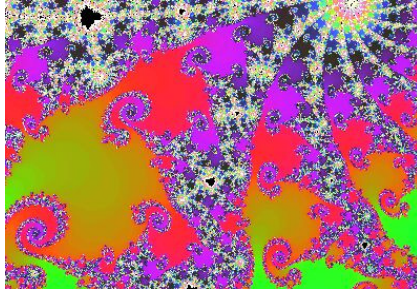


Datenbanken

5. Erweitertes ER-Modell



Fachhochschule Darmstadt
Fachbereich Informatik

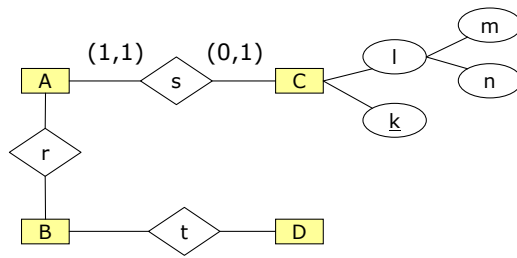
Dr. Peter Nevermann

Rückblick

■ ER-Modell

- ◆ *Entity* (Entität, Datenspeicher)
- ◆ *Relationship* (Beziehung)
- ◆ *Attribute* (Eigenschaft, Attribut [einfach, komplex])

■ ER-Diagramm



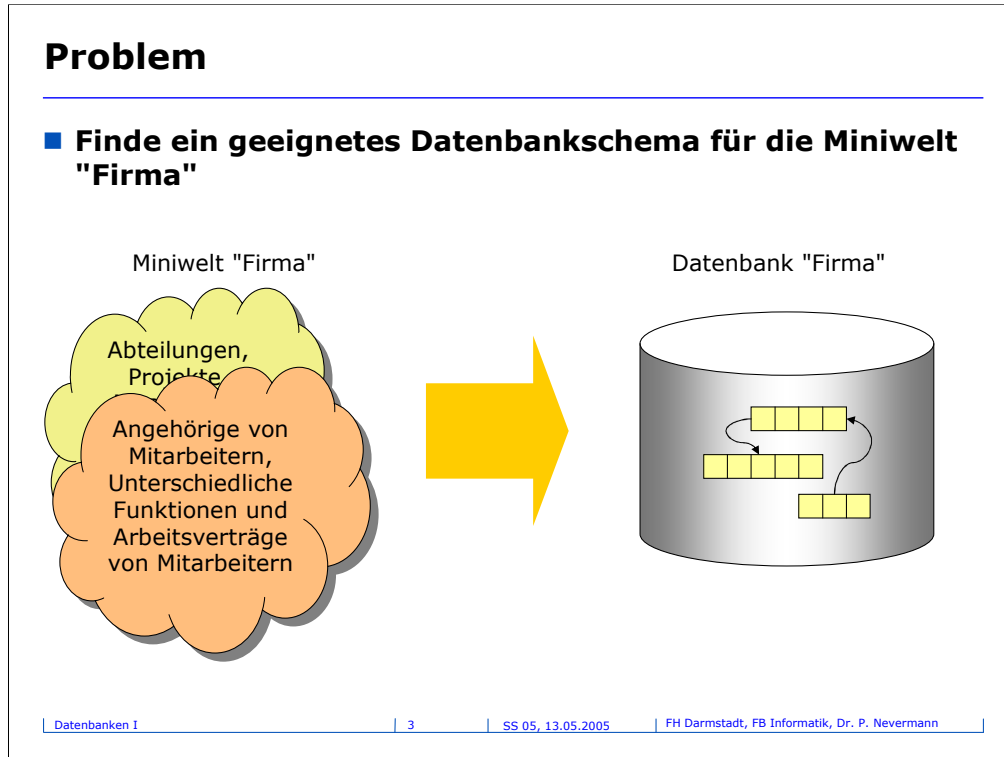
Kardinalitäten
von Beziehungen:

$(0,1)$

$(1,1)$

$(0,n)$

$(1,n)$



Die Anforderungen haben sich geändert. Die Beschreibung der Miniwelt "Firma" muß noch um folgende zwei Punkte erweitert werden:

Mitarbeiter

- Zu jedem Mitarbeiter werden auch Informationen zu den Angehörigen (Partner, Kinder) festgehalten
- Zu einem Angehörigen sind folgende Daten von Interesse: Vorname, Geschlecht, Geburtsdatum und ob es sich um den Partner oder ein Kind handelt

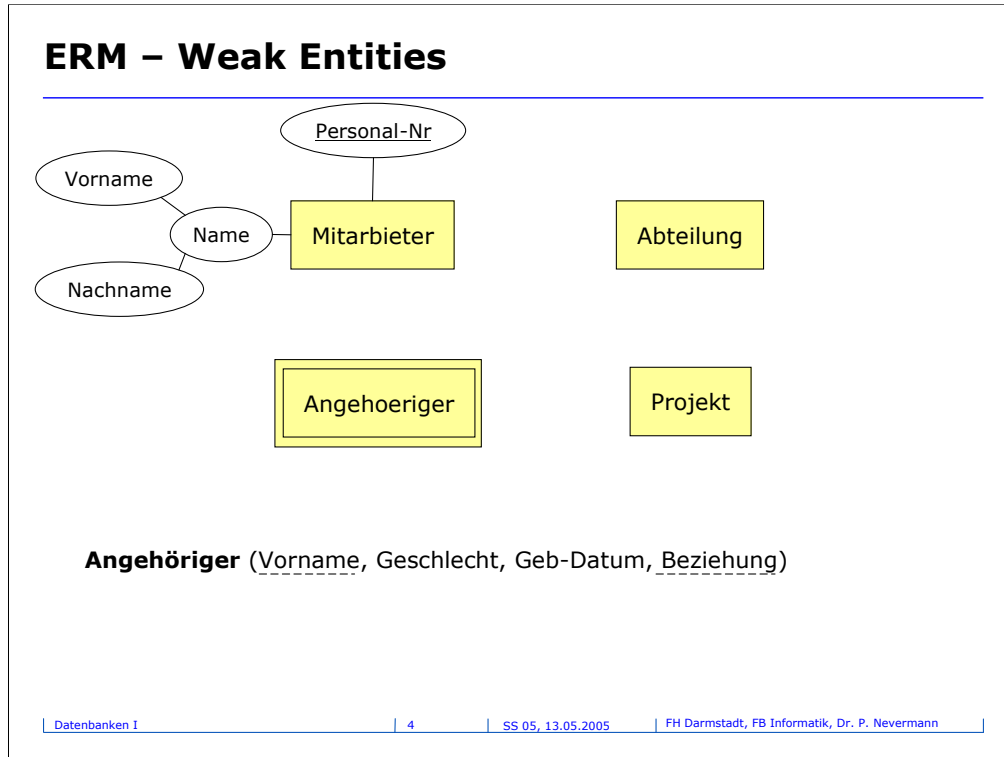
Funktionen und Arbeitsvertrag

- Es soll noch in der Datenbank verwaltet werden, daß Mitarbeiter unterschiedliche Funktionen ausüben (Manager, Entwickler, Techniker, ...)
- Es gibt in der Firma noch unterschiedliche Arbeitsverträge (Teilzeit, freie Mitarbeiter, ...)

Erweitertes ER-Modell

Das ursprüngliche ER Modell wurde nachträglich durch Modellelemente erweitert, welche z.B. aus der Objektorientierung entnommen wurden. Dadurch erhöht sich die Fähigkeit des ER Modells, die Semantik der Miniwelt präzise zu beschreiben. Zwei dieser Erweiterungen sollen in dieser Vorlesungsstunde vorgestellt werden:

- Weak Entities
- Spezialisierung/Generalisierung von Entities



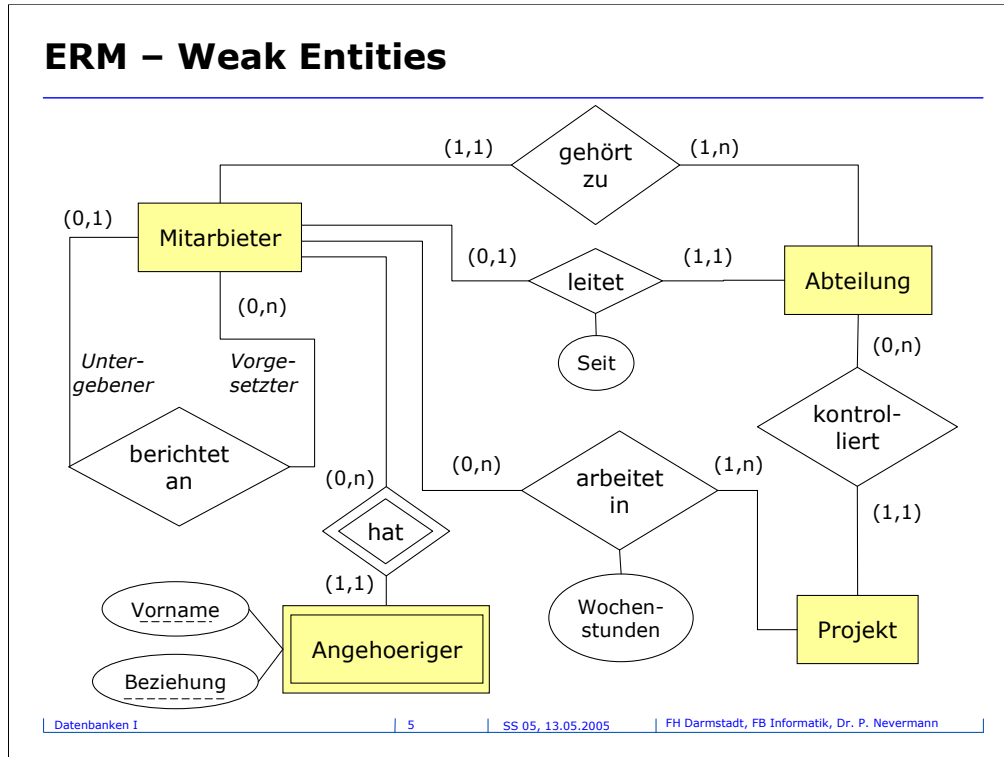
Weak Entity

Entities ohne Schlüssel nennt man Schwache Entitäten (*weak entity*). Im Sinne der Miniwelt, hat eine Instanz des Weak-Entities keine eigene Identität, sondern nur im Kontext einer weiteren Instanz eines gewöhnlichen Entities (*strong entity*, starke Entität).

Im ER Diagramm werden Weak-Entities mit doppelter Umrandung dargestellt.

Beispiel:

In unserem Beispiel ist "Angehöriger" (Partner, Kind) als Weak-Entity modelliert. Natürlich haben das Kind und der Partner eine eigene Identität in der realen Welt ☺ ... aber im Kontext der Miniwelt "Firma" interessieren nur Vorname, Geschlecht, Geburtsdatum und Art der Beziehung (P=Partner oder K=Kind) des Angehörigen. Rein zufällig könnten zwei Angehörige unterschiedlicher Mitarbeiter identisch in diesen Merkmalen sein. Darum hat das Entity "Angehöriger" keinen Schlüssel und ist somit ein Weak-Entity. Im Kontext des Mitarbeiters sollte z.B. Vorname + Beziehung identifizierend sein (Monogamie vorausgesetzt und unter der Annahme, daß es ungewöhnlich wäre zwei Kindern denselben Namen zu geben).



Identitätgebende Beziehung, Owner Entity

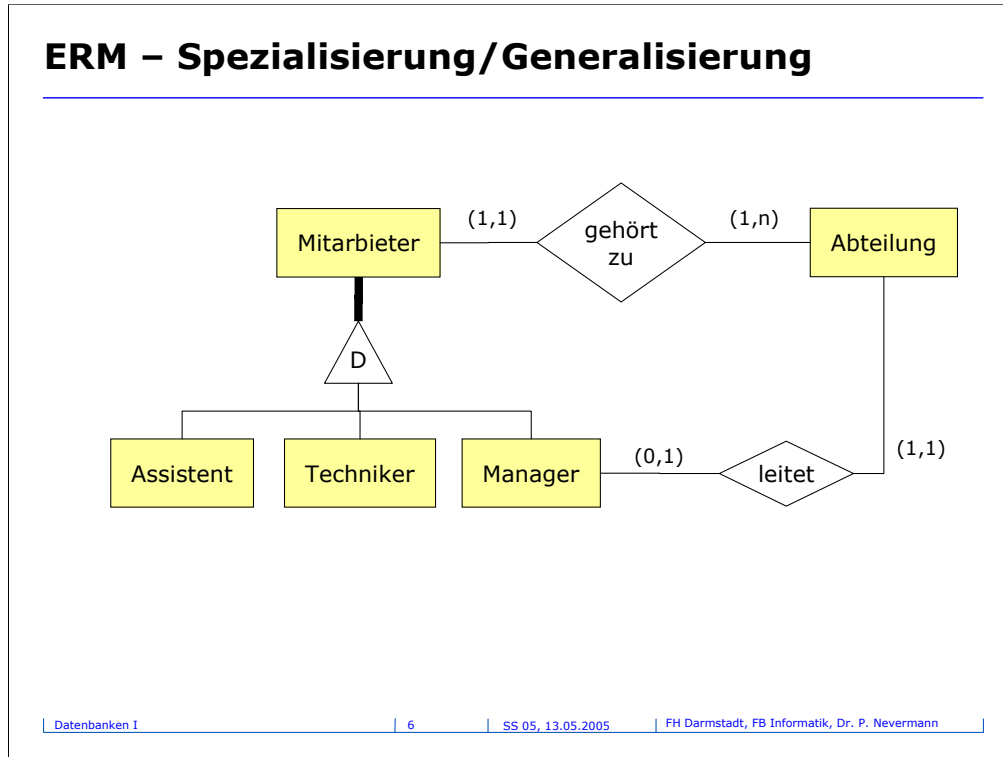
Ein Weak-Entity benötigt also ein anderes Entity mit dem es in Beziehung steht, in dessen Kontext das schwache Entity seine Identität bekommt. Man nennt das identitätgebende Entity auch *Owner Entity* und stellt die *identitätgebende Beziehung* im Diagramm ebenfalls mit doppelter Umrandung dar. Die Kardinalität der identitätgebenden Beziehung ist immer (1,1) auf der Seite des Weak-Entities.

Partieller Schlüssel

Das Weak-Entity besitzt einen partiellen Schlüssel (gestrichelt unterstrichen), der nur zusammen mit einem Schlüssel des Owner-Entities identifizierend ist. Ein Angehöriger kann also mit der Kombination (Personal-Nr, Vorname, Beziehung) eindeutig identifiziert werden.

Weak-Entity Hierarchien

Ein Weak-Entity kann ein anderes Weak-Entity als Owner Entity haben, wenn man über die identitätgebenden Beziehungen schließlich zu einem starken Entity gelangt.



Spezialisierung, Sub- und Supertypen

Wenn die Extension (Entity-Menge, Instanzenmenge) eines Entities sich nach einem bestimmten *Aspekt* aufteilen läßt, derart daß die Teilmengen von Bedeutung für die Miniwelt sind, kann man das Entity *spezialisieren*, indem man *Subtypen* des Entities einführt. Das aufgeteilte Entity ist dann der *Supertyp*. Die Aufteilung des Supertypen in 1..n Subtypen nach einem bestimmten Aspekt nennt man eine *Spezialisierung*.

Beispiel: Mitarbeiter werden aufgrund ihrer Funktion (→ Aspekt) in Assistenten, Techniker und Manager aufgeteilt, indem entsprechend neue Entities als Subtypen des Mitarbeiter Entities definiert werden. Die Aufteilung von Mitarbeiter in Assistent, Techniker und Manager ist eine Spezialisierung.

"is-a" Beziehung, disjoint/overlapping, partiell/total

Die Beziehung zwischen dem Sub- und dem Supertyp wird *Sub-/Supertyp-Beziehung* oder auch *is-a* Beziehung ("ist-ein") genannt, weil eine Instanz des Subtypen auch immer eine Instanz des Supertypen ist (ein Manager "ist-ein" Mitarbeiter). Die Beziehung hat also eine besondere Semantik und wird im Diagramm nicht durch die Raute dargestellt. Stattdessen wird eine Spezialisierung durch ein Dreieck dargestellt, wobei die Spitze des Dreiecks zum Supertypen zeigt. Je nach dem, ob die Subtypen überschneidungsfrei oder überlappend sind, wird das Dreieck mit "D" (*disjoint*) oder mit "O" (*overlapping*) beschriftet.

Während jede Subtyp-Instanz auch eine Instanz der Supertypen ist, gilt nicht unbedingt umgekehrt, daß jede Supertyp-Instanz auch Instanz eines Subtypen sein muß. Ist das jedoch der Fall, dann spricht man von einer *totalen* Spezialisierung (im Diagramm mit einem Doppelstrich [oder einem fetten Strich] zwischen Dreieck und Supertyp dargestellt). Gibt es Instanzen des Supertyps, die in keinem Subtyp der Spezialisierung Instanz sind, dann handelt es sich um eine *partielle* Spezialisierung (im Diagramm mit einem einfachen Strich [oder einem dünnen Strich] zwischen Dreieck und Supertyp dargestellt). Im Diagramm auf der nächsten Folie ist eine überlappende, partielle Spezialisierung dargestellt.

Generalisierung

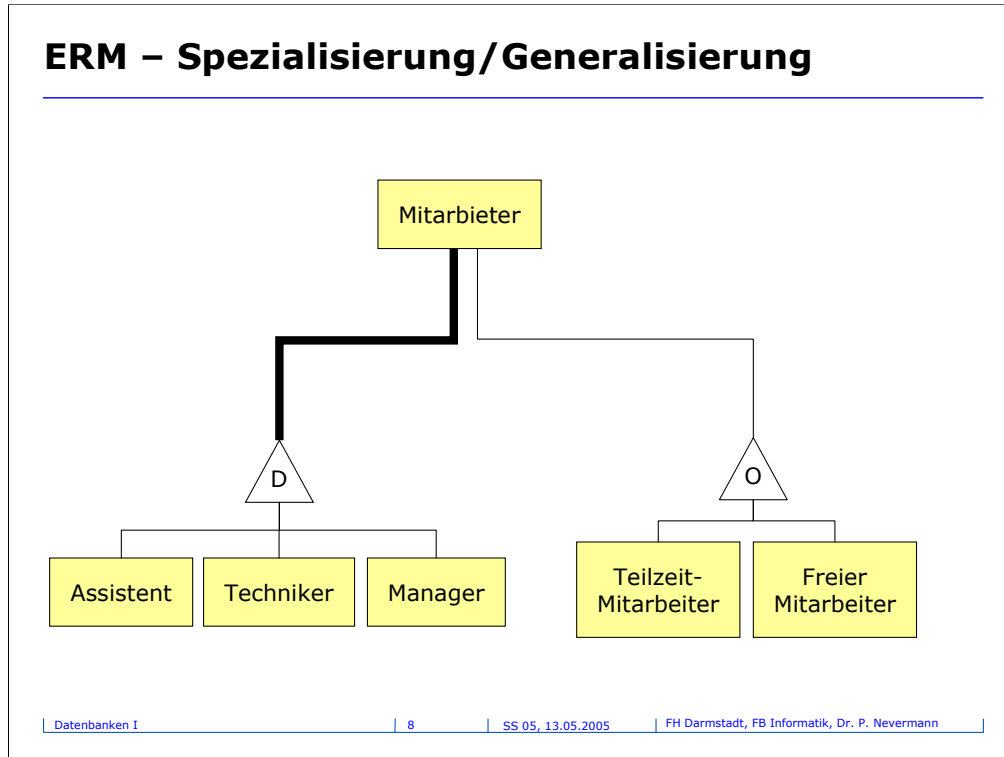
Generalisierung beschreibt den zur Spezialisierung umgekehrten Vorgang: zwei oder mehr Entities werden zusammengefaßt (*verallgemeinert*, *generalisiert*) indem ein gemeinsamer Supertyp definiert wird. Gebe es zum Beispiel in unserer Firmen-Miniwelt auch Kunden (durch Entity "Kunde" repräsentiert), dann könnte es sinnvoll sein "Mitarbeiter" und "Kunde" durch einen Supertyp "Person" zu generalisieren, z.B. um die gemeinsamen Merkmale von Mitarbeitern und Kunden (Name, Adresse, ...) nur einmal beschreiben zu müssen. Im Diagramm ist später nicht mehr zu erkennen, ob ein Dreieck eine Spezialisierung oder eine Generalisierung darstellt (was letztendlich auch egal ist, da der Unterschied im Entwurfsprozeß und nicht im Ergebnis liegt).

Vererbung

Ein wesentliches Prinzip bei Spezialisierung und Generalisierung ist, daß Merkmale (Attribute und Beziehungen) vom Supertyp auf die Subtypen *vererbt* werden, und dadurch die allgemeinen Merkmale nur im Supertyp, während die speziellen Merkmale nur in den Subtypen beschrieben werden müssen. So ist z.B. die Beziehung "leitet" ein spezielles Merkmal des Subtyps "Manager", während die Beziehungen "gehört zu", "arbeitet in" und "berichtet an" allgemeine Merkmale des Supertyps Mitarbeiter sind. Ebenso verhält es sich mit den allgemeinen Attributen "Name", "Adresse", etc. bzw. mit einem speziellen Attribut "Budget" des Managers oder "Skills" des Technikers.

Hierarchie

Ein Subtyp einer Spezialisierung kann selbst wieder spezialisiert werden, wodurch Spezialisierungshierarchien entstehen. Zum Beispiel könnte der Techniker wieder in "Hardwarespezialist" und "Software-Engineer" spezialisiert werden. Merkmale (Attribute und Beziehungen) vererben sich dann natürlich über alle Hierarchie-Ebenen hinweg.



Mehrfache Spezialisierung

Ein Entity kann auch mehrfach ("orthogonal" zueinander) spezialisiert werden, falls das in der Miniwelt von Bedeutung ist. Das Diagramm auf der Folie zeigt eine zweite Spezialisierung von Mitarbeiter nach dem Aspekt "Art des Arbeitsvertrags". Während die erste Spezialisierung nach dem Aspekt "Funktion" disjunkt und total ist, ist die neue Spezialisierung überlappend (es gibt freie Mitarbeiter mit Teilzeitvertrag) und partiell (die meisten Mitarbeiter sind festangestellte Mitarbeiter mit Vollzeitvertrag, also nicht Instanz eines der beiden Subtypen).

Mehrfachvererbung

Mehrfachvererbung liegt vor, wenn ein Entity Subtyp in mehr als einer Spezialisierung ist, d.h. Merkmale von mehreren direkten Supertypen erbt.

Entwurfsrichtlinien

- **Weak Entity oder gegliedertes Attribut?**
- **Gründe für Spezialisierung**
- **Gründe für Generalisierung**
- **Spezialisierung oder Typ-Attribut?**

Weak Entity oder komplexes Attribut?

Ein Weak Entity kann auch als komplexes Attribut modelliert werden. Zum Beispiel kann das Weak Entity "Angehöriger" auch durch das multiple, gegliederte Attribut:

{Angehöriger (Vorname, Geschlecht, Geb-Datum, Beziehung)} ersetzt werden. Ein Grund sich für Weak-Entity zu entscheiden könnte bestehen, wenn das Weak-Entity selbstständig an Beziehungen teilnimmt (z.B. Angehöriger "ist versichert bei" Krankenkasse).

Gründe für Spezialisierung

- Die Instanzen eines Entities lassen sich (für die Miniwelt bedeutungsvoll) derart typisieren (kategorisieren, in Gruppen aufteilen), daß einzelne Subtypen durch spezifische Merkmale (Attribute und Beziehungen) beschrieben werden können
- Beispiele:
 1. "Manager" als Subtyp von "Mitarbeiter" zeichnet sich durch manager-spezifische Merkmale aus (Beziehung: "leitet" Abteilung, Attribut: "Budget")
 2. "Mitarbeiter" in "Männlicher Mitarbeiter" und "Weiblicher Mitarbeiter" zu spezialisieren wäre dagegen wenig sinnvoll, da keine Miniwelt-relevanten Merkmale gefunden werden können, welche die beiden Subtypen trennen. In diesem Fall reicht also ein Attribut "Geschlecht" im Entity "Mitarbeiter" aus.

Gründe für Generalisierung

• Mehrere Entities weisen Gemeinsamkeiten auf, derart daß es im Kontext der Miniwelt sinnvoll erscheint einen Supertypen zu definieren, der die gemeinsamen Merkmale trägt

1. Einen Supertypen "Person" (mit den Merkmalen "Name", "Adresse") über "Mitarbeiter" und "Kunde" zu definieren erscheint sinnvoll

2. Einen Supertypen "Benanntes Etwas" (mit Merkmal "Name") über "Mitarbeiter" und "Abteilung" nur deshalb einzuführen, weil beide Entities ein Attribut "Name" tragen scheint wiederum wenig sinnvoll

Spezialisierung oder Typ-Attribut?

Wie das Beispiel mit den männlichen und weiblichen Mitarbeitern zeigt, gibt es manchmal die Modellierungsalternative Spezialisierung vs. Typ-Attribut. Später werden wir sehen, daß Spezialisierungen - bei der Überführung des ER Modells in ein Datenbankschema - häufig mit Hilfe eines Typ-Attributs realisiert werden. [Daneben gibt es natürlich auch andere Realisierungsmöglichkeiten.] Allerdings sollte man die Entscheidung für das Typ-Attribut im ER Modell i.d.R. nicht vorwegnehmen, um später noch die freie Wahl zu haben, d.h. im Zweifel sollte man im ER Modell die Spezialisierung dem Typ-Attribut vorziehen. [Im Falle der männlichen und weiblichen Mitarbeiter würde ich trotzdem das Typ-Attribut schon im ER Modell vorsehen.]

Ausblick

■ SQL DDL (Data Definition Language)

- ◆ Eine standardisierte Sprache für die Definition von Datenbank-Schemas
- ◆ Von DBMS interpretierbar