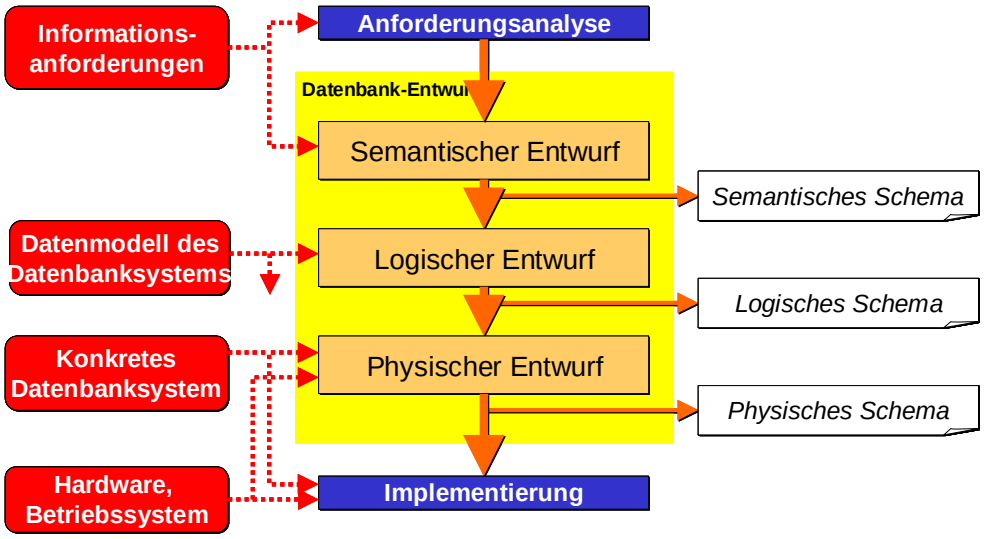
WP Datenbankdesign

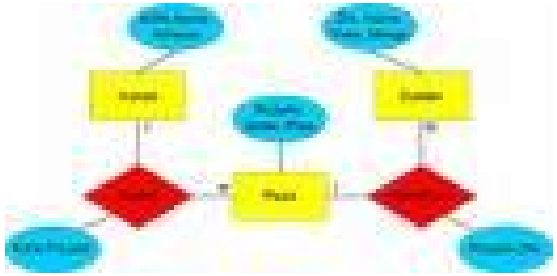
Kapitel 4: Semantische und logische Modellierung

- **Semantische Modellierung**
- **Logische Modellierung**



Semantisches bzw. konzeptionelles Modell

- Vereinfachte Beschreibung eines betrieblichen Realitätsausschnitts (=Miniwelt)
- Definition der relevanten Objekttypen mit ihren **sachlogischen** und **strukturellen** Zusammenhängen
- Basis für das logische Datenmodell
- Neutralität gegenüber Einzelanwendungen
- Unabhängig von der (späteren) Implementierungs-datenbank
- Beschreibung meist durch eine grafisch orientierte, formale Modellierungssprache (bspw. das Entitiy-Relationship-Modell oder UML)
- Sprachliche Basis für die Kommunikation zwischen Entwicklern und Anwendern



Entwicklung eines semantischen Datenmodells

1. Analyse bestehender Informationselemente
2. Festlegung der relevanten Objekte/Entitytypen
3. Festlegung der identifizierenden Attribute
4. Festlegung der relevanten Beziehungen, einschließlich Typ
5. Überprüfung des vorliegenden Datenmodells (Entity-Relationship-Darstellung)
6. Überführung des ER- / UML-Modells in eine normalisierte Form. Insbesondere: Auflösung von (m:n)-Beziehungen
7. Auflösung von Inkonsistenzen und Redundanzen
8. Festlegung der die Objekte und Beziehungen beschreibenden Attribute (zusätzlich zu Nr. 3)
9. Festlegung von Integritätsbedingungen

Vgl. SE, **Glossar**!

Festlegung der identifizierenden Attribute

Einige Richtlinien zur Definition von Schlüsselattributen

1. Konstanz über die Lebensdauer des Objektes
2. Eindeutigkeit über die Lebensdauer der Anwendung
3. Ausreichender Wertebereich
4. Vollständige Nutzung des Wertebereichs
5. Minimalität
6. Maschinelle Generierbarkeit
7. Absicherung durch Prüfziffer
8. Zusammengesetzte Schlüssel sind zu vermeiden
9. Schlüsselattribute, bei denen einzelne Teile davon eine unterschiedliche kontextabhängige Bedeutung haben, sind verboten.

Hinweis: Existenzabhängige Objekte erfordern als Schlüssel eine Kombination von Attributen.

AUFTRAG(AUFTRAGS-NR,...)
AUFTRAGSPOSITION(AUFTRAGS-NR, POSITIONS-NR, ...)

Nummernsysteme und Verschlüsselung

Identifikationsschlüssel: fortlaufende Nummerierung, ohne inhaltliche Bedeutung.

Beispiel: Personal-Nr, Matrikel-Nr

Klassifikationsschlüssel: Verschlüsselung von Attributen, Gruppierung nach Merkmalen

Beispiel: Konto-Nr in Fibu, Artikel-Nr

Informationsschlüssel: unverschlüsselte Attribute

Beispiel: Personal-Nr = Kostenstelle + Geburtsdatum + 1. Buchstabe NN
Probleme: Schlüssel kann sich ändern, kann platzen,
Eindeutigkeit nicht gewährleistet

Verbundschlüssel: Klassifikations-/Informationsteil + davon abhängiger Identifikationsteil

Beispiel: Kfz-Kennzeichen, Flug-Nr, ISBN
Kunden-Nr = Branche + fortlaufende Nr.
BfA-Nr = 1.Buchstabe NN + Geburtsdatum + fortlaufende Nr.

Prüfung des vorliegenden Datenmodells

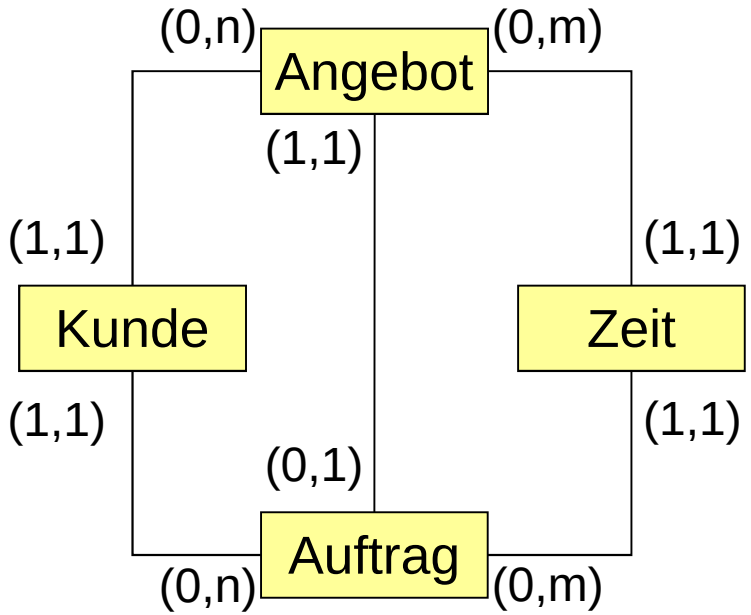
In ERM- bzw. UML-Notation

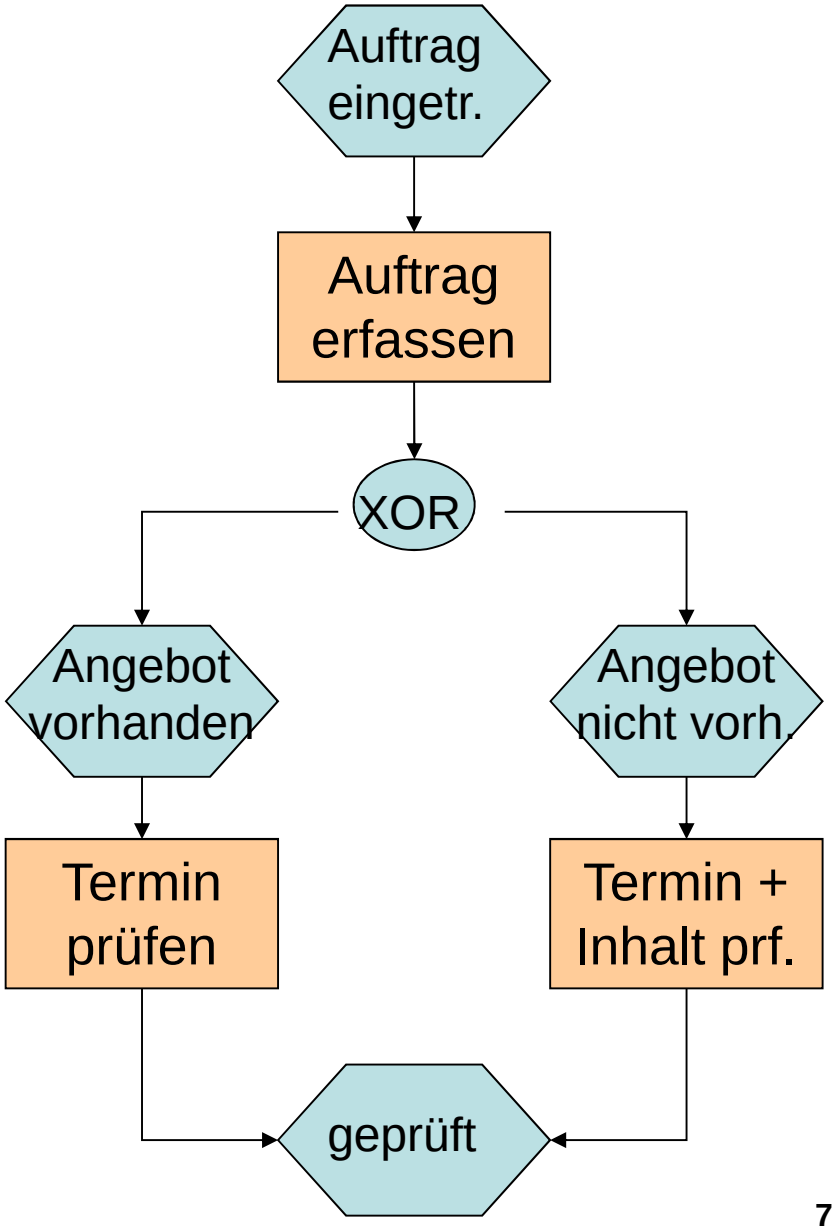
1. Prüfung der Vollständigkeit und der Verständlichkeit der Objekttyp-, Beziehungstyp- und Schlüsselattribut-Beschreibungen.
2. Prüfung der Bedeutung von Objekttypen, Beziehungstypen und Attributen mit Fachspezialisten.
3. Vergleich des Datenmodells mit den Funktionen des analysierten Systems (Methodenkonsistenz, Funktions-Daten-Matrix).

Kritische / ungültige Beziehungen:

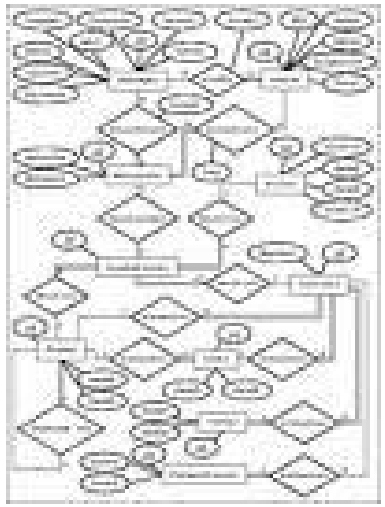
(1,1) - (1,1)
(1,n) - (1:m)
rekursive (1,1) - (0,1), (1,n) - (0,1), (1,n) - (1,m) Beziehungen

Beispiel: Vergleich EPK - ERM





Probleme bei der ER-Notation



ER- bzw. UML-Notation bietet keine Eindeutigkeit bei der Modellbildung

- bzgl. graphischer Anordnung der Symbole
- bzgl. der Detaillierungstiefe des Modells
- bzgl. der Abgrenzung zwischen Objekt- und Beziehungstypen.

Die resultierenden individuellen Unterschiede in der Modellierung erschweren den Vergleich und die Interpretation von Modellen, die Einbindung von Abstimminstanzen zur Überprüfung fachlicher Korrektheit und die generelle Akzeptanz.

Hinzu kommt: Mehrere Schreibweisen
(Chen- bzw. umgekehrte Chen-Notation)

Normalisiertes ER-Modell

Normalisiertes ER-Modell (nach Zehnder, Sinz)

1. Nur drei Arten von Beziehungs-Typen:
   (d. h. keine m:n-Beziehungen mehr)
   a) 1-muss : 1-kann
   b) 1-muss : N-muss
   c) 1-muss : N-kann
2. Nur Objekte werden durch Attribute beschrieben
3. Alle Beziehungen sind wenn möglich vom Grad 2
4. Der Objekttyp mit Kardinalität 1 sollte immer links stehen
5. Zur Vermeidung von Nullwerten: Bildung von Sub- bzw. Supertypen (Generalisierung)

Auflösung von Inkonsistenzen und Redundanzen

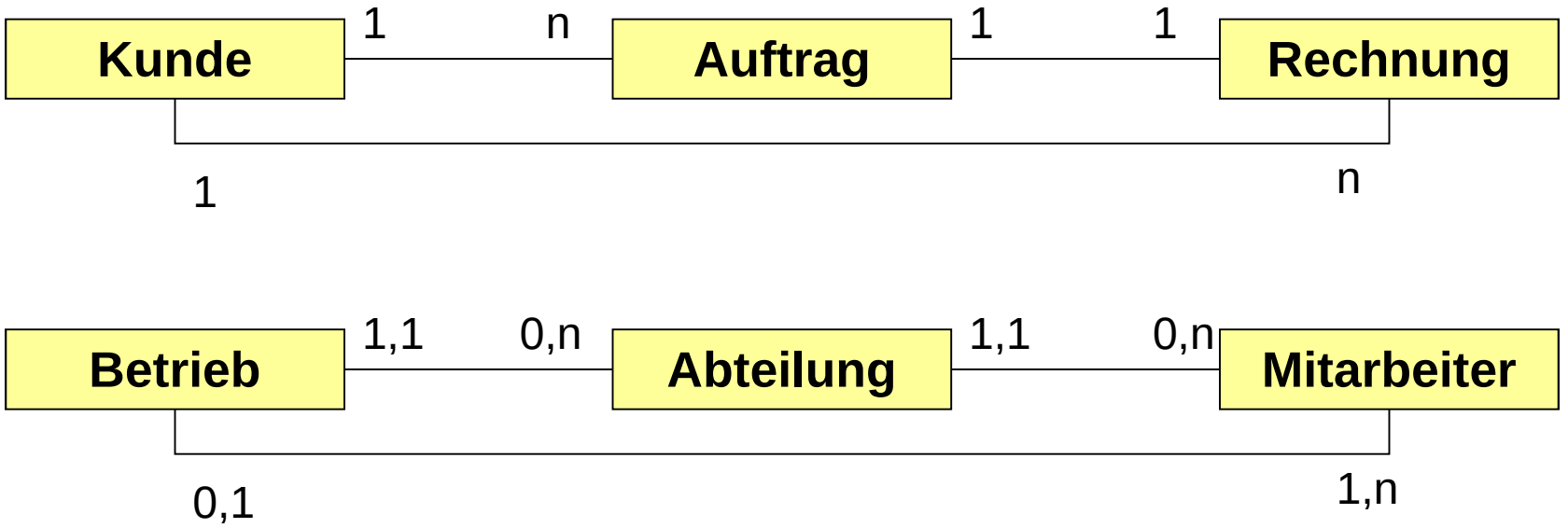
Sind einzelne Beziehungstypen redundant?
Analysiert werden Beziehungstypen, die graphisch Kreise bilden oder parallel sind.

Ableitbare Attributwerte oder Beziehungen können aus anderen gespeicherten Daten berechnet werden. Möglichkeiten zur Behandlung ableitbarer Daten:

- **nur die originären Daten werden in Basistabellen gespeichert,**
- **sowohl die ableitbaren als auch die originären Daten werden gespeichert.**

Ableitbare Daten werden vielfach aus Performance-Gründen gespeichert. Im ER-Modell besonders zu beachten: ableitbare, redundanten Beziehungen

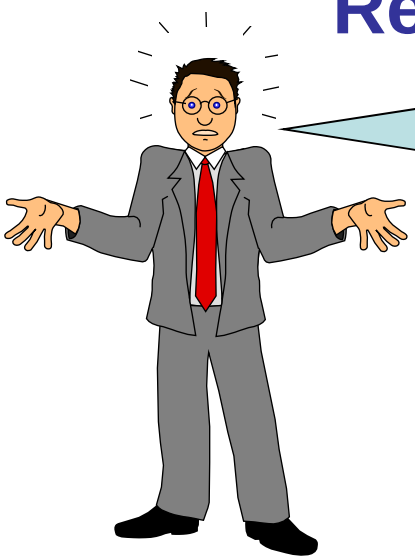
Beispiele



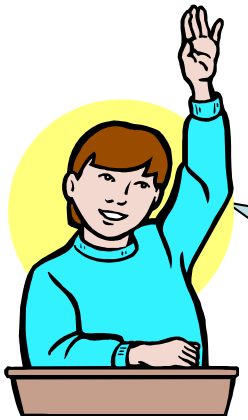
Wenn Mitarbeiter m einer Abteilung a zugeordnet ist und Abteilung a zum Betrieb b gehört, dann ist Mitarbeiter m dem Betrieb b zugeordnet.

Ein Mitarbeiter einer Abteilung eines Betriebes kann dort als Sicherheits-beauftragter bestellt sein.

Redundante Beziehungen



Wie kann man redundante Beziehungsarten (BA) erkennen?



Redundante BA gehen i. d. R. einher mit Verletzungen der 2. bzw. 3. NF. Ist eine lückenlose Folge von gleich-gerichteten 1:n-Beziehungen zu einem Ring 'kurzgeschlossen', dann **kann** es sich um eine redundante BA handeln.

Aber: Nicht jeder Ring enthält eine redundante BA.

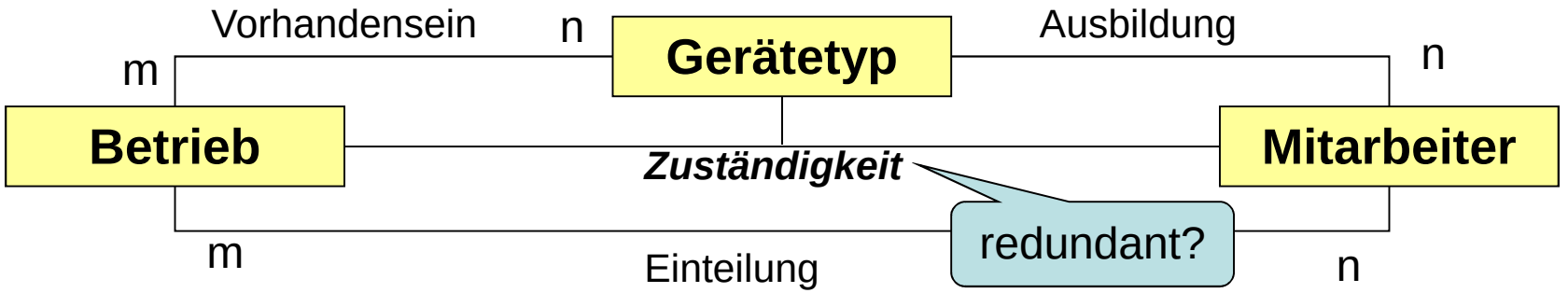
Beispiel: Beziehung Betrieb-Mitarbeiter als Inspektor interpretiert.

Indirekte bzw. redundante Attribute

- Entity-Typ *Mitarbeiter* enthält die Fax-Nr des Betriebs
- Auftrag enthält die Anzahl der Auftragspositionen
- Beim Kunden wird die Summe seiner Rechnungsbeträge gespeichert.

Die Liste ließe sich beliebig verlängern.

Übung



Ein Mitarbeiter m ist im Betrieb b für den Gerätetyp g genau dann zuständig, wenn
- m für den Einsatz im Betrieb b eingeteilt ist und
- m am Gerätetyp g ausgebildet ist und
- im Betrieb b Geräte des Typs g vorhanden sind.

Ein Mitarbeiter m ist im Betrieb b für den Gerätetyp g genau dann zuständig, wenn
- m für den Einsatz im Betrieb b eingeteilt ist und
- m am Gerätetyp g ausgebildet ist und
- im Betrieb b Geräte des Typs g vorhanden sind und
- Mitarbeiter m mindestens 1 Jahr im Unternehmen tätig ist.

Ein Mitarbeiter m kann im Betrieb b für den Gerätetyp g zuständig sein, wenn
- m für den Einsatz im Betrieb b eingeteilt ist und
- m am Gerätetyp g ausgebildet ist und
- im Betrieb b Geräte des Typs g vorhanden sind.

Fallstudie „Automobil-Meldewesen“

Während der Lebensdauer eines Automobils ist der Typ, der jeweilige Besitzer (das kann ein Händler oder eine Person sein) und der Hersteller zu speichern. Es gibt folgende Relationen:

Hersteller (HerstellerNr, Herstellername, ...)
Auto (SerienNr, Herstelljahr, Zulassungsdatum, Besitzer, ModellNr, HerstellerNr, PS, Hubraum, ...)
Modell (ModellNr, Bezeichnung, HerstellerNr, ...)
Leistung (ModellNr, Hubraum, PS, Kmh)
Händler (HändlerNr, Firmenname, Anzahl_Autos, ...)
Person (PersNr, Name, ...)

1. Erstellen Sie dazu ein ERM
2. Wie lautet der Primärschlüssel der Relation Leistung?
3. Gibt es bei diesem Relationsschema Redundanzen?
4. Wo ist das DB-Schema evtl. unsauber modelliert?
5. Welche Fremdschlüsselbeziehungen existieren?
6. Welche Integritätsbedingungen sind notwendig?

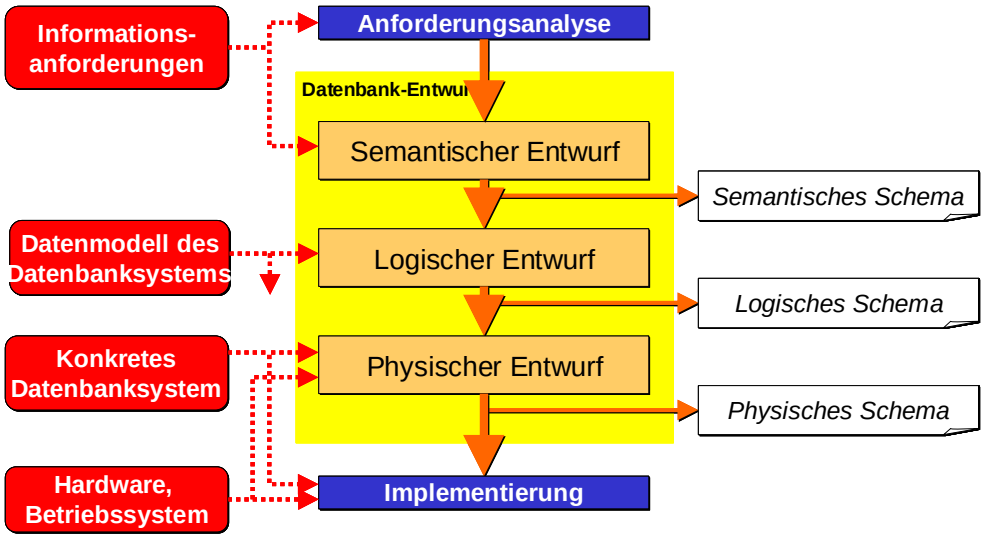
Integritätsbedingungen

1. Primärschlüsselbedingungen
2. Fremdschlüsselbeziehungen
3. Zulassungsdatum ≥ Herstelljahr
4. Besitzer ≠ NULL
5. Besitzerwechsel nur von Händler an Person oder
   Person an Händler oder Person
6. Modell- und SerienNr eines Autos dürfen nicht geändert werden.
7. Modell- und HerstellerNr bei Auto und Modell müssen übereinstimmen.
8. Anzahl_Autos beim Händler muss stimmen.
9. Herstellername muss eineindeutig sein.
10. PS und Hubraum beim Auto müssen ein für das Modell gültiges Wertepaar sein.

Logisches Datenbankdesign

Umsetzung des semantischen (konzeptionellen) Entwurfs in das Datenmodell des verwendeten DBMS.

Hierarchisches Datenbankmodell
Netzwerkartiges Datenbankmodell
Relationales Datenbankmodell
Objektrelationales Datenbankmodell
Objektorientiertes Datenbankmodell



Das bedeutet für ein RDBMS: **Relationsschema erstellen**
Tabellen, Attribute, PK, FK
Normalisierung

Abbildung von Domänen

Jede im ER-Schema auftauchende **atomare** Domäne D_{ER} wird auf eine geeignete Domäne D_R des relationalen Modells abgebildet. Z. B. String(25) auf Varchar(25)

Nicht-atomare Domänen:

Nicht-atomare Domänen müssen transformiert werden.

Einfachster Ansatz: Die Domänen werden direkt auf einzelne, atomare Attribute abgebildet.

Beispiel: (**Kartographie):**

Domäne Kreis als Kombination aus Mittelpunkt und Radius

Tabellen: eigene Relation mit FK-Beziehung

Mengen: eigene Relation mit FK-Beziehung

Abbildung von Entitäten

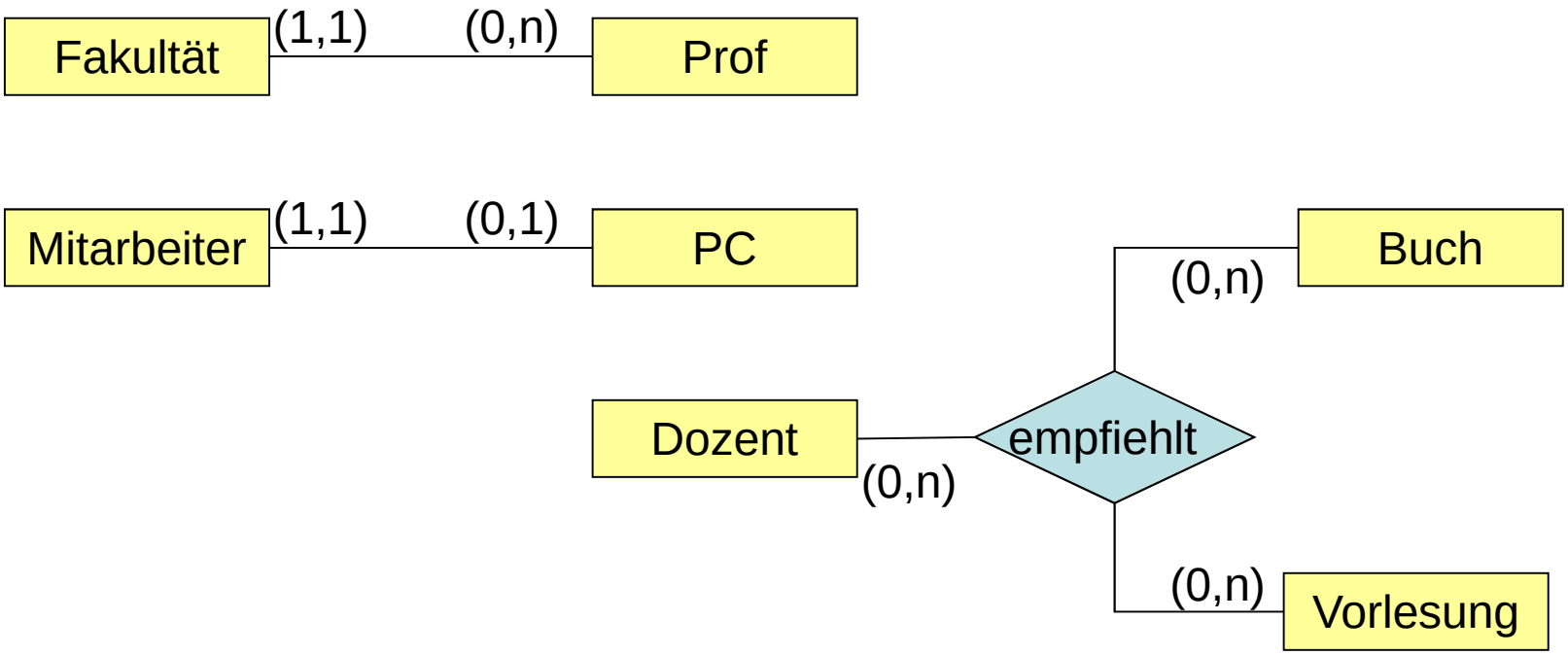
Jede (starke) Entität wird auf eine Relation abgebildet

- Schlüssel der Entität wird auch Schlüssel der Relation (können aus mehreren Attributen zusammengesetzt sein!)
- Falls im ER-Modell mehrere Schlüssel vorhanden sind, wird ein *Primärschlüssel* für die Relation gewählt
- Alle anderen Schlüsselkandidaten werden als *unique* def.
- Überprüfung auf Einhaltung der geforderten Normalform (3. NF, BCNF, 4. NF)

Abbildung von Beziehungen

Beziehungen im ER-Modell werden abgebildet auf

- Fremdschlüsselbeziehung oder Beziehungsrelation
- Attribute der Beziehung werden zu Attributen der Relation
- Und was passiert bei dreistelligen Beziehungen?



Und wie geht‘s weiter?

- Kardinalitäten z. B. (**1**, x)
- ON DELETE {cascade, restrict, set null, set default}
- Prozeduren in der Datenbank

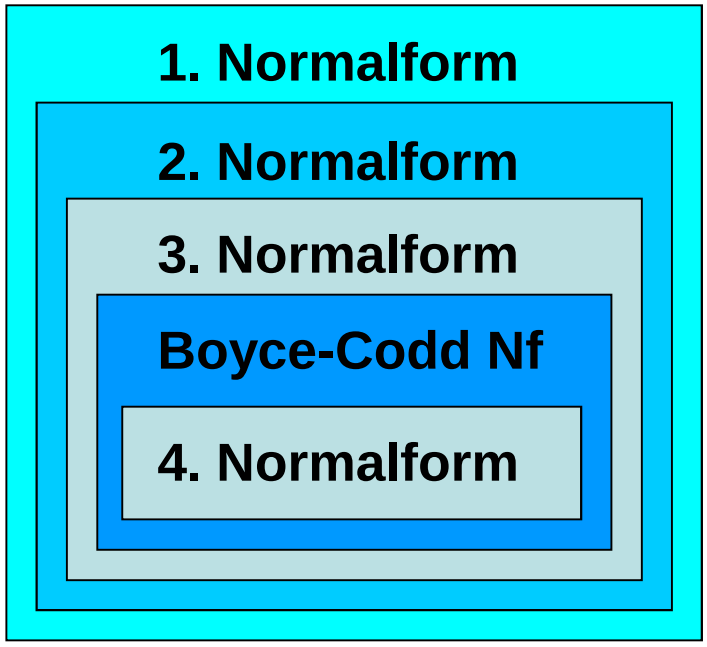
⇒ Das geht erst im physischen Design, da DBMS-abhängig

⇒ Create table bei Oracle, PL/SQL (für Trigger, SP)

Normalisierung

Es gibt mehrere Gründe, warum Normalisierung wichtig ist:

- Verständlicheres Datenmodell für Anwender und Entwickler
- Vermeidung von Anomalien beim Einfügen, Löschen oder Ändern von Daten!
- Eliminierung von Redundanzen
- Robusteres Datenmodell gegenüber Änderungen oder Erweiterungen
- Korrekte Abbildung der Realität
- man kann sich mal so richtig schön systematisch mit den Daten beschäftigen...

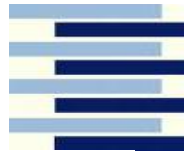


- Allerdings: nicht immer Normalisierung (z. B. DWH)

Boyce-Codd Normalform

- Eine Relation ist in Boyce-Codd Normalform (BCNF), wenn für jede funktionale Abhängigkeit FD: $X \rightarrow Y$ mindestens eine der beiden Bedingungen gilt:
  1. $Y \subseteq X$ (d.h. die FD ist trivial)
  2. X ist ein Schlüsselkandidat (der nicht minimal sein muss)
- Die BCNF ist verletzt, wenn es eine funktionale Abhängigkeit gibt, deren linke Seite keine Schlüsseleigenschaft hat
- Die BCNF ist nicht in jeden Fall abhängigkeitserhaltend erzeugbar

Beispiel 1 zur BCNF



Gegeben sei die Relation

Stadt(Ort, Bundesland, Chef, Einwohner)

für die die folgenden FD‘s gelten:

(Ort, Bundesland) → Einwohner
Bundesland → Chef
Chef → Bundesland

Schlüsselkandidaten:

- (Ort, Bundesland)
- (Ort, Chef)

Diese Relation Stadt ist in 3. NF, aber nicht in BCNF

Dekomposition (verlustfrei und abhängigkeitserhaltend)

Stadt1(Ort, Bundesland, Einwohner)
Regiert(Bundesland, Chef)

Beispiel 2 zur BCNF

Relation **PLZBuch (Ort, Bundesland, Straße, PLZ)**

Es gelten die FD‘s

PLZ → (Ort, Bundesland)

(Straße, Ort, Bundesland) → PLZ

Diese Relation ist ebenfalls in 3. NF, aber nicht in BCNF.

Die Zerlegung in die zwei Relationen

Straßen (PLZ, Straße)

Orte (PLZ, Ort, Bundesland)

mit den FD‘s für Straßen: (PLZ, Straße) → (PLZ, Straße)

für Orte: PLZ → (Ort, Bundesland)

erzeugt die BCNF, hat aber Probleme ...

Beispiel 2 …

PLZBuch			
Ort	**Bundesland**	**Straße**	**PLZ**
Oldenburg	Niedersachsen	Goethestraße	26123
Oldenburg	Niedersachsen	Gildestraße	26133
Oldenburg	Schleswig-H.	Goethestraße	15234

$\Pi_{PLZ,Straße}$

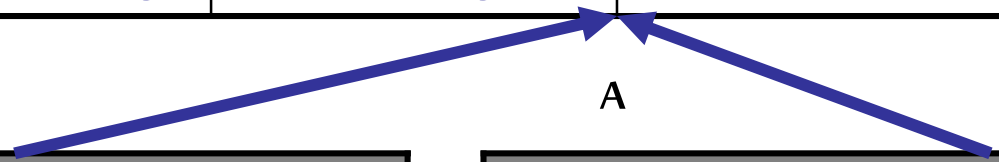
$\Pi_{Ort,Bland,PLZ}$

Straßen	
PLZ	**Straße**
15234	Goethestraße
26123	Goethestraße
26133	Gildestraße
23755	Goethestrasse

Orte		
Ort	**Bundesland**	**PLZ**
Oldenburg	Niedersachsen	26123
Oldenburg	Niedersachsen	26133
Oldenburg	Schleswig-H.	15234
Oldenburg	Schleswig-H.	23755

Beispiel 2 …

PLZBuch			
Ort	**Bundesland**	**Straße**	**PLZ**
Oldenburg	Niedersachsen	Goethestraße	26123
Oldenburg	Niedersachsen	Gildestraße	26133
Oldenburg	Schleswig-H.	Goethestraße	15234
Oldenburg	Schleswig-H.	Goethestraße	23755



A

Straßen	
PLZ	**Straße**
15234	Goethestraße
26123	Goethestraße
26133	Gildestraße
23755	Goethestrasse

Orte		
Ort	**Bundesland**	**PLZ**
Oldenburg	Niedersachsen	26123
Oldenburg	Niedersachsen	26133
Oldenburg	Schleswig-H.	15234
Oldenburg	Schleswig-H.	23755

Mehrwertige Abhängigkeiten

Neben den funktionalen Abhängigkeiten existiert eine weitere wichtige Art von Abhängigkeiten:
Mehrwertige Abhängigkeiten (Multi-valued dependencies (MVD))

Mehrwertige Abhängigkeiten beschreiben die Abhängigkeit einer Menge von Attributwerten von einem anderen Attributwert

Notation: X →→ Y für „Y ist mehrwertig von X abhängig“

Die mehrwertige Abhängigkeit ist Grundlage der vierten Normalform.

Mehrwertige Abhängigkeiten

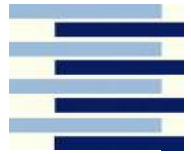
Personalprofil		
PersonalNr	**Sprache**	**Programmierspr**
4711	englisch	C
4711	plattdeutsch	Java
4711	englisch	Java
4711	plattdeutsch	C
0815	deutsch	Prolog

Mehrwertige Abhängigkeiten dieser Relation:

- {PersonalNr} →→ {Sprache} und
- {PersonalNr} →→ {Programmierspr}

Hintergrund: Die Werte von *Sprache* zu einer bestimmten PersonalNr hängen nur von dieser Nr. und nicht von den Programmiersprachen ab (und umgekehrt).

Mehrwertige Abhängigkeiten



Seien x, y, z Attributmengen einer Relation r mit $z = r - (x \cup y)$, so gilt $x \rightarrow\rightarrow y$ genau dann wenn:

- Wenn es zwei Tupel t1 und t2 mit gleichen x–Werten gibt
- Dann muss es auch zwei Tupel t3 und t4 geben mit
  - $\pi[x]\,(t1) = \pi[x]\,(t2) = \pi[x]\,(t3) = \pi[x]\,(t4)$ und
  - $\pi[y]\,(t3) = \pi[y]\,(t1)$ und $\pi[y]\,(t4) = \pi[y]\,(t2)$ und
  - $\pi[z]\,(t3) = \pi[z]\,(t2)$ und $\pi[z]\,(t4) = \pi[z]\,(t1)$

Wenn also zwei Tupel mit unterschiedlichen Y-Werten existieren, die aber die gleichen X-Werte aufweisen, so müssen die Y-Werte in unterschiedlichen Tupeln mit jedem unterschiedlichen Z-Wert kombiniert werden, die auch den gleichen X-Wert haben.

Die vierte Normalform

Eine Relation r ist in vierter Normalform, wenn sie in dritter Normalform ist und es keine mehrwertigen Abhängigkeiten (außer trivialen) gibt.

Eine MVD x →→ y ist trivial genau dann, wenn

- $y \subseteq x$ oder
- $y = r - x$ (es also kein z gibt)

MVD treten paarweise auf oder sind trivial.

Die vierte Normalform

Personalprofil		
PersonalNr	**Sprache**	**Programmierspr**
4711	englisch	C
4711	plattdeutsch	Java
4711	englisch	Java
4711	plattdeutsch	C
0815	deutsch	Prolog

Mehrwertige Abhängigkeiten dieser Relation:

{PersonalNr} →→ {Sprache} und

{PersonalNr} →→ {Programmierspr}

Also ist diese Relation nicht in 4. NF und wird umgewandelt.

Die vierte Normalform

Personalprofil		
Personalnr	**Sprache**	**Programmierspr**
4711	englisch	C
4711	plattdeutsch	Java
4711	englisch	Java
4711	plattdeutsch	C
0815	deutsch	Prolog

π[Personalnr, Sprache]

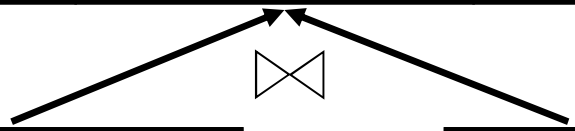
Sprachen	
Personalnr	**Sprache**
4711	englisch
4711	plattdeutsch
0815	deutsch

π[Personalnr, Programmierspr]

Sprachen	
Personalnr	**Programmierspr**
4711	C
4711	Java
0815	Prolog

Die vierte Normalform

Personalprofil		
Personalnr	**Sprache**	**Programmierspr**
4711	englisch	C
4711	plattdeutsch	Java
4711	englisch	Java
4711	plattdeutsch	C
0815	deutsch	Prolog



Sprachen	
Personalnr	**Sprache**
4711	englisch
4711	plattdeutsch
0815	deutsch

Sprachen	
Personalnr	**Programmierspr**
4711	C
4711	Java
0815	Prolog

Zusammenfassung

- Die ersten drei Normalformen sind aus der Veranstaltung „Datenbanken“ schon bekannt
- Für alle erzeugten Normalformen gibt es mit der Verlustlosigkeit und Abhängigkeitstreue zwei anzustrebende Kriterien
- Die BCNF ist verlustlos zu erzeugen, aber nicht immer abhängigkeitstreu
- Während Grundlage der ersten drei NF die funktionalen Abhängigkeiten sind, existiert mit den mengenwertigen Abhängigkeiten eine zusätzliche Art von Integritätsbedingung
- Die vierte Normalform basiert auf den mengenwertigen Abhängigkeiten
- In der Praxis wird die vierte Normalform derzeit nicht genutzt
- Es existiert noch eine fünfte Normalform, die in dieser Veranstaltung nicht behandelt wird
- Bei Interesse: Es existiert eine schöne Theorie mit diversen Lehrbüchern rund um den relationalen Datenbankentwurf