
Datenbanken und Informationssysteme

Material for CLIL included

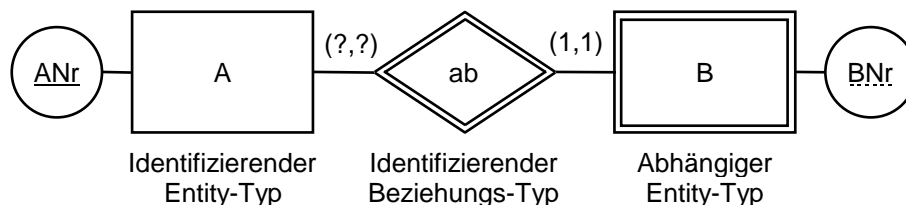
Kurt Hillebrand

September 2014

4 Weiterführende ER-Modellierung

4.1 Abhängige Entity-Typen

- Kern-Entity-Typ (Kernel-Entity-Type, Regular Entity-Type, Fundamental Entity-Type, Dominant Entity-Type):
 - Jedes Entity existiert unabhängig von der Existenz anderer Entities
- Abhängiger Entity-Typ (Dependent Entity-Type, Weak Entity-Type, Subordinate Entity-Type):
 - Kein Entity kann ohne einem übergeordneten Entity (Owner-Entity) eines bestimmten Entity-Typs (Identifizierender Entity-Typ, Identifying Entity-Type) existieren. Wird auch Existenzabhängigkeit (Existence Dependency) genannt.
 - Zwischen dem übergeordneten und dem abhängigen Entity-Typ besteht ein Beziehungs-Typ bei dem der (min,max)-Wert beim abhängigen Entity-Typ immer (1,1) ist (muss vollständig am Beziehungs-Typ teilnehmen). Dieser Beziehungs-Typ wird Identifizierender Beziehungs-Typ (Identifying Relation-Type) genannt.
Hinweis: Die Tatsache, dass der (min,max)-Wert bei einem Entity-Typ (1,1) ist, bedeutet nicht das Vorliegen eines abhängigen Entity-Typs.
 - Die Beziehung vom abhängigen Entity zum identifizierenden Entity ändert sich während der Existenz des abhängigen Entities nicht, non-transferable Relationships (d.h. der Fremdschlüsselwert ändert sich nicht).
 - Der Primärschlüssel des Identifizierenden Entity-Typs ist im Primärschlüssel des Abhängigen Entity-Typs als Fremdschlüssel enthalten (ist Teilschlüssel im Abhängigen Entity-Typ).
 - Üblicherweise werden mit dem Identifizierenden Entity alle von ihm Abhängigen Entities mitgelöscht (siehe SQL: ON DELETE CASCADE).
- Graphische Darstellung im ERD:

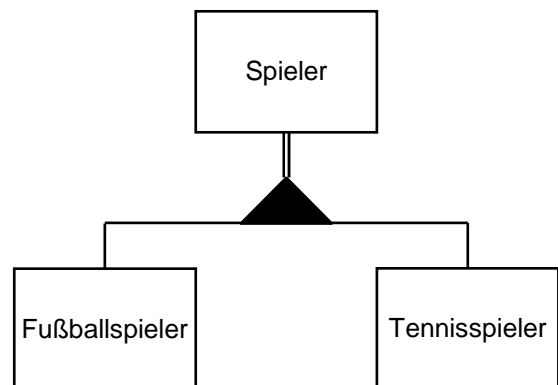
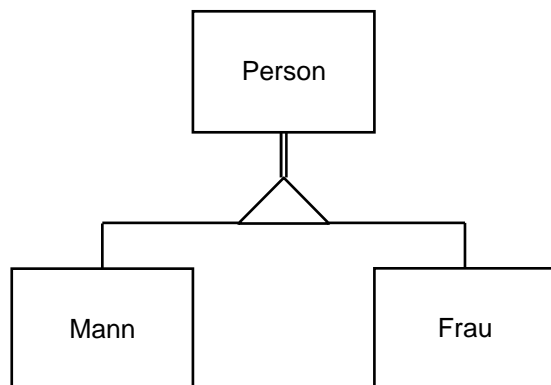


- Tabellen:
 - A (ANr, ...)
 - B (ANr > A, BNr, ...)
- Strichliert unterstrichener Primärschlüssel symbolisiert, dass er alleine nicht identifizierend ist
- Ein Abhängiger Entity-Typ kann auch von mehreren Entity-Typen abhängen, es gibt dann mehrere Identifizierende Beziehungs-Typen alle entsprechenden Fremdschlüssel werden Teil des Primärschlüssels.
- Beispiele:
 - Auftrag – Auftragsposition, Gebäude – Raum, Angestellter – Angehöriger des Angestellten, Geschäft – Geschäftskündigung

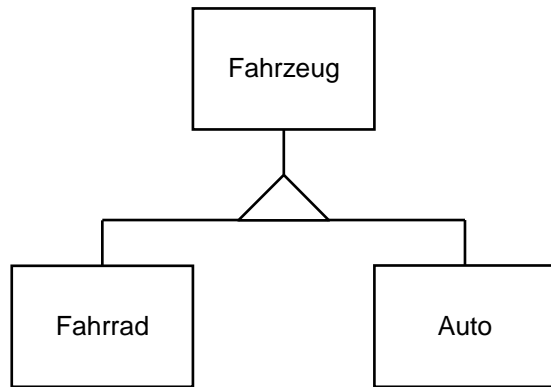
4.2 Überlagerte Entity-Typen

- Sub- und Super-Entity-Typen, Unter- und Ober-Entity-Typen (abgekürzt auch Sub- und Super-Typen)
 - Es gibt einen Super-Entity-Typ und dazu einen oder mehrere Sub-Entity-Typen
 - Jedes Entity eines Sub-Entity-Typs muss auch als Entity im Super-Entity-Typ enthalten sein (die Sub-Entity-Typen sind echte Teilmengen des Super-Entity-Typs)
 - Der Super-Entity-Typ enthält alle Attribute, die den Entities gemeinsam sind.
 - Die Sub-Entity-Typen enthalten die individuellen Attribute (haben in der Regel denselben Primärschlüssel wie der Super-Entity-Typ).
 - Zwischen dem Super-Entity-Typ und den einzelnen Sub-Entity-Typen besteht jeweils ein 1:1-Beziehungs-Typ, der als 'ist ein' ('is-a') bezeichnet wird.
- Übung: Welche Werte treten bei der (min,max)-Notation auf?

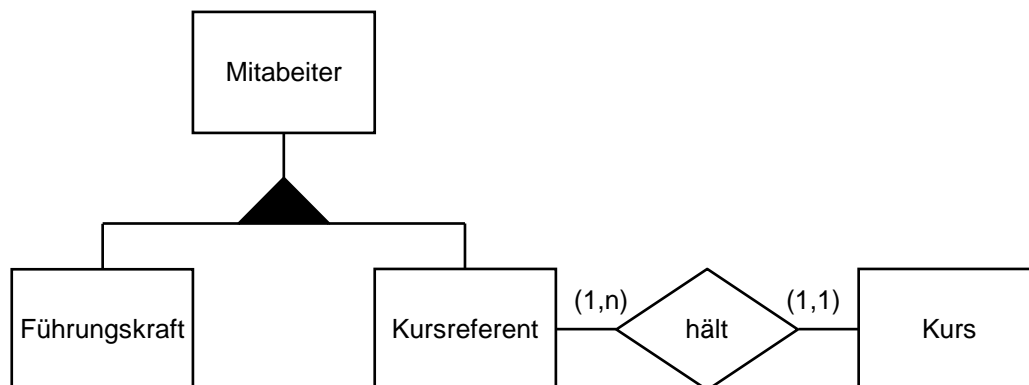
- Beispiele:
 Person - Mann, Frau
 Mitarbeiter - Vollbeschäftigter Mitarbeiter, Teilbeschäftigter Mitarbeiter
 Person - Lehrer, Student
 Geschäft - Kreditgeschäft, Dokumentengeschäft
 Turbine - Axialturbine, Francis turbine, Pelton turbine
 Führerscheinbesitzer - PKW-Fahrer, LKW-Fahrer, Motorradfahrer
- Die Aufteilung in Sub-Entity-Typen oder die Sub-Entity-Typen selbst werden als Spezialisierung(en) (Specialization) des Super-Entity-Typs bezeichnet.
 Beispiel: Mann und Frau sind Spezialisierungen von Person
- Die Zusammenfassung zum Super-Entity-Typ oder der Super-Entity-Typ selbst wird als Generalisierung (Generalization, Verallgemeinerung) der Sub-Entity-Typen bezeichnet.
 Beispiel: Person ist die Generalisierung von Mann und Frau
- Sub-Entity-Typen können selbst wieder Sub-Entity-Typen haben, sodass eine ganze Typ-Hierarchie entsteht (vergleiche Objektorientierte Konzepte wie Vererbung, etc.)
 Beispiel: Mitarbeiter - Sekretär, Ingenieur, Techniker;
 Ingenieur - Kfz-Ingenieur, Elektronik-Ingenieur, Flugzeugbau-Ingenieur;
 Beispiel: Luftfahrzeug - Flugzeug, Hubschrauber, Luftkissenfahrzeug, anderes Luftfahrzeug
 Flugzeug - Segelflieger, Motorflieger, Düsenflugzeug;
- Wenn eine Spezialisierung in mehr als einen Sub-Entity-Typ vorliegt, so können zwei Bedingungen angegeben werden, die die Art der Spezialisierung näher beschreiben:
 - Disjointness Constraint: Gibt es Entities des Super-Entity-Typs, die in mindestens zwei Sub-Entity-Typen enthalten sind?
 nein: disjunkt, disjoint
 ja: nicht disjunkt, overlapping
 - Completeness Constraint (Vollständigkeitsbedingung): Ist jedes Entity des Super-Entity-Typs in mindestens einem Sub-Entity-Typen enthalten?
 ja: total
 nein: partiell, partial
- Sei E der Super-Entity-Typ und seien E1, E2, ..., En die Sub-Entity-Typen für eine bestimmte Spezialisierung:
 - Die Spezialisierung ist disjunkt wenn
 $E_i \cap E_j = \{\}$ für $i \neq j$ und $i, j \in \{1, 2, \dots, n\}$
 - Die Spezialisierung ist total wenn
 $E_1 \cup E_2 \cup \dots \cup E_n = E$
- Für den 'ist ein'-Beziehungs-Typ werden spezielle graphische Darstellungsarten verwendet (Dreieck), mit denen auch die oben angeführten Bedingungen ausgedrückt werden können:
 - total, disjunkt:
 - total, nicht disjunkt (Sportverein mit Sektionen Fußball und Tennis):



- partiell, disjunkt:



- partiell, nicht disjunkt:



Mitarbeiter (Personal#, Name, Adresse, ...)

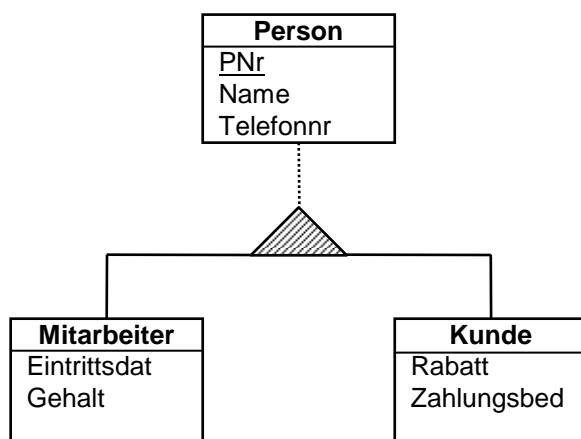
Führungskraft (Personal# > Mitarbeiter, Überstundenpauschale, Anz Sekretärinnen, Geleitete Abteilung, ...)

Kursreferent (Personal# > Mitarbeiter, Stundenkosten, Gehaltene Kursstunden, ...)

Kurs (Kurs#, Titel, ..., Personal# > Kursreferent)

4.3 Implementierung von Überlagerten Entity-Typen in Tabellen

- Beispiel



- Übung: Interpretieren Sie die verschiedenen Möglichkeiten der Disjointness- und Completeness-Constraints

1) Eine Tabelle pro Entity-Typ

Die Primärschlüssel der Sub-Entity-Typen sind dieselben wie der Primärschlüssel des Super-Entity-Typs, gleichzeitig sind sie auch Fremdschlüssel zum Super-Entity-Typ.

a) Die Tabellen für die Sub-Entity-Typen übernehmen den Primärschlüssel vom Super-Entity-Typ (Vertical Mapping)

Person (PNr, Name, Telefonnr)

Mitarbeiter (PNr > Person, Eintrittsdat, Gehalt)

Kunde (PNr > Person, Rabatt, Zahlungsbed)

- bevorzugte Lösung
- Vorteile: strukturell sauber, keine Redundanzen
- Nachteile: Vollständige Information (alle Attribute) über einen Sub-Typ auf zwei Tabellen verteilt (view mit join), Information über Sub-Typ-Zuordnung in eigenen Tabellen, viele Tabellen

b) Die Tabellen für die Sub-Entity-Typen übernehmen alle Attribute vom Super-Entity-Typ (Horizontal Mapping)

Person (PNr, Name, Telefonnr)

Mitarbeiter (PNr > Person, Name, Telefonnr, Eintrittsdat, Gehalt)

Kunde (PNr > Person, Name, Telefonnr, Rabatt, Zahlungsbed)

- Vorteile: Vollständige Information (alle Attribute) über einen Sub-Typ in einer Tabelle
- Nachteile: Redundanzen (sogar mehrfach, wenn overlapping), Information über Sub-Typ-Zuordnung in eigenen Tabellen, viele Tabellen

2) Eine Tabelle pro Sub-Entity-Typ

Mitarbeiter (PNr, Name, Telefonnr, Eintrittsdat, Gehalt)

Kunde (PNr, Name, Telefonnr, Rabatt, Zahlungsbed)

- Vorteile: Vollständige Information (alle Attribute) über einen Sub-Typ in einer Tabelle
- Nachteile: Speicherung der Attribute des Super-Typs bei Completeness Constraint partial, Redundanzen (sogar mehrfach, wenn overlapping), Information über Sub-Typ-Zuordnung in eigenen Tabellen, Zugriff auf alle Entities des Super-Typs (union)
- unübliche Lösung (nur überlegenswert, wenn total und disjoint)

3) Eine Tabelle für alle Entity-Typen (Filtered Mapping)

Person (PNr, Name, Telefonnr, Eintrittsdat, Gehalt, Rabatt, Zahlungsbed)

Zuordnung zu den Sub-Entity-Typen über Abfrage der NULL-Werte aufwendig, daher

a) zusätzlich ein Attribut: ist von welchem Sub-Typ (wenn disjoint)

Person (PNr, Name, Telefonnr, PArt, Eintrittsdat, Gehalt, Rabatt, Zahlungsbed)

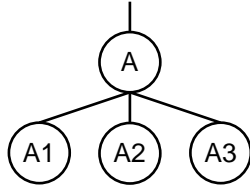
b) zusätzlich pro Sub-Typ ein Attribut (Datentyp Logical): ist von diesem Sub-Typ (wenn overlapping)

Person (PNr, Name, Telefonnr, is-Mitarbeiter, Eintrittsdat, Gehalt, is-Kunde, Rabatt, Zahlungsbed)

- Vorteile: Vollständige Information (alle Attribute) über einen Sub-Typ in einer Tabelle, Information über Sub-Typ-Zuordnung direkt beim Super-Typ, nur eine Tabelle
- Nachteile: viele NULL-Werte in den Attributen der Sub-Typen (speziell, wenn partial und disjoint)
- überlegenswert, wenn Sub-Typen wenige zusätzliche Attribute enthalten (und diese überwiegend nicht NULL sind)
- Mit anderer Anzahl von Attributen in den Entity-Typen oder mehr als zwei Sub-Entity-Typen analoge Vorgangsweise
- Übung: Untersuchen Sie die Beispiele aus 4.2 auf sinnvolle Implementierungsmöglichkeiten
- Übung: Wie verändern sich die Überlegungen, wenn es nur einen Sub-Entity-Typ gibt (welche Disjointness- und Completeness-Constraints sind dann überhaupt sinnvoll)?
- Übung: Erweitern Sie die Überlegungen auf Hierarchien von Super-Sub-Entity-Typen

4.4 Arten von Attributen

- Einfaches Attribut (Simple Attribute, Atomic Attribute): Ein Attributwert ist nicht weiter unterteilt.
- Zusammengesetztes Attribut (Structured Attribute, Composite Attribute): Der Attributwert besteht aus mehreren Teilen.
Beispiel: Adresse ist unterteilt in Straße, Hausnummer, Stadt, Land. Name ist unterteilt in Vorname, Nachname.
Darstellung: Adresse(Straße,Hausnummer,Stadt,Land), Name(Vorname,Nachname)
Graphisch:



- Einwertiges Attribut (Singlevalued Attribute): Bei einem Entity hat das Attribut einen Wert.
- Mehrwertiges Attribut (Multivalued Attribute): Bei einem Entity treten mehrere Werte in diesem Attribut auf.
Beispiel: Farbe eines Autos, Titel einer Person, Vornamen einer Person, Telefonnummern eines Angestellten, Qualifikation eines Angestellten
Darstellung: In geschwungenen Klammern
Graphisch:



- Basis-Attribut (Base Attribute): Der Attributwert eines Entities kann von keinem (keinen) anderen Wert eines Attributs (Werten von Attributen) abgeleitet werden.
- Abgeleitetes Attribut (Derived Attribute): Der Attributwert eines Entities kann von einem Wert (mehreren Werten) eines Attributs (mehrerer Attribute) abgeleitet werden.
Führt zu Redundanzen in der Datenverwaltung und ist im allgemeinen zu vermeiden.
Beispiele: Alter - Geburtsdatum, Auftragswert - Summe Positionswerte des Auftrags
Graphisch:

