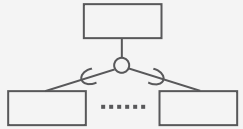
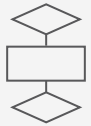
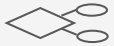
- Top-down
  - Sukzessiver Verfeinerungen initialer (abstrakter) Konzepte der Miniwelt
- Bottom-up
  - Zerlegung der Miniwelt-Darstellung in kleinste Konzepte
  - Abschließend Integration in ein Gesamtschema
- Inside-out
  - wie bottom-up, aber ausgehend von einem zentralen Konzept
    - Vom „Wichtigen“ zum „Un-wichtigen“
- Mixed
  - Mischform des Top-down- und des Bottom-up-Ansatzes
- Für ein gegebenes ER-Modell: verschiedene Transformationen

Top-down

- Sukzessive Verfeinerungen initialer (abstrakter) Konzepte der Miniwelt

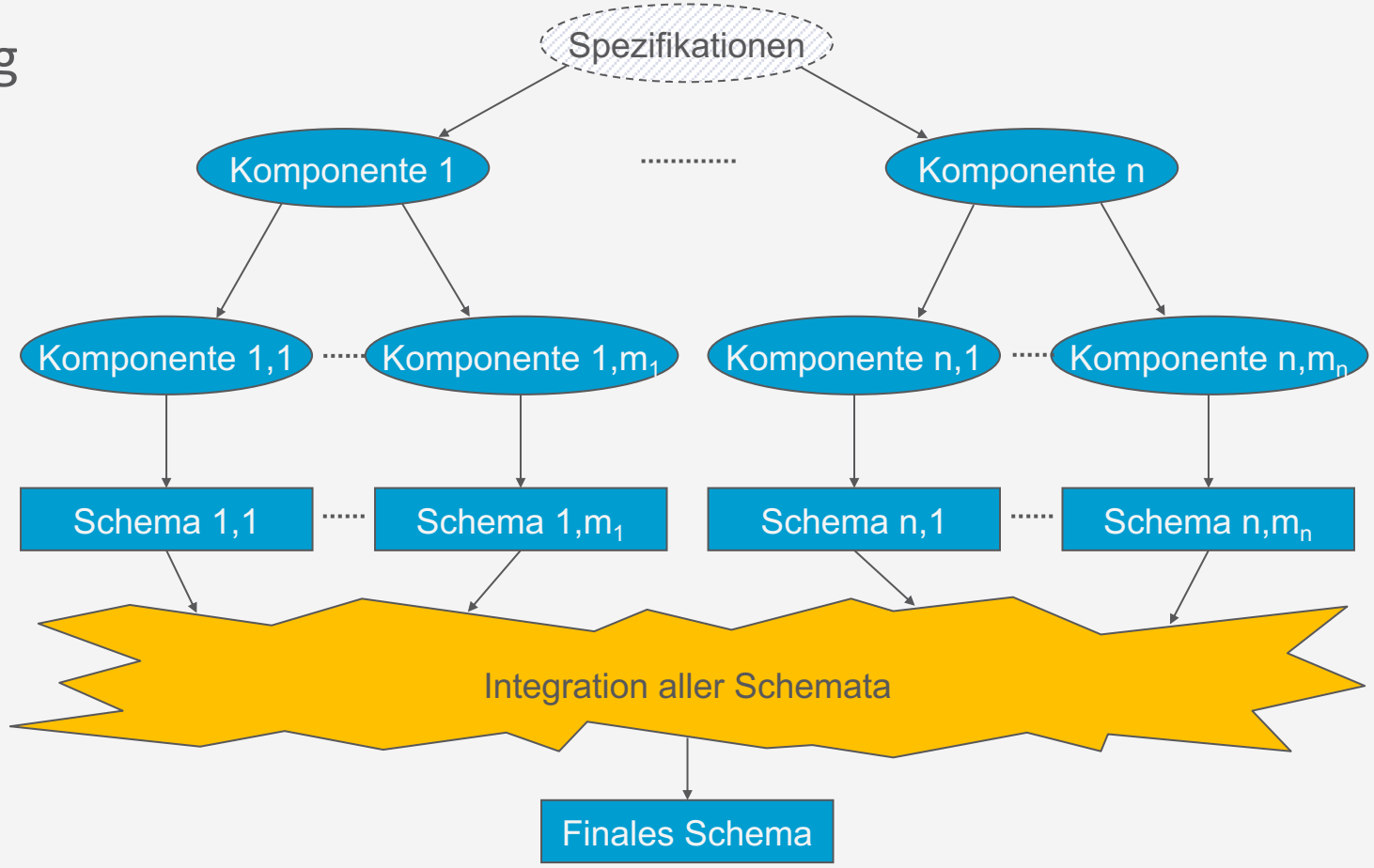


Top-down EER-Transformationen

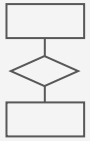
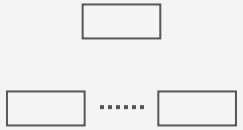
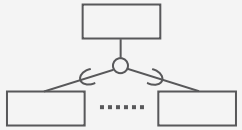
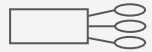
Transformation	Initiales Konzept	Ergebnis
T-TD$_1$:Von einer Entität zu zwei Entitäten mit einer Beziehung		
T-TD$_2$: Von einer Entität zu Entität und Spezialisierungen		
T-TD$_3$: Von einer Beziehung zu mehreren Beziehungen		
T-TD$_4$: Von einer Beziehung zu einer Entität mit Beziehungen		
T-TD$_5$: Hinzufügen von Attributen zu einer Entität		
T-TD$_6$: Hinzufügen von Attributen zu einer Beziehung		

Bottom-up

- Zerlegung der Miniwelt-Darstellung in kleinste Konzepte; abschließend Integration in ein Gesamtschema

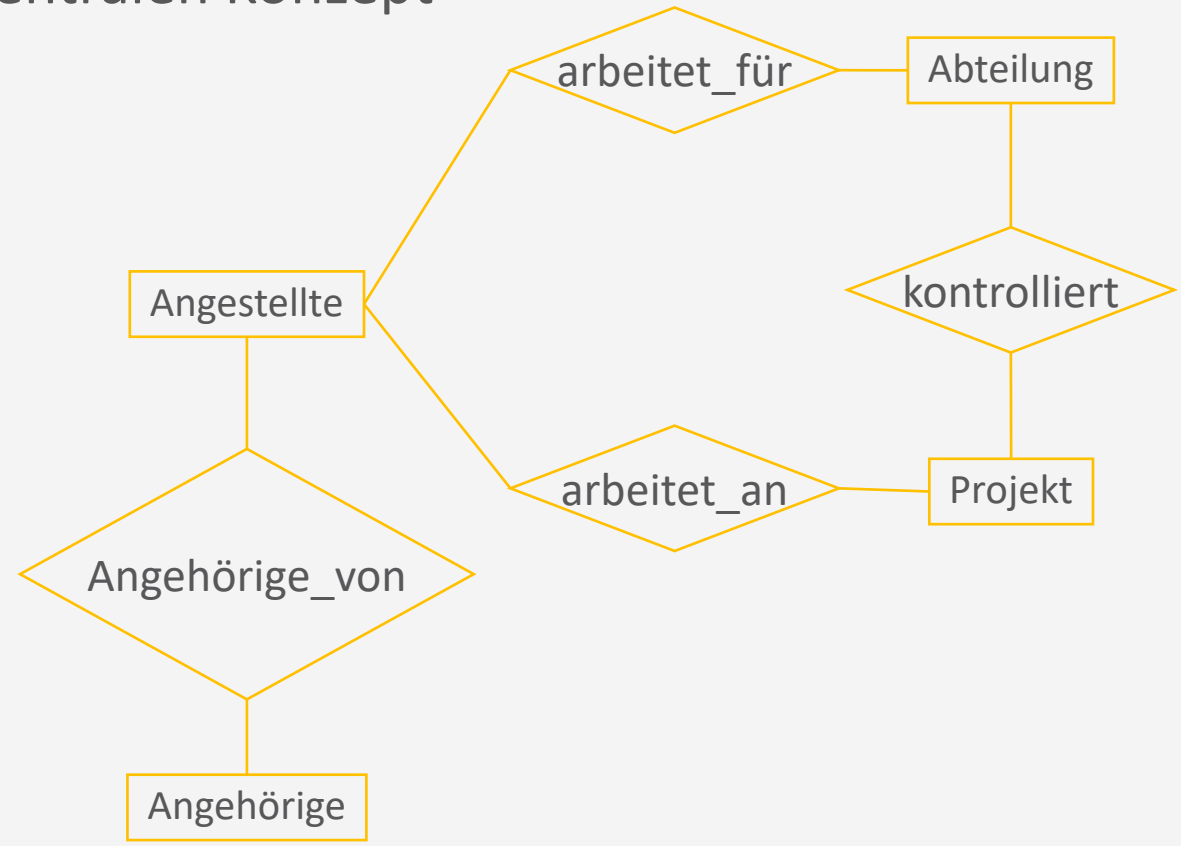


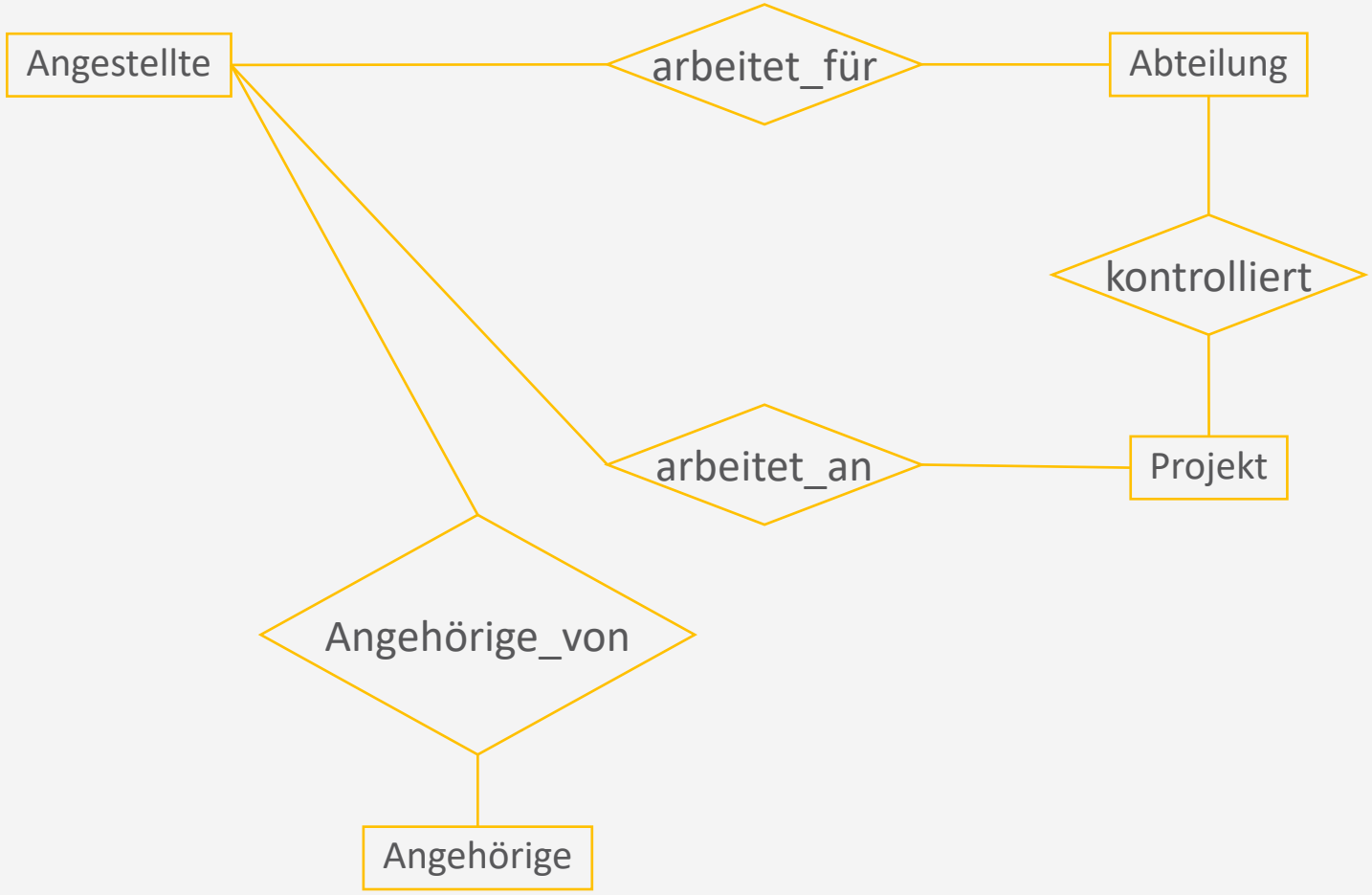
Bottom-up EER-Transformationen

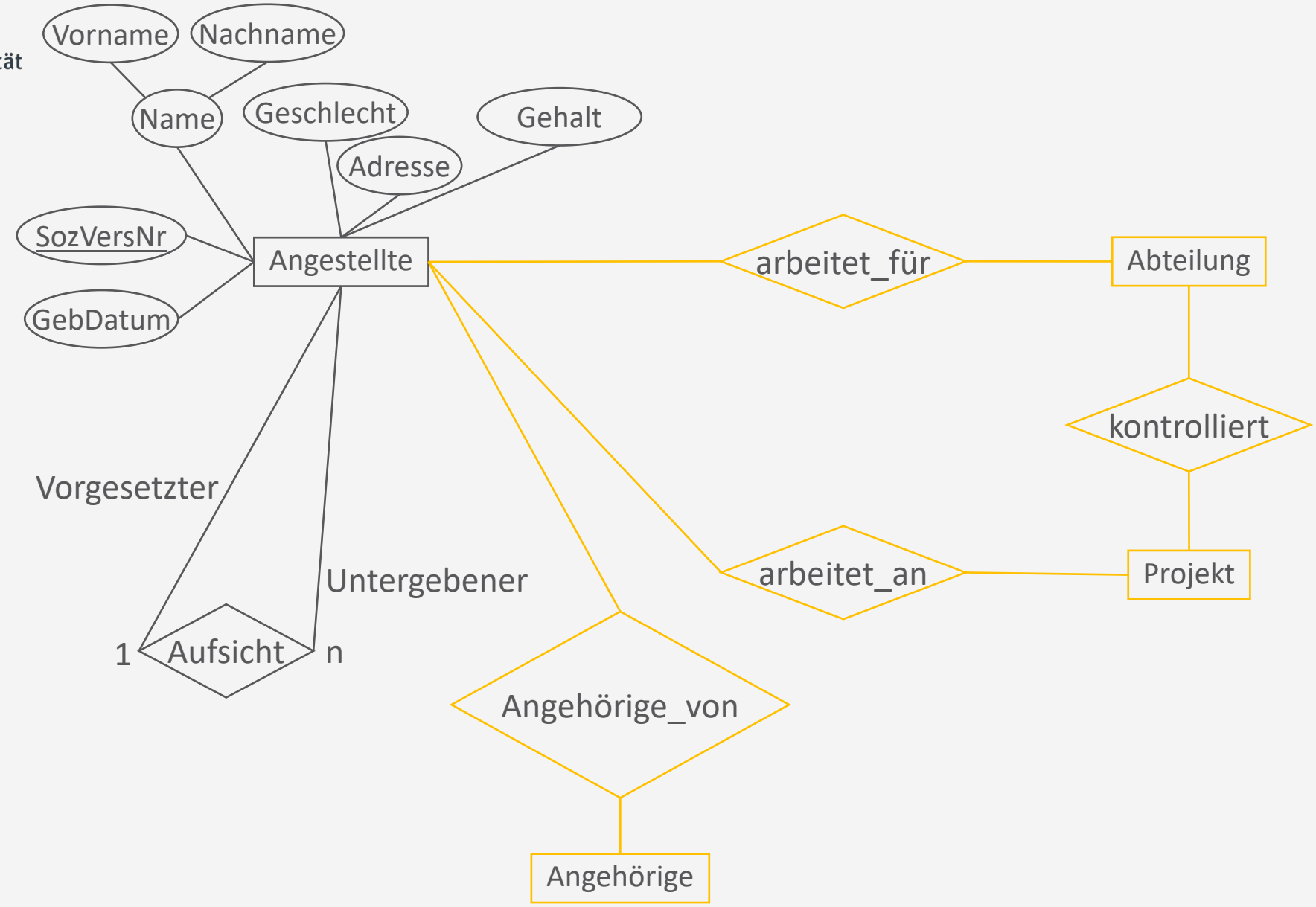
Transformation	Initiales Konzept	Ergebnis
T-BU$_1$: Erzeugung einer Entität		
T-BU$_2$: Erzeugung einer Beziehung		
T-BU$_3$: Erzeugung einer Generalisierung		
T-BU$_4$: Zusammenfassung und Zuordnung von Attributen zu einer Entität		
T-BU$_5$: Zusammenfassung und Zuordnung von Attributen zu einer Beziehung		

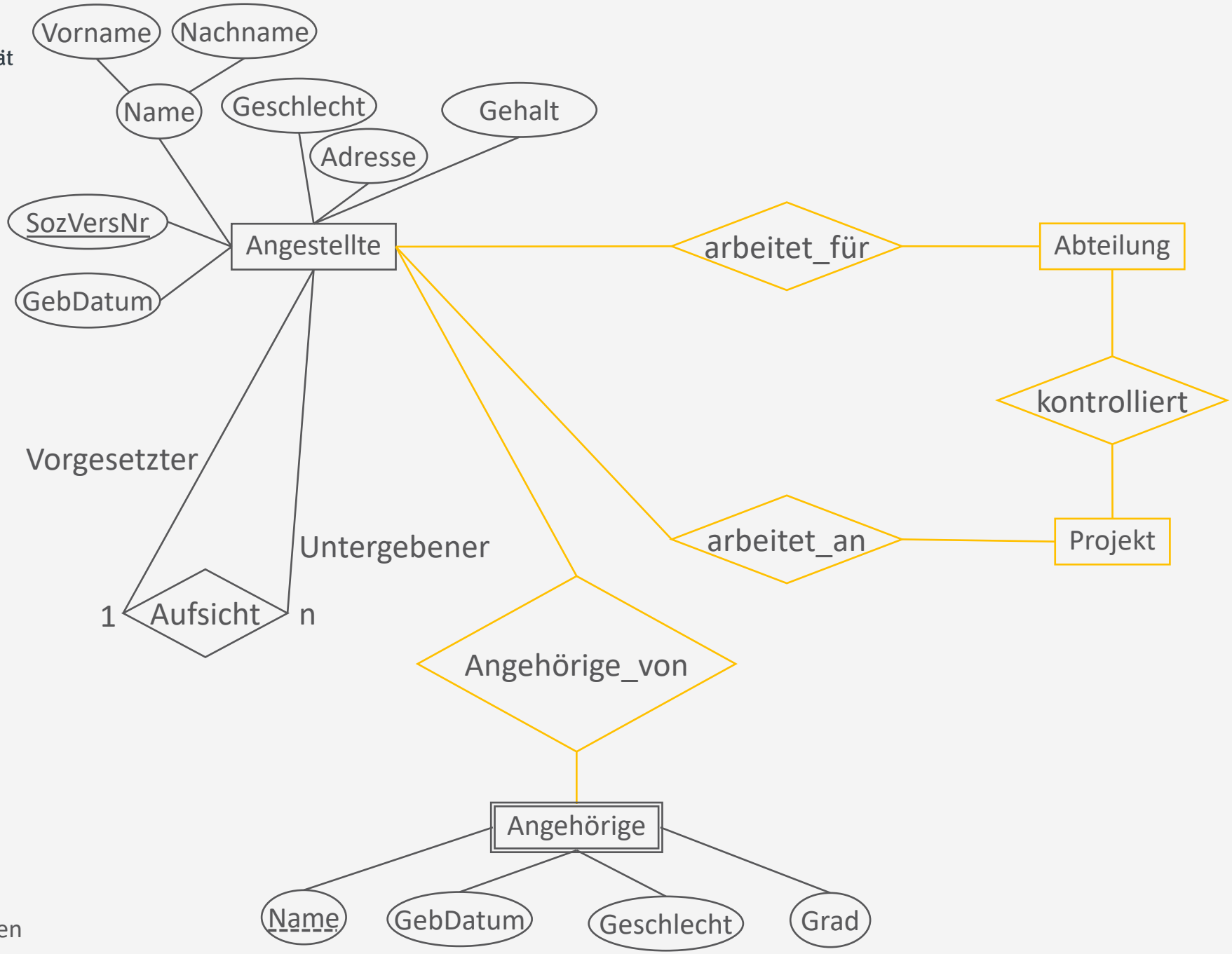
Inside-out

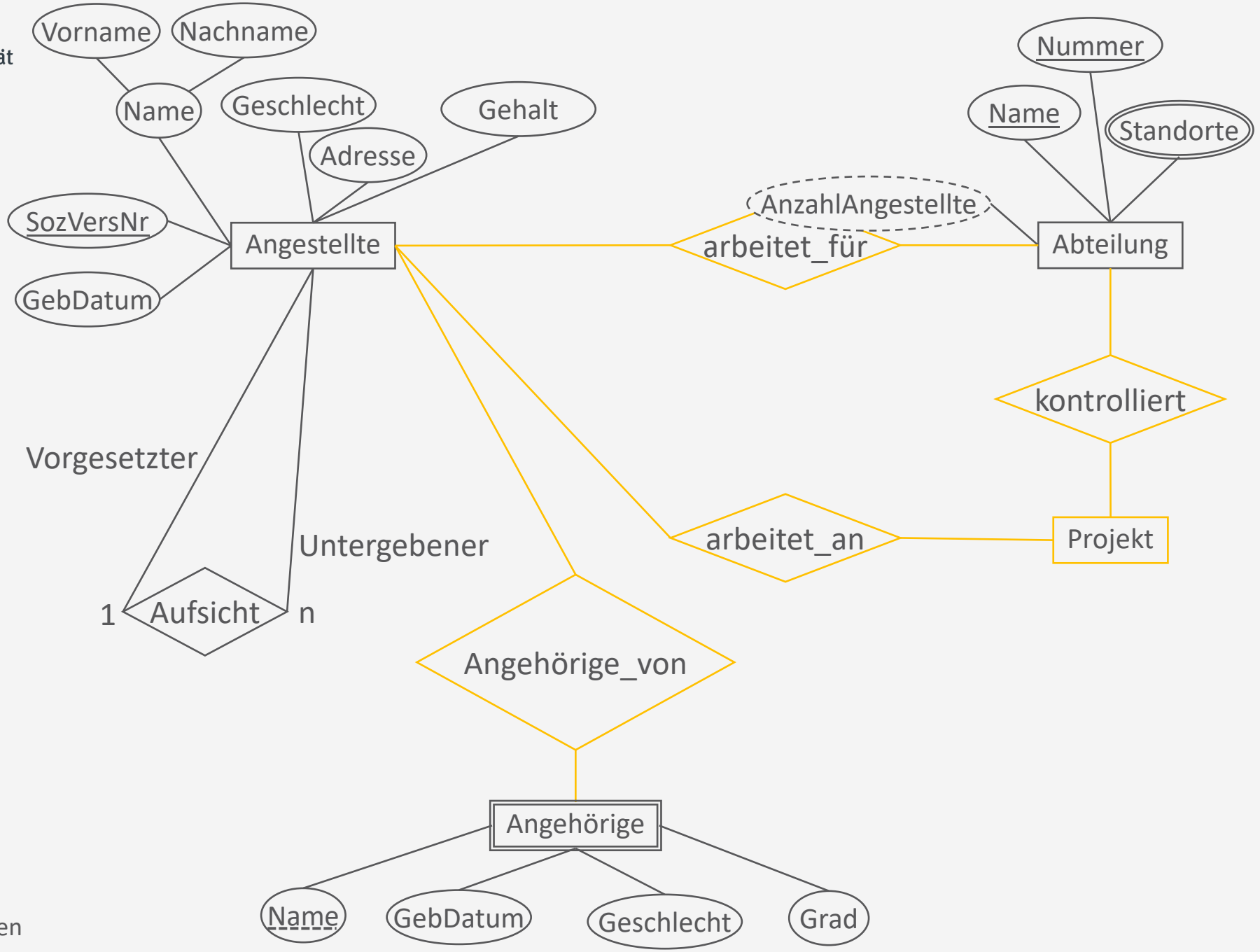
- Wie bottom-up, aber ausgehend von einem zentralen Konzept
  - Vom „Wichtigen“ zum „Un-wichtigen“
  - Beispiel
    - Entitäten
      - Angestellte
      - Abteilung
      - Projekt
      - Angehörige
    - Beziehungen
      - Angestellte ist Abteilung zugeordnet
      - Angestellte arbeitet für Projekt
      - Abteilung kontrolliert Projekt
      - Angestellte haben Angehörige

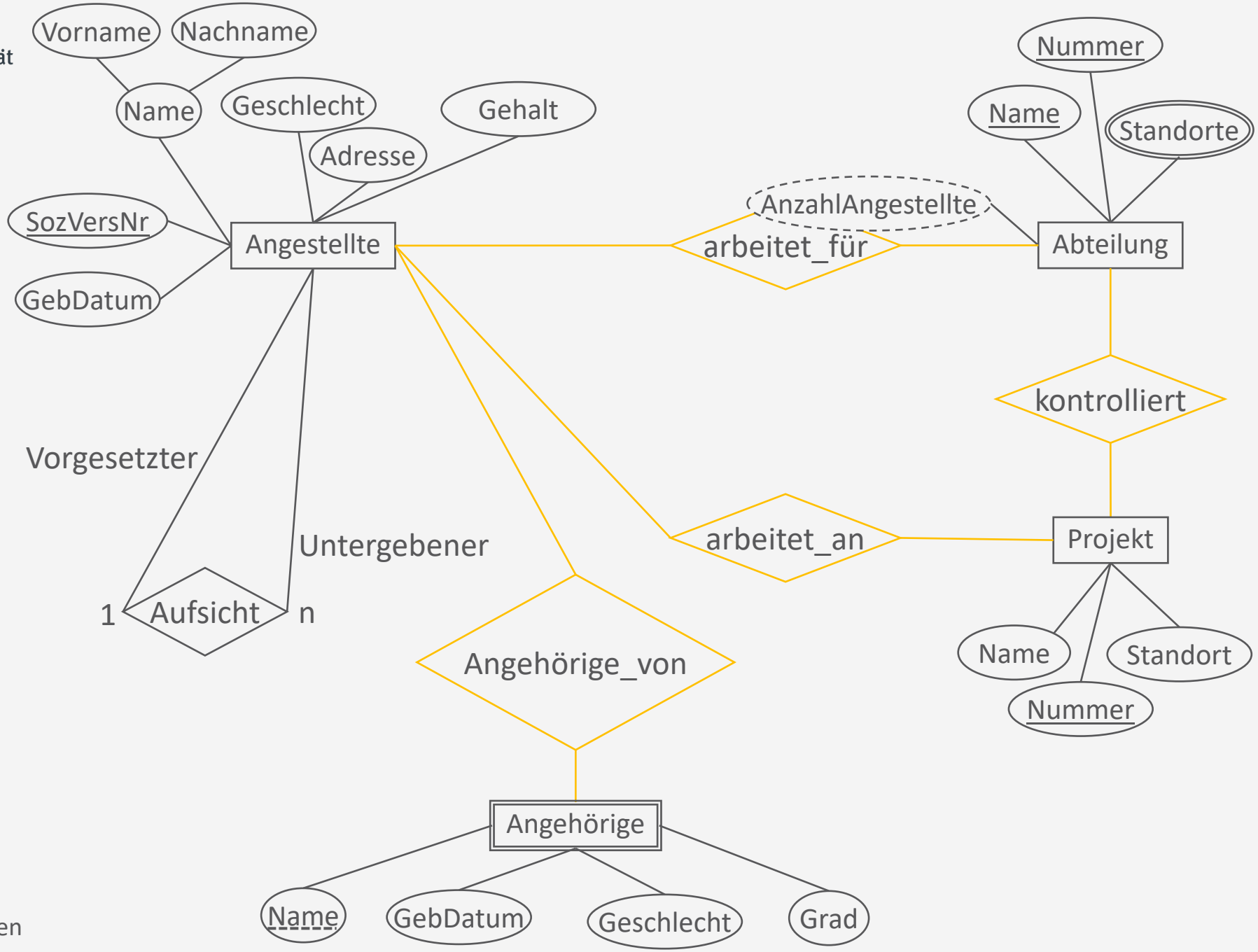


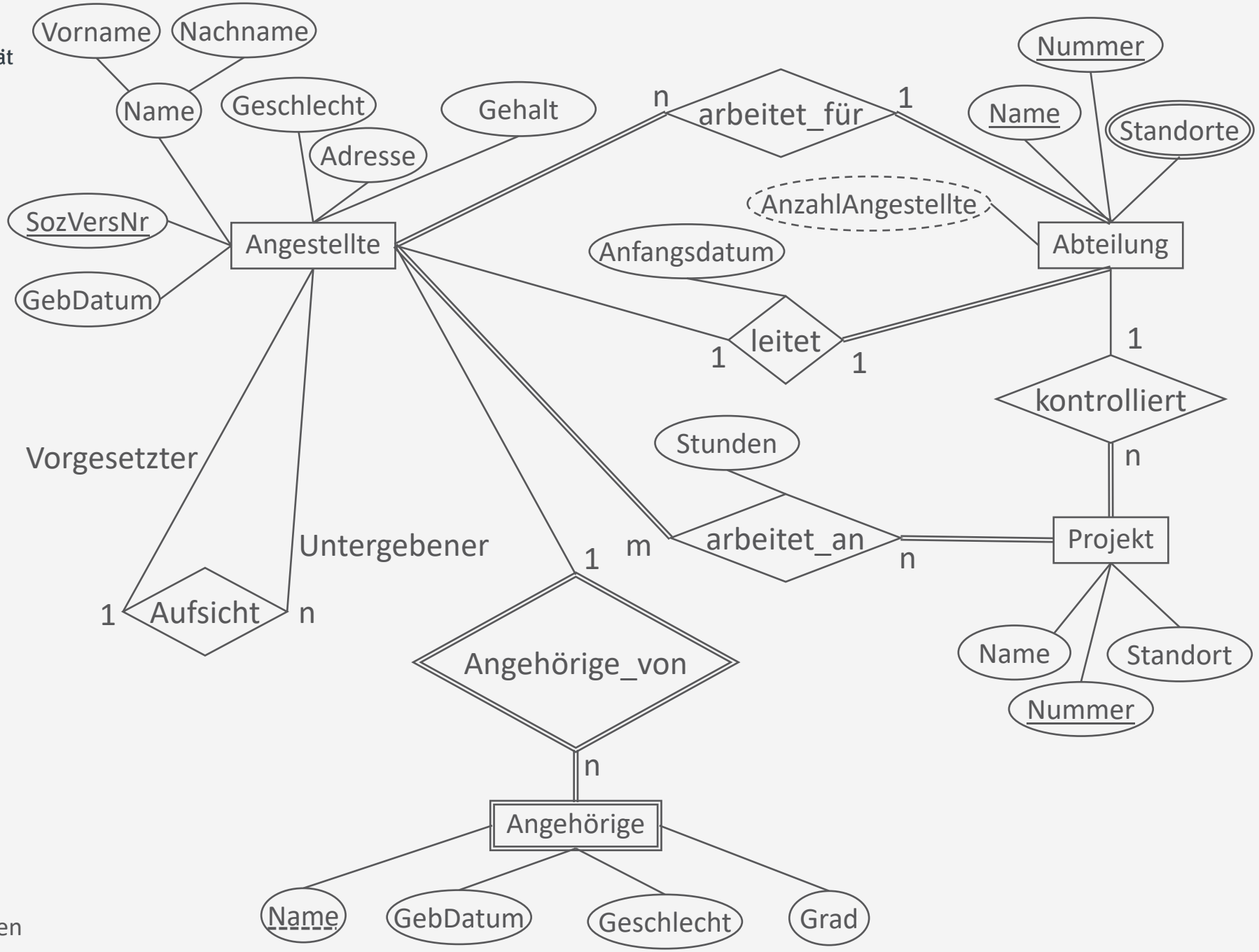

Angestellte
arbeitet_für
Abteilung
kontrolliert
arbeitet_an
Projekt
Angehörige_von
Angehörige


Vorname
Nachname
Name
Geschlecht
Gehalt
Adresse
SozVersNr
Angestellte
GebDatum
arbeitet_für
Abteilung
kontrolliert
Vorgesetzter
Untergebener
arbeitet_an
Projekt
1
Aufsicht
n
Angehörige_von
Angehörige


Vorname
Nachname
Name
Geschlecht
Gehalt
Adresse
SozVersNr
Angestellte
GebDatum
arbeitet_für
Abteilung
kontrolliert
Vorgesetzter
Untergebener
arbeitet_an
Projekt
1
Aufsicht
n
Angehörige_von
Angehörige
Name
GebDatum
Geschlecht
Grad


Vorname
Nachname
Name
Geschlecht
Gehalt
Adresse
SozVersNr
Angestellte
GebDatum
Vorgesetzter
Untergebener
1
Aufsicht
n
Nummer
Name
Standorte
AnzahlAngestellte
arbeitet_für
Abteilung
kontrolliert
arbeitet_an
Projekt
Angehörige_von
Angehörige
Name
GebDatum
Geschlecht
Grad

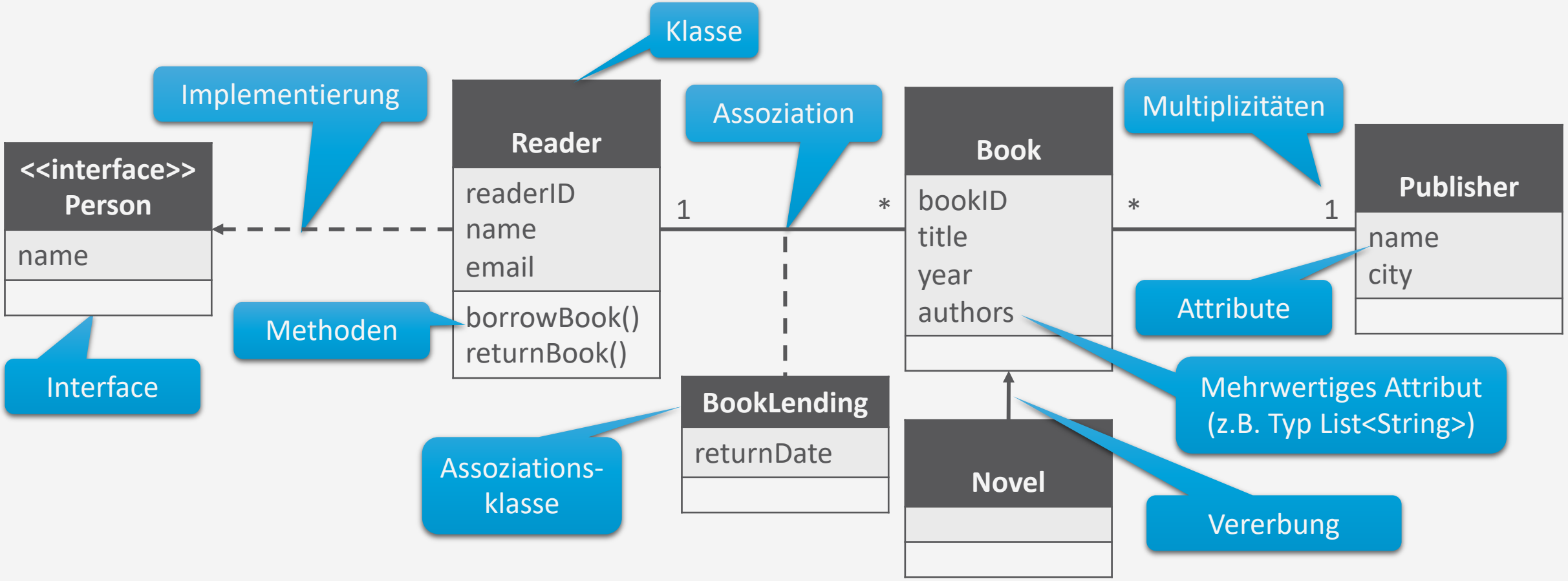

Vorname
Nachname
Name
Geschlecht
Gehalt
Adresse
SozVersNr
GebDatum
Angestellte
Vorgesetzter
Untergebener
1
Aufsicht
n
Nummer
Name
Standorte
AnzahlAngestellte
arbeitet_für
Abteilung
kontrolliert
arbeitet_an
Projekt
Name
Standort
Nummer
Angehörige_von
Angehörige
Name
GebDatum
Geschlecht
Grad


Vorname
Nachname
Name
Geschlecht
Adresse
Gehalt
SozVersNr
GebDatum
Angestellte
n
arbeitet_für
1
Nummer
Name
Standorte
AnzahlAngestellte
Abteilung
Anfangsdatum
leitet
1
1
1
kontrolliert
n
Stunden
Vorgesetzter
Untergebener
1
Aufsicht
n
m
arbeitet_an
n
Projekt
1
Angehörige_von
n
Name
Standort
Nummer
Angehörige
Name
GebDatum
Geschlecht
Grad

UML als Datenmodell

Für UML-Bekannte

Unified Modeling Language (UML)



Lena Wiese: Advanced Data Management, DeGruyter, 2015.

(E)ER vs. UML

(E)ER	UML
Entität	Klasse
Relation	Assoziation
Kardinalität: 1:1, 1:n, n:m oder min/max	Multiplizität: 0, 1, eine Zahl, Intervalle, *
Attribute	Attribute
Mehrwertige Attribute	Attribute des Typs List<type>
Schlüsselattribute	-- (als normales Attribut)
--	Methoden (im Allgemeinen)
Abgeleitete Attribute	Methoden (die ein berechnetes Attribut zurückgeben)
Spezialisierung/Gen.	Vererbung
--	Interface, Implementierung

gleiches Konzept, gleicher Name

gleiches Konzept, anderer Name

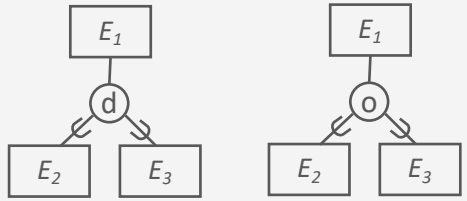
Konzept fehlt in ER

Konzept fehlt in UML

Ähnliches Konzept

Zwischenzusammenfassung

- EER-Modellierung
  - Spezialisierung
    - Disjunkt vs. überlappend
    - Partiell vs. total
    - Aufzählungstypen für definierende Attribute
  - Generalisierung als inverse Spezialisierung
    - is-a Beziehung als alternative Darstellung
- Vorgehensmodelle zur konzeptuellen Modellierung
  - Top-down
  - Bottom-up
  - Inside-out
- UML: einige gleiche bzw. ähnliche Konzepte zu (E)ER, Schlüsselattribut kein Konzept



Überblick: 2. Datenbank-Modellierung

A. Motivation
- Modelle und Modellierung

B. Entity-Relationship-Modell (ER)
- Komponenten
- Von Anforderungen zum ER-Modell
- Interpretation von ER-Modellen

C. Enhanced ER-Modell (EER)
- Spezialisierung, Generalisierung
- Modellierungsvorgehen
- Beziehung zu UML

D. Dokumentation & Bewertung
- Dokumentation
- Qualitätskriterien EER

Dokumentation

- Ein (E)ER-Schema ist i.d.R. nicht ausreichend, um alle interessierenden Details einer betrachteten Miniwelt zu beschreiben.
- Beispiele:
  - Jede Abteilung hat wenigstens fünf Mitarbeiter, die der Firma jeweils mehr als zehn Jahre angehören.
  - In keiner Abteilung gibt es Mitarbeiter, die ein höheres Gehalt bekommen als ihr jeweiliger Abteilungsleiter.
  - „JahresSonderzahlung“ ist ein abgeleitetes Attribut, welches sich stets wie folgt errechnet:
    - JahresSonderzahlung = 0,01 · Projektbudget
  - Ein Projektbudget liegt nie unter 50T€
- Lösung → Geschäftsregeln (Business Rules)

Geschäftsregeln (Business Rules)

- Beschreibend (*descriptions*):
  - strukturierte Dokumentation: z.B. Tabellen
- Einschränkungen (*constraints*):
  - <Konzept> muss / darf nicht <formale Einschränkung>
- Spezifikation abgeleiteter Größen (*derivations*):
  - <Konzept> wird gebildet über <Operation zur Gewinnung des Konzepts>

Beispiel: Beschreibende Regeln

Entität	Beschreibung	Attribute	Identifikator
ANGESTELLTE	Angestellter der Firma.	SozVersNr, GebDatum, Name, Geschlecht, Adresse, Gehalt	SozVersNr
PROJEKT	Ein Projekt der Firma, bei dem Angestellte mitwirken.	Name, Nummer, Standort	Nummer
...	...	...	...

Relation	Beschreibung	Beteiligte Entitäten	Attribute
LEITET	Ordnet einen Angestellten in der Rolle des Abteilungsleiters einer Abteilung zu.	Angestellter (1, 1) Abteilung (1, 1)	Anfangsdatum
ARBEITET_FÜR	Ordnet einen Angestellten einer Abteilung in der Rolle des Mitarbeiters zu.	Angestellter (1, N) Abteilung (1, 1)	
...	...	...	...

Beispiel: Einschränkungen & Ableitungen

Einschränkungen (Constraints)	
(BR1)	Ein Abteilungsleiter muss auch Mitarbeiter der Abteilung sein.
(BR2)	Ein Mitarbeiter darf kein höheres Gehalt beziehen als das des Abteilungsleiters der Abteilung, zu der er gehört.
(BR3)	Eine Abteilung mit Standort in Schweden muss von einem Mitarbeiter geleitet werden, der mindestens zehn Jahre durchgehend bei der Firma angestellt ist.
(BR4)	Ein Mitarbeiter, der keiner Abteilung zugeordnet ist, darf an keinem Projekt mitwirken.
...	
Ableitungen (Derivations)	
(BR7)	Die Anzahl der Angestellten einer Abteilung ergibt sich aus der Anzahl der Angestellten, die über die Relation ARBEITET_FÜR mit der jeweiligen Abteilung in Beziehung stehen.
...	

Qualitätskriterien für (E)ER-Schemata

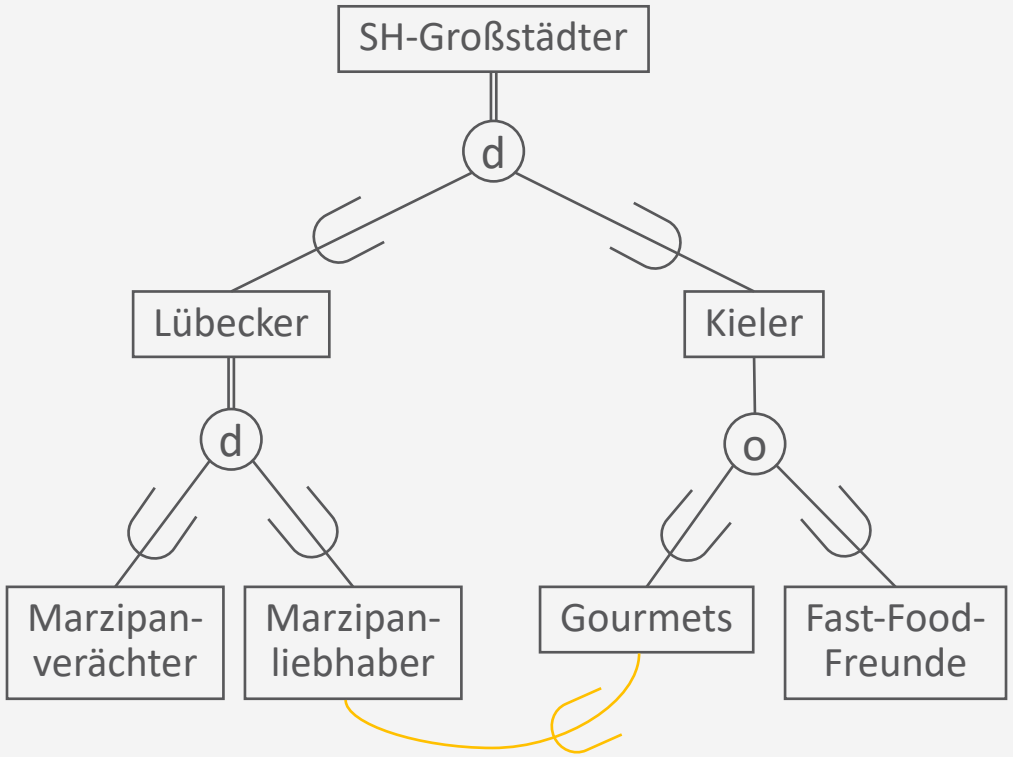
- Korrektheit
  - Modellierte Entitäten, Beziehungen, Attribute sind in der zugrunde liegenden Miniwelt enthalten, werden getreu der gegebenen Miniwelt-Darstellung in einen Zusammenhang gesetzt
- Vollständigkeit
  - Miniwelt-Darstellung wird (soweit möglich) vollständig durch das (E)ER-Schema wiedergegeben
- Minimalität
  - Im (E)ER-Schema berücksichtigte Konzepte, Beziehungen, Attribute der Miniwelt sind möglichst redundanzfrei modelliert
- Lesbarkeit
  - (E)ER-Schema ist übersichtlich organisiert, zeigt einen systematischen Aufbau
- Verständlichkeit
  - Modellierte (logische) Zusammenhänge entsprechen einer „natürlichen“ Intuition

Konsistenzprüfungen möglich

- Ein Beispiel aus dem Norden

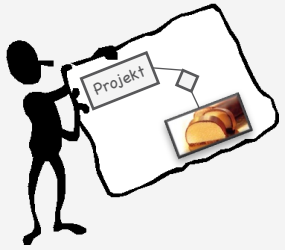
- Es gibt keine Marzipanliebhaber mehr
  → Alle Lübecker sind Marzipanverächter?!

- Konsistenzprüfungen
  - Automatische Bestimmung von leeren Begriffen
  - Prüfung auf globale Konsistenz



Zwischenzusammenfassung

- Dokumentation von (E)ER-Schemata: Business Rules
  - Beschreibungen
  - Einschränkungen
  - Ableitungen
- Qualitätskriterien für (E)ER-Schemata
  - Korrektheit, Vollständigkeit, Minimalität, Lesbarkeit, Verständlichkeit
  - EER-Schemata erlauben Konsistenzprüfungen



Überblick: 2. Datenbank-Modellierung

A. Motivation
- Modelle und Modellierung

B. Entity-Relationship-Modell (ER)
- Komponenten
- Von Anforderungen zum ER-Modell
- Interpretation von ER-Modellen

C. Enhanced ER-Modell (EER)
- Spezialisierung, Generalisierung
- Modellierungsvorgehen
- Beziehung zu UML

D. Dokumentation & Bewertung
- Dokumentation
- Qualitätskriterien EER

→ Das relationale Modell