

Datenbanken 1

Datenbankentwurf

Nikolaus Augsten

nikolaus.augsten@sbg.ac.at

FB Computerwissenschaften
Universität Salzburg

Sommersemester 2014

Inhalt

- 1 Datenbankentwurf und ER-Modell
- 2 Entitäten und Attribute
- 3 Beziehungen
 - Was sind Beziehungen?
 - Funktionalitäten
 - Rollen und Attribute
- 4 Generalisierung

Literatur und Quellen

Lektüre zum Thema "Datenbankentwurf":

- Kapitel 2 (außer 2.7.3, 2.13) aus Kemper und Eickler: Datenbanksysteme: Eine Einführung. 8. Auflage, Oldenbourg Verlag, 2011.

Literaturquellen

- Peter P. Chen: The Entity-Relationship Model - Toward a Unified View of Data. ACM TODS 1(1): 9-36 (1976)
- Silberschatz, Korth, and Sudarashan: Database System Concepts, McGraw Hill, 2006.
- Elmasri and Navathe: Fundamentals of Database Systems. Fourth Edition, Pearson Addison Wesley, 2004.

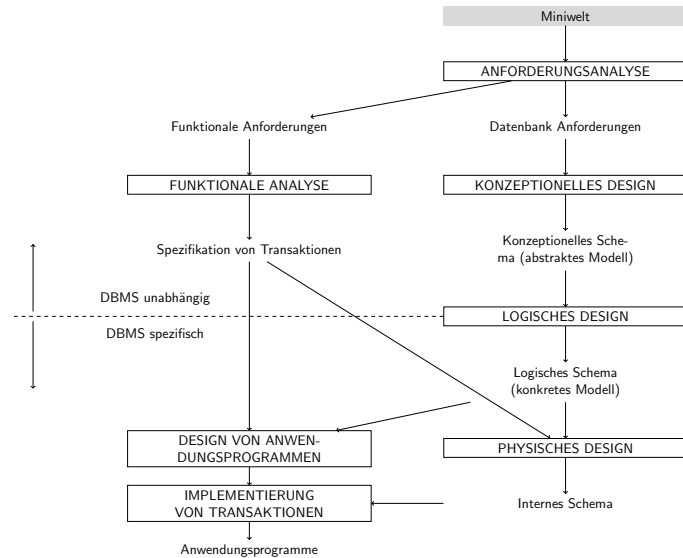
Danksagung Die Vorlage zu diesen Folien wurde entwickelt von:

- Michael Böhlen, Universität Zürich, Schweiz
- Johann Gamper, Freie Universität Bozen, Italien

Inhalt

- 1 Datenbankentwurf und ER-Modell
- 2 Entitäten und Attribute
- 3 Beziehungen
 - Was sind Beziehungen?
 - Funktionalitäten
 - Rollen und Attribute
- 4 Generalisierung

Der Datenbank Entwurfsprozess im Überblick



Beispiel: NAWI Datenbank

- Aufgrund der folgenden **Anforderungen** soll ein konzeptionelles Datenbankschema für eine NAWI Fakultätsdatenbank erstellt werden:
 - Die NAWI ist in Fachbereiche gegliedert. Jeder Fachbereich hat einen Namen, eine Nummer und einen Mitarbeiter, der den Fachbereich führt. Fachbereichsleitung beginnt ab einem bestimmten Datum. Ein Fachbereich kann an unterschiedlichen Standorten untergebracht sein.
 - Jeder Fachbereich ist für eine Anzahl von Projekten verantwortlich. Jedes Projekt hat einen eindeutigen Namen, eine eindeutige Nummer und wird an einem einzigen Standort durchgeführt.
 - Von jedem Mitarbeiter erfassen wir Sozialversicherungsnummer, Adresse, Lohn, Geschlecht und Geburtsdatum. Jeder Mitarbeiter arbeitet für nur einen Fachbereich, kann aber an mehreren Projekten arbeiten. Die Anzahl der Wochenstunden pro Projekt werden erfasst. Jeder Mitarbeiter hat einen direkten Vorgesetzten.
 - Jeder Mitarbeiter kann eine Anzahl von abhängigen Personen haben. Von jeder abhängigen Person erfassen wir Name, Geschlecht, Geburtstag und Art der Beziehung.

Das ER-Modell

- ER steht für Entity-Relationship
- Das ER-Modell hat **drei Hauptkonstrukte**:
 - Entitäten (entities)
 - Attribute (attributes)
 - Beziehungen (relationships)
- ER-Modell ist **konzeptionelles Datenmodell**
 - Datendefinitionssprache (DDL)
 - keine Datenmanipulationssprache (DML)
 - beschreibt Schema, nicht Instanzen
 - ähnliches Modell: Klassendiagramme in UML
- **Entwurfsprozess** ist eine *schrittweise Verfeinerung*.
 - Der erste Entwurf ist typischerweise nicht komplett (das ist vernünftig und kein Problem).
 - Der erste Entwurf wird iterativ verfeinert.

Tools für die Datenmodellierung

- **Tools für konzeptionelle Modellierung**
 - unterstützen Erstellung konzeptioneller Modelle
 - bilden konzeptionelles Modell auf relationales Modell ab
 - Beispiele: ERWin, Rational Rose, ER/Studio
- **Vorteile:**
 - dient als Dokumentation der Anforderungsanalyse
 - einfache Benutzerschnittstelle: graphische Unterstützung durch Editor
 - einfache graphische Modelle sind sehr intuitiv
- **Nachteile:**
 - Graphische Modelle werden schnell komplex und mehrdeutig
- **Einfache Zeichentools**
 - Graphikprogramme mit Erweiterung für ER-Diagramme
 - Beispiele: dia (Zeichenprogramm), tikz-er (Latex Package)

Inhalt

- 1 Datenbankentwurf und ER-Modell
- 2 Entitäten und Attribute
- 3 Beziehungen
 - Was sind Beziehungen?
 - Funktionalitäten
 - Rollen und Attribute
- 4 Generalisierung

Entitäten und Attribute

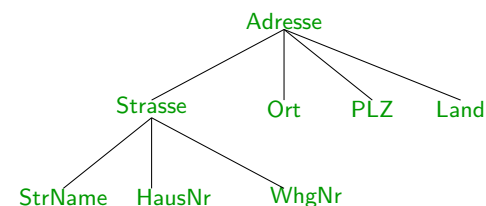
- **Entitäten** sind konkrete Gegenstände oder Konzepte der Miniwelt.
 - Bsp: "Mitarbeiter John Smith", "Fachbereich für Computerwissenschaften", "Projekt SyRA"
- **Attribute** sind Eigenschaften der Entitäten.
 - Bsp: Mitarbeiter John Smith hat Attribute **Name**, **SVN**, **Geschlecht**, **Geburtstag**, ...
 - Eine Entität hat einen Wert für jedes seiner Attribute.
 - Bsp: Mitarbeiter John Smith hat folgende Werte für seine Attribute:
 - Name** = 'John Smith',
 - Adresse** = '731, Fondren, Houston, TX',
 - Geburtstag** = '09-JAN-55'
 - Jedes Attribut hat eine zugehörige **Domäne** (Wertemenge, Datentyp).
 - Bsp: **Name** ist eine Zeichenkette, **Geburtstag** ist vom Typ Datum

Attribute/1

- **Einfache Attribute**
 - Jede Entität hat einen einfachen, atomaren Wert für das Attribut
 - Bsp: **SVN**, **Geschlecht**.
- **Zusammengesetzte Attribute**
 - Attribute sind aus mehreren Komponenten zusammengesetzt.
 - Notation: **Attribut(Komponente1, Komponente2, ..., KomponenteN)**
 - Beispiele:
 - **Adresse(Strasse, HausNr, WhgNr, PLZ, Ort, Land)**
 - **Name(Vorname, Nachname)**
- **Mehrwertige Attribute**
 - Eine Entität kann mehrere Werte für ein Attribut haben.
 - Notation: { **Attribut** }
 - Bsp: { **Telefonnummer** }, { **Farbe** }
- **Abgeleitete Attribute**
 - Attribute können abgeleitet (berechnet) sein.
 - Bsp: **AnzahlMitarbeiter** kann berechnet werden

Attribute/2

- Zusammengesetzte und mehrwertige Attribute können **beliebig verschachtelt** werden.
- **Beispiel:** zusammengesetztes, mehrwertiges Attribut **Abschlüsse**:
 { **Abschlüsse (Institution, Jahr, Diplom, Fachgebiet)** }
 - mehrere Abschlüsse sind möglich
 - jeder Abschluss hat vier Attribute: **Institution, Jahr, Diplom, Fachgebiet**
- **Beispiel:** hierarchisch zusammengesetztes Attribut **Adresse**:
Adresse(Strasse(StrName, HausNr, WhgNr), Ort, PLZ, Land)



Entitätstypen und Schlüsselattribute

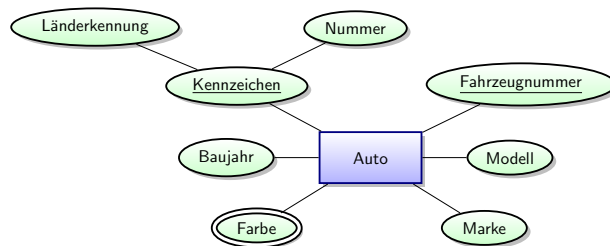
- **Entitätstyp**: fasst Entitäten mit den gleichen Attributen zusammen.
 - Bsp: Entitätstypen **Mitarbeiter** und **Projekte**
- **Schlüsselattribut**: Attribut eines Entitätstyps, für den jede Entität einen eindeutigen Wert hat.
 - Bsp: Schlüsselattribut **SVN** von **Mitarbeiter**
- Ein Schlüsselattribut **kann zusammengesetzt sein**.
 - **Kennzeichen** mit den Komponenten (**Länderkennung**, **Nummer**) ist ein Schlüssel des **Auto** Entitätstyps.
- Ein Entitätstyp kann **mehr als einen Schlüssel** haben.
 - Der **Auto** Entitätstyp hat zwei mögliche Schlüssel:
 - **Fahrzeugnummer**
 - **Kennzeichen (Länderkennung, Nummer)**
- Schlüsselattribute werden unterstrichen.

Darstellung von Entitätstypen in ER-Diagrammen

- **ER-Diagramm**: graphische Darstellung der ER-Modellierung
- **Entitätstyp** wird als **Rechteck** dargestellt.
- **Attribut** wird als **Oval** dargestellt.
 - Attribut ist mit einem Entitätstyp verbunden
 - **zusammengesetzten Attribute**: Komponenten werden mit zusammengesetztem Attribut verbunden
 - **mehrwertige Attribute**: werden in doppelten Ovalen dargestellt
 - **abgeleitete Attribute**: werden als gepunktete Ovale dargestellt
 - **Schlüssel**: werden unterstrichen

ER-Diagramm – Beispiel

- **Bsp**: Entitätstyp **Auto** mit Attributen **Kennzeichen**(**Länderkennung**, **Nummer**), **Fahrzeugnummer**, **Marke**, **Modell**, **Baujahr**, { **Farbe** }



Integrierte Übung 2.1

Gegeben ist ein Entitätstyp R mit Attributen A , B , C , D und E . Schlüssel sind $\{A\}$, $\{B, E\}$ und $\{C\}$.
Stellen Sie den Entitätstyp R mit Hilfe der ER-Notation dar.

Entitätsmengen

- Eine **Entitätsmenge** besteht aus der Menge aller Entitäten eines bestimmten Entitätstypen.
- Für eine Entitätsmenge und einen Entitätstyp wird der **gleiche Name** verwendet (z.B. **Auto**).
- Eine Entitätsmenge stellt den **aktuellen Zustand** eines Teiles der Datenbank dar.
- Entitätsmenge **Auto**:

```

Auto1
((ABC 123, TEXAS), TK629, Ford Mustang, convertible, 2004, {red, black})

Auto2
((ABC 123, NEW YORK), WP9872, Nissan Maxima, 4-door, 2005, {blue})

Auto3
((VSY 720, TEXAS), TD729, Chrysler LeBaron, 4-door, 2002, {white, blue})

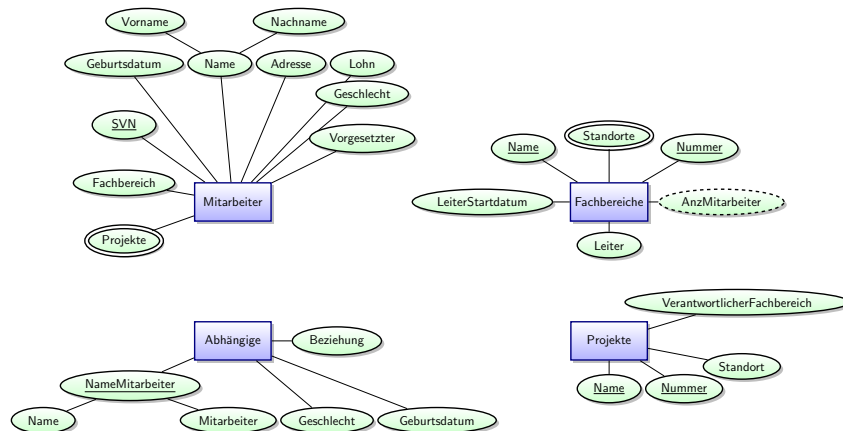
.
.
.

```

Entitätstypen der NAWI Datenbank/1

- Ausgehend von der Anforderungsanalyse identifizieren wir vier Entitätstypen der NAWI Datenbank:
 - **Fachbereiche**
 - **Projekte**
 - **Mitarbeiter**
 - **Abhängige**
- Die Attribute werden aus den Anforderungen abgeleitet.
- Richtlinien für die Bestimmung von Entitätstypen und Attributen:
 - *Substantive* in einer Beschreibung werden als Entitätstypen abgebildet.
 - *Substantive die Entitätstypen beschreiben* werden als Attribute abgebildet.

Entitätstypen der NAWI Datenbank/2



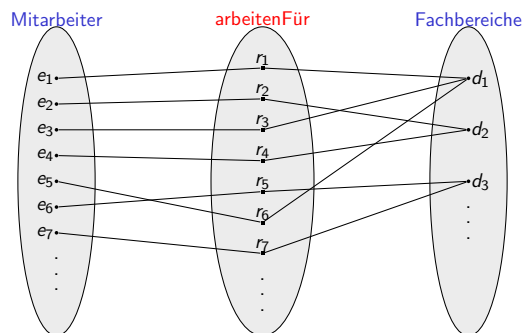
Inhalt

- 1 Datenbankentwurf und ER-Modell
- 2 Entitäten und Attribute
- 3 **Beziehungen**
 - Was sind Beziehungen?
 - Funktionalitäten
 - Rollen und Attribute
- 4 Generalisierung

Beziehungen und Beziehungstypen/1

- **Beziehung** erstellt Zusammenhang mit spezifischer Bedeutung zwischen mehreren Entitäten:
 - Mitarbeiter John Smith **arbeitet an** Projekt Syra
 - Mitarbeiter Andreas Uhl **leitet** den Fachbereich für Computerwissenschaften
- **Beziehungstypen** gruppieren Beziehungen des gleichen Typs:
 - der **arbeitenAn** Beziehungstyp zwischen **Mitarbeiter** und **Projekten**
 - der **leiten** Beziehungstyp zwischen **Mitarbeiter** und **Fachbereichen**
- **Ordnung** des Beziehungstyps: Anzahl der involvierten Entitätstypen
 - sowohl **leiten** als auch **arbeitenAn** sind *binäre* Beziehungstypen
 - binäre Beziehungstypen sind weitaus die häufigsten

Die arbeitenFür Beziehung



Beispiele für

- Entitätsmenge: **Mitarbeiter** = {e₁, e₂, e₃, ...}
- Entität: e₁, e₆, d₃
- Fachbereiche = {d₁, d₂, d₃, ...}
- Beziehungsmenge: **arbeitenFür**
- Beziehung: r₁ = (e₁, d₁), r₅ = (e₆, d₃)

Beziehungen und Beziehungstypen/2

- **Beziehungstyp:**
 - identifiziert Namen der Beziehung und involvierten Entitätstypen
 - identifiziert Einschränkungen (constraints) für Beziehungen
 - beschreibt eine Beziehung auf Schemaebene
- **Beziehungsmenge:**
 - Menge von Beziehungen, die in der Datenbank dargestellt sind
 - beschreibt Beziehungen auf Instanzebene
 - Beziehungsmenge R für Entitätsmengen E_1, E_2, \dots, E_n ist definiert als:

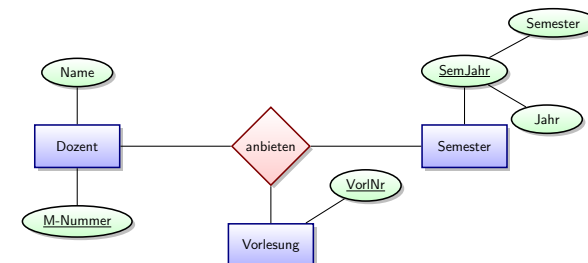
$$R \subseteq E_1 \times E_2 \times \dots \times E_n$$

- **Element der Beziehungsmenge (Beziehung):**
 - stellt Beziehung zwischen Entitäten her
 - genau 1 Entität jeder involvierten Entitätsmenge ist Teil der Beziehung
 - Beziehung r für Entitätsmengen E_1, E_2, \dots, E_n ist definiert als:

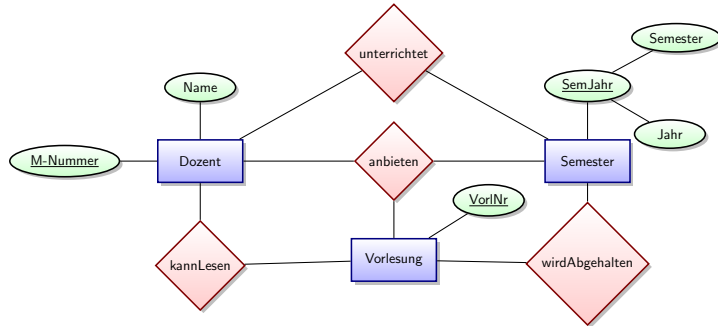
$$r \in E_1 \times E_2 \times \dots \times E_n$$

Beziehungen höherer Ordnung/1

- Beziehungstypen 2. Ordnung sind **binäre** Beziehungstypen.
- Beziehungstypen 3. Ordnung sind **ternäre** und jene der n-ten Ordnung sind **n-wertige** Beziehungstypen.



Beziehungen höherer Ordnung/2

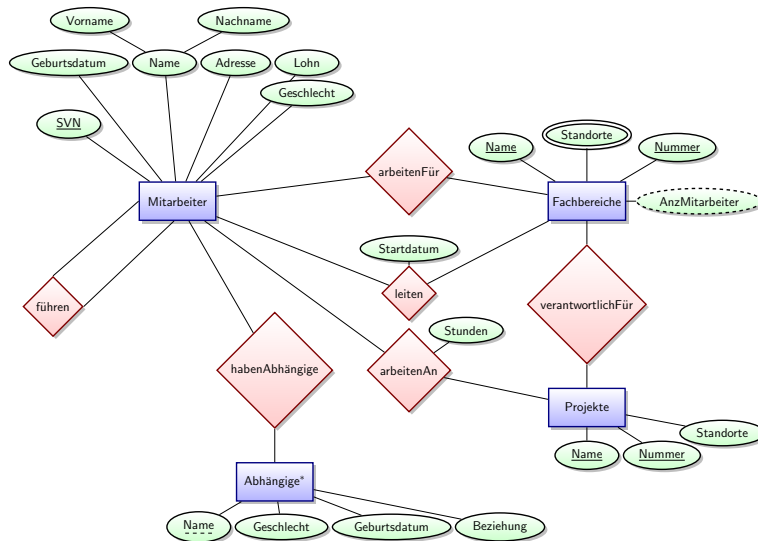


- n -wertige Beziehung ist **nicht äquivalent** zu n binären Beziehungen:
 - Beziehung **unterrichtet** kann von ternärer Beziehung **anbieten** hergeleitet werden
 - Beziehung **kannLesen** kann *nicht* von ternären Beziehung **anbieten** hergeleitet werden
 - Beziehung **anbieten** kann nicht aus **unterrichtet**, **kannLesen**, **wirdAbgehalten** hergeleitet werden

Beziehungen in der NAWI Datenbank/1

- Aus den Anforderungen lassen sich **6 Beziehungstypen** ableiten
- Alle Beziehungen sind **binär** (d.h. stellen eine Beziehung zwischen zwei Entitäten her)
- Beziehungstypen mit **involvierten Entitätstypen**:
 - **arbeitenFür** (zwischen **Mitarbeiter** und **Fachbereiche**)
 - **leiten** (zwischen **Mitarbeiter** und **Fachbereiche**)
 - **verantwortlichFür** (zwischen **Fachbereiche** und **Projekte**)
 - **arbeitenAn** (zwischen **Mitarbeiter** und **Projekte**)
 - **führen** (zwischen **Mitarbeiter** (als **Unterstellter**) und **Mitarbeiter** (als **Vorgesetzter**))
 - **habenAbhängige** (zwischen **Mitarbeiter** und **Abhängige**)
- In ER-Diagrammen werden **Beziehungstypen wie folgt dargestellt**:
 - ein Rhombus wird verwendet um einen Beziehungstypen darzustellen
 - der Rhombus ist mit den involvierten Entitätstypen verbunden

Beziehungstypen in der NAWI Datenbank/2



* Die Entität **Abhängige** hat keinen vollständigen Schlüssel. Diese Art von Entitäten heißt **existenzabhängig** und wird später behandelt.

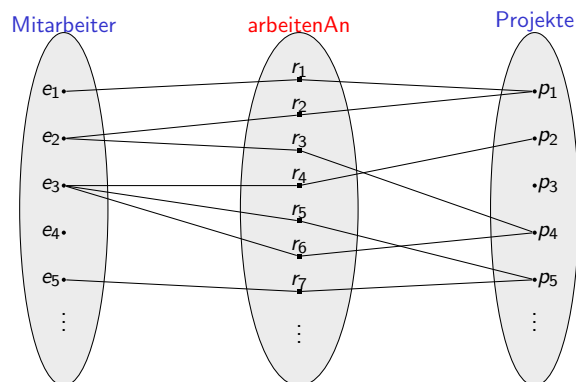
Diskussion von Beziehungstypen

- Im verfeinerten Entwurf werden einige **Attribute** von Entitätstypen **durch Beziehungen dargestellt**:
 - Leiter eines Fachbereichs -> **leiten**
 - Projekte von Mitarbeiter -> **arbeitenAn**
 - Fachbereich von Mitarbeiter -> **arbeitenFür**
 - usw.
- Zwischen Entitätstypen können **mehrere Beziehungstypen** existieren:
 - **leiten** und **arbeitenFür** sind unterschiedliche Beziehungstypen zwischen **Mitarbeiter** und **Fachbereiche**
 - diese Beziehungstypen haben unterschiedliche Bedeutung und unterschiedliche Beziehungsinstanzen

Funktionalitäten von Beziehungstypen

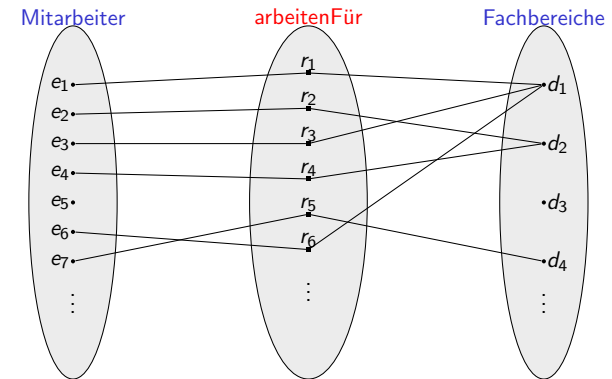
- **Funktionalitäten** schränken die Anzahl der möglichen Kombinationen von Entitäten in einer Beziehungsmenge ein.
- **Kardinalitätseinschränkung** spezifiziert *Obergrenze* für Häufigkeit einer Entität in Beziehungen
 - 1:1 Beziehung
 - 1:N Beziehung (analog eine N:1 Beziehung)
 - M:N Beziehung
- **Teilnahmebeschränkung** spezifiziert *Untergrenze* für Häufigkeit einer Entität in Beziehungen
 - 0 (optionale Teilnahme)
 - 1 oder mehr (zwingende Teilnahme)

M:N Beziehung



- Mitarbeiter:Projekte = M:N
- Ein Mitarbeiter arbeitet an maximal N Projekten.
- Ein Projekt wird von maximal M Mitarbeitern bearbeitet.

N:1 Beziehung



- Mitarbeiter:Fachbereiche = N:1
- Ein Mitarbeiter arbeitet für maximal 1 Fachbereich.
- Ein Fachbereich beschäftigt maximal N Mitarbeiter.

Notation für Funktionalitäten/1

Kardinalitätseinschränkungen binärer Beziehungen:

- **Notation:** durch Beschriftung der Kanten mit Zahlen



- **Interpretation:**
 - für einen bestimmten Mitarbeiter gibt es 1 Fachbereich, mit dem er in der *arbeitetFür* Beziehung stehen kann
 - für einen bestimmten Fachbereich gibt es N Mitarbeiter, mit denen er in der *arbeitetFür* Beziehung stehen kann
- **n-wertiger Beziehungstyp:**
 - *Kardinalitätseinschränkung* spezifiziert, wie oft eine Entität für eine konkrete Instanz aller anderen Entitäten vorkommen darf
- **Richtlinie für Leserichtung:** links nach rechts, oben nach unten
 - Ein Mitarbeiter arbeitet für 1 Fachbereich.
 - umgekehrte Leserichtung: Verb wird geändert, z.B. Ein Fachbereich beschäftigt N Mitarbeiter.

Formale Definition von Funktionalitäten

- Beziehungen mit Funktionalitäten definieren **partielle Funktionen** zwischen Entitätsmengen.
 - eine Funktion $F : X \rightarrow Y$ ist *partiell*, wenn nicht jedem Element von X ein Element von Y zugeordnet werden muss
 - eine nicht-partielle Funktion ist *total*
- Gegeben eine **n-wertige Beziehung** R zwischen den Entitätsmengen E_1, E_2, \dots, E_n mit den jeweiligen Kardinalitätseinschränkungen K_1, K_2, \dots, K_n , wobei K_i ein eindeutiger Buchstabe zugeordnet ist (z.B. M, N), oder $K_i = 1$. R definiert folgende partielle Funktion:

$$R : E_1 \times E_2 \times \dots \times E_{k-1} \times E_{k+1} \times \dots \times E_n \rightarrow E_k$$

für jedes $k, 1 \leq k \leq n$ mit $K_k = 1$.

Beispiel: Formale Definition von Funktionalitäten

- Die $1 : M : N$ -Beziehung "prüfen" zwischen Professor, Student, und Vorlesung definiert die partielle Funktion:

$$\text{prüfen} : \text{Student} \times \text{Vorlesung} \rightarrow \text{Professor}$$

- Die $1 : 1 : N$ -Beziehung "betreuen" zwischen Professor, Seminarthema und Student definiert folgende partielle Funktionen:

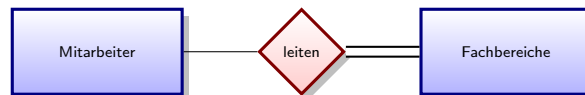
$$\text{betreuen} : \text{Student} \times \text{Seminarthema} \rightarrow \text{Professor}$$

$$\text{betreuen} : \text{Student} \times \text{Professor} \rightarrow \text{Seminarthema}$$

Notation für Funktionalitäten/2

Teilnahmebeschränkung einer Beziehung:

- **Notation:**
 - total (zwingend); Notation: doppelte Linie
 - partiell (optional); Notation: einfache Linie



- **Interpretation:**
 - Jeder Fachbereich muss eine Beziehung **leiten** eingehen, d.h., jeder Fachbereich wird von einem Mitarbeiter geleitet.
 - Ein Mitarbeiter muss nicht unbedingt die Beziehung **leiten** eingehen, d.h., muss keinen Fachbereich leiten.
- Nicht alle real existierenden Einschränkungen lassen sich mit Hilfe von ER-Diagrammen modellieren.

Integrierte Übung 2.2

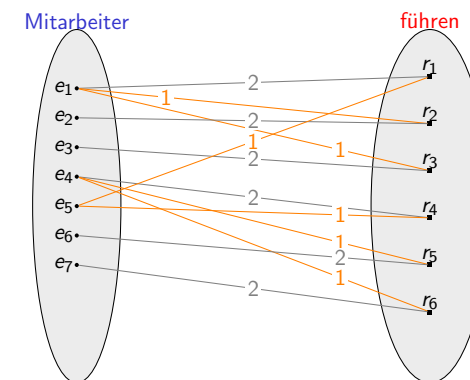
Ein Flugreservierungssystem erfasst Daten von Passagieren, die über ihre Ticket Nummer identifiziert werden. Zu jedem Flug werden das Abfluggate und die Sitzplatzreservierungen erfasst.

Erstellen Sie die Beziehungstypen eines ER-Schemas und deren Funktionalitäten.

Rekursive Beziehungen/1

- In einem rekursiven Beziehungstypen kommt der gleiche Beziehungstyp in **unterschiedlichen Rollen** vor.
- Zum Beispiel der **führen** Beziehungstyp zwischen
 - einem Mitarbeiter in der Rolle als Vorgesetzter und
 - einem Mitarbeiter in der Rolle des Unterstellten.
- Jede Beziehung (Element der Beziehungsmenge) verbindet zwei Mitarbeiter:
 - ein Angestellter in der Rolle des *Vorgesetzten*
 - ein Mitarbeiter in der Rolle des *Unterstellten*
- Zwei Rollen eines Entitätstypen werden in Beziehungen **wie zwei verschiedene Entitätstypen** betrachtet, das heißt:
 - eine Entität kann in einer Beziehung (Element der Beziehungsmenge) mehrfach in verschiedenen Rollen aufscheinen.
 - die Funktionalität bezieht sich auf die Rolle, d.h., eine Entität kann in einer Rolle mit "1" beschränkt sein, in einer anderen Rolle mit *N*.
- In **ER-Diagrammen** geben wir die Rollennamen an, um die unterschiedlichen Rollen des Entitätstypen zu identifizieren.

Rekursive Beziehungen/2



- **Annotation 1** steht für die Rolle des **Vorgesetzten**
- **Annotation 2** steht für die Rolle des **Unterstellten**
- e_1 ist der Vorgesetzte von e_2
- e_1 ist der Unterstellte von e_5

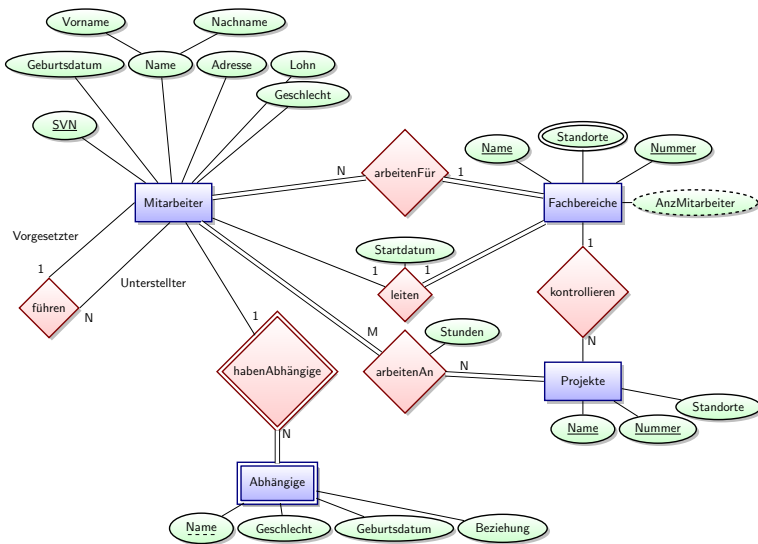
Attribute eines Beziehungstyps

- Ein **Beziehungstyp** kann **Attribute** haben.
 - Beispiel: **Wochenstunden** von **arbeitenAn**
 - der Wert von Wochenstunden gibt für jede Beziehung an, für viele Wochenstunden der Mitarbeiter auf das Projekt angestellt ist
 - der Wert von Wochenstunden hängt von den involvierten Entitäten (**Mitarbeiter**, **Projekt**) ab
- Attribut des Beziehungstypen **zum Entitätstypen verschieben?**
 - in 1:1 Beziehungen können die Attribute zu einem der beiden Entitätstypen verschoben werden
 - in 1:N Beziehungen können die Attribute zum Entitätstyp auf der N-Seite verschoben werden
 - in M:N Beziehungen können die Attribute nicht verschoben werden
- **Beachte:** Attribut beim Entitätstyp muss auch für Entitäten, die nicht in Beziehung stehen, einen Wert haben.

Existenzabhängige Entitätstypen

- **Existenzabhängiger Entitätstyp** (weak entity type):
 - hat keinen (vollständigen) Schlüssel
 - muss **übergeordneten Entitätstypen** haben
 - geht identifizierende Beziehung mit übergeordnetem Entitätstyp ein
- **Schlüssel:** Existenzabhängige Entitäten werden wie folgt identifiziert:
 - den partiellen Schlüssel des existenzabhängigen Entitätstypen und
 - die übergeordnete Entität
- **Beispiel:**
 - eine Entität von **Abhängige** wird identifiziert durch **Name** der abhängigen Person *und* **SVN** des dazugehörigen Mitarbeiters
 - **Name** von **Abhängige** ist ein *partieller Schlüssel* (im ER-Schema unterstrichen mit gepunkteter Linie)
 - **Abhängige** ist ein *existenzabhängiger Entitätstyp*
 - **Mitarbeiter** ist der *übergeordnete Entitätstyp*

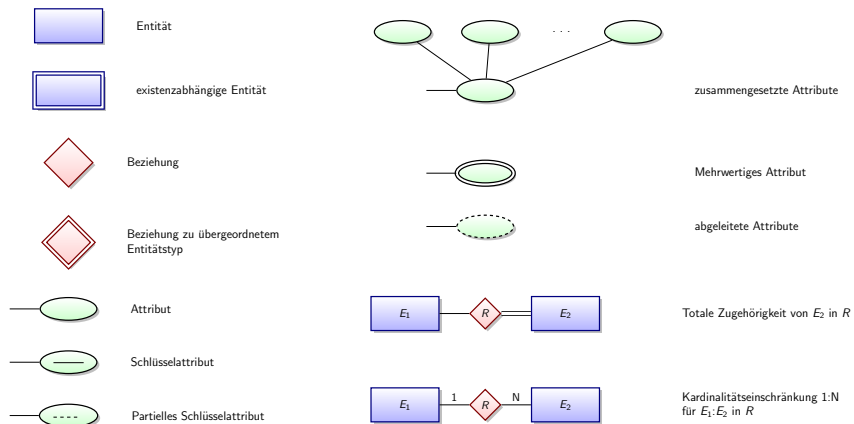
Das NAWI ER-Diagramm



Integrierte Übung 2.3

Ein Flugreservierungssystem erfasst Daten von Passagieren, die über ihre Ticket Nummer identifiziert werden. Zu jedem Flug werden das Abfluggate und die Sitzplatzreservierungen erfasst. Erstellen Sie ein ER-Schema.

Zusammenfassung der ER-Notation



Inhalt

- 1 Datenbankentwurf und ER-Modell
- 2 Entitäten und Attribute
- 3 Beziehungen
 - Was sind Beziehungen?
 - Funktionalitäten
 - Rollen und Attribute
- 4 Generalisierung

Unter- und Obertypen/1

- Entitätstypen können **sinnvolle Untergruppen** haben.
- Beispiel: **Mitarbeiter** können weiter unterteilt werden in
 - **Administrative, Techniker, Ingenieure, ...**
 - abhängig von der Aufgabe der Mitarbeiter
 - **Leiter**
 - **Mitarbeiter** die Leiter sind
 - **GehaltsAng, StundenAng**
 - abhängig von der Vertragsart
- Jede Unterteilung ist eine **Teilmenge** von **Mitarbeiter**
- Erweiterte ER-Diagramme stellen diese Unterteilungen anhand von **Untertypen** und **Obertypen** dar.
 - jede Unterteilung ist ein Untertyp von **Mitarbeiter**
 - **Mitarbeiter** ist der Obertyp über allen Untertypen

Unter- und Obertypen/2

- Verhältnis zwischen Mitarbeiter und Teilmengen wird **Obertyp/Untertyp Beziehung** genannt:
 - **Mitarbeiter/Administrative**
 - **Mitarbeiter/Techniker**
 - **Mitarbeiter/Leiter**
 - ...
- Obertyp/Untertyp Beziehungen auch als **IS-A Beziehung** bezeichnet:
 - **Administrative IS-A Mitarbeiter**
 - **Techniker IS-A Mitarbeiter**
 - ...
- **Notation:** Dreieck mit Bezeichnung "ISA"

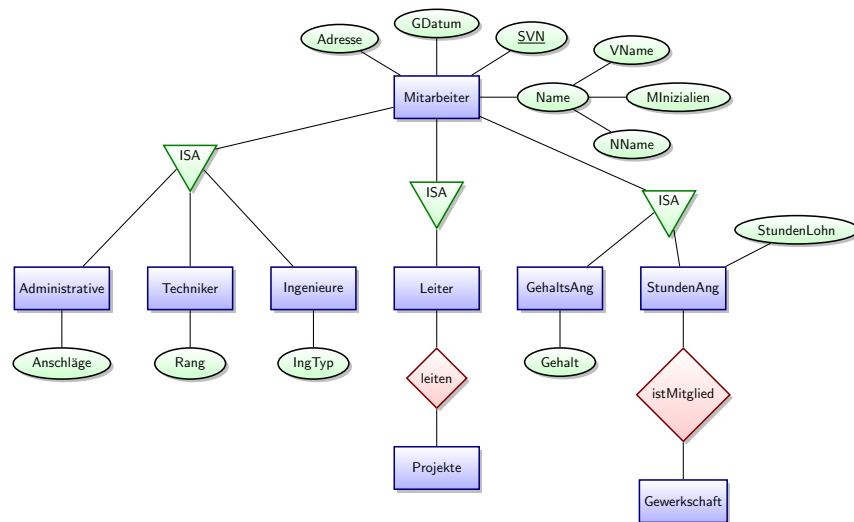
Unter- und Obertypen/3

- Die **Entität eines Untertypen ist dasselbe Objekt** in der realen Welt wie das des Obertypen:
 - die Untertypen-Entität ist dasselbe Objekt in einer *speziellen Rolle*
 - eine Entität kann nicht als reiner Untertyp existieren, es ist immer auch eine Entität des Obertypen
- Eine Entität kann in **mehreren Untertypen** vorkommen.
- Beispiel: Ein Mitarbeiter der Techniker ist und mit Gehalt angestellt ist, gehört zu den Untertypen: **Techniker** und **GehaltsAng**

Vererbung

- Die Entitäten der Untertypen **erben** vom Obertypen
 - alle Attribute
 - alle Beziehungstypen
- **Beispiel:**
 - **Administrative** (sowie **Techniker** und **Ingenieure**) erben die Attribute Name, SVN, ..., von **Mitarbeiter**
 - Jeder **Administrative** hat Werte für die geerbten Attribute.

Beispiel: Unter- und Obertypen



Spezialisierung und Generalisierung

- **Spezialisierung:**
 - aus einem Obertyp eine Menge von Untertypen generieren
 - die Untertypen basieren auf unterschiedlichen Charakteristiken
 - Bsp: **Administrative**, **Ingenieur**, **Techniker** ist eine Spezialisierung von **Mitarbeiter** aufgrund der verrichteten Arbeit
 - mehrere Spezialisierungen eines Obertyps sind möglich
- **Generalisierung:**
 - umgekehrter Prozess zu Spezialisierung
 - verschiedene Typen mit gemeinsamen Eigenschaften werden zu Obertyp zusammengefasst
 - Bsp: **PKW** und **Lastwagen** werde zu **Fahrzeug** generalisiert
 - **PKW** und **Lastwagen** werden Untertypen des Obertyps **Fahrzeug**
 - **Fahrzeug** ist Generalisierung von **PKW** und **Lastwagen**, **PKW** und **Lastwagen** sind Spezialisierung von **Fahrzeug**

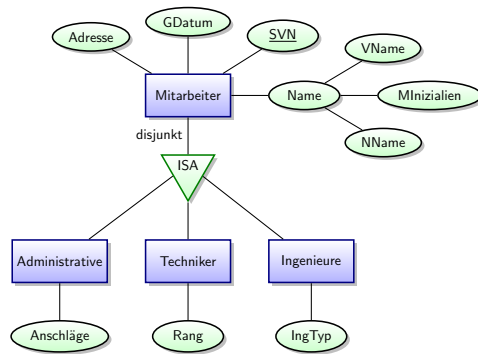
Einschränkungen der Spezialisierung/1

- **Zwei Arten** von Einschränkungen auf Spezialisierung:
 - disjunkte Spezialisierung
 - vollständige Spezialisierung
- **Disjunkte Spezialisierung:**
 - Entität kann zu *höchstens einem* Untertyp gehören
 - Notation: Verbindungslinie zu Obertyp mit "disjunkt" bezeichnen
 - nicht-disjunkte Spezialisierung wird *überlappend* genannt (keine Beschriftung auf Verbindungslinie)
- **Vollständige Spezialisierung:**
 - Entität muss zu *mindestens einem* Untertyp gehören, d.h. *keine* Entität kann *nur* zu Obertyp gehören
 - Notation: Doppelte Linie als Verbindung zu Obertyp
 - nicht-vollständige Spezialisierung wird *partiell* genannt (einfache Verbindungslinie)

Einschränkungen der Spezialisierung/2

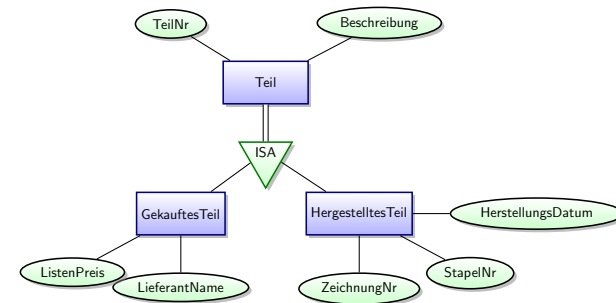
- **Alle Kombinationen** erlaubt:
 - überlappend/partiell (Standard): Entität *kann* zu einem oder mehreren Untertypen gehören
 - disjunkt/partiell: Entität *kann* zu maximal einem Untertypen gehören
 - überlappend/vollständig: Entität *muss* zu einem oder mehreren Untertypen gehören
 - disjunkt/vollständig: Entität *muss* zu genau einem Untertypen gehören

Beispiel: Disjunkte partielle Spezialisierung



- Ein Mitarbeiter muss weder Administrativer, Techniker, noch Ingenieur sein (partielle Spezialisierung).
- Ein Mitarbeiter kann nicht Administrativer und Techniker, Administrativer und Ingenieur, oder Techniker und Ingenieur zugleich sein (disjunkte Spezialisierung).

Beispiel: Überlappende vollständige Spezialisierung



- Es gibt kein Teil, das weder hergestellt noch gekauft ist (vollständige Spezialisierung).
- Ein Teil kann zugleich hergestellt und gekauft sein (überlappende Spezialisierung).

Zusammenfassung

- Das ER-Modell hat **drei Hauptkonstrukte**:
 - **Entitäten** (mit Entitätstypen und Entitätsmengen)
 - **Attribute** (einfach, zusammengesetzt, mehrwertig, usw.)
 - **Beziehungen** (mit Beziehungstypen und Beziehungsmengen)
- **Funktionalitäten** schränken die möglichen Beziehungen ein:
 - Kardinalitätseinschränkung: 1:1, 1:N, M:N
 - Teilnahmebeschränkung: zwingende oder optionale Teilnahme
- **Generalisierung und Spezialisierung**:
 - Obertypen erlauben Zusammenfassung ähnlicher Entitäten
 - jede Entität eines Untertyps ist auch im Obertyp
 - Einschränkung der Spezialisierung: disjunkt, vollständig