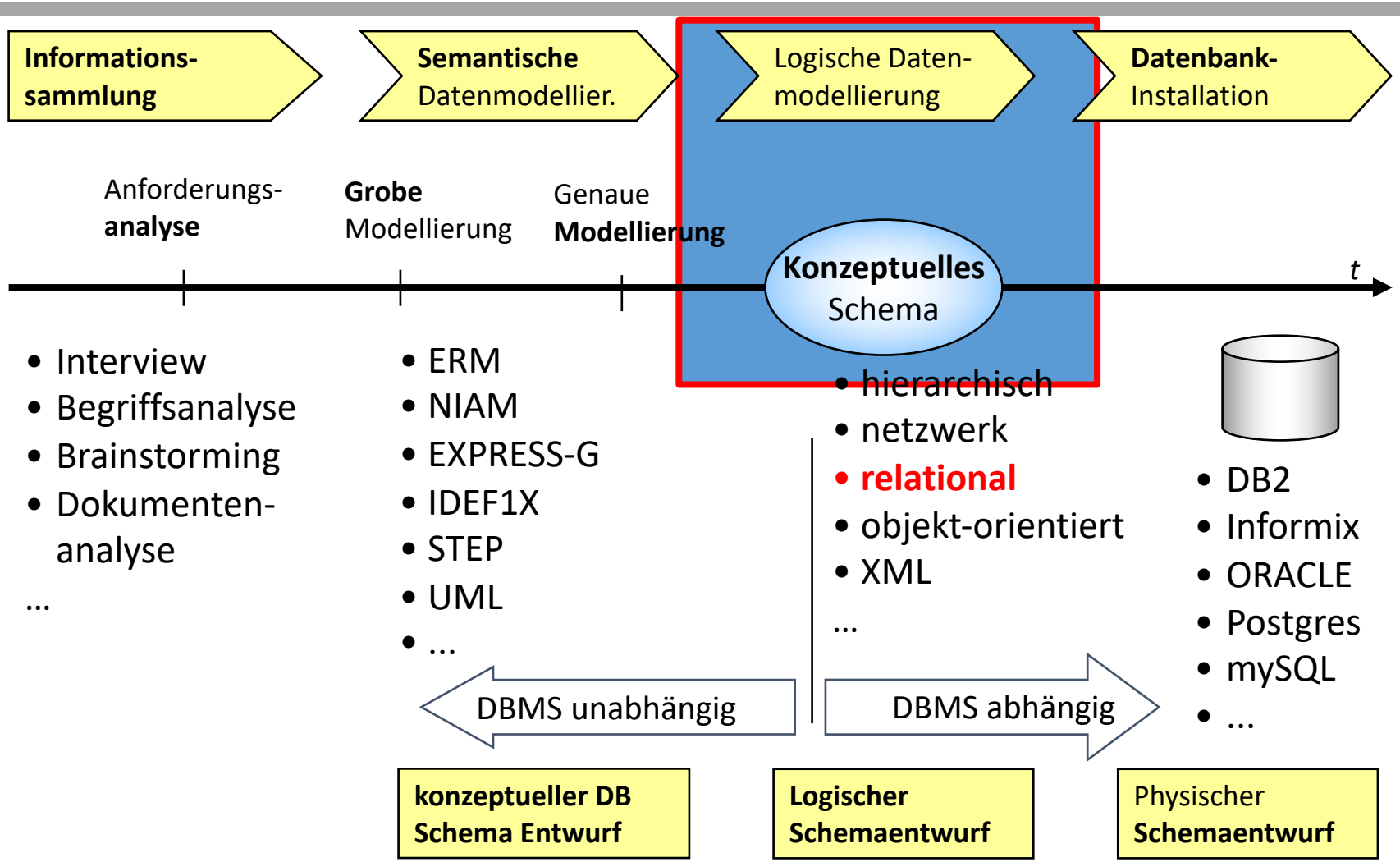
MOBI-DBS-B: Datenbanksysteme Datenbankentwurf

Vorlesung Sommersemester 2019

Tanya Braun, Universität zu Lübeck

Lehrauftrag SoSe 19, Universität Bamberg

Die Phasen des DB-Entwurfs



Datenbankentwurf

Inhalte

- Qualität von Schemata
- Probleme in Schemata
- Funktionale Abhängigkeiten
- Normalformen
- Abhängigkeitswahrung
- Nicht-additiver Join
- Wenn die Zeit es zulässt: Datenqualität

Kompetenzen

- Schlechte relationale Schemata erkennen und reparieren können
- Aus gegebenen Abhängigkeiten relationale Schemata bestimmter Güte erstellen können

Bezug zu Phasen des DB-Entwurfs

Qualität und Problembereiche

Relationenschemata

Qualität von DB-Schemata

- Korrektheit
  - Entitäten, Beziehungen und Attribute entsprechen denen der Miniwelt
- Vollständigkeit
  - Miniwelt (=relevanter Ausschnitt) ist vollständig enthalten
- Minimalität
  - Konzepte der Miniwelt sind möglichst redundanzfrei enthalten
- Lesbarkeit
  - Schema ist übersichtlich und systematisch aufgebaut

… und wie misst man das?

→ Normalformen

„Schlechte“ Relation

ANAME	SSN	GDATUM	ADRESSE	ABTNUMMER	ABT	AMGRSSN
Smith, John B.	123456789	1965-01-09	731 Fondren, Houston TX	5	Research	333445555
Wong, Franklin T.	3334445555	1955-12-08	638 Voss, Houston TX	5	Research	333445555
Zelaya, Alicia J.	9998887777	1968-07-19	3321 Castle,Spring, TX	4	Administration	987654321
Wallace, Jennifer S.	987654321	1941-06-20	291, Berry, Bellaire, TX	4	Administration	987654321
Narayan, Ramesh K.	666884444	1962-09-15	975, FireOak, Humble, TX	5	Research	333445555
English, Joyce A.	453453453	1972-07-31	Rice, Houston, TX	5	Research	333445555
Jabber, Ahmad V.	987987987	1969-03-29	Dallas, Houston, TX	4	Administration	987654321
Borg, James E.	888665555	1937-11-10	Stone, Houston, TX	1	Headquarters	888665555

„Schlechte“ Relation: Einfügeanomalie

ANAME	SSN	GDATUM	ADRESSE	ABTNUMMER	ABT	AMGRSSN
Smith, John B.	123456789	1965-01-09	731 Fondren, Houston TX	5	Research	333445555
Wong, Franklin T.	3334445555	1955-12-08	638 Voss, Houston TX	5	Research	333445555
Zelaya, Alicia J.	9998887777	1968-07-19	3321 Castle,Spring, TX	4	Administration	987654321
Wallace, Jennifer S.	987654321	1941-06-20	291, Berry, Bellaire, TX	4	Administration	987654321
Narayan, Ramesh K.	666884444	1962-09-15	975, FireOak, Humble, TX	5	Research	333445555
English, Joyce A.	453453453	1972-07-31	Rice, Houston, TX	5	Research	333445555
Ja						
Borg, James E.	888665555	1937-11-10	Stone, Houston, TX	1	Headquarters	888665555
Grawunder, M	007	1971-02-17	Barßel	1	**Hädquarters**	**886665555**
				42	Geheim	007

neuer Angestellter → Information zu Abteilung (konsistent!) → Tippfehler?

keine neue Abteilung ohne Mitarbeiter

„Schlechte“ Relation: Update-Anomalie

ANAME	SSN	GDATUM	ADRESSE	ABTNUMMER	ABT	AMGRSSN
Smith, John B.	123456789	1965-01-09	731 Fondren, Houston TX	5	Research	732456732
					earch	732456732
					inistration	987654321
Wallace, Jennifer S.	987654321	1941-06-20	291, Berry, Bellaire, TX	4	Administration	987654321
Narayan, Ramesh K.	666884444	1962-09-15	975, FireOak, Humble, TX	5	Research	**333445555**
English, Joyce A.	453453453	1972-07-31	Rice, Houston, TX	5	Research	732456732
Jabber, Ahmad V.	987987987	1969-03-29	Dallas, Houston, TX	4	Administration	987654321
Borg, James E.	888665555	1937-11-10	Stone, Houston, TX	1	Headquarters	888665555

Abteilung Research bekommt neuen Leiter ... hoffentlich keine Zeile vergessen ☹

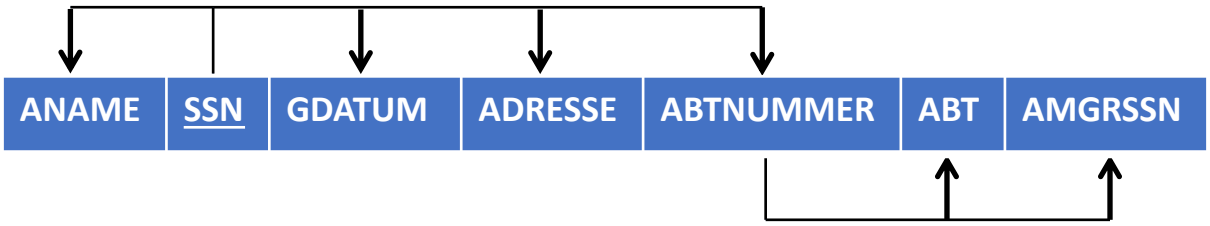
„Schlechte“ Relation: Löschanomalie

ANAME	SSN	GDATUM	ADRESSE	ABTNUMMER	ABT	AMGRSSN
Smith, John B.	123456789	1965-01-09	731 Fondren, Houston TX	5	Research	333445555
Wong, Franklin T.	3334445555	1955-12-08	638 Voss, Houston TX	5	Research	333445555
Zelaya, Alicia J.	9998887777	1968-07-19	3321 Castle,Spring, TX	4	Administration	987654321
Wallace, Jennifer S.	987654321	1941-06-20	291, Berry, Bellaire, TX	4	Administration	987654321
Narayan, Ramesh K.	666884444	1962-09-15	975, FireOak, Humble, TX	5	Research	333445555
English, Joyce A.	453453453	1972-07-31	Rice, Houston, TX	5	Research	333445555
Jabber, Ahmad V.	987987987	1969-03-29	Dallas, Houston, TX	4	Administration	987654321
~~Borg, James E.~~	~~888665555~~	~~1937-11-10~~	~~Stone, Houston, TX~~	~~1~~	~~Headquarters~~	~~888665555~~

James E. Borg geht in den Ruhestand. Es gibt aber noch keinen Nachfolger ... Wo ist die Abteilung Headquarters hin??

„Schlechte“ Relation

- Problem: Kombination aus Angestellten- mit Abteilungsinformation
- Einfüge-Anomalie
  - Neuer Angestellter → Information zu Abteilung (konsistent!)
  - Keine neue Abteilung ohne Mitarbeiter
- Update-Anomalie
  - Änderung des Abteilungsleiters → Update in allen Angestellten
- Lösch-Anomalie
  - Wird der letzte Angestellte einer Abteilung gelöscht, verschwindet auch die Abteilung
- Was fällt auf?
  - Es gibt in der Relation verschiedene Abhängigkeiten!



Normalformen

- Ermöglichen den objektiven Vergleich zwischen DB-Schemata bezüglich der Qualität
  - Hauptziel: Vermeidung von Redundanz
- Ermöglichen automatisierte Verfahren zur Generierung guter DB-Schemata
- „Hartes“ Kriterium: Funktionale Abhängigkeiten

- Ergänzung durch „weiche“ Kriterien:
  - Abgrenzung der Relationenschemata
  - Reduktion redundanter Werte in Tupeln
  - Reduktion der NULL-Werte in Tupeln
  - Vermeidung unechter Tupel

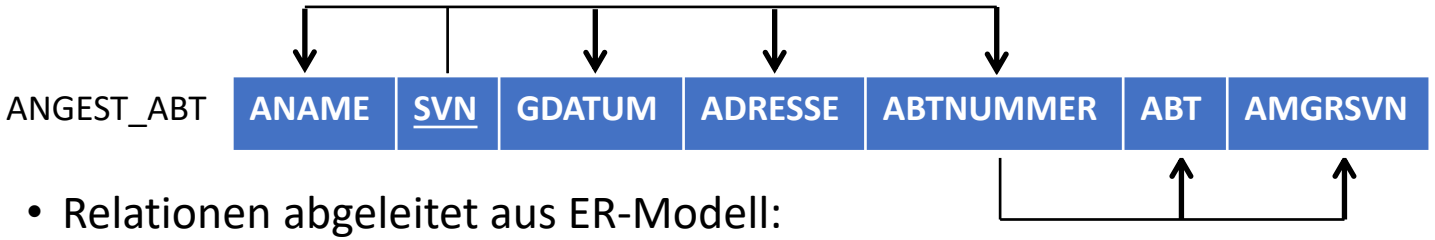
- Gute Qualität: aus validen ER-Diagrammen erstellte DB-Schemata

Funktionale Abhängigkeiten

Relationenschemata

Funktionale Abhängigkeiten

- R = {A_1, …, A_n} sei ein Schema, X und Y seien Attributteilmengen
- FD: X → Y sei eine funktionale Abhängigkeit (functional dependency), wenn ∀ Tupel t_1, t_2 gilt: wenn t1[X] = t2[X] gilt, dann gilt auch t1[Y] = t2[Y]
- Die Werte von X bestimmen also eindeutig die Werte von Y
  - FD heißt trivial, wenn Y ⊆ X
  - Und wenn X ein Schlüssel ist? Dann gilt X → Y für alle möglichen Y aus R
  - Folgt aus X → Y auch Y → X? Nein!
- Für ANGEST_ABT gilt:
  - F1: {SVN} → {ANAME, GDATUM, ADRESSE, ABTNUMMER}
  - F2: {ABTNUMMER} → {ABT, AMGRSVN}



- Relationen abgeleitet aus ER-Modell:

ANGESTELLTE	SozVersNr	Nachn.	Vorn.	Geschlecht	Adresse	Gehalt	GebDatum	AbtNr	Vorges.

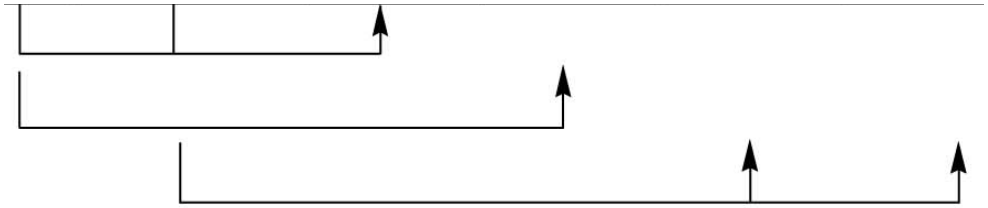
ABTEILUNG	Name	Nummer	Leitung	AnfDatum

Funktionale Abhängigkeiten

- Für ANGEST_PROJ gilt
  - F3: {SSN, PNUMMER} → {STUNDEN}
  - F4: {SSN} → {ANAME}
  - F5: {PNUMMER} → {PNAME, PSTANDORT}

ANGEST_PROJ

SVN	PNUMMER	STUNDEN	ANAME	PNAME	PSTANDORT



- Relationen abgeleitet aus ER-Modell:

ARBEITET_AN

ProjNr	SozVersNr	Stunden

ANGESTELLTE

SozVersNr	Nachn.	Vorn.	Geschlecht	Adresse	Gehalt	GebDatum	AbtNr	Vorges.

PROJEKT

Nummer	Name	Standort	AbtNr

Wo kommen die FDs her?

- Können die nicht einfach aus den Daten abgeleitet werden?
- Beispiel: Tierart → Farbe?

Datum	Tierart	Farbe
12.3.2010	Schwan	weiß
14.3.2010	Fuchs	rot
17.3.2010	Schwan	weiß

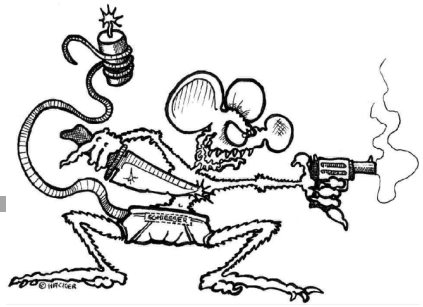
"Not all swans are white."

- FDs können nicht aus den Daten (Extension) abgeleitet werden!
- FDs sind Eigenschaften des Relationenschemas (Intension)
- Sie müssen vom DB-Designer definiert werden

FDs und Inferenzregeln

- Problem: Für große DB-Schemata ist es praktisch unmöglich, alle FDs „von Hand“ zu spezifizieren
- Aber: Aus gegebenen FDs können weitere FDs abgeleitet werden
  - Inferenzregeln!
- Closure / Transitive Hülle F^+:
  - Gegeben eine Menge von FDs F
  - F^+: Menge aller FDs, die mit Inferenzregeln abgeleitet werden können
- Es gibt sechs Inferenzregeln …

Die sechs Inferenzregeln (RATZAP)



- **R**eflexivitätsregel: IR1 : Falls $Y \subseteq X$, dann $X \rightarrow Y$

  Eine Attributmenge bestimmt sich immer selbst oder eine ihrer Teilmengen

- **A**ugmentationsregel: IR2 : Falls $X \rightarrow Y$, dann $XZ \rightarrow YZ$

  Hinzufügen von Attributen auf beiden Seiten führt zu weiterer Regel

- **T**ransitivitätsregel: IR3: Falls $\{X \rightarrow Y, Y \rightarrow Z\}$, dann $X \rightarrow Z$

- **Z**erlegungsregel: IR4: Falls $X \rightarrow YZ$, dann $X \rightarrow Y$ und $X \rightarrow Z$

  Attribute auf der rechten Seite können entfernt und FDs in Teilmengen zerlegt werden

- **A**dditive oder Vereinigungsregel: IR5: Falls $\{X \rightarrow Y, X \rightarrow Z\}$, dann $X \rightarrow YZ$

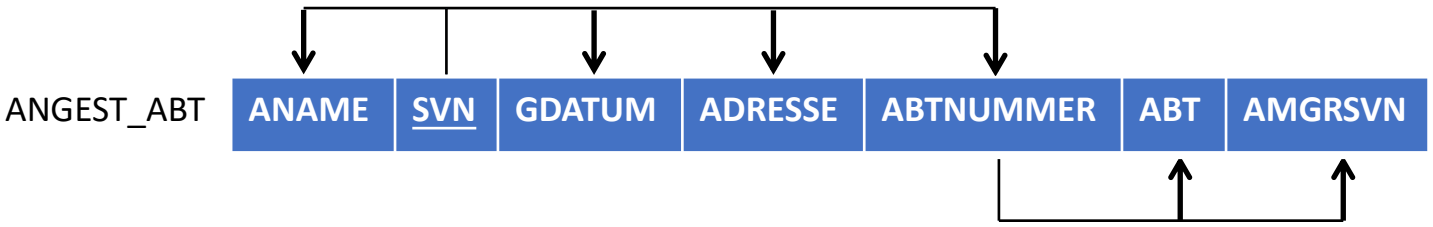
  Gegenstück zu Z: Regel wieder zusammenfassen

- **P**seudotransitive Regel: IR6: Falls $\{X \rightarrow Y, WY \rightarrow Z\}$, dann $WX \rightarrow Z$

  Transitivität im Kontext

Beispiel für die Anwendung von Inferenzregeln

- Regeln
  - IR1: Falls $Y \subseteq X$, dann $X \rightarrow Y$
  - IR2: Falls $X \rightarrow Y$, dann $XZ \rightarrow YZ$
  - IR3: Falls $\{X \rightarrow Y, Y \rightarrow Z\}$, dann $X \rightarrow Z$
  - IR4: Falls $X \rightarrow YZ$, dann $X \rightarrow Y$ und $X \rightarrow Z$
  - IR5: Falls $\{X \rightarrow Y, X \rightarrow Z\}$, dann $X \rightarrow YZ$
  - IR6: Falls $\{X \rightarrow Y, WY \rightarrow Z\}$, dann $WX \rightarrow Z$
- F = { F1, F2 }
  - F1: {SVN} → {ANAME, GDATUM, ADRESSE, ABTNUMMER}
  - F2: {ABTNUMMER} → {ABT, AMGRSVN}
- … zusätzlich können u.a. abgeleitet werden …
  - F3: {SVN} → {ABT, AMGRSVN}
    - IR3
  - F4: {SVN} → {ADRESSE,ABTNUMMER}
    - IR4
  - F5: {SVN} → {SVN}
    - IR1



Weiche Kriterien

Abgrenzung, Reduktion redundanter Werte und NULL-Werte, Vermeidung unechter Tupel

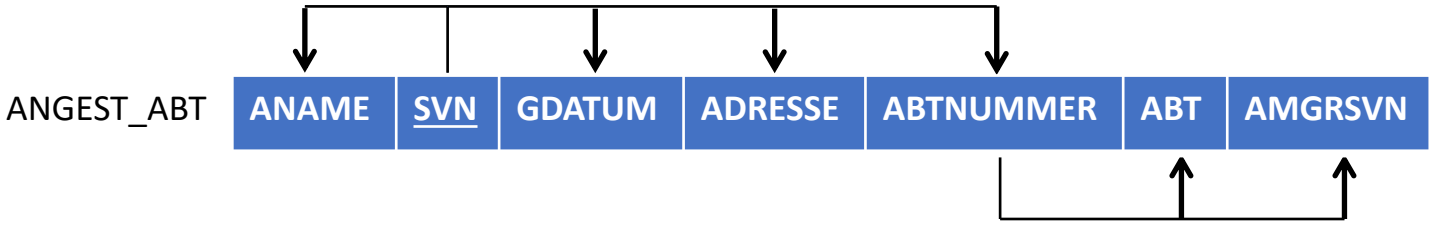
Entwurfsempfehlungen für Relationenschemata

- Jede Relation sollte eine klar abgrenzbare Bedeutung haben
  - Keine Kombination von Entitäten in einer Relation
  - Positiv formuliert: jede Relation entspricht einer Entität
    - die Methode E(E)R→ Relationales Modell erzeugt "gute" DB-Schemata

ANGESTELLTE	SozVersNr	Nachn.	Vorn.	Geschlecht	Adresse	Gehalt	GebDatum	AbtNr	Vorges.

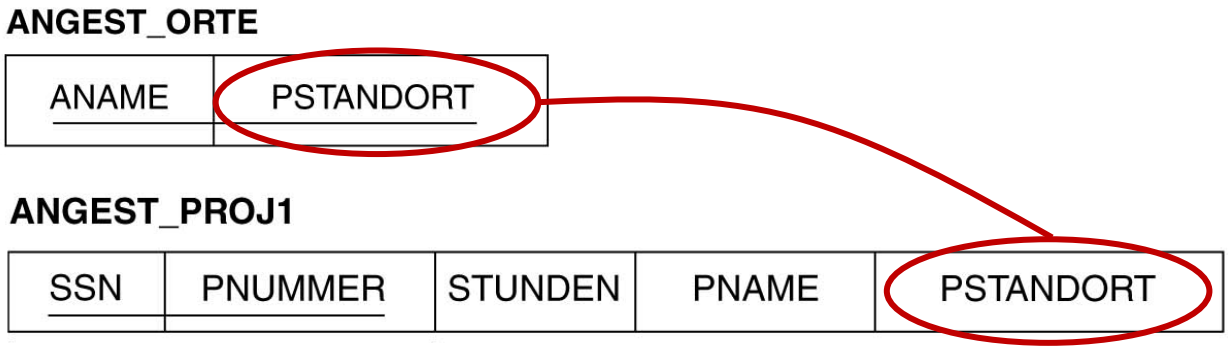
ABTEILUNG	Name	Nummer	Leitung	AnfDatum

- Warum ist das gut?
  - Vermeidung von Redundanzen → Vermeidung von Anomalien
  - Vermeidung von NULL-Werten
- Schlechtes Beispiel:
  - Kombination aus Angestellten-Informationen mit Abteilungsinformation



Keine unechten Tupel

- Aus Equijoins über Nicht-Schlüssel-Attribute sollen keine unechten (spurious) Tupel entstehen
  - Vermeidung gleichlautender Attribute, die keine Schlüssel sind
  - Wenn dies unvermeidbar ist: kein Join darüber
- Beispiel:
  - ANGEST_ORTE * ANGEST_PROJ1 (Natural Join)



Beispiel

ANGEST_PROJ1

SSN	PNUMMER	STUNDEN	PNAME	PSTANDORT
123456789	1	32.5	Product X	Bellaire
123456789	2	7.5	Product Y	Sugarland
666884444	3	40.0	Product Z	Houston
453453453	1	20.0	Product X	Bellaire
453453453	2	20.0	Product Y	Sugarland
333445555	2	10.0	Product Y	Sugarland
333445555	3	10.0	Product Z	Houston
333445555	10	10.0	Computerization	Stafford
333445555	20	10.0	Reorganization	Houston
999887777	30	30.0	Newbenefits	Stafford
999887777	10	10.0	Computerization	Stafford
987987987	10	35.0	Computerization	Stafford
987987987	30	5.0	Newbenefits	Stafford
987654321	30	20.0	Newbenefits	Stafford

ANGEST_ORTE

ANAME	PSTANDORT
Smith, John B.	Bellaire
Smith, John B.	Sugarland
Narayan, Ramesh K.	Houston
English, Joyce A.	Bellaire
English, Joyce A.	Sugarland
Wong, Franklin T.	Sugarland
Wong, Franklin	Houston
Wong, Franklin	Houston
Zelaya, Alicia J	
Jabbar, Ahmad	
Wallace, Jenni	
Wallace, Jenni	
Borg, James E	

	SSN	PNUMMER	STUNDEN	PNAME	PSTANDORT	ANAME
	123456789	1	32.5	ProductX	Bellaire	Smith,John B.
*	123456789	1	32.5	ProductX	Bellaire	English,Joyce A.
	123456789	2	7.5	ProductY	Sugarland	Smith,John B.
*	123456789	2	7.5	ProductY	Sugarland	English,Joyce A.
*	123456789	2	7.5	ProductY	Sugarland	Wong,Franklin T.
	666884444	3	40.0	ProductZ	Houston	Narayan,Ramesh K.
*	666884444	3	40.0	ProductZ	Houston	Wong,Franklin T.
*	453453453	1	20.0	ProductX	Bellaire	Smith,John B.
	453453453	1	20.0	ProductX	Bellaire	English,Joyce A.
*	453453453	2	20.0	ProductY	Sugarland	Smith,John B.
	453453453	2	20.0	ProductY	Sugarland	English,Joyce A.
*	453453453	2	20.0	ProductY	Sugarland	Wong,Franklin T.
*	333445555	2	10.0	ProductY	Sugarland	Smith,John B.
*	333445555	2	10.0	ProductY	Sugarland	English,Joyce A.
	333445555	2	10.0	ProductY	Sugarland	Wong,Franklin T.
*	333445555	3	10.0	ProductZ	Houston	Narayan,Ramesh K.
	333445555	3	10.0	ProductZ	Houston	Wong,Franklin T.
	333445555	10	10.0	Computerization	Stafford	Wong,Franklin T.
*	333445555	20	10.0	Reorganization	Houston	Narayan,Ramesh K.
	333445555	20	10.0	Reorganization	Houston	Wong,Franklin T.

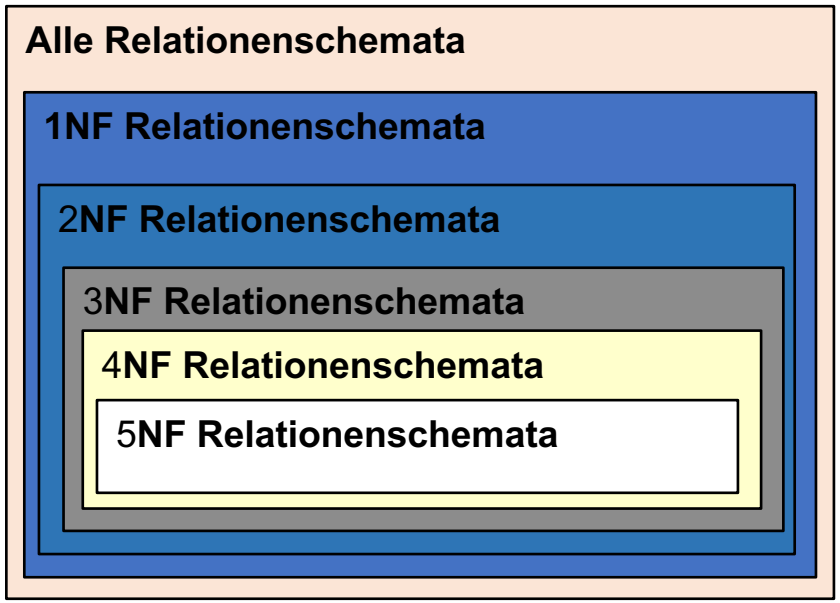
* unechte Tupel

Normalformen

Relationenschemata

Normalformen - Historie

- Normalisierungsverfahren: 1972 von Codd vorgeschlagen
  - unterzieht DB-Schemata einer Reihe von Tests, ob sie einer bestimmten NF genügen
  - ursprünglich 1. – 3. Normalform (1NF, 2NF, 3NF)
  - dann Verschärfung der 3. NF: Boyce Codd Normalform (BCNF)
  - später: 4. und 5. Normalform
- NFs enthalten einander:



Prozess der Normalisierung

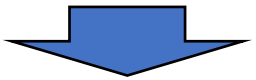
- Ziele:
  - Redundanzfreiheit oder zumindest kontrollierte Redundanz herstellen
  - Einfüge-/Lösch-/Update-Anomalien eliminieren oder zumindest reduzieren
- Normalisierung bietet:
  - Normalformbedingungen, um NF zu testen
  - Vorgehen, um NF zu erreichen
- Grundsätzliches Vorgehen:
  - Prüfung eines Relationenschemas auf eine Normalform
  - Wenn nicht erfüllt: Zerlegung in neue Relationenschemata, bis die gewünschte Normalform erreicht ist
- Vorsicht:
  - Normalisierung alleine garantiert noch keinen guten DB-Entwurf!
    - Nach Zerlegung: keine unechten Tupel durch NATURAL JOIN
    - Möglichst alle funktionalen Abhängigkeiten erhalten

1. Normalform

- Ein Relationenschema R ist in 1. Normalform (1NF), wenn die Domänen der Attribute von R ausschließlich atomare Werte (einfache bzw. unteilbare Werte) enthalten.
  - Erzeugung: Erstelle für jedes nicht-atomare Attribut oder die verschachtelten Relationenschemata ein neues Relationenschema

ABTEILUNG

ABT	ABTNUMMER	AMGRSSN	ASTANDORT
Research	5	333445555	{Bellaire, Sugarland, Houston}
Administration	4	987654321	{Stafford}
Headquarters	1	888665555	{Houston}



ABTEILUNG

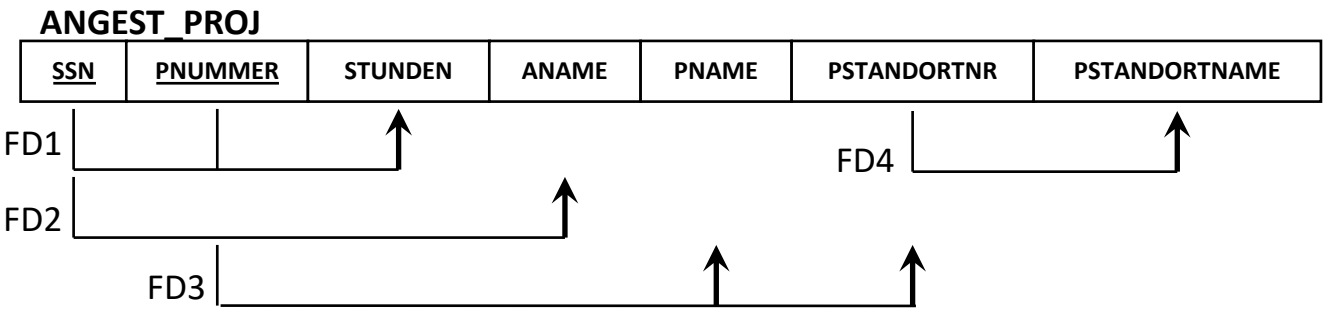
ABT	ABTNUMMER	AMGRSSN
Research	5	333445555
Administration	4	987654321
Headquarters	1	888665555

ABTSTAN

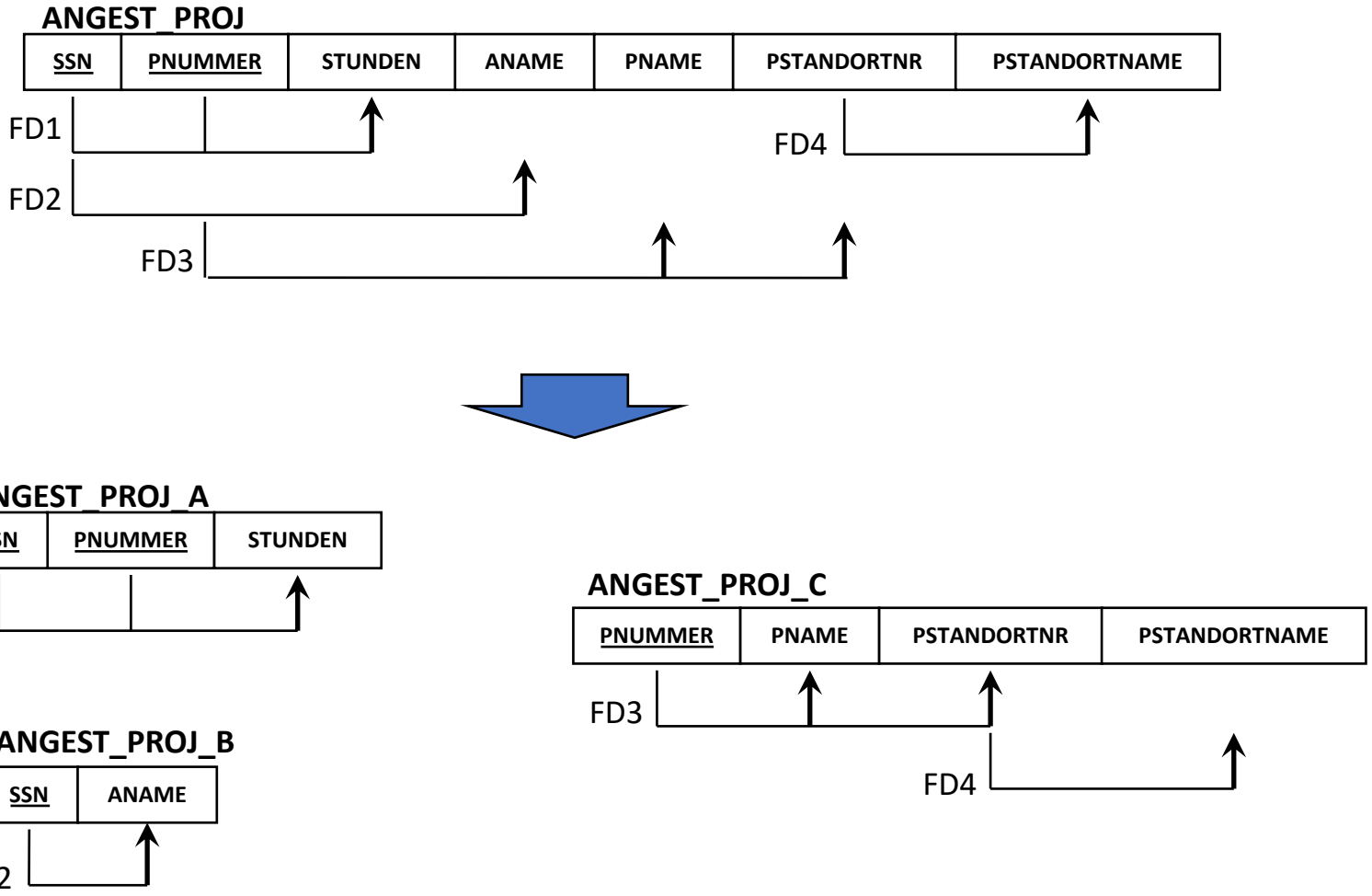
ABTNUMMER	ASTANDORT
5	Bellaire
5	Sugarland
5	Houston
4	Stafford
1	Houston

2. Normalform

- Ein Relationenschema R ist in 2. Normalform (2NF), wenn es 1NF ist und kein nicht-primes Attribut von nur einem Teil des Primärschlüssels abhängt.
- Erzeugung:
  - Zerlege das Relationenschema und erstelle ein neues für jeden partiellen Schlüssel mit seinen abhängigen Attributen
  - Erhalte das (restliche) Relationenschema mit dem ursprünglichen Primärschlüssel und Attributen, die von diesem voll funktional abhängig sind.

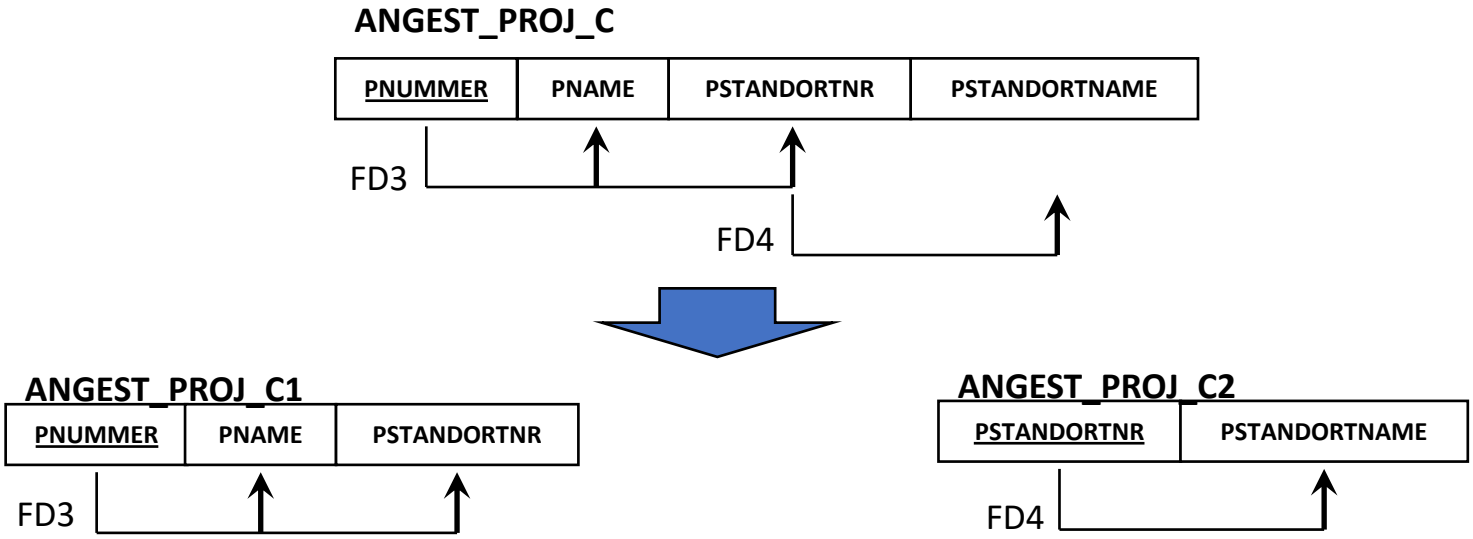


2.Normalform - Beispiel



3. Normalform

- Ein Relationenschema R ist in 3. Normalform (3NF), wenn es 2NF ist und kein nicht-primes Attribut von R transitiv vom Primärschlüssel abhängt.
- Erzeugung:
  - Zerlege das Relationenschema und erstelle ein neues, welches das nicht-prime Attribut bzw. die nicht-primen Attribute beinhaltet, die funktional von anderen nicht-primen Attributen bestimmt werden.



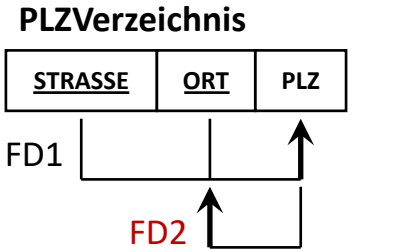
Probleme in der 3NF

- Die meisten 3NF-Schemata weisen in der Praxis keine dramatischen Probleme auf
- Trotzdem nicht frei von problematischen Eigenschaften
- Beispiel: **Kunde(KNr, KName, Adr, VkNr, VkName)**
  - Abhängigkeiten: KNr $\rightarrow$ KName, Adr, VkNr und VkNr $\rightarrow$ VKName
  - In 3NF: **Kunde(KNr, KName, Adr, VkNr)** und **Verkäufer(VkNr, VkName)**
  - Problem: Folgende Zerlegung ebenfalls in 3NF
    Kunde(KNr, KName, Adr, VkNr) und **Verkäufer(KNr, VkName)**
  - Erzeugt u.a. folgende Anomalien:
    - Änderungsaufwand
      - Änderung eines VkNamen zu einer VkNr
    - Unvollständiges Einfügen
      - Einfügen eines neuen Verkäufers mit Nummer und Name, der noch keinen Kunden betreut
- Ursache: FDs Abhängigkeiten KNr $\rightarrow$ VkNr und VkNr $\rightarrow$ VKName werden auf zwei Relationen "transitiv" verteilt

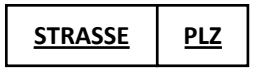
Lösung: Boyce Codd Normalform

- BCNF erzeugt "informationserhaltende Zerlegung"
  - Damit keine Anomalien mehr
  - Leider kann nicht jedes 3NF Schema in BCNF überführt werden, da es nicht immer eine informationserhaltende Zerlegung gibt (unechte Tupel entstehen)
- Ein Relationenschema R ist in Boyce-Codd-Normalform (BCNF), wenn sie in 3NF ist und für jede nicht-triviale funktionale Abhängigkeit X → A in R gilt: X ist ein Superschlüssel von R

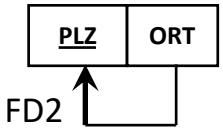
PLZ kein Super-schlüssel



STRASSE	ORT	PLZ
Baumstr	Biel	2500
Parkstr	Bern	3000
Wiesenstr	Bern	3018
Baumstr	Bern	3018



STRASSE	PLZ
Baumstr	2500
Parkstr	3000
Wiesenstr	3018
Baumstr	3018



ORT	PLZ
Biel	2500
Bern	3000
Bern	3018

FD1 geht verloren!

Zusammenfassung

- Normalformen sind Qualitätsmaß für Relationenschemata
- Normalformen (hier: 1NF bis 3NF und BCNF) …
  - Ein Relationenschema R ist in 1NF, wenn die Domänen der Attribute von R ausschließlich atomare Werte (einfache bzw. unteilbare Werte) beinhalten.
  - Ein Relationenschema R ist in 2NF, wenn es 1NF ist und kein nicht-primes Attribut von nur einem Teil des Primärschlüssels abhängt.
  - Ein Relationenschema R ist in 3NF, wenn es in 2NF ist und kein nicht-primes Attribut von R transitiv vom Primärschlüssel abhängt.
  - Ein Relationenschema R ist in Boyce-Codd-Normalform (BCNF), wenn für jede nicht-triviale funktionale Abhängigkeit $X \rightarrow A$ in R gilt: X ist ein Superschlüssel von R.
- 3NF immer erreichbar; aber BCNF nicht immer erreichbar
- Aus Performance-Gründen manchmal keine Normalisierung
  ABER: Es ist wichtig, die Probleme zu kennen und zu behandeln!!

1NF BIS 3NF & BCNF – Genauere Betrachtung

Die Überprüfung eines einzelnen Relationenschemas daraufhin, ob es einer höheren Normalform genügt, garantiert noch kein insgesamt gutes relationales DB-Schema.

Vielmehr ist zu prüfen, ob eine Menge von Relationenschemata, die insgesamt ein DB-Schema bilden, folgende zwei zusätzliche Eigenschaften aufweist:

- Die Eigenschaft der Abhängigkeitswahrung, die gewährleistet, dass jede funktionale Abhängigkeit nach der Zerlegung in (mindestens) einem der resultierenden Relationenschemata dargestellt wird.
- Die Eigenschaft eines verlustfreien bzw. nicht-additiven JOIN (informationserhaltende Zerlegung), die gewährleistet, dass in Bezug auf die nach einer Zerlegung gebildeten Relationenschemata keine unechten Tupel erzeugt werden.

Abhängigkeitswahrung und nicht-additiver JOIN

Relationenschemata

Notation

- $R = \{A_1, \ldots, A_n\}$: (Universal-)Relationenschema
  - Hypothetische Relation mit allen Attributen
- F: Menge von FDs, die für die Attribute von R gelten
  - durch den DB-Designer explizit vorgegeben werden
  - Erinnerung FD (funktionale Abhängigkeit) $X \rightarrow Y$:
    $\forall$ Tupel t_1, t_2 gilt: wenn t1[X] = t2[X] gilt, dann gilt auch t1[Y] = t2[Y]
- F^+: Hülle von F
  - Ableitbare FDs
  - Inferenzregeln (RATZAP)
- $D = \{R_1, \ldots, R_m\}$: Zerlegung („Decomposition“) von R
  - Attributerhaltung der Zerlegung D:
    - Jedes Attribut aus R erscheint in mindestens einem R_i
    - Vereinigung der Attributmengen aller R_i entspricht damit der Attributmenge von R

Abhängigkeitswahrung

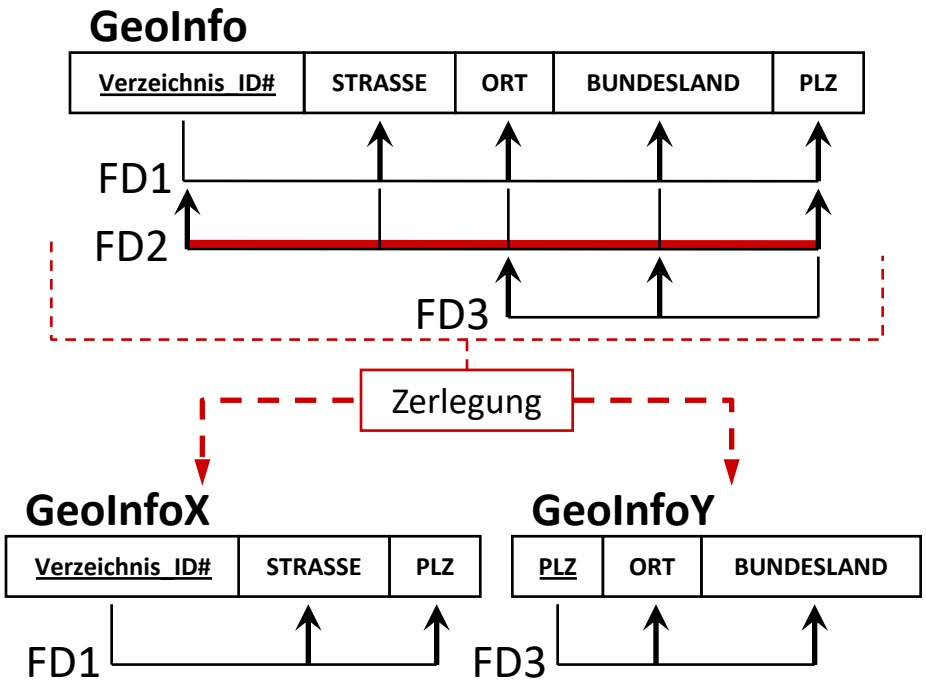
- Jede in F spezifizierte FD $X \rightarrow Y$
  - soll direkt in einem der R_i aus der Zerlegung D erscheinen oder
  - (indirekt) aus den Abhängigkeiten, die in den R_i gelten, abgeleitet werden können

- Formale Betrachtung
  - Gegeben sei eine Menge von FDs F in R.
  - Dann ist die Projektion $\pi_{Ri}(F)$ von F auf R_i (für alle R_i aus D) die Menge von Abhängigkeiten $X \rightarrow Y$ in F^+, für die alle Attribute $X \cup Y$ in R_i vorkommen.
  - Eine Zerlegung $D = \{R_1, \ldots, R_m\}$ von R heißt in Bezug auf F abhängigkeits-wahrend, wenn die Vereinigung der Projektionen von F auf jedes R_i in D mit F^+ äquivalent ist. Dies bedeutet:

$$\left(\pi_{R_1}(F) \cup \cdots \cup \pi_{R_m}(F)\right)^+ = F^+$$

Abhängigkeitswahrung – Beispiel (1)

- Zerlegung, die nicht alle FDs erhält …
  - Bei der unten gezeigten Zerlegung von GeoInfo in die Relationenschemata GeoInfoX und GeoInfoY geht FD2 verloren
  - (Verzeichnis_ID# könnte z.B. für GPS-Koordinaten stehen.)



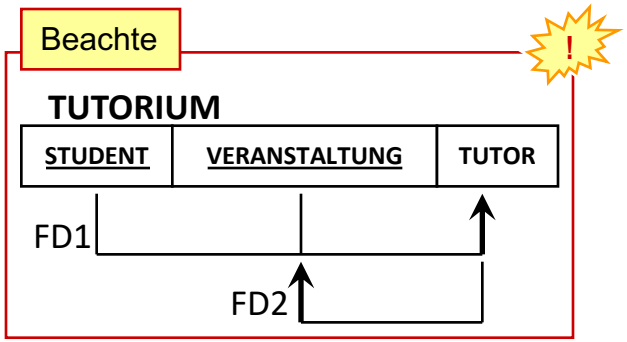
Abhängigkeitswahrung – Beispiel (2)

- Gegeben ist die 3NF-Relation TUTORIUM mit den FDs
  - FD1 : {STUDENT,VERANSTALTUNG} → TUTOR
  - FD2 : TUTOR → VERANSTALTUNG
- Mögliche Zerlegungen:
  - (STUDENT, TUTOR) und (STUDENT, VERANSTALTUNG)
  - (VERANSTALTUNG, TUTOR) und (STUDENT, VERANSTALTUNG)
  - (VERANSTALTUNG, TUTOR) und (STUDENT, TUTOR)
- FD1 geht natürlich bei allen möglichen Zerlegungen verloren, denn man bräuchte für sie alle drei Attribute in genau einem Relationenschema.

Relation TUTORIUM

TUTORIUM

STUDENT	VERANSTALTUNG	TUTOR
Narayan	Informationssysteme	Mark
Smith	Informationssysteme	Navathe
Smith	Betriebssysteme	Ammar
Smith	Formale Sprachen	Schulz
Wallace	Informationssysteme	Mark
Wallace	Betriebssysteme	Ahamad
Wong	Informationssysteme	Otte
Zelaya	Informationssysteme	Navathe



Nicht-additiver JOIN

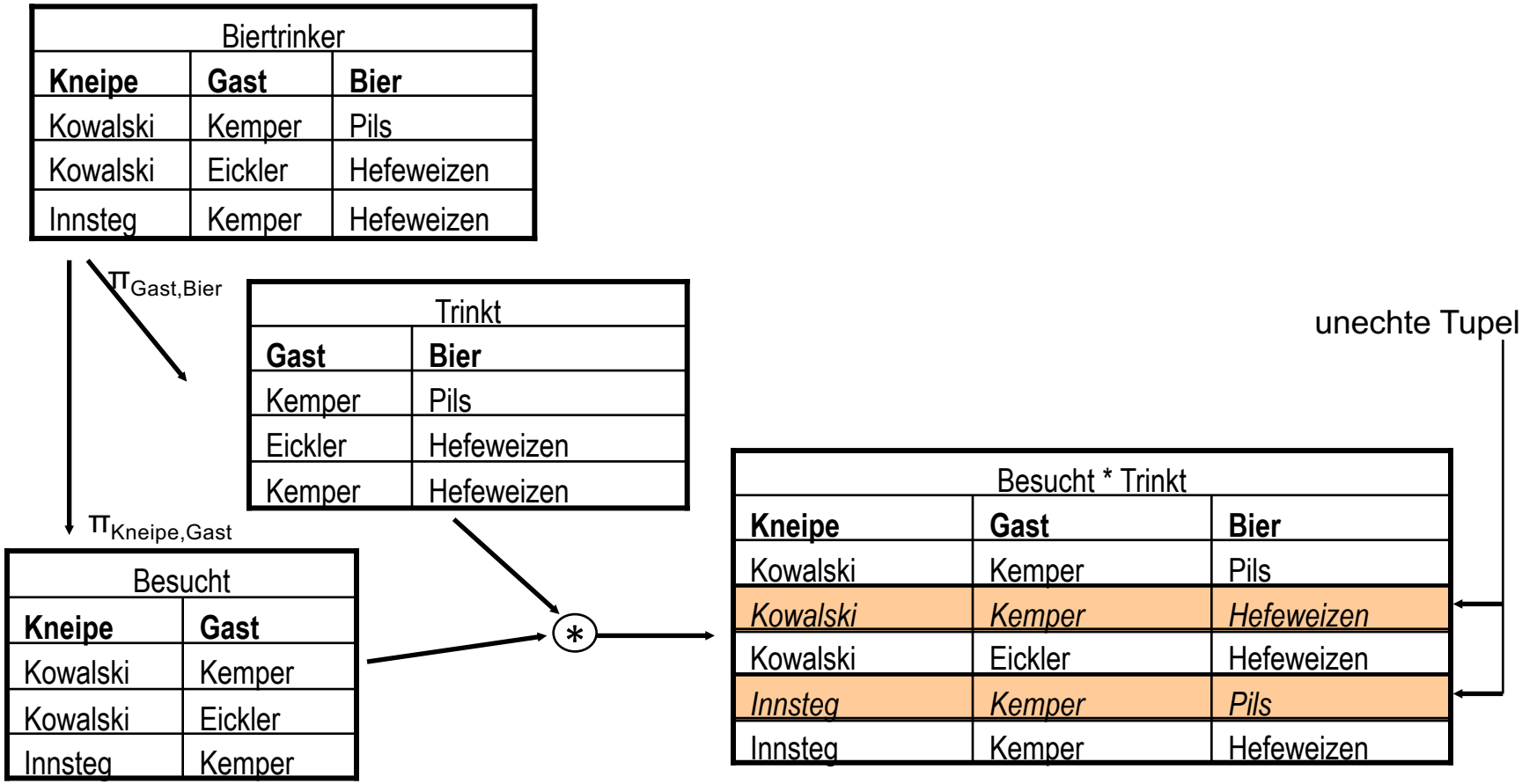
- Eine Zerlegung D eines Relationenschemas R sollte auch die Eigenschaft des nicht-additiven (informationserhaltenden) JOIN erfüllen (etwas missverständlich oft auch verlustfreier JOIN genannt)
  - Eine Zerlegung D = {R_1, …, R_m} von R weist die Eigenschaft des nicht-additiven JOIN in Bezug auf die Abhängigkeitsmenge F in R auf, wenn für jede Relation r(R), die F erfüllt, gilt (* bezeichnet hier den NATURAL JOIN):

$$*\left(\pi_{R_1}(r), \ldots, \pi_{R_m}(r)\right) = r$$

- D.h. JOINs über die zerlegten Relationen erzeugen keine „unechten“ Tupel
  - „Informationsgehalt“ von R und D bleibt gleich

Nicht-additiver JOIN: Beispiel

- Zerlegung, die nicht die Eigenschaft des nicht-additiven JOINs erfüllt



Synthesealgorithmus

Zerlegung eines Relationenschemas

Relationaler Synthesealgorithmus

- Es ist für ein Relationenschema R immer möglich, eine abhängigkeits-wahrende Zerlegung D in Bezug auf F zu finden, so dass jede Relation R_i in 3NF ist und die nicht-additive JOIN Eigenschaft erfüllt ist.

→ Relationaler Synthesealgorithmus

- Input: Universal-Relationenschema R, Menge von FDs F für die Attribute von R
- Output: Zerlegung D = $\{R_1, \ldots, R_m\}$ von R

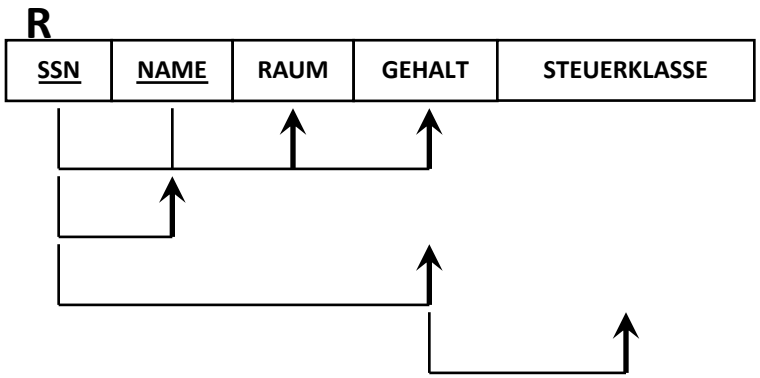
1. Finde eine minimale Hülle G für F.
2. Für jedes linke X (Quelle) einer FD aus G, erzeuge ein Relationenschema in D mit Attributen $\{X \cup \{A_1\} \cup \{A_2\} \cup \ldots \cup \{A_k\}\}$,
   wobei $X \rightarrow A_1$, $X \rightarrow A_2$, …, $X \rightarrow A_k$ die einzigen Abhängigkeiten in G mit X auf der linken Seite sind
   - X ist Schlüssel dieses Relationenschemas
3. Wenn keines der resultierenden Relationenschemata in D einen Schlüssel von R enthält, dann erzeuge ein weiteres Relationenschema in D, das die Attribute enthält, die einen Schlüssel von R bilden
4. Eliminiere diejenigen Schemata $R(A_1,\ldots, A_n)$ in D, die in einem anderen Schema S aus D enthalten sind, d.h. $\pi_{A1,\ldots, An}(S) = R$

Schritt 1: Ermittlung der minimalen Hülle

- Minimale Hülle:
  - Für jede FD X → Y: Y ist ein Attribut, X lässt sich nicht weiter reduzieren
  - Es lässt sich keine FD entfernen, ohne Abhängigkeiten zu verlieren
- Vorgehen
  a) Setze $G := F$.
  b) Ersetze jede FD $X \rightarrow \{A_1, A_2, \ldots, A_n\}$ in G durch n FDs $X \rightarrow A_1, X \rightarrow A_2, \ldots, X \rightarrow A_n$
     - Zerlegungsregel IR4
  c) Für jede FD $X \rightarrow A$ in G, und für jedes Attribut $B \in X$:
     - Wenn $G - \{X \rightarrow A\} \cup \{(X - \{B\}) \rightarrow A\}$ äquivalent zu G, dann ersetze $X \rightarrow A$ in G durch $\{(X - \{B\}) \rightarrow A\}$
       - Reduziert Abhängigkeiten mit mehreren Attributen als Quelle auf die zwingend notwendigen Attribute
  d) Für jede noch verbleibende FD $X \rightarrow A$ in G:
     - Wenn $G - \{X \rightarrow A\}$ äquivalent zu G, dann entferne $X \rightarrow A$ aus G
       - Streicht überflüssige Abhängigkeiten ganz

Algorithmus – Beispiel

- Sei folgendes Relationenschema R mit funktionalen Abhängigkeiten gegeben:



- Schritt 1: Ermittlung der minimalen Hülle G für F
  - 1.a) Setze G := F
    - G = {
      {SSN,NAME} → {RAUM,GEHALT},
      {SSN} → {NAME},
      {SSN} → {GEHALT},
      {GEHALT} → {STEUERKLASSE}
      }

Algorithmus – Beispiel

- Schritt 1: Ermittlung der minimalen Hülle G für F
  - G nach Schritt 1.a)
    - G = {
      {SSN,NAME} → {RAUM,GEHALT},
      {SSN} → {NAME},
      {SSN} → {GEHALT},
      {GEHALT} → {STEUERKLASSE}
      }
  - 1.b) Ersetze jede FD $X \rightarrow \{A_1, A_2, \ldots, A_n\}$ in G durch n FDs $X \rightarrow A_1, X \rightarrow A_2, \ldots, X \rightarrow A_n$
    - G = {
      {SSN,NAME} → {RAUM},
      {SSN,NAME} → {GEHALT},
      {SSN} → {NAME},
      {SSN} → {GEHALT},
      {GEHALT} → {STEUERKLASSE}
      }

Algorithmus – Beispiel

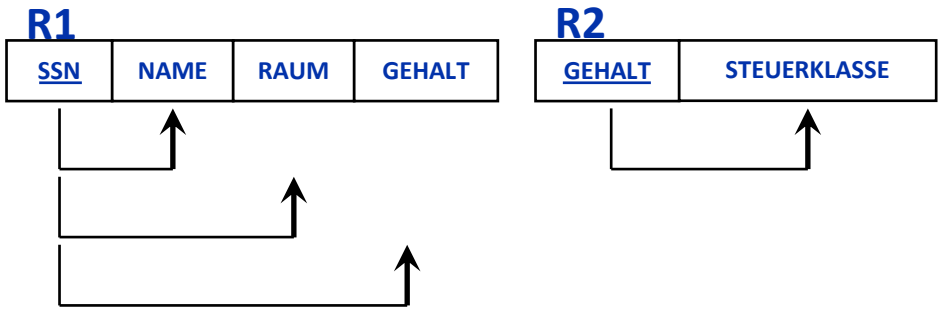
- Schritt 1: Ermittlung der minimalen Hülle G für F
  - G nach Schritt 1.b)
    - G = {
      {SSN,NAME} → {RAUM},
      {SSN,NAME} → {GEHALT},
      {SSN} → {NAME},
      {SSN} → {GEHALT},
      {GEHALT} → {STEUERKLASSE}
      }
  - 1.c) Für jede FD $X \rightarrow A$ in G, und für jedes Attribut $B \in X$:
    - Wenn $G - \{X \rightarrow A\} \cup \{(X - \{B\}) \rightarrow A\}$ äquivalent zu G, dann ersetze $X \rightarrow A$ in G durch $\{(X - \{B\}) \rightarrow A\}$
    - Aus der Abhängigkeitsquelle {SSN,NAME} wird NAME gestrichen, d.h. NAME ist B. Man hätte aber auch SSN streichen können.
    - G = {
      {SSN} → {RAUM},
      ~~{SSN} → {GEHALT},~~
      {SSN} → {NAME},
      {SSN} → {GEHALT},
      {GEHALT} → {STEUERKLASSE}
      }

Algorithmus – Beispiel

- Schritt 1: Ermittlung der minimalen Hülle G für F
  - G nach Schritt 1.c)
    - G = {
      {SSN} → {RAUM},
      {SSN} → {NAME},
      {SSN} → {GEHALT},
      {GEHALT} → {STEUERKLASSE}
      }
  - 1.d) Für jede noch verbleibende FD $X \rightarrow A$ in G:
    - Wenn $G - \{X \rightarrow A\}$ äquivalent zu G,
      dann entferne $X \rightarrow A$ aus G
    - Hier nichts zu entfernen
    - G = {
      {SSN} → {RAUM},
      {SSN} → {GEHALT},
      {SSN} → {NAME},
      {GEHALT} → {STEUERKLASSE}
      }
- Schritt 1 abgeschlossen, es folgen die Schritte 2, 3 und 4 ...

Algorithmus – Beispiel

- Schritt 2: Erzeuge Relationenschemata für FDs
  - Für jedes linke X (Quelle) einer FD aus G, erzeuge ein Relationenschema in D mit Attributen $\{X \cup \{A_1\} \cup \{A_2\} \cup \ldots \cup \{A_k\}\}$,
    wobei $X \rightarrow A_1$, $X \rightarrow A_2$, …, $X \rightarrow A_k$ die einzigen Abhängigkeiten in G mit X auf der linken Seite sind
  - G nach Schritt 1
    - G = {
      {SSN} → {RAUM},
      {SSN} → {GEHALT},
      {SSN} → {NAME},
      {GEHALT} → {STEUERKLASSE}
      }



Algorithmus – Beispiel

- D nach Schritt 2

- Schritt 3: Abschließendes Relationenschema bei fehlendem Schlüssel
  - Wenn keines der Relationenschemata in D einen Schlüssel von R enthält, dann erzeuge ein weiteres Relationenschema in D, das Attribute enthält, die einen Schlüssel von R bilden.
    - Betrifft Attribute, die in keiner FD vorkommen
  - R_1 und R_2 enthalten einen Schlüssel von R (keine Änderungen in D)

- Schritt 4: G reduzieren
  - Eliminiere diejenigen Schemata in D, die in einem anderen Schema aus D enthalten sind
  - R_1 und R_2 sind nicht im jeweils anderen enthalten (keine Änderungen in D) d.h. der Algorithmus terminiert

Erzeugung BCNF

- Zerlegungsalgorithmus
  - Nicht-additive JOIN Eigenschaft erfüllt, aber:
  - Nicht abhängigkeitswahrend!
  - Ein Schema, das die BCNF nicht erfüllt, wird in zwei Schemata zerlegt, so dass die beiden zerlegten Schemata in BCNF sind
    - Durch die Dekomposition können Abhängigkeiten verloren gehen, da die Attribute einer FD möglicherweise nicht mehr in einem Schema vorkommen

- Nicht Teil dieser Vorlesung

(Zwischen-)Rückblick

- Funktionale Abhängigkeiten
- Maß für Qualität von Relationenschemata
- Problembereiche:
  - Inhaltliche Abgrenzung der Relationenschemata
  - Reduktion redundanter Werte in Tupeln
  - Reduktion der NULL-Werte in Tupeln
  - Vermeidung „unechter“ Tupel
- Normalformen (in dieser Lerneinheit)
  - 1. bis 3. Normalform
  - Boyce-Codd-Normalform
  - Abhängigkeitswahrung
  - Nicht-additiver Join

Nicht klausurrelevant :-)

Höhere Normalformen

4NF und 5NF

Höhere Normalformen

- 4. Normalform
  - Mehrwertige Attribute (geht nach 1NF nicht, sind also möglicherw. versteckt)
  - Mehrwertige Abhängigkeiten (multi-valued dependency, MVD)
    - Falls in einem Relationenschema $R(A_1, \ldots, A_n)$ wenigstens zwei 1:N-Beziehungen der Form $A_i:A_j$ und $A_i:A_k$ existieren, bei denen A_j und A_k „unabhängig“ sind, dann kann eine MVD entstehen → zu zerlegen für 4NF
  - Beispiel
    - Ein Angestellter arbeitet an Projekten und hat Angehörige. Er kann an mehreren Projekten arbeiten und ebenfalls mehrere Angehörige haben. Projekte und Angehörige sind aber voneinander unabhängig.
    - In (a) sind die beiden 1:N-Beziehungen in einer Relation, in (b) auf zwei Relationen verteilt.

(a) **ANGEST**

ANAME	PNAME	AANAME
Smith	P1	John
Smith	P2	Anna
Smith	P1	Anna
Smith	P2	John

(b) **ANGEST_PROJEKTE**

ANAME	PNAME
Smith	P1
Smith	P2

ANGEST_ANGEHÖRIGE

ANAME	AANAME
Smith	John
Smith	Anna

Höhere Normalformen

- 5. Normalform
  - Es gibt Fälle, in denen keine nicht-additive JOIN-Zerlegung eines Schemas R in zwei Relationenschemata möglich ist, sondern nur in drei oder mehrere.
  - Beispiel:
    - Zerlegung in R1, R2 und R3 ist notwendig, um die Informationen aus R verlustfrei aufzunehmen; man will aber wiederum eine nicht-additive JOIN-Zerlegung erreichen (man sagt auch „JOIN-Abhängigkeit“ verhindern).

R	Repräsentant	Firma	Produkt
	Schmidt	Ford	PKW
	Schmidt	Ford	LKW
	Schmidt	VW	Transporter
	Müller	Porsche	PKW
	Müller	Ford	PKW

Zerlegung

R1	Repräsentant	Firma
	Schmidt	Ford
	Schmidt	VW
	Müller	Porsche
	Müller	Ford

R2	Repräsentant	Produkt
	Schmidt	PKW
	Schmidt	LKW
	Schmidt	Transporter
	Müller	PKW

R3	Firma	Produkt
	Ford	PKW
	Ford	LKW
	VW	Transporter
	Porsche	PKW

Die 5. Normalform – Motivation

Beispiel (Test 1): JOIN über *R1* und *R2* (bzgl. Repräsentant)

R1	Repräsentant	Firma
	Schmidt	Ford
	Schmidt	VW
	Müller	Porsche
	Müller	Ford

R2	Repräsentant	Produkt
	Schmidt	PKW
	Schmidt	LKW
	Schmidt	Transporter
	Müller	PKW

R3	Firma	Produkt
	Ford	PKW
	Ford	LKW
	VW	Transporter
	Porsche	PKW

NATURAL JOIN

R	Repräsentant	Firma	Produkt
	Schmidt	Ford	PKW
	Schmidt	Ford	LKW
	Schmidt	Ford	Transporter
	Schmidt	VW	PKW
	Schmidt	VW	LKW
	Schmidt	VW	Transporter
	Müller	Porsche	PKW
	Müller	Ford	PKW

Unechte Tupel!

Die 5. Normalform – Motivation

Beispiel (Test 2): JOIN über *R1* und *R3* (bzgl. Firma)

R1	Repräsentant	Firma
	Schmidt	Ford
	Schmidt	VW
	Müller	Porsche
	Müller	Ford

R2	Repräsentant	Produkt
	Schmidt	PKW
	Schmidt	LKW
	Schmidt	Transporter
	Müller	PKW

R3	Firma	Produkt
	Ford	PKW
	Ford	LKW
	VW	Transporter
	Porsche	PKW

NATURAL JOIN

R	Repräsentant	Firma	Produkt
	Schmidt	Ford	PKW
	Schmidt	Ford	LKW
	Schmidt	VW	Transporter
	Müller	Porsche	PKW
	Müller	Ford	PKW
	Müller	Ford	LKW

Unechtes Tupel!

Die 5. Normalform – Motivation

Beispiel (Test 3): JOIN über *R2* und *R3* (bzgl. Produkt)

R1	Repräsentant	Firma
	Schmidt	Ford
	Schmidt	VW
	Müller	Porsche
	Müller	Ford

R2	Repräsentant	Produkt
	Schmidt	PKW
	Schmidt	LKW
	Schmidt	Transporter
	Müller	PKW

R3	Firma	Produkt
	Ford	PKW
	Ford	LKW
	VW	Transporter
	Porsche	PKW

NATURAL JOIN

R	Repräsentant	Firma	Produkt
	Schmidt	Ford	PKW
	Schmidt	Porsche	PKW
	Schmidt	Ford	LKW
	Schmidt	VW	Transporter
	Müller	Ford	PKW
	Müller	Porsche	PKW

Unechtes Tupel!

Die 5. Normalform – Motivation

Beispiel (Test 4): `JOIN` über *R1*, *R2* und *R3*

R1	Repräsentant	Firma
	Schmidt	Ford
	Schmidt	VW
	Müller	Porsche
	Müller	Ford

R2	Repräsentant	Produkt
	Schmidt	PKW
	Schmidt	LKW
	Schmidt	Transporter
	Müller	PKW

R3	Firma	Produkt
	Ford	PKW
	Ford	LKW
	VW	Transporter
	Porsche	PKW

NATURAL JOIN

R	Repräsentant	Firma	Produkt
	Schmidt	Ford	PKW
	Schmidt	Ford	LKW
	Schmidt	VW	Transporter
	Müller	Porsche	PKW
	Müller	Ford	PKW

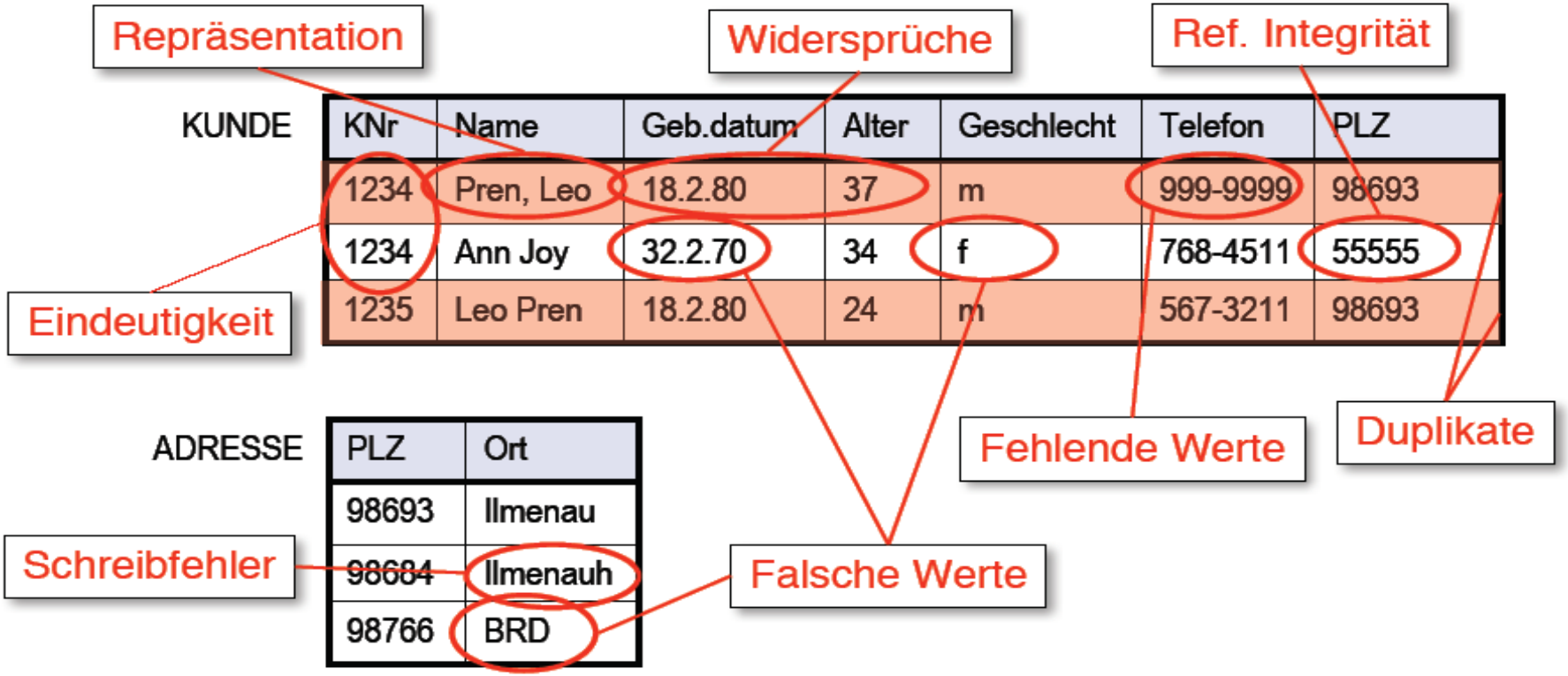
Datenqualität

Qualität der Extension

Das Problemfeld Datenqualität

- Normalformen zielen auf die strukturelle Qualität von Relationenschemata
  - „Intension“
- Datenqualität umfasst Beiträge zur Verbesserung der konkreten „inhaltlichen“ Dateninstanzen auf der Schemaebene
  - „Extension“
- Der Aspekt der Datenqualität ist als problematisch einzuschätzen, wenn Daten vor dem Hintergrund einer Anwendungssituation, z.B.
  - nicht die angenommene Bedeutung haben,
  - nicht der Spezifikation entsprechen oder
  - unverständlich sind.

Beispiele für mindere Datenqualität



(Quelle: K. Sattler; DB-Tutorientage 2005)

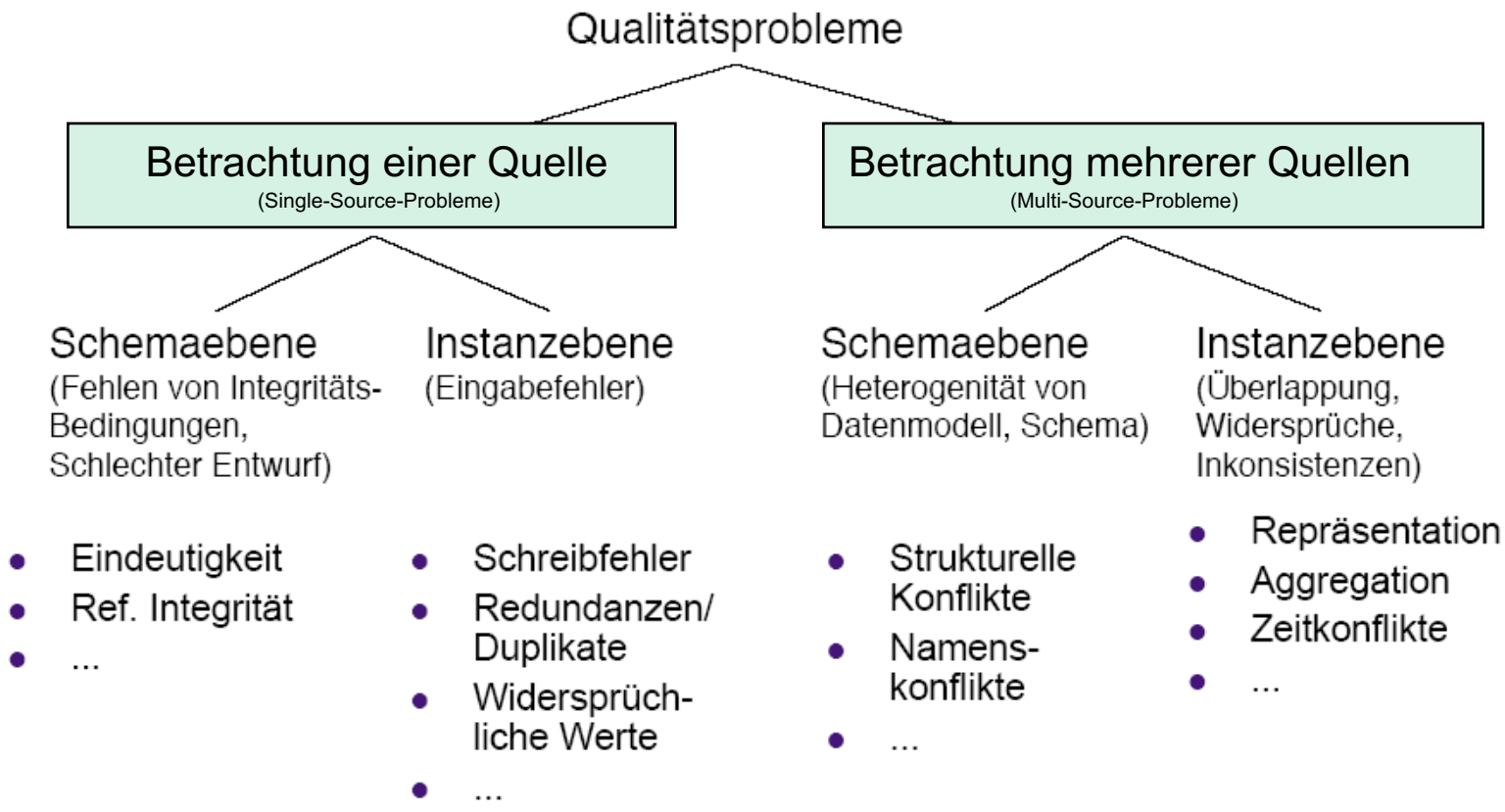
Ursachen für Datenqualitätsprobleme

- Ursachen liegen auf verschiedenen Ebenen, u.a. bei der
  - Datenproduktion, z.B.
    - Verschiedene Quellen repräsentieren gleiche Realwelt-Objekte in unterschiedlicher Form
    - Datenerfassung unterliegt „subjektiven Einflüssen“
    - Systematische Probleme bei Datenerfassung (Messung, Kodierung etc.)
  - Datenspeicherung, z.B.
    - Unterschiedliche oder ungeeignete Formate
  - Datennutzung, z.B.
    - Unzureichende Analyse- und Verarbeitungsmöglichkeiten
    - Veränderung der Nutzerbedürfnisse
    - Sicherheits- und Zugriffsprobleme

(Quelle: K. Sattler; DB-Tutorientage 2005)

Arten von Datenqualitätsproblemen

Systematische Betrachtung von Datenqualitätsproblemen …



(Quelle: K. Sattler; DB-Tutorientage 2005)

Dimensionen der Datenqualität

- Datenqualität wird anhand verschiedener Dimensionen beurteilt, z.B.
  - Vollständigkeit: Verhältnis tatsächlicher Werte zu gespeicherten Werten, u.a.
    - Wertebelegungen verschieden von Null
    - Repräsentation aller in der Realwelt vorkommenden Objekte
  - Genauigkeit: Verhältnis der Anzahl der korrekten Werte zur Gesamtanzahl, d.h. prozentualer Anteil an Daten ohne Datenfehler
    - Umfang, in dem Attributwerte im jeweils „optimalen“ Detaillierungsgrad vorliegen
    - Nähe eines Wertes zum korrekten Wert innerhalb der Realwelt
  - Zeitnähe: Aktualität, in der Attributwerte dem sich dynamisch ändernden Realwelt-Zustand entsprechen
    - Alter: Zeit seit dem Erfassen / Laden der Daten
    - Volatilität: Häufigkeit der Änderungen
  - Relevanz: Grad, in dem der Informationsgehalt den Nutzerbedürfnissen entspricht

(Quelle: K. Sattler; DB-Tutorientage 2005)

Dimensionen der Datenqualität (Fortsetzung)

- Weitere Dimensionen, u.a.
  - Verständlichkeit
    - Grad, in dem Daten in Inhalt und Struktur mit der „Vorstellungswelt“ des Nutzers übereinstimmen
  - Konsistenz
    - Grad, in dem Daten frei von logischen Widersprüchen sind (Integritätsbedingungen, Geschäftsregeln, ...)
  - Verfügbarkeit
    - Grad, in dem Daten für einen Nutzer in einem bestimmten Zeitraum nutzbar sind
  - Glaubwürdigkeit
    - Grad, in dem Daten vom Nutzer als korrekt akzeptiert werden
  - Kosten
    - Preis für Datenzugriff, Anfrage, Datenübertragung, ...

(Quelle: K. Sattler; DB-Tutorientage 2005)

Rückblick

- Funktionale Abhängigkeiten
- Maß für Qualität von Relationenschemata
- Problembereiche:
  - Inhaltliche Abgrenzung der Relationenschemata
  - Reduktion redundanter Werte in Tupeln
  - Reduktion der NULL-Werte in Tupeln
  - Vermeidung „unechter“ Tupel
- Normalformen
  - 1. bis 3. Normalform
  - Boyce-Codd-Normalform
  - Abhängigkeitswahrung
  - Nicht-additiver Join
  - Motivation für höhere Normalformen (4. und 5.)
- Datenqualität