

# Erweiterte Entity-Relationship- und UML-Modellierung

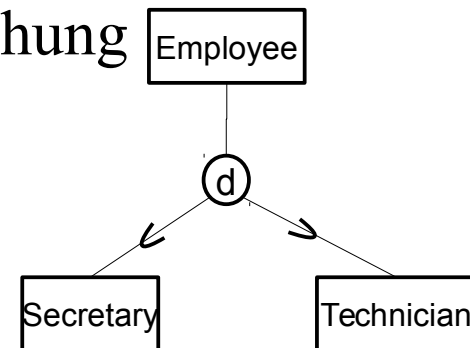


# Erweitertes-ER (EER) Modellkonzept

- Beinhaltet alle Aspekte des Basis-ER-Modellkonzeptes
- Weitere Konzepte: Unterklassen/Oberklassen, Spezialisierung/Generalisierung, Kategorien, Attributvererbung
- Das resultierende Modell wird als Enhanced-ER oder Extended ER (E2R or EER) Modell bezeichnet
- Es wird verwendet, wenn es nötig ist, Anwendungen vollständiger und genauer zu modellieren
- Es beinhaltet objektorientierte Konzepte, wie z.B. Vererbung

# Unter- und Oberklassen(1)

- Ein Entitäts-Typ hat vielleicht zusätzliche sinnvolle Untergruppierungen seiner Entitäten.
- Beispiel: EMPLOYEE kann vielleicht eingeteilt werden in SECRETARY, ENGINEER, MANAGER, TECHNICIAN, SALARIED\_EMPLOYEE, HOURLY\_EMPLOYEE, ...
  - Jede dieser Gruppierungen ist eine Untermenge der EMPLOYEE Entität
  - Jede wird als Unterklasse von EMPLOYEE bezeichnet
  - EMPLOYEE ist eine Oberklasse für jede dieser Unterklassen
- Dies wird als Oberklassen-/Unterklassenbeziehung bezeichnet.
- Beispiel: EMPLOYEE/SECRETARY, EMPLOYEE/TECHNICIAN

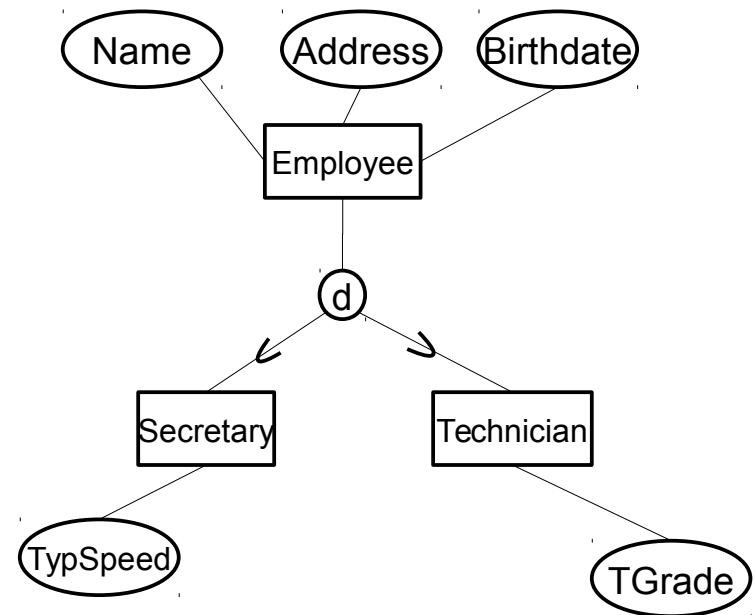


# Unter- und Oberklassen(2)

- Dies Beziehung wird auch als IS-A Beziehung bezeichnet (SECRETARY IS-A EMPLOYEE, TECHNICIAN IS-A EMPLOYEE, ...).
- Merke: Eine Entität die Mitglied einer Unterklasse ist, steht für die selbe Entität der realen Welt wie die Mitglieder der Oberklasse
  - Das Mitglied der Unterklasse ist die selbe Entität in einer unterschiedlichen Rolle
  - Eine Entität kann in der Datenbank nicht lediglich als ein Mitglied der Unterklasse existieren; sie muss auch ein Mitglied der Oberklasse sein
  - Ein Mitglied der Oberklasse kann wahlweise für ein Mitglied seiner Unterklassen eingesetzt werden
- Beispiel: Ein Angestellter, der auch ein Ingenieur ist, gehört gleichzeitig zu den zwei Unterklassen ENGINEER und SALARIED\_EMPLOYEE
  - Es ist nicht nötig, dass jede Entität einer Oberklasse Mitglied in einer ihrer Unterklassen ist.

# Attributvererbung in Oberklassen / Unterklassen Beziehungen

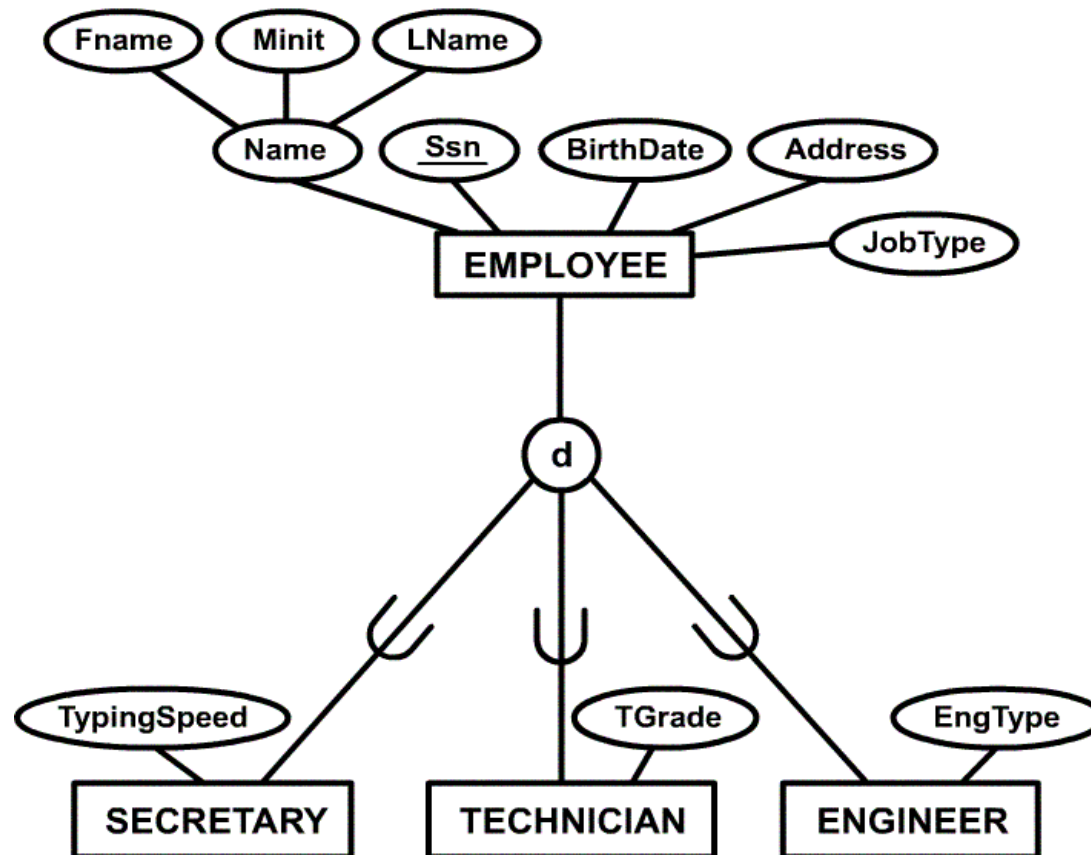
- Eine Entität, die Mitglied einer Unterklasse ist, erbt alle Attribute der Oberklasse
- Sie erbt weiterhin alle Beziehungen



# Spezialisierung

- Ist der Prozess, eine Menge von Unterklassen einer Oberklasse zu definieren
- Die Menge von Unterklassen basiert auf Unterscheidungsmerkmalen der Entitäten in der Oberklasse
- Beispiel: {SECRETARY, ENGINEER, TECHNICIAN} ist eine Spezialisierung von EMPLOYEE basierend auf *JobType*.
  - Vielleicht mehrere Spezialisierungen der selben Superklasse
- Beispiel: Eine andere Spezialisierung von EMPLOYEE basierend auf der Bezahlungsmethode {SALARIED\_EMPLOYEE, HOURLY\_EMPLOYEE}.
  - Oberklasse/Unterklassen-Beziehung und Spezialisierung kann grafisch durch ein EER-Diagramm dargestellt werden
  - Attribute von Unterklassen werden spezifische Attribute genannt. Zum Beispiel TypingSpeed von SECRETARY
  - Die Unterklasse kann an bestimmten Beziehungstypen beteiligt sein, z.B.: BELONGS\_TO von HOURLY\_EMPLOYEE

# Beispiel einer Spezialisierung



# Generalisierung

- Das Entgegengesetzte zum Spezialisierungsprozess
- Mehrere Klassen mit gemeinsamen Eigenschaften werden zu einer gemeinsamen Oberklasse generalisiert; Originalklasse wird zu einer Unterklasse
- Beispiel: CAR, TRUCK generalisiert in VEHICLE; CAR und TRUCK werden zu Unterklassen von VEHICLE.
  - Wir sehen: {CAR, TRUCK} ist eine Spezialisierung von VEHICLE
  - Alternativ können wir sehen, dass VEHICLE eine Generalisierung ist von CAR und TRUCK



# Generalisierung und Spezialisierung

- Schematische Notation wird manchmal verwendet, um zwischen Generalisierung und Spezialisierung zu unterscheiden
  - Pfeilspitze zur generalisierten Oberklasse repräsentiert eine Generalisierung
  - Pfeilspitze zur spezialisierten Unterklasse repräsentiert eine Spezialisierung
  - Wir benutzen diese Notation nicht, da es oft subjektiv ist welches Verfahren für eine bestimmte Situation angewendet wird
  - Wir empfehlen, in diesen Situationen keine Pfeile zu zeichnen
- Datenmodellierung mit Spezialisierung und Generalisierung
  - Eine Ober- oder Unterklasse repräsentiert eine Menge von Entitäten
  - Dargestellt durch Rechtecke in EER Diagrammen (wie Entitäts-Typen)
- Manchmal werden alle Entitätsmengen als Klassen bezeichnet, egal ob sie Entitäts-Typen, Ober- oder Unterklassen sind.

# Arten von Generalisierungen und Spezialisierungen (2)

- **Prädikats- oder Bedingungs-definiert:** wir können die exakten Entitäten einer Unterklasse anhand einer Bedingung bestimmen
  - Voraussetzung ist eine Einschränkung, die die Mitglieder der Unterklasse bestimmt
  - Anzeigen einer Prädikats-definierten Unterklasse durch das Notieren der Bedingung an der Linie, welche von der Unterklasse zur Oberklasse führt
- **Attribut-definiert:** alle Unterklassen in einer Spezialisierung haben die selbe Mitgliedsbedingung auf dem selben Oberklassenattribut
  - Attribut wird als definierendes Attribut für Spezialisierung bezeichnet
  - Beispiel: JobType ist das definierende Attribut der Spezialisierung {SECRETARY, TECHNICIAN, ENGINEER} of EMPLOYEE
- **Benutzerdefiniert:** Wenn keine Bedingung die Mitgliedschaft bestimmt
  - Die Mitgliedschaft in einer Unterklasse wird von den Datenbank-Benutzern bestimmt durch das Anwenden einer Operation, um eine Entität zur Unterklasse hinzuzufügen
  - Die Mitgliedschaft in einer Unterklasse ist individuell spezifiziert für jede Entität in der Oberklasse

# Einschränkungen auf Spezialisierungen und Generalisierungen (2)

## ● **Disjunktheit:**

- Die Unterklassen der Spezialisierung müssen disjunkt sein (Eine Entität kann nur Mitglied in einer Subklasse der Spezialisierung sein).  
Spezifiziert durch **d** im EER Diagramm
- Wenn nicht disjunkt, überlappend; die selbe Entität kann Mitglied in mehreren Unterklassen einer Spezialisierung sein.  
Spezifiziert durch **o** im EER Diagramm

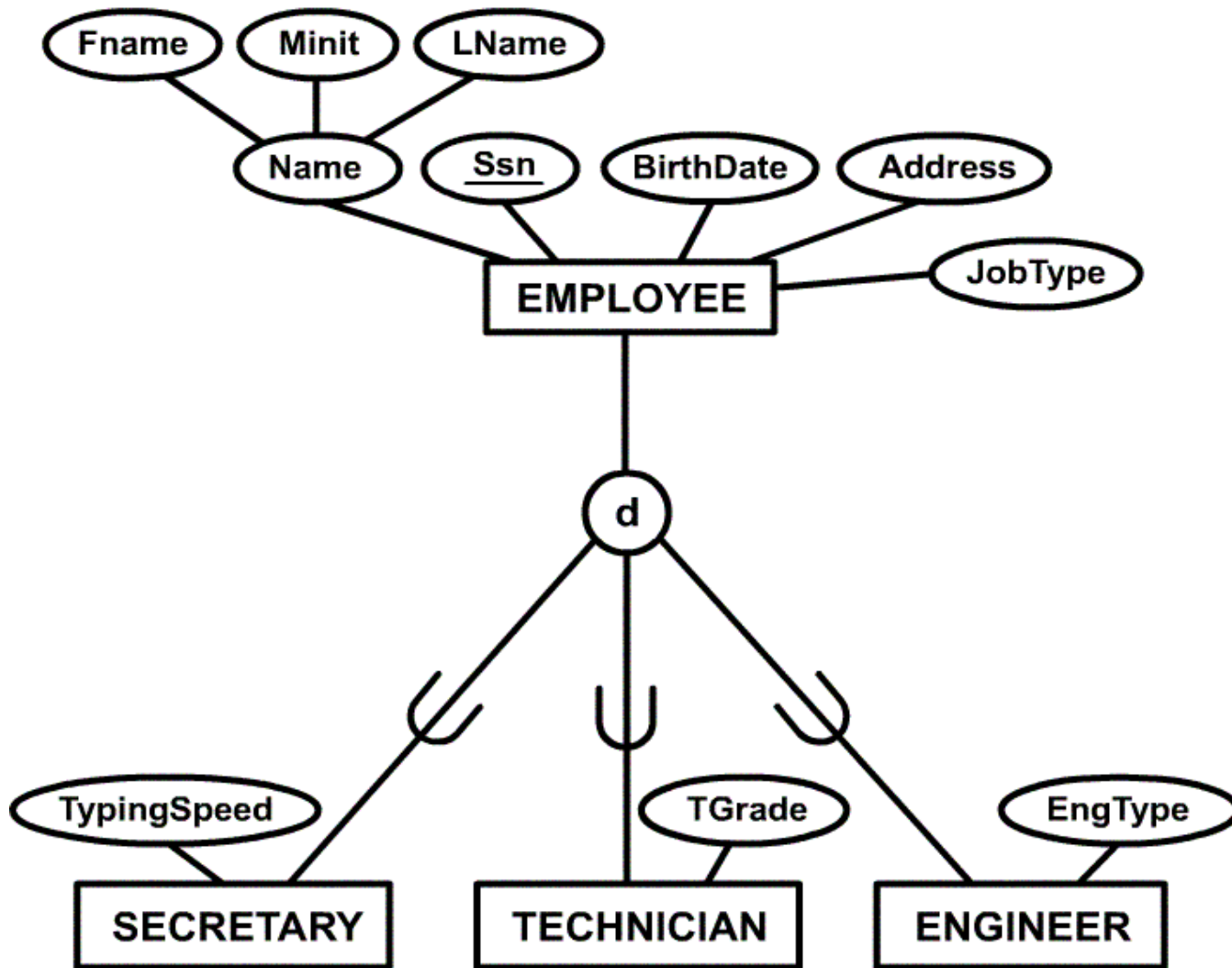
## ● **Vollständigkeit:**

- *Total* : jede Entität in einer Oberklasse muss ein Mitglied einer Unterklasse in der Spezialisierung/Generalisierung sein  
Dargestellt im EER Diagramm durch eine **doppelte Linie**
- *Partiell* : Eine Entität muss keiner Unterklasse angehören.  
Dargestellt im EER Diagramm durch eine **einzelne Linie**

# Einschränkungen auf Spezialisierungen und Generalisierungen (3)

- Vier Arten der Spezialisierung / Generalisierung:
  - Disjunkt, total
  - Disjunkt, partiell
  - Überlappend, total
  - Überlappend, partiell
- Generalisierung ist in der Regel total, weil die Oberklasse von der Unterklassen abgeleitet ist

# Beispiel einer disjunkten partiellen Spezialisierung



# Spezialisierung / Generalisierung: Hierarchien, DAGs und Geteilte Unterklassen (1)

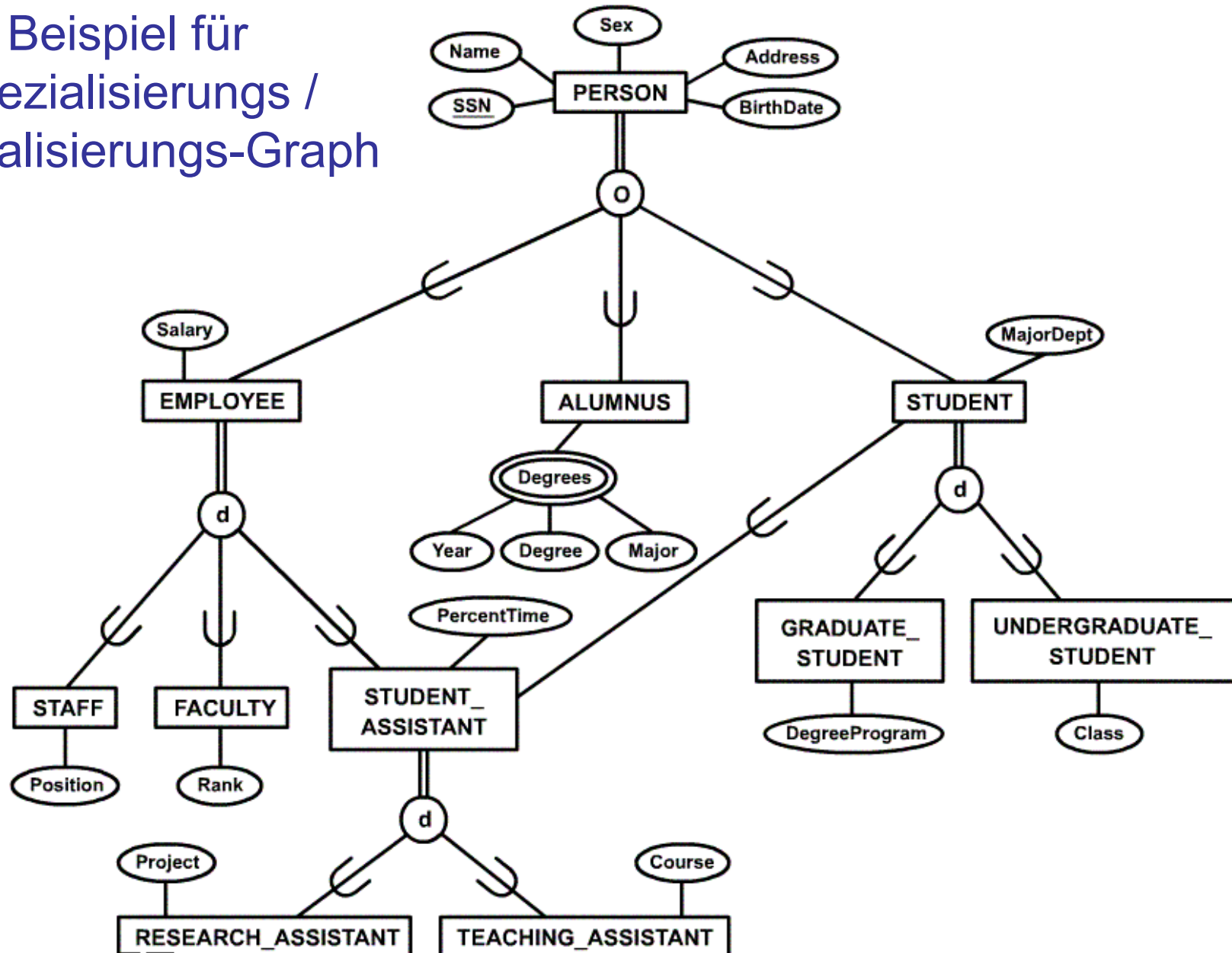
- Eine Unterklasse kann auf sich selbst weitere Unterklassen spezifizieren
- Formt eine Hierarchie oder einen gerichteten azyklischen Graph (directed acyclic graph, DAG)
- Hierarchie ist eine Einschränkung, dass jede Unterklasse nur eine Oberklasse hat (so genannte einfache Vererbung )
- In einem DAG kann eine Klasse Unterklasse von mehr als einer Oberklasse sein (genannt Mehrfachvererbung)
- In einem DAG oder einer Hierarchie erbt eine Unterklasse nicht nur die Attribute ihrer direkten Oberklasse, sondern auch alle der Vorgänger ihrer Oberklassen

# Spezialisierung / Generalisierung: Hierarchien, DAGs und geteilte Unterklassen (2)

- Eine Unterklasse mit mehr als einer Oberklasse wird geteilte Unterklasse genannt
- Kann spezialisierte Hierarchien oder DAGs oder generalisierte Hierarchien oder DAGs haben
- Spezialisierung startet mit einem Entity-Typen und definiert dann die Unterklassen vom Entity-Typen durch sukzessive Spezialisierung (top-down, konzeptioneller Verfeinerungsprozess)
- Generalisierung startet mit vielen Entitytypen und generalisiert diese mit allgemeinen Eigenschaften (bottom-up, konzeptioneller Syntheseprozess)
- In der Praxis wird eine Kombination von beiden Prozessen angewendet



# Beispiel für Spezialisierungs / Generalisierungs-Graph





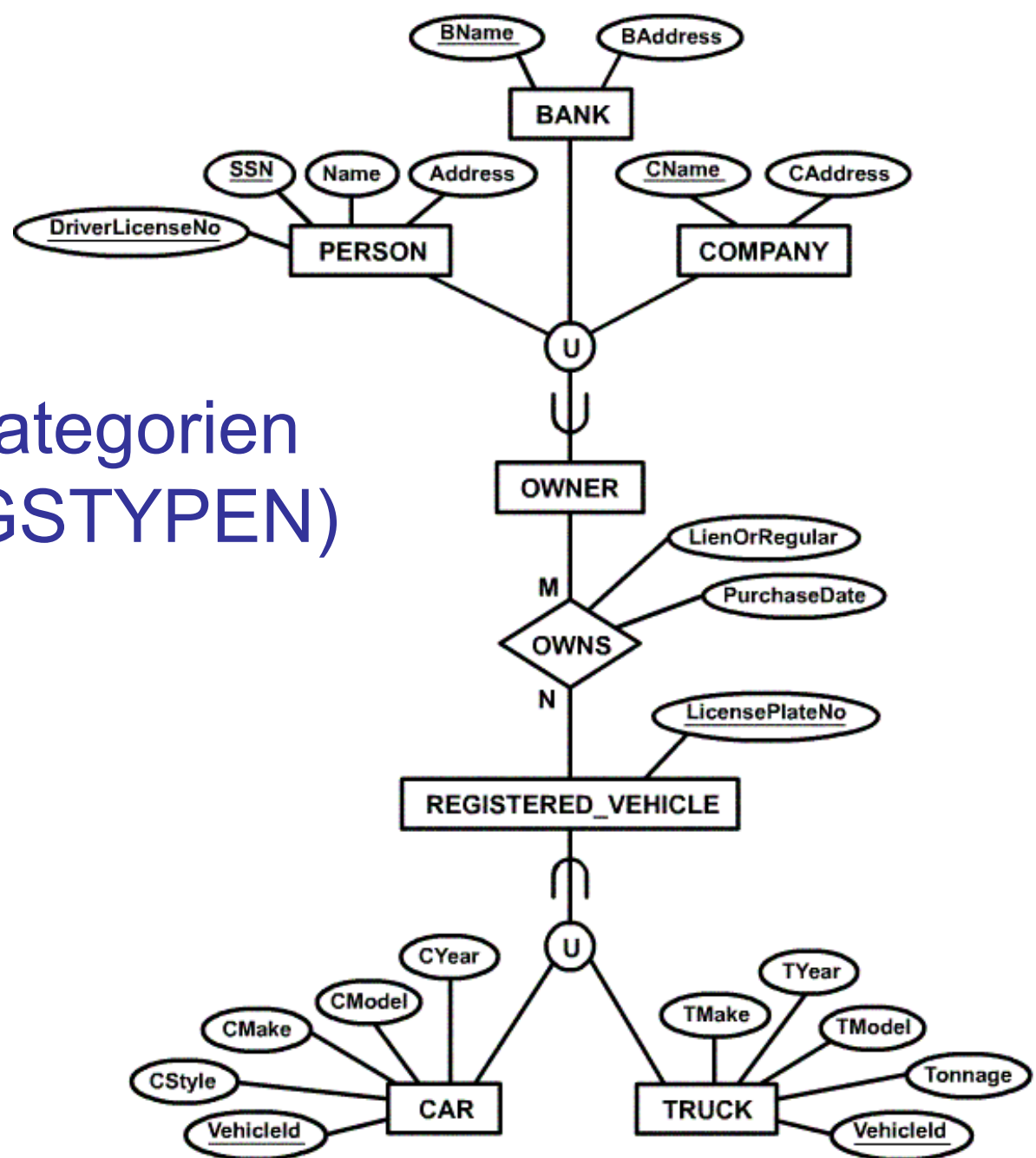
# Kategorien (VEREINIGUNGSTYPEN) (1)

- Bisher: Ober/Unterklassen-Beziehungen haben eine einzige Oberklasse
  - **geteilte Unterklasse** ist eine Unterklasse mit mehr als einer eindeutigen Oberklassen/Unterklassenbeziehung, wo jede Beziehung eine einzige Oberklasse hat (**multiple Vererbung**)
- In machen Fällen ist es nötig, eine **einzige Oberklassen/Unterklassenbeziehung** zu modellieren, mit mehr als einer Oberklasse
  - Oberklassen repräsentieren verschiedene Entity-Typen
  - Solche Unterklassen werden Kategorie oder **Vereinigungstypen** bezeichnet
  - Spezifiziert durch **U** im EER-Diagramm

# Kategorien (VEREINIGUNGSTYPEN) (2)

- Beispiel: Datenbank für Kfz-Zulassungsstellen, Fahrzeugeigentümer kann eine Person, eine Bank oder eine Firma sein.
  - Kategorie (Unterklasse) OWNER ist eine Untermenge der Vereinigung von den drei Oberklassen COMPANY, BANK, and PERSON
  - Ein Kategoriemitglied muss in **mindestens einer** seiner **Oberklassen** existieren
- Hinweis: Im Gegensatz dazu ist eine geteilte Unterklassen eine Teilmenge des Durchschnitts seiner Oberklassen (geteilte Unterklassenmitglieder müssen in **allen** ihren **Oberklassen** existieren).

# Beispiele für Kategorien (VEREINIGUNGSTYPEN)



# Formelle Definition des EER-Modells (1)

- **Klasse C**: Eine Menge von Entitäten; kann ein Entity-Typ sein, Unterklasse, Oberklasse, Kategorie.
- **Unterklasse**: Eine Klasse dessen Entitäten immer eine Untermenge von Entitäten einer anderen Klasse  $C$  sein müssen, bezeichnet die Unterklasse  $S$  der Oberklassen/Unterklassen- (oder IS-A) Beziehung  $C/S$   
 $S \subseteq C$
- **Spezialisierung Z**:  $Z = \{S_1, S_2, \dots, S_n\}$  eine Menge von Unterklassen mit der selben Oberklasse  $G$ ; daher,  $G/S_i$  eine Oberklassenbeziehung für  $i = 1, \dots, n$ .
  - $G$  wird als eine Generalisierung einer Unterklasse bezeichnet  $\{S_1, S_2, \dots, S_n\}$
  - $Z$  ist **total** wenn gilt:  $S_1 \cup S_2 \cup \dots \cup S_n = G$ ;  
anderenfalls ist  $Z$  **partiell**.
  - $Z$  ist **disjunkt** wenn immer gilt:  $S_i \cap S_j = \emptyset$  für  $i \neq j$ ;  
anderenfalls ist  $Z$  **überlappend**.

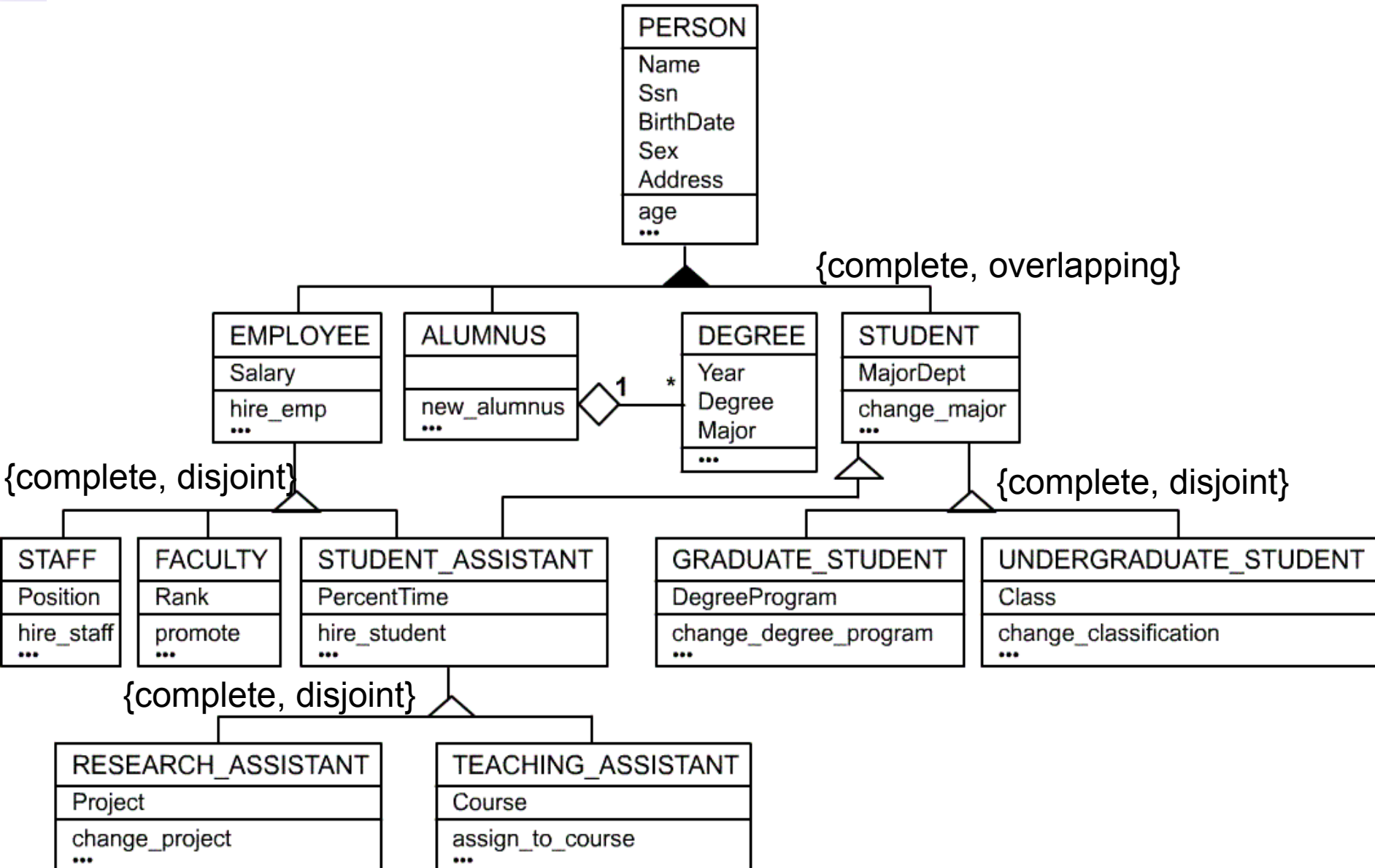
# Formale Definition des EER-Modells (2)

- Unterklasse  $S$  von  $C$  ist **Prädikats-definiert**, wenn Prädikat  $p$  auf den Attributen von  $C$  benutzt wird, um die Mitgliedschaft in  $S$  zu spezifizieren; d.h.,  $S = C[p]$ , wobei  $C[p]$  die Menge der Entitäten in  $C$  ist, die  $p$  erfüllen
- **Attribut-definierte** Spezialisierung: wenn ein Prädikat  $A = c_i$  (wobei  $A$  ein Attribut von  $G$  ist und  $c_i$  ein konstanter Wert der Domäne von  $A$ ) benutzt wird um die Mitgliedschaft in jeder Unterklasse  $S_i$  in  $Z$  zu spezifizieren
- Eine Unterklasse, die nicht durch ein Prädikat definiert ist, bezeichnet man als **Benutzer-definiert**
- Hinweis: Wenn  $c_i \neq c_j$  für  $i \neq j$ , und  $A$  ist ein atomares Attribut, dann ist die Attribut-spezifizierte Spezialisierung disjunkt .

# Formale Definition des EER-Modells (3)

- Kategorie oder Vereinigungstyp  $T$ 
  - Eine Klasse, die eine Untermenge der Vereinigung von  $n$  definierten Oberklassen ist  
 $D_1, D_2, \dots, D_n, n > 1$ :  
$$T \subseteq (D_1 \cup D_2 \cup \dots \cup D_n)$$
  - Sei  $p_i$  ein Prädikat auf den Attributen von  $T$ , das die Entitäten von  $D_j$ , welche Mitglieder von  $T$  sind, spezifizieren kann
  - Dann ist  $T = (D_1[p_1] \cup D_2[p_2] \cup \dots \cup D_n[p_n])$
  - Hinweis: Die Definition des Beziehungstypen sollte 'Entity-Typen' ersetzen durch 'Klasse'.

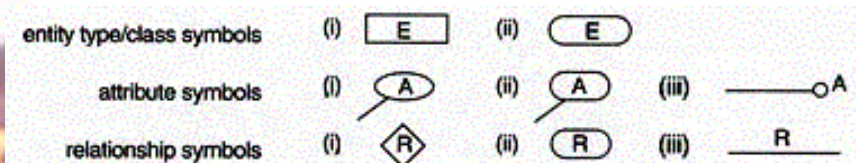
# UML-Beispiel mit Spezialisierung / Generalisierung



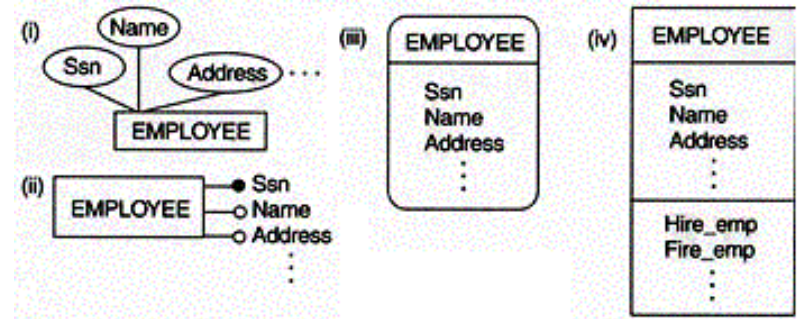


# Alternative Darstellungsnotationen

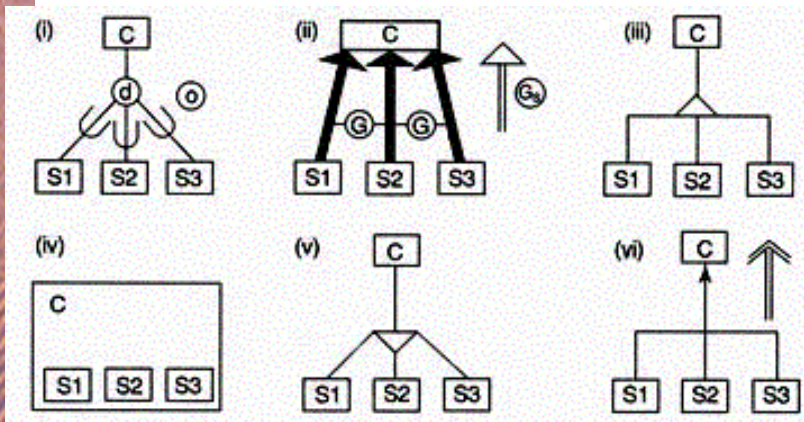
Symbols for entity type / class, attribute and relationship



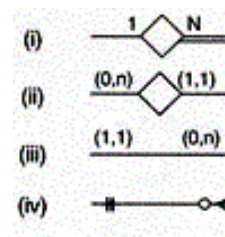
Displaying attributes



Notations for displaying specialization / generalization



Various (min, max) notations



Displaying cardinality ratios

