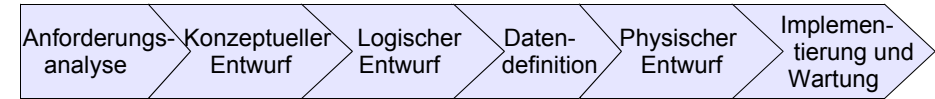


Datenbankdesign – Vom Modell zum Datenbankentwurf

- Vorgehensmodell zum Datenbankdesign
- Vom ER Modell zum relationalen Modell
 - Anforderungen
 - Entity-Typen, Beziehungstypen, Wertebereiche
- Entwurfsziele
 - Inhaltliche Ziele
 - Einfaches Ableiten aller Anwendungsdaten
 - Konzentration auf wesentliche Daten
 - Keine redundante Daten
 - Datenbankdesign entspricht Modell
 - Informationserhaltung
 - Konsistenzerhaltung
 - Weitere Gütekriterien
 - Verständlichkeit und Lesbarkeit
 - Erweiterbar, modular, wieder verwendbar

Vorgehensmodell

Phasen

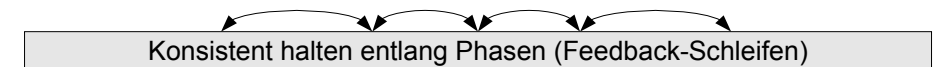


Werkzeuge

Entity Relationship Modell	Relationales Modell	SQL DDL DML	SQL (DB-spez.)
----------------------------	---------------------	-------------	----------------

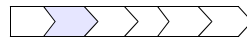
Aufgaben

Welche Objekte, Beziehungen, Attribute?
 Übersetzung ERM → Rel. M.
 Normalisierung
 Umsetzung Rel. M. → SQL
 Tuning
 Optimierung
 Änderung



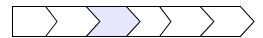
Konzeptueller Gesamt-Entwurf

- Modellierung verschiedener Sichten auf gleiche Objekte
 - Verschiedene Miniwelten, Anforderungen, Anwendungen
 - Modellierung von Bedingungen
- Probleme
 - Homonyme, Synonyme
 - Typkonflikte, Attributkonflikte, Wertebereichskonflikte
 - Bedingungskonflikte
- Integration in Gesamtschema
 - Einheitliche Miniwelt für existierende und zukünftige Anwendungen
 - Balance Flexibilität, Erweiterbarkeit und Rigidität
- Ziel: ER-Modell
 - Aussagekräftig, einfach und verständlich
 - Minimal, nicht überlappend, formal eindeutig
 - Darstellung in Diagrammform



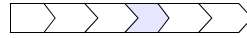
Logischer Entwurf

- Umsetzung von Entity-Relationship Modell in relationales Modell
 - Weitestgehend automatisierbar
 - Ergebnis: Erstes relationales Modell
- Verbesserung des relationalen Modells
 - Nicht vollständig automatisierbar
 - Normalisierung
 - Einfluss von Kontextwissen (Bedingungen)
 - Ergebnis: Optimiertes relationales Modell
- Einheitliche relationale Miniwelt
 - Gesamtsicht und anwendungsspezifische Sichten
 - Dokumentiert
 - Von allen Anwendungen effizient steuerbar
 - Datenbankunabhängig



Datendefinition

- Umsetzung der konzeptuellen Gesamtsicht
 - Definition des Datenbankschemas
- Konkretisierung
 - Nutzen einer Data Definition Language (DDL)
 - Nutzen einer Data Manipulation Language (DML)
 - [Relationales Modell und SQL]
- Unterstützung der Anwendungen
 - Externe Sichten
- Noch keine Optimierungen
 - Noch nicht die interne Ebene
 - Physischer Entwurf zur Optimierung (idealerweise nicht notwendig)



Phasenbegleitende Methoden

- Verifikation
 - Formale Überprüfung von Eigenschaften der Relationenschemas
- Prototyping
 - Teilimplementierungen
 - Ausprobieren
- Testen
 - Validieren durch (extreme) Testdaten
 - Kontinuierlich weiter gepflegt
 - Automatisierbar



Abbildung ER auf relationales Modell

Halbautomatisches Abbilden von ER auf relationales Modell

Entity Relationship Modell

- Entity
 - Attribute
 - Primärschlüssel
- Beziehung R_1, R_2
 - Attribute
 - Schlüssel
 - 1:1
 - 1:N
 - N:M
 - IST

Relationales Modell

- Relationenschema
 - Attribute des zugehörigen Relationenschemas
 - Primärschlüssel
- Relationenschema
 - Attribute
 - Weitere Attribute
 - Primärschlüssel von R_1 oder R_2
 - Primärschlüssel von R_2
 - Vereinigung Primärschlüssel von R_1 und R_2
 - R_1 erhält zusätzlich Schlüssel von R_2

Laufendes Beispiel Ergebnis Konzeptueller Entwurf

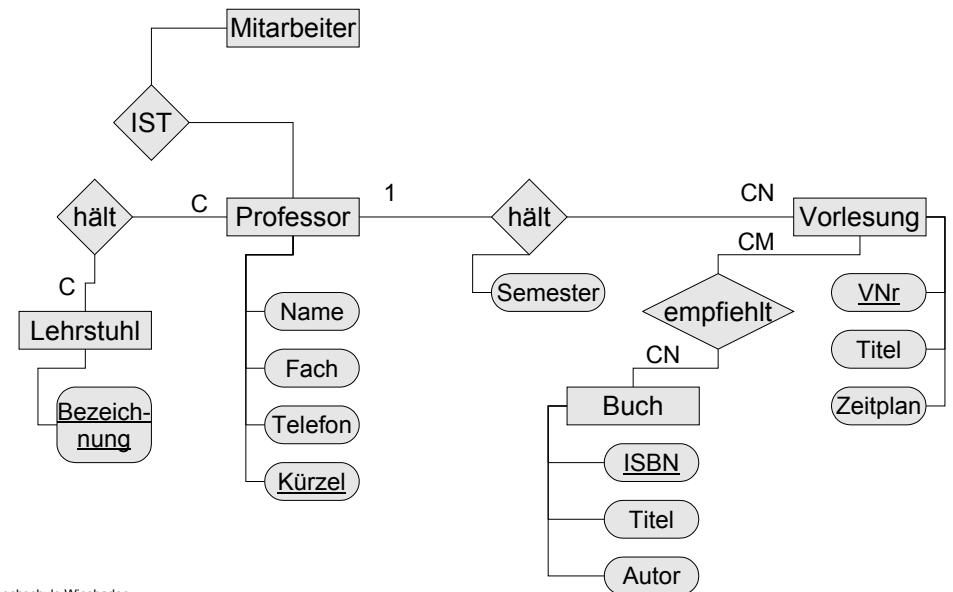


Abbildung Entity-Typ

- Relationenschema
 - Alle Attribute werden übernommen
 - Primärschlüssel wählen aus Schlüsselkandidaten
- Wahl Primärschlüssel
 - So einfach wie möglich
 - Ein Attribut (statt mehrere)
 - Integer (statt String)

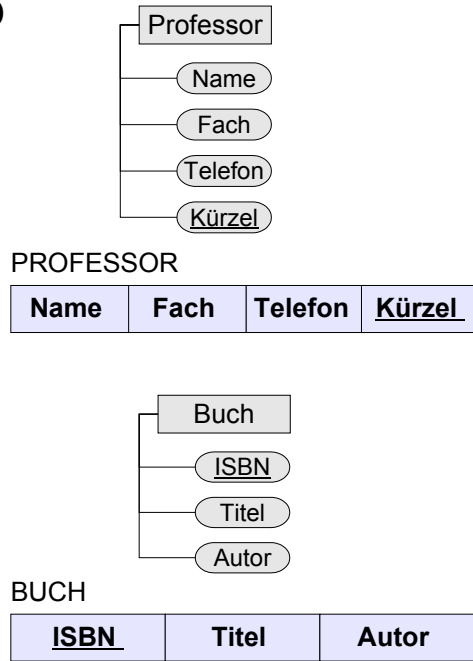
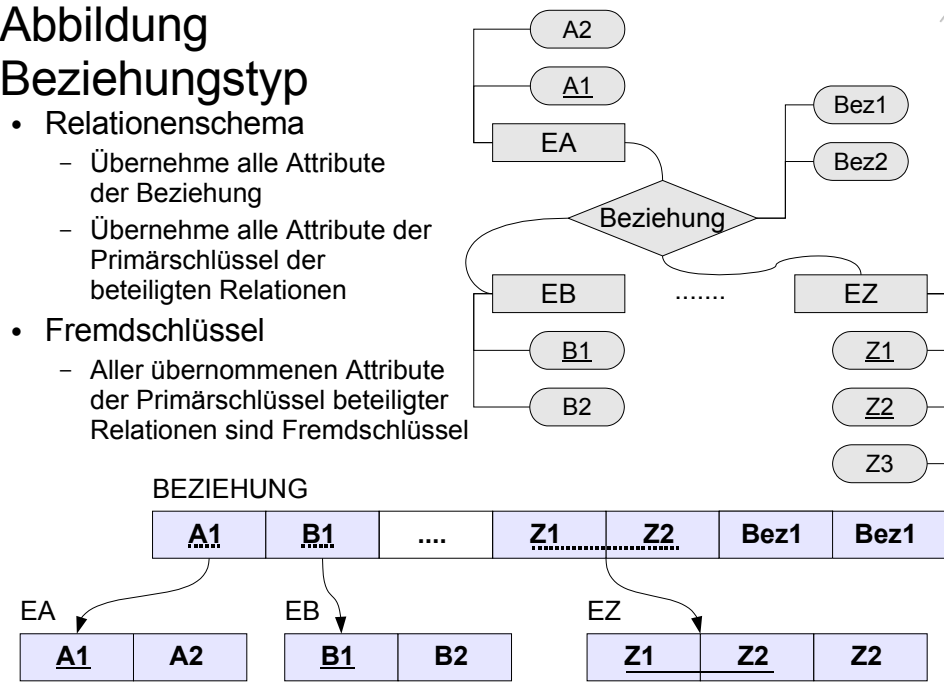


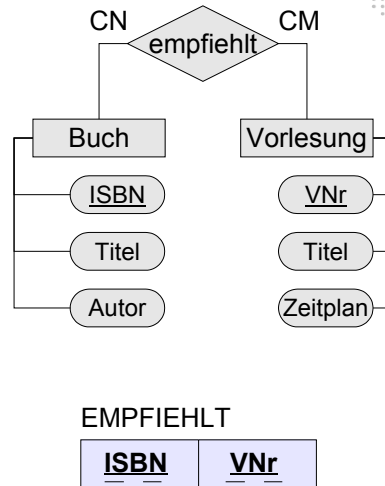
Abbildung Beziehungstyp

- Relationenschema
 - Übernehme alle Attribute der Beziehung
 - Übernehme alle Attribute der Primärschlüssel der beteiligten Relationen
- Fremdschlüssel
 - Alle übernommenen Attribute der Primärschlüssel beteiligter Relationen sind Fremdschlüssel



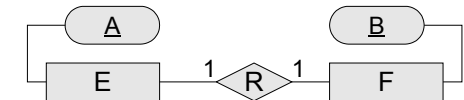
Beziehungstyp – Wahl Primärschlüssel

- 1:1
 - Beide Primärschlüssel der Relationen des Entity-Typs sind Schlüsselkandidat, einer davon als Primärschlüssel für die Beziehungsrelation wählen
- 1:N
 - Der Primärschlüssel der N-Seite wird Primärschlüssel der Beziehungsrelation
- M:N
 - Beide Primärschlüssel der Relation des Entity-Typs werden gemeinsam Primärschlüssel der Beziehungsrelation
- Beispiel
 - (C)N:(C)M Relation empfiehlt zwischen Buch und Vorlesung

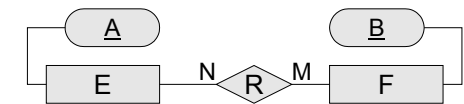


Ziel – Erhaltung Informationskapazität

- Keine Kapazitätserhöhung
 - Nicht mehr darstellbar als durch ER-Modell vorgeben
- Keine Kapazitätsverminderung
 - Alles was im ER-Modell möglich ist soll darstellbar sein
- Beispiel: (K Menge Schlüsselkandidaten)
 - Relation für Beziehung R in einer 1:1 Beziehung
 - Auswahl eines Primärschlüssels erzwingt Kapazitätserhöhung
 - {A,B} ist nur Super-Schlüssel, kein Schlüssel
 - Relation für Beziehung R in einer N:M Beziehung



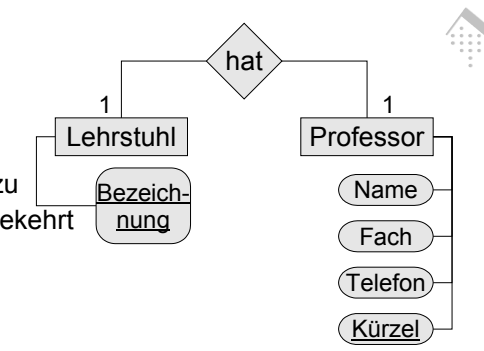
(1,3) und (2,3) können **nicht** beide in R(A,B) sein
 $K = \{A, \{B\}\}$: geht nicht, ok
 $K = \{A\}$: geht, erhöhend



{(1,3), (2,3)} kann in R(A,B) sein
 $\{\{A\}, \{B\}\}$: geht nicht, mindernd
 $\{\{B\}\}$: geht nicht
 $\{\{A\}\}$: nein ((1,3), (1,4)) nicht
 $K = \{A, B\}$: geht, ok

Beziehungstyp – 1:1

- 1:1 Beziehung R_1, R_2
 - Jedes Tupel in R_1 hat Beziehung zu genau einem Tupel in R_2 und umgekehrt
 - Jeder Prof genau ein Lehrstuhl
Jeder Lehrstuhl ist besetzt
- Vorgehen für „hat“
 - Übernahme alle Attribute
 - „hat“ hat keine Attribute
 - Beide Primärschlüssel
 - Kürzel, Bezeichnung
 - der Entity-Relationen
 - Lehrstuhl und Professor
 - werden Schlüsselkandidat, einer davon
 - Kürzel
 - wird Primärschlüssel



PROFESSOR

Name	Fach	Telefon	<u>Kürzel</u>
------	------	---------	---------------

LEHRSTUHL

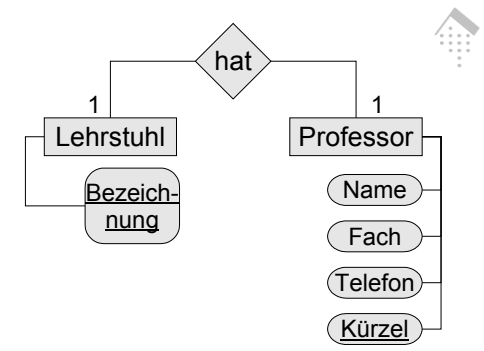
<u>Bezeichnung</u>

HAT

<u>Kürzel</u>	<u>Bezeichnung</u>
---------------	--------------------

Beziehungstyp 1:1

- Achtung
 - Transformation ist nicht informationserhaltend, Informationserhöhung
 - Ein Lehrstuhl kann in diesem Relationenmodell von zwei Professoren gehalten werden
- Lösung
 - Zusätzliche Einschränkung (UNIQUE-Constraint)
 - Zusätzlich Relation zusammenfallen lassen



PROFESSOR

Name	Fach	Telefon	<u>Kürzel</u>
------	------	---------	---------------

LEHRSTUHL

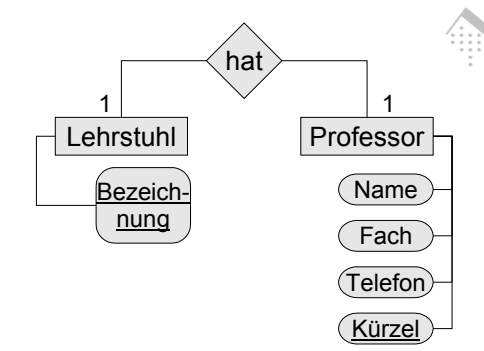
<u>Bezeichnung</u>

HAT

<u>Kürzel</u>	<u>Bezeichnung</u>
---------------	--------------------

Beziehungstyp 1:1

- Falls echte 1:1 Beziehung
 - Jeder Professor einen Lehrstuhl haben muss
 - Jeder Lehrstuhl einen Professor haben muss
- Vereinfachung
 - Eine Relation statt drei Relationen
- Zusammenlegen der Attribute
 - Alle Attribute in eine neue Relation
 - Primärschlüssel werden Schlüsselkandidaten
 - Einen Schlüssel auswählen als Primärschlüssel

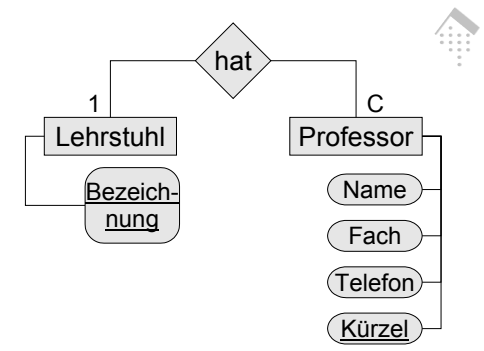


PROFESSOR

Name	Fach	Telefon	<u>Kürzel</u>	<u>Bezeichnung</u>
------	------	---------	---------------	--------------------

Beziehungstyp 1:C

- Was passiert
 - Bei einer Tabelle, bei optionalen Werten?
 - Beispiel: Unbesetzte Lehrstühle
- Auffüllen mit NULL-Werten
 - Alle Instanzen verwalten zwingt andere Attribute mit NULL zu füllen
- Extra-Relation
 - NULL-Werte meist unerwünscht
 - Nur eine Seite zusammenführen



PROFESSOR

Name	Fach	Telefon	Kürzel	Bezeichnung
Schwan	Inf	605	Se	CompGraf
NULL	NULL	NULL	NULL	Faulenzen

PROFESSOR

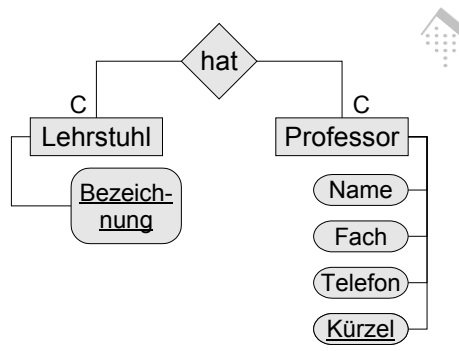
Name	Fach	Telefon	<u>Kürzel</u>	<u>Bezeichnung</u>
------	------	---------	---------------	--------------------

LEHRSTUHL

<u>Bezeichnung</u>

Beziehungstyp C:C

- C:C
 - Bei Zusammenführung überall NULL-Werte möglich
 - Kein Primärschlüssel definierbar
 - Beispiel
 - Lehrstuhl ohne Prof
 - Prof ohne Lehrstuhl
- C:C Vorgehen
 - Überführung in drei Relationen wie bei 1:1 ohne Zusammenlegung der Attribute

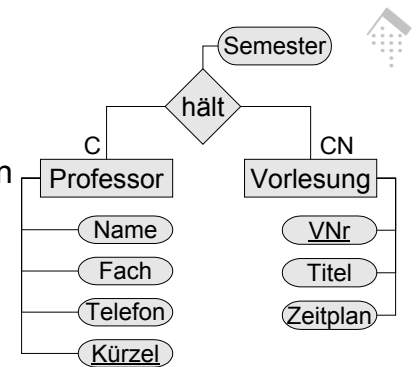


PROFESSOR

Name	Fach	Telefon	Kürzel	Bezeichnung
Schwan	Inf	606	Se	CompGraf
NULL	NULL	NULL	NULL	Faulenzen
Gates	Extend	666	Gs	NULL

Beziehungstyp – (C)1:(C)N

- Attribute und Primärschlüssel der beteiligten Relationen übernehmen
- Primärschlüssel ist übernommener Primärschlüssel der N-Seite
- Beispiel
 - Vorlesung wird von einem Prof gehalten
 - Professor kann mehrere Vorlesungen halten
- Vorgehen für „hält“
 - Übernahme alle Attribute
 - Semester
 - Übernahme alle Primärschlüssel
 - Kürzel, VNr
 - Der Primärschlüssel der N-Seite (VNr) wird neuer Primärschlüssel



HAELT

<u>VNr</u>	<u>Kürzel</u>	Semester
------------	---------------	----------

PROFESSOR

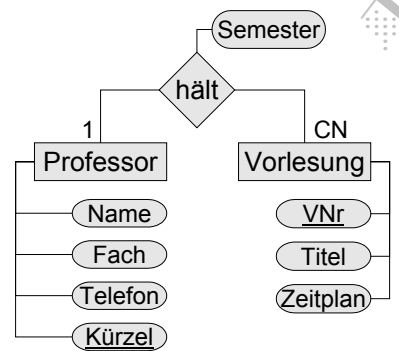
Name	Fach	Telefon	<u>Kürzel</u>
------	------	---------	---------------

VORLESUNG

<u>VNr</u>	Titel	Zeitplan
------------	-------	----------

Beziehungstyp – 1:(C)N

- Zusammenlegung
 - Integration der Beziehung in N-Seite
 - Wenn die 1-Seite nicht NULL sein kann
 - Beispiel: Jede Vorlesung wird von einem Professor gehalten
 - NULL Werte für Fremdschlüssel zulässig (unschön)
- Durchführung
 - Integration der Attribute in Vorlesung
 - Anderer Primärschlüssels als Fremdschlüssel
 - Beispiel: Semester in Vorlesung, Kürzel des Professors als Fremdschlüssel



PROFESSOR

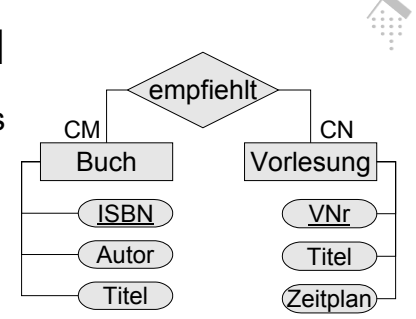
Name	Fach	Telefon	<u>Kürzel</u>
------	------	---------	---------------

VORLESUNG

<u>VNr</u>	Titel	Zeitplan	<u>Kürzel</u>	Semester
------------	-------	----------	---------------	----------

Beziehungstyp – (C)N:(C)M

- Neue Relation mit allen Attributen plus Primärschlüssel beteiligter Relationen
- Beide Primärschlüssel zusammen werden neuer Primärschlüssel der Beziehungsrelation
- Kein Möglichkeit zusammenzufassen
- Beispiel
 - Ein Buch für verschiedene Vorlesungen empfohlen
 - Mehrere Bücher für eine Vorlesung empfohlen
- Durchführen Beispiel
 - Übernahme alle Attribute: {}
 - Übernahme alle Primärschlüssel, die dann VORLESUNG neuer Primärschlüssel sind: ISBN, VNr



EMPFIEHLT

<u>VNr</u>	<u>ISBN</u>
------------	-------------

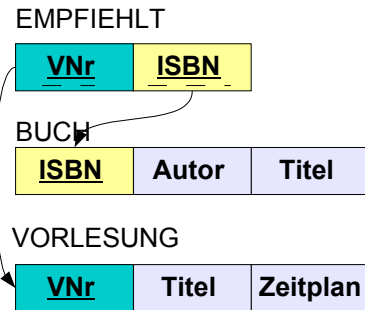
BUCH

<u>ISBN</u>	Autor	Titel
-------------	-------	-------

<u>VNr</u>	Titel	Zeitplan
------------	-------	----------

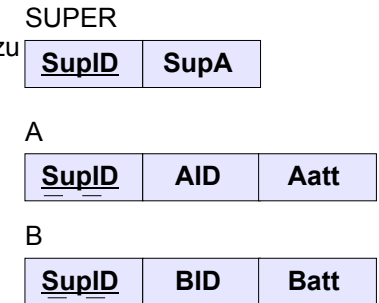
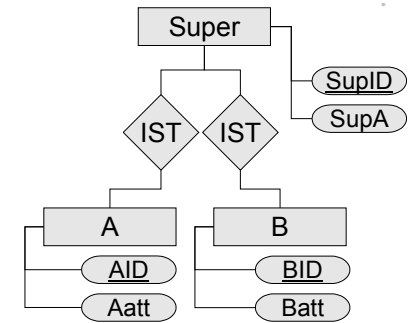
Fremdschlüssel bei Transformation

- Primärschlüssel werden bei Beziehungstypen als Attribute übernommen
- Diese Attribute sind Fremdschlüssel in der Beziehungsrelation
- Im ER-Modell gibt es keine Fremdschlüssel!!!
 - Nur Beziehungen
- Beispiel:
 - ISBN ist Fremdschlüssel in der Relation "empfiehlt" und referenziert auf Buch
 - VNr ist Fremdschlüssel in der Relation "empfiehlt" und referenziert auf Vorlesung



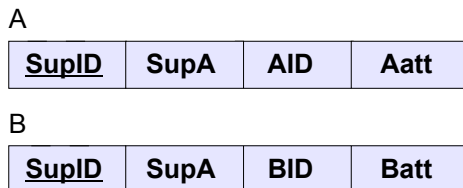
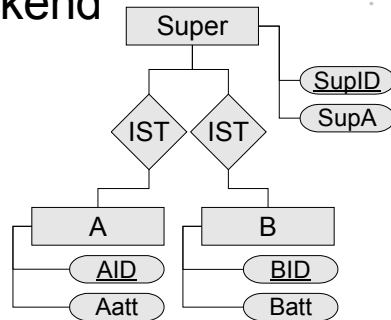
Beziehungstyp – IST

- Generalisierung / Spezialisierung
 - Vererbung der Primärschlüssel an untergeordnete Relation
 - Neue Auswahl des Primärschlüssels möglich
 - Inspektion der Attribute
- Allgemeine Umsetzung in Relationenmodell
 - Erstelle Relation zu jedem Entity-Typ
 - Vererbe Schlüssel von Super-Entity-Typ zu Sub-Entity-Typen
 - Etabliere Fremdschlüsselbeziehung von Sub-Entity-Typen zu Super-Entity-Typen
 - Vorteil: klappt, Kardinalitäten abbildbar
 - Nachteil: Anfragen verbinden Relationen



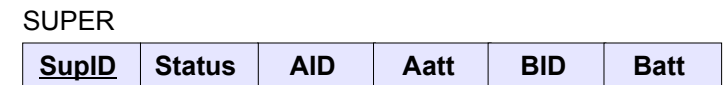
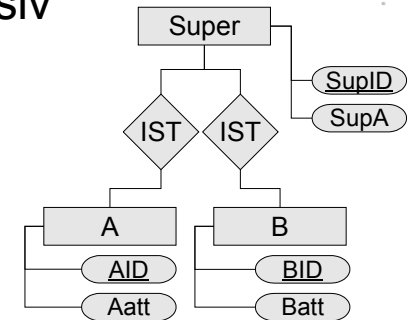
Beziehungstyp – IST, Abdeckend

- Abdeckende Generalisierung
 - Jede Instanz des Entity-Typs muss A oder B (oder beides) sein
- Optionale Umsetzung im Relationenmodell mit Abdeckung
 - Erstelle eine Relationen je Sub-Entity-Typ
 - Vereinige alle Attribute des Super- und des jeweiligen Sub-Entity-Typs
- Bewertung
 - Vorteil: Weniger Relationen, Anfragen die SupA und Aatt verbinden sind effizienter
 - Nachteil: Redundanz wenn sowohl A als auch B, Kardinalitäten zu Super schwierig



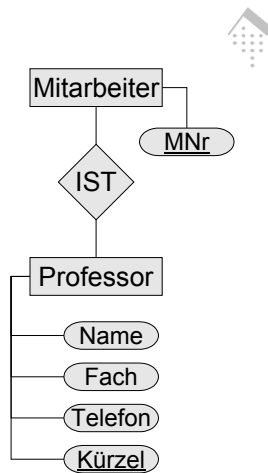
Beziehungstyp – IST, Exklusiv

- Exklusive Spezialisierung
 - Jede Instanz des Entity-Typs Super ist entweder A oder B
- Optionale Umsetzung im Relationenmodell mit Exklusivität
 - Erstelle eine Relation SUPER
 - Vereinige alle Attribute des Super- und der Sub-Entity-Typen
 - Ein neues Status Attribut zur Klassifizierung ob A oder B
- Bewertung
 - Vorteil: Nur eine Relation, Kardinalitäten klar
 - Nachteil: NULL-Werte



Beziehungstyp – IST

- Kein eigenes Relationenschema
- Vererbung der Primärschlüssel an untergeordnete Relation
- Neue Auswahl des Primärschlüssels möglich
- Attribute sollten überprüft werden
- Beispiel
 - Mitarbeiter ist übergeordnet
 - Primärschlüssel von Mitarbeiter sei MNr
 - MNr wird zu Professor-Relation hinzugefügt
 - MNr kann neuer Primärschlüssel von Professor werden



PROFESSOR

Name	Fach	Telefon	Kürzel	<u>MNr</u>
------	------	---------	--------	------------

Grenzen – Beispiel Generalisierung

- Umsetzung Generalisierung
 - Im Relationenmodell nicht intuitiv und natürlich
 - Problematisch beim Zusammenspiel mit objekt-orientierten Design
- Beispiel: Ist-Beziehungen und Nebenbedingungen
 - "Entweder ein Studierender oder ein Laboringenieur darf in die Vorlesung gehen. Es dürfen max. 80 Personen in die Vorlesung gehen. Wie modellieren?"
- Zusätzliche Bedingungen und Überprüfungen notwendig
 - Ausserhalb des Relationenmodells

Generalisierung – Kard. Beispiel

- Anforderung
 - "Entweder ein Studi oder ein LabIng darf in die Vorlesung gehen. Es dürfen max. 80 Personen in die Vorlesung gehen."
- Wie modellieren?
 - Zwei Relationen für Studierender und Laboringenieur
 - Redundanz
 - Anzahl der Leute nicht aus einer Relation ablesbar
 - Eine Relation für Personen generalisiert
 - Auch Nicht-Studierende und Nicht-Laboringenieure sind Personen
 - Jede Person darf in die Vorlesung
- Möglichkeiten
 - Vorherige Umsetzung und weitere Bedingungen
 - Extra Entity-Typ „Vorlesungsteilnehmer“

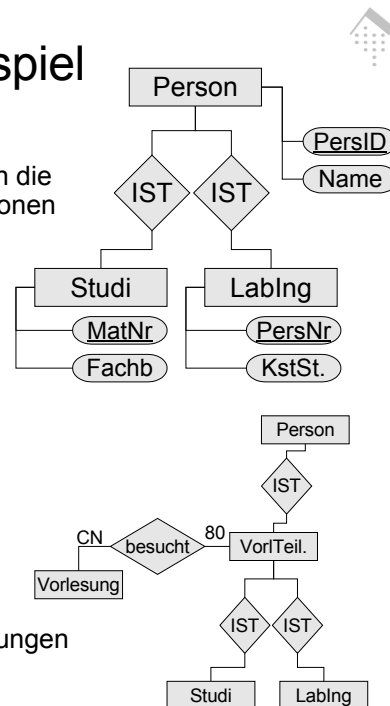


Abbildung von Wertebereichen

- Abbildung auf Wertebereich des Relationenmodells/DBMS
 - Die von Programmiersprachen bekannten abstrakten Datentypen
 - Beispiel:
 - Zeichenketten, String: Char, Varchar
 - Zahlen: int, ...
 - Für Aufzählungen optional
 - Separate Relation mit erlaubten Werten
 - Fremdschlüsselbeziehung zu Relation mit erlaubten Werten
- Nicht atomare Wertebereiche
 - Müssen transformiert werden
 - Direkt auf einzelne atomare Attribute
 - Beispiel
 - Adresse (Strasse, Hausnummer, PLZ, Ort)
 - Aufspaltung von Tupel
 - Neue Relation

ÜBUNGSGRUPP

<u>Bez</u>
A
B
C