

Grundlagen der Datenbanknormalisierung

Referent: Michael Zimmermann

Firma:

S&Z DataWare
Friedrich-Naumann-Straße 25
55131 Mainz

06131 - 2 77 8 55

Zimmermann@SZWeb.de

Privat:

Friedrich-Naumann-Straße 21
55131 Mainz
06131 - 83 31 79

www.szweb.de

Weitere Artikel

Die Inhalte des Skriptes werden auch auf **www.info.szweb.de** veröffentlicht und dort weiter überarbeitet und um weitere Beispiele ergänzt.

Weiterführende Themen werden beispielsweise die Modellierung zeitlich variabler Daten sein (Historisierungsansatz, Archivansatz) und Artikel über ERM.

Ebenso sind Artikel über SQL- und VBA-Performance geplant.

Übersicht

- Entwurfsziele
- Anomalien
- 0. Normalform
- 1. Normalform
- Theorie der Abhängigkeiten
- Der Schlüsselbegriff
- 2. Normalform
- Kentsche 2. Normalform
- 3. Normalform
- Boyce-Codd-Normalform
- 4. Normalform
- Noch einmal: Abhängigkeitstheorie
- 5. Normalform
- Domain-Key-Normalform
- Entwurfsempfehlungen

Warum Relationalität?

Der Ansatz der relationalen Datenbank ist weder Selbstzweck noch von rein akademischem Interesse, sondern verfolgt konkrete praktische Ziele. Einige Vorgehensweisen bei den einzelnen Normalformen werden einem reinen Praktiker, der sich bisher wenig mit Theorie beschäftigt hat, vielleicht übertrieben streng vorkommen. Aber die oft praktizierten Ersatzlösungen entsprechen in der Regel den Strukturen der historisch vorangegangenen Textdatenbanken (Trennzeichen- und Festbreitenprinzip) oder hierarchischen Datenbanken.

Die dabei in der Praxis zutage getretenen Probleme, die insbesondere das Textprinzip sich als wenig geeignet erweisen ließen, haben gerade zur Entwicklung der relationalen Theorie in den 70er Jahren geführt.

Man sollte sich gut überlegen, ob man durch Verzicht auf Normalisierungsregeln alte und längst als untauglich abgeschnittene Zöpfe durch die Hintertür in einen Datenbankentwurf wieder einlassen will.

Entwurfsziele

An erster Stelle ist die Sicherung der Konsistenz der Datenbank zu nennen.

Konsistenzsicherung

- Es muß ausgeschlossen sein, daß die DB in einen unzulässigen Zustand geraten kann

Unzulässige Zustände sind solche, die logisch widersprüchlich sind. Die Möglichkeit logischer Widersprüche wird in erster Linie durch das Abspeichern folgerbarer Werte (Berechnungen) und mehrfaches Abspeichern derselben Daten (Redundanz) eröffnet.

Informationserhalt/Abhängigkeitserhalt

- Zerlegungen, durch die Information verlorengeht, werden nicht durchgeführt; es sind nur verlustlose Zerlegungen zulässig
- Zerlegungen, durch die Abhängigkeiten/ Schlüsselbedingungen verlorengehen, werden nur durchgeführt, wenn es keine abhängigkeitsbewahrende Zerlegung gibt und Redundanzen vermieden werden müssen

Um das Ziel der Redundanzfreiheit zu erreichen, wird es notwendig sein, Tabellen, die redundante Informationen enthalten, in mehrere kleinere Tabellen aufzuteilen. Dies nennt man Zerlegung. Diese Zerlegung muß so erfolgen, daß die ursprünglichen Informationen durch Abfragen rekonstruierbar sind.

Im wesentlichen wird das dadurch erreicht, daß beide Tabellen einer Zerlegung über gleiche Schlüsselfelder verfügen, über die sich mit einem Join die ursprüngliche Tabelle rekonstruieren läßt.

Zur Abhängigkeitsbewahrung bzw. -verlust werden wir bei der Boyce-Codd-Normalform ein Beispiel sehen.

Regelbildung

- Nicht mit Wahrscheinlichkeiten
- Zulässige Kriterien sind:
 - ▶ logische Gesetze
 - ▶ mathematische Gesetze
 - ▶ Naturgesetze
 - ▶ sonstige stringente Regeln (Vertrag, Gesetz, Satzung, Geschäftsregel)
- ... durch die das Kriterium einwandfrei allgemeingültig im Rahmen der möglichen Werte wird

Bei der Kriteriensuche, z. B. für Eindeutigkeit von Wertkombinationen oder Existenz von Abhängigkeiten usw., wird **ausschließlich** mit stringenten Regeln und niemals mit Wahrscheinlichkeiten gearbeitet.

Um beispielsweise zu beurteilen, ob eine Feldgruppe

(Vorname, Nachname, Wohnort, Straße, Geburtsdatum)

eindeutig ist, was später bei der Schlüsselfindung eine Rolle spielen wird, ist es völlig irrelevant, daß man einen solchen Fall für „ziemlich unwahrscheinlich“ oder „praktisch ausgeschlossen“ hält.

Normalisierung alleine nicht ausreichend

- Entity-Relationship-Modell (ERM) bringt weitere notwendige Einschränkungen

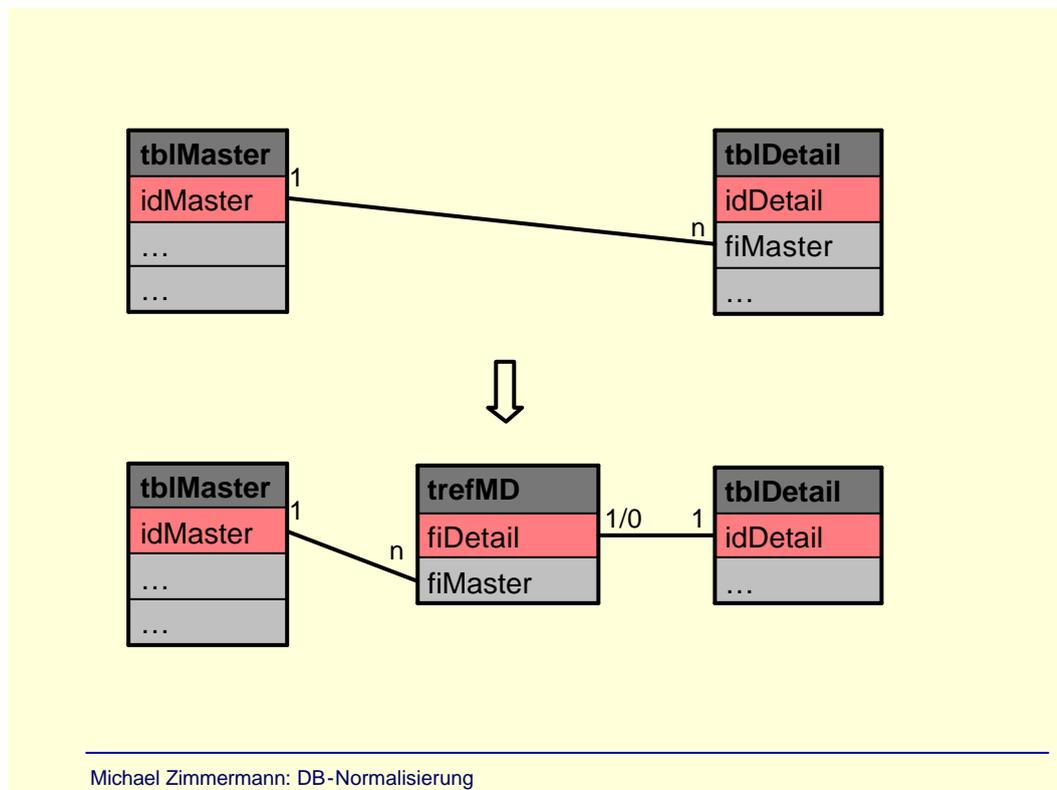
- Weitere Entwurfsvorschriften aus

- ▶ Relationaler Algebra
- ▶ Semantik

Als solche Einschränkungen wären z. B. zu nennen:

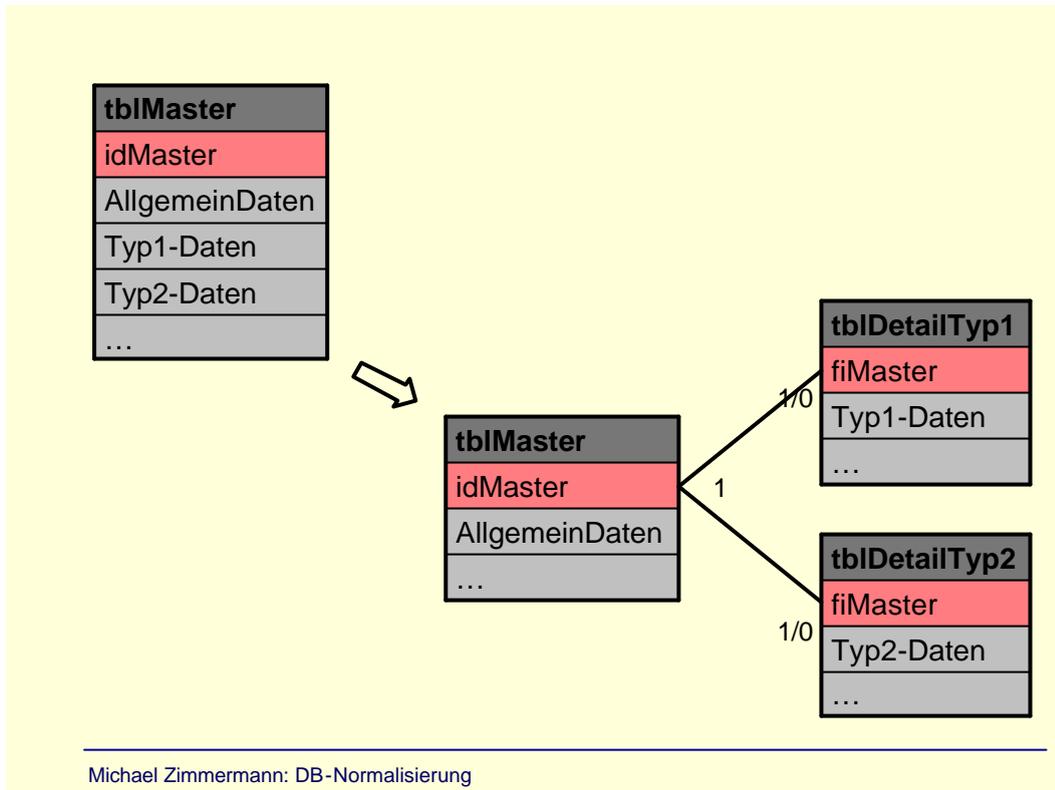
- Die Vorschrift, NULL-Werte in Fremdschlüsselfeldern von 1:n-Beziehungen zu vermeiden.

In der Folge müssen Fremdschlüsselfelder mit Eingabepflicht versehen werden. Wenn die Beziehung sachlich keine strenge 1:n-Beziehung ist, muß sie analog einer m:n-Beziehung mit einer Referenztabelle aufgebaut werden, nach dem Muster



- Die Vorschrift, systematische NULL-Werte in normalen Datenfeldern zu vermeiden.

Wenn sich ergibt, daß bestimmte Felder bei bestimmten Datensatztypen nur NULL-Werte enthalten können, ist eine **Spezialisierung** über eine 1:1/0-Beziehung vorzunehmen.



Beispiel Tiere (Gewicht; Fellfarbe; Schuppenfarbe)

Bei einem Säugetier bedeutet ein NULL-Wert in der Fellfarbe, daß er zwar eine Fellfarbe hat, wir aber deren Wert noch nicht kennen (eigentliche Bedeutung des NULL-Wertes). Bei allen Fischen bleibt das Feld aber grundsätzlich leer, da sie gar kein Fell haben. Für die Schuppenfarbe entsprechend umgekehrt.

- Die Vorschrift, fehlende Daten tatsächlich einheitlich über alle Datentypen hinweg durch NULL-Werte auszudrücken und niemals durch Scheindaten wie „Keine Angabe“, „Nicht verfügbar“ o. ä.

Normalisierung notwendig zur Qualitätssicherung

- ... „intuitiver“ Entwürfe
- ... von ERM-Entwürfen

Insbesondere sind Entity-Relationship-Modellierung und Normalisierung zwei Verfahren, die einander **ergänzen** und **nicht ersetzen**. Intuitiver Entwurf soll hier auch nicht als theoriefreies „Darauflosetwerfen“ verstanden werden, sondern meint, daß der Entwickler die ERM-Prinzipien so sehr verinnerlicht hat, daß er sie blind, auch ohne Diagramme zu zeichnen, direkt als Tabellenentwürfe umsetzen kann.

Um es noch einmal deutlich zu machen: Die beiden oben vorgestellten Vorgehensweisen werden durch keine Normalisierungsregel gefordert, sind aber dennoch unbedingt einzuhalten. Die ERM-Regeln haben ebenso normativen Charakter wie die Normalisierungsregeln.

Algorithmisches Normalisieren alleine kein Garant für „guten“ Entwurf

Es gibt Algorithmen, um Feldgruppen programmatisch in bestimmte Normalformen zu bringen. So erzeugte Entwürfe sind häufig, wiewohl formal einwandfrei, wenig brauchbar. Insbesondere haben algorithmische Entwürfe oft mit „unschönen Semantiken“ zu kämpfen, d. h. die Kombinationen der Felder zu Tabellen kann zu Konstrukten führen, denen nur schwer oder gar nicht ein anschauliches Objekt der abzubildenden Realität zugeordnet werden kann.

Ein an den Objekten der Realität orientierter anschaulicher Entwurf, der danach über Normalisierungsregeln die notwendige Nachjustierung erhält, ist vorzuziehen

Anomalien

Einfügeanomalie

- Datensätze können nicht angelegt werden, ohne Scheindaten zu produzieren

Wenn beispielsweise Auftraggeberdaten direkt und nur beim Auftrag erfaßt werden, kann ein potentieller Kunde nicht eingegeben werden, da er noch keinen Auftrag erteilt hat. Wenn man, um die Kundendaten eingeben zu können, einen leeren Auftrag anlegt, stimmt die Anzahl der Aufträge nicht mehr.

Löschanomalie

- Datensätze können nicht gelöscht werden, ohne dadurch weitere Daten zu verlieren

Das Gegenstück zur Einfügeanomalie: Werden alle Aufträge eines Kunden gelöscht, gehen damit auch die Kundendaten verloren, wenn diese nur als Felder der Auftrags-tabelle existieren.

Änderungsanomalie

- Ein Faktum muß an vielen Stellen geändert werden
 - ▶ Widersprüchliche Daten
 - ▶ Fehlerhafte Abfrageergebnisse

Wenn dieselben Kundendaten wiederholt eingegeben werden – bei jedem Auftrag des Kunden –, besteht das Risiko, daß durch Eingabefehler Widersprüche in den Daten entstehen. Bei einer Änderung der Kundendaten werden eventuell nicht alle Datensätze erfaßt. Abfragen, die z. B. die Anzahl der Aufträge eines Kunden liefern sollen liefern dann möglicherweise falsche Ergebnisse.

Strukturanomalie

- Tabellen geben nicht die Abhängigkeiten der Daten wider
 - ▶ Objektbeziehungen nicht ersichtlich
 - ▶ Integritätsregeln nicht durchsetzbar

0. Normalform

Tableauschema, falsch

| Zahl 1 | Zahl 2 | Summe |
|--------|--------|--------|
| 20.000 | 30.000 | 50.000 |
| 40.000 | 50.000 | 70.000 |

Noch ist die Welt in Ordnung...
 Welche der Zahlen ist falsch?

| Anrede | VName | Name | Briefanrede |
|--------|-------|-------|-------------------|
| Herr | P. | Meier | Sehr geehrte Frau |

Herr oder Frau Meier?

| Von | Bis | Beitrag |
|------|------|---------|
| 1997 | 2002 | 500 |
| 2001 | 2005 | 700 |

Welcher Beitrag gilt nun 2001 und 2002?

Michael Zimmermann: DB-Normalisierung

Definition

- Alle für das zu modellierende Thema relevanten Daten und nur diese sind erfaßt und bringen echten Informationsgewinn

Bedeutung

- Informationen sind sowohl Werte selbst als auch die Abhängigkeiten zwischen ihnen
- Keine Werte speichern, die sich aus anderen Werten folgern lassen
 - ▶ Keine Zeilenberechnungen (Kalkulation)
 - ▶ Keine Spaltenberechnungen (Aggregation)
- Abhängigkeiten nicht mehrfach modellieren
- Aber auch keine relevanten Werte oder Wertbeziehungen auslassen

0 NF

Die 0 NF beinhaltet im wesentlichen die Vorschrift, keine berechneten Werte zu speichern.

In den Beispielen oben können z. B. die Summen innerhalb eines Datensatzes oder die Briefanrede aus anderen Feldern in einer Abfrage berechnet werden. Die Speicherung, die die Möglichkeit der Fehleingabe eröffnet und damit Widersprüche ermöglicht, ist daher obsolet.

Ebenso im dritten Beispiel, wo sich das Ende einer Gültigkeitsperiode als der nächstgrößere Beginn ergibt. Realisieren kann man das hier mit einer Aggregatfunktion (Min) über der Teilmenge der Datensätze, bei denen [Von] > 1997 ist

Berechnete Werte in derselben Tabelle zu speichern, die auch die Berechnungsgrundlagen enthält, verstößt obendrein gegen die 3. bzw. Boyce-Codd-Normalform; die 0 NF verbietet darüber hinaus das Speichern berechneter Werte generell.

1. Normalform

Tableauschema, falsch

| Name | Adresse | Kinder | Eltern |
|------------|--------------------------|-----------------------|------------------|
| Hans Meier | Bachweg 12 55127 Kaub | Klaus; Petra; Uwe | Heinz; Gertrud |
| oder | | | |
| ... | ... | Klaus Petra Uwe | Heinz Gertrud |

The diagram below the table shows four callout boxes connected to the table cells:

- A box labeled "Molekül Vor-, Nachname" points to the "Name" column.
- A box labeled "Molekül Straße, Nr, PLZ, Ort" points to the "Adresse" column.
- A box labeled "Liste, variabel Kind1, Kind2, Kind3" points to the "Kinder" column.
- A box labeled "Liste, konstant Elter1, Elter2" points to the "Eltern" column.

Michael Zimmermann: DB-Normalisierung

Definition:

- Alle Werte müssen **elementar** sein, d. h.
 - ▶ skalar: keine Listen/Mengen/Vektoren
d. h. mehrere Werte gleicher Struktur
 - ▶ atomar: keine Moleküle/Aggregate
d. h. mehrere Werte verschiedener Struktur

Listen

- werden wenn möglich (konstante bzw. nach oben begrenzte Anzahl) in Spalten verteilt
- werden sonst (variable Anzahl, Normalfall) in Zeilen verteilt
 - ▶ Beim Übergang in 3NF werden daraus eigene Tabellen

Moleküle

- werden immer in Spalten aufgelöst
- Ob ein Wert zerlegt werden muß oder als Atom anzusehen ist, hat kein formales Kriterium
- Die Entscheidung ist vom späteren Verwendungszweck abhängig

Tableauschema, richtig

| VN | NN | Str/Nr | PLZ | Ort | Kinder | Vater | Mutter |
|------|-------|-------------|-------|------|--------|-------|---------|
| Hans | Meier | Bachweg 12 | 55127 | Kaub | Klaus | Heinz | Gertrud |
| Hans | Meier | Bachweg 12 | 55127 | Kaub | Petra | Heinz | Gertrud |
| Hans | Meier | Bachweg 12 | 55127 | Kaub | Uwe | Heinz | Gertrud |
| ... | ... | Bachweg 12 | ... | ... | ... | ... | ... |
| ... | ... | Bach weg 12 | ... | ... | ... | ... | ... |

Alle Werte skalar
Aber: Wann ist Atomarität erreicht?

Michael Zimmermann: DB-Normalisierung

Listen sind leicht erkennbar

- Trennzeichen
- sich wiederholende gleichartige Daten

Moleküle werden leicht verkannt

- Indizien sind oft „Aktenzeichen“, häufige gleichartige Wortbestandteile in gleichen Positionen, trennende Sonderzeichen, Bedarf für Textfelder mit über 255 Zeichen
- kein inhaltliches formales Kriterium für Atomarität

- pragmatisches formales Kriterium: 1 NF ist verletzt, wenn ...
 - ▶ in irgendeinem Zusammenhang erforderlich, nach Teilen von Feldinhalten abzufragen oder zu sortieren
 - ▶ vom Datenmodell geforderte Abhängigkeiten (Indizes) nicht bildbar, weil auf Teile von Feldinhalten bezogen
- → Vorliegen von 1 NF anforderungsabhängig

Atomgrenzen können durch Wörter/Datentypen verlaufen

| Namensstamm | Suffix |
|--------------------|--------|
| Majest | ix |
| Gutemi | ne |
| Verleihn | ix |
| Yellowsubmari | ne |
| Schlagdraufundschl | us |
| Handzumgr | us |
| Maul | af |
| Klammer | af |

Michael Zimmermann: DB-Normalisierung

- Beispiel Asterixinische Namen:
Die Suffixe enthalten Information über Geschlecht und Nationalität

Ein ähnlich gelagertes etwas alltäglicheres Beispiel mit verdächtigen Wortbestandteilen: Wenn als Produktbezeichnungen wiederholt Plusbrief, Pluspäckchen, Pluspaket, Wertbrief, Wertpaket, Einschreibebrief usw. auftauchen, ist anzunehmen, daß „Plus“, „Wert“ und „Einschreibe“ keine zufälligen Namensbestandteile sind, sondern eigentlich Produktkategorien.

Daher wäre zu zerlegen:

| | |
|-------------|----------|
| Plus | brief |
| Plus | päckchen |
| Plus | paket |
| Wert | brief |
| Wert | paket |
| Einschreibe | brief |

- Aber: Einteilung in Morpheme (kleinste bedeutungstragende Einheiten der Sprache) ist i. d. R. überflüssig und unsinnig
- Beispiel Regelmodellierung
Wenn ein eindeutiger Index über Prüfung/Jahr zu legen ist, weil Prüfungen nur einmal jährlich wiederholt werden dürfen, muß Prüfungsdatum in T/M/J zerlegt werden

Atome können aber auch sehr umfangreich werden

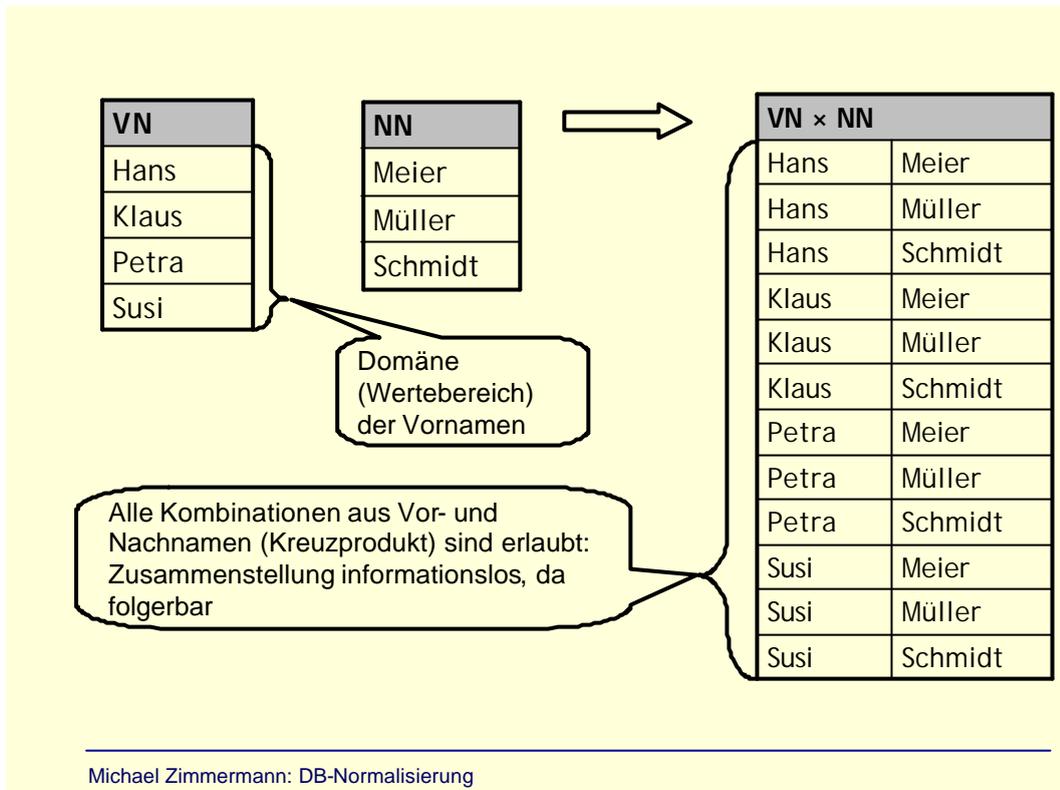
- Vollständiger Buchtext in Memofeld in Literaturdatenbank
- Adresse: Wenn keine Sortierung/Abfrage nach PLZ/Ort/Straße erfolgen müßte und Adressen nur für Anschreiben benötigt würden, wäre
 Straße Nr
 PLZ Ort
 durchaus atomar

Im Zweifel lieber aufteilen

- Unterlassen notwendiger Teilungen **ist** ein Entwurfsfehler
- Vornehmen überflüssiger Teilungen ist **kein** Entwurfsfehler
- Unnötig unterteilte Felder sind problemlos und performant verkettbar
- Die Empfehlung „Im Zweifel lieber aufteilen“ gilt nicht für die höheren Normalformen

Theorie der Abhängigkeiten

Unabhängigkeit



- Definition:
Jede Zeile aus dem Kreuzprodukt der Domänen **kann** vorkommen. Falls auch nur **eine** Kombination durch Integritätsregeln verboten ist, liegt keine Unabhängigkeit vor

- Bedeutung:
 - ▶ Die Kenntnis eines Wertes eines Feldes verrät nichts über Werte in anderen Feldern

- Symbolisch: VN •--• NN

Man verwechsle Unabhängigkeit nicht mit Zusammenhanglosigkeit. Vorname und Nachname stehen durchaus als Bestandteile des Namens in einem Sachzusammenhang miteinander. Es gibt aber keine Regel (höchstens statistische Unwahrscheinlichkeiten ibs. aufgrund fremdländischer Namen), die bestimmte Kombinationen von zugelassenen Vor- und Nachnamen verböte.

Abhängigkeiten

Funktionale Abhängigkeit (FA, FD):



| Fzg | Eigner |
|-----------|---------|
| MZ-RC 406 | Meier |
| MZ-KV 103 | Meier |
| WI-FA 205 | Meier |
| F-FM 230 | Schmidt |
| DA-GG 103 | Müller |
| AZ-C 1004 | Müller |

Michael Zimmermann: DB-Normalisierung

- Jedes Fahrzeug hat genau einen Eigentümer
- Aus der Kenntnis des Fahrzeugs kann der Eigentümer gefolgert werden

- **Definition:**

Wenn für zwei Felder X, Y aus $x_1 = x_2$ folgt, daß $y_1 = y_2$, dann gilt: Y ist funktional abhängig von X (X impliziert Y, X identifiziert Y)

- **Bedeutung:**

- ▶ Zu jedem X gehört genau ein Y, allerdings darf das gleiche Y auch bei verschiedenen X herauskommen
- ▶ Wer X kennt, kennt auch Y

- **Symbolisch: $X \rightarrow Y$**

Das entspricht im großen und ganzen dem aus der Schule bekannten $y = f(x)$

Für ein einfaches Einführungsbeispiel nehmen wir den Nachnamen auch einmal wider besseres Wissen als eindeutig an.

Abhängigkeiten

Volle funktionale Abhängigkeit

| Personenname | | Anrede | Namenslänge |
|--------------|----------|--------|-------------|
| Vorname | Nachname | | |
| Petra | Meier | Frau | 10 |
| Hermann | Meier | Herr | 12 |
| Hans | Meier | Herr | 9 |
| Petra | Schmidt | Frau | 12 |
| Susi | Schulze | Frau | 11 |
| Hermann | Schulze | Herr | 14 |

Michael Zimmermann: DB-Normalisierung

- **Definition:**

Wenn $X \rightarrow Y$ und X eine Gruppe aus mehreren Feldern X_1, X_2 ist, also $(X_1, X_2) \rightarrow Y$, dann ist Y voll funktional abhängig von X , wenn es nicht schon von X_1 oder X_2 alleine abhängt. Man sagt auch: X determiniert Y

- **Bedeutung:**

- ▶ Nur wer alle Teile von X (X_1, X_2, \dots) kennt, kennt auch Y
- ▶ Wenn schon Teile von X ausreichen, um Y zu bestimmen, heißt die funktionale Abhängigkeit von X partiell

- **Hinweis:**

- ▶ Jede FA ist entweder voll oder partiell
- ▶ Eine FA, die von einem einzelnen Feld ausgeht, ist immer voll

Im Beispiel folgt die Anrede allein schon aus der Kenntnis des Vornamens. Das ist im wahren Leben nicht so, es geht hier jedoch zunächst um ein leicht verständliches Beispiel. Ebenso würde man die Namenslänge als berechneten Wert eigentlich nicht in eine Tabelle aufnehmen. Auch hier geht es nur darum, die volle funktionale Abhängigkeit an einem nachvollziehbaren Beispiel zu demonstrieren.

Abhängigkeiten

Mehrwertige Abhängigkeit (MWA, MVD)



| Eigner | Fzg |
|---------|-----------|
| Meier | MZ-RC 406 |
| Meier | MZ-KV 103 |
| Meier | WI-FA 205 |
| Schmidt | F-FM 230 |
| Müller | DA-GG 103 |
| Müller | AZ-C 1004 |

Michael Zimmermann: DB-Normalisierung

- Eine Person kann beliebig viele Fahrzeuge besitzen
- Aus der Kenntnis des Eigentümers kann zwar nicht auf ein bestimmtes Fahrzeug geschlossen, aber die Menge der fraglichen Fahrzeuge eingeschränkt werden

- **Definition:**

Wenn für zwei Felder X, Y aus $x_1 = x_2$ folgt, daß die Mengen $\{y_{11} \dots y_{1n}\} = \{y_{21} \dots y_{2n}\}$, dann gilt: Y ist mehrwertig abhängig von X (X bestimmt Y mehrwertig, X reduziert Y)

- **Bedeutung:**

- ▶ Zu jedem X gehören beliebig viele Y , das kann natürlich auch nur eins sein
- ▶ Wer X kennt, kennt zwar nicht Y , schränkt aber die Menge der möglichen Y ein

- **Symbolisch: $X \rightarrow Y$**

- ▶ Das gängigste Symbol ist $X \twoheadrightarrow Y$; ich benutze stattdessen \rightarrow für MWA gegenüber \rightarrow für FA (Pfeilspitze beachten!)

Vorsicht bei MWA!

Wenn nicht beliebig viele, sondern immer genau gleich viele Werte bestimmt werden, liegt keine MWA vor, sondern mehrere FA

Abhängigkeiten

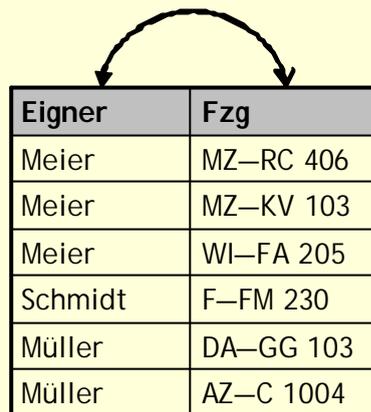
Nicht

- ▶ Kind → Eltern (1 MWA)

sondern

- ▶ Kind → Vater, Kind → Mutter (2 FA)

Abhängigkeiten sind zweiseitig:



| Eigner | Fzg |
|---------|-----------|
| Meier | MZ-RC 406 |
| Meier | MZ-KV 103 |
| Meier | WI-FA 205 |
| Schmidt | F-FM 230 |
| Müller | DA-GG 103 |
| Müller | AZ-C 1004 |

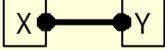
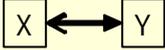
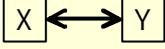
Michael Zimmermann: DB-Normalisierung

... wie die meisten am Eigner-Fahrzeug-Beispiel schon gemerkt haben dürften.

Abhängigkeiten

Folgende Varianten sind möglich:

- Folgende Varianten sind möglich:

- o  Unabhängigkeit
- o  Mehreindeutig, meist kurz eindeutig (1:n)
- o  Eineindeutig, reziprok eindeutig (1:1)
- o  Mehrmehrdeutig, reziprok mehrdeutig (m:n)

- o  Abkürzung für unabhängig
- o  Abkürzung für (mehr)eindeutig

Michael Zimmermann: DB-Normalisierung

Darstellen von Abhängigkeiten

- Abhängigkeiten können nur modelliert werden durch ...
 - ▶ Eindeutige Indizierung: Existentielle Integrität
 - ▶ Beziehungen zwischen Tabellen: Referentielle Integrität
- Indizierung wie folgt:
 - ▶ $A \rightarrow B, B \rightarrow A$: A eindeutig indiziert, B mehrdeutig
 - ▶ $A \leftrightarrow B$: A, B beide eindeutig indiziert
 - ▶ $A \Leftrightarrow B$: eindeutiger Mehrfeldindex über (AB)
 - ▶ $A \bullet \rightarrow \bullet B$: beide mehrdeutig, normale Datenfelder
- Restriktionen aus Abhängigkeiten, die sich so nicht durchsetzen lassen, müssen getriggert werden (BCNF)
- Wenn Restriktionen, die im realen Vorbild bestehen, nicht modelliert werden, gilt der Entwurf als fehlerhaft

Abhängigkeiten

Ziel der Normalisierung:

- Alle Abhängigkeiten außer voll funktionalen Schlüsselabhängigkeiten umwandeln in Beziehungen zwischen Tabellen

Es gibt weitere Arten von Abhängigkeiten

- **Projektions-Verbund-Abhängigkeit (JD)**, aus ring- oder netzförmigen Strukturen von FA und MWA zusammengesetzt
 - ▶ wird bei 5 NF besprochen
- Nur der Vollständigkeit halber seien genannt die
 - ▶ Inklusionsabhängigkeit (Inclusion Dependency)
 - ▶ Vorbildabhängigkeit (Template Dependency)

Trivialität:

- Einige Abhängigkeiten sind „selbstverständlich“ und liegen automatisch immer vor. Solche Abhängigkeiten heißen trivial
- Trivial sind z. B.

- ▶ $A \rightarrow A$

Wer den Vornamen weiß, weiß den Vornamen (kein Tippfehler sondern Logik)

- ▶ $A \rightarrow A$

Dito. Da für Mathematiker eins ein Spezialfall von mehreren ist, gilt der erste Satz auch für MWA

- ▶ $(AB) \rightarrow A, (AB) \rightarrow B$

Wer den vollen Namen kennt, kennt auch den Vornamen (Nachnamen)

- ▶ $(AB) \rightarrow A, (AB) \rightarrow B$

Dito. Da für Mathematiker eins ein Spezialfall ...

- ▶ und weitere ...

Abhängigkeiten

Inferenzregelarithmetik: Rechnen mit Abhängigkeiten

- $B \subseteq A \quad \Rightarrow A \rightarrow B$ (Reflexivität/ Projektivität)
- $A \rightarrow B \quad \Rightarrow AC \rightarrow BC$ (Erweiterungsregel)
- $A \rightarrow B, B \rightarrow C \quad \Rightarrow A \rightarrow C$ (Transitivität)
- $A \rightarrow B, A \rightarrow C \quad \Rightarrow A \rightarrow BC$ (Vereinigungsregel)
- $A \rightarrow B, BC \rightarrow D \quad \Rightarrow AC \rightarrow D$ (Pseudotransitivität)
- $A \rightarrow B, C \subseteq B \quad \Rightarrow A \rightarrow C$ (Zerlegungsregel)
- $A \rightarrow B \quad \Rightarrow A \rightarrow T-(AB)$ (Komplement)
- $A \rightarrow B, D \subseteq C \quad \Rightarrow AC \rightarrow BD$ (Erweiterungsregel)
- $A \rightarrow B, B \rightarrow C \quad \Rightarrow A \rightarrow C - B$ (Transitivität)
- $A \rightarrow B \quad \Rightarrow A \rightarrow B$ (Replikation)
- $A \rightarrow B, AB \rightarrow C \quad \Rightarrow A \rightarrow C - B$ (Pseudotransitivität)
- $A \rightarrow B, C \rightarrow B, C \cap D = \{ \}, D \subseteq B \Rightarrow A \rightarrow B$ (Koaleszenz)
- Folgerbare Abhängigkeiten sind ebenfalls **trivial**

A, B, C stehen für Gruppen (Mengen) von Feldern, wobei nur ein Feld auch eine Gruppe ist. T steht für alle Felder einer Tabelle. T - A steht für die Mengendifferenz:

Angenommen $T = (ABCD)$, dann bedeutet $T - (AB)$ dasselbe wie (CD)

Die ersten drei Regeln sind auch unter dem Namen Armstrong-Axiome bekannt.

Die Regeln werden insbesondere benutzt, um beim algorithmischen Normalisieren aus bekannten Abhängigkeiten die Menge aller Abhängigkeiten – die sogenannte **Hülle** – herzuleiten. Diese wird dann mit denselben Regeln wieder reduziert auf die kleinstmögliche Menge von Abhängigkeiten, aus denen sich alle Abhängigkeiten erzeugen lassen – die sogenannte **kanonische Überdeckung**. Dadurch hat man es nur noch mit nichttrivialen, also sozusagen „echten“ Abhängigkeiten zu tun, die sachlich aus der Datenstruktur folgen.

Für Interessierte sei noch angemerkt, daß die obigen Regeln im wesentlichen ihre Entsprechungen im mehrstelligen Prädikatenkalkül finden.

Der Schlüsselbegriff

Definition Oberschlüssel (Schlüssel, Super Key)

- Eine Feldgruppe (auch ein einzelnes Feld), von der **alle** Felder einer Tabelle **funktional** abhängen, heißt Oberschlüssel (OS)
- Man nennt OS auch **Identifikator**

Definition Kandidatschlüssel (Schlüssel, Candidate Key)

- Ein Oberschlüssel, der **minimal** ist, d. h. von dem man keine Felder weglassen kann, ohne daß er seine Schlüsseleigenschaft verliert, heißt Kandidatschlüssel (KS). Von einem KS hängt die ganze Tabelle **voll funktional** ab
- Man nennt KS auch **Determinator** (=minimaler Identifikator)

Hinweis

- Kandidatschlüssel ist eine wesentlich stärkere Forderung als Ober(Super)schlüssel

Man stelle sich den OS als Schlüsselbund vor:

Daß sich damit die Haustür öffnen läßt, ist nicht das Verdienst des Bundes an sich, sondern der Tatsache, daß sich der Haustürschlüssel darunter befindet.

Eigenschaften von Kandidatschlüsseln

- Die Wertkombinationen von KS sind in einer Tabelle immer eindeutig
- Technisch wird ein KS als Eindeutiger Index **ohne** NULL-Werte realisiert
- Der Primärschlüssel ist ein (willkürlich) ausgewählter KS
- Weitere KS heißen Parallel- oder Sekundärschlüssel
- Zusätzliches künstliches Feld heißt Surrogatschlüssel
- Logisch gibt es zwischen Primär- und Sekundärschlüsseln keinen Unterschied. Der Primärschlüssel ist keinesfalls irgendwie „wichtiger“ als die Parallelschlüssel

Schlüsselfelder

- Felder, die Teil **irgendeines** KS sind, heißen Schlüsselfelder (SF)
- Alle anderen heißen Nichtschlüsselfelder (NSF)

Bedeutung von Schlüsseln/eindeutigen Indizes

- Identifikation von Datensätzen
 - ▶ Oft fälschlich als Hauptbedeutung angesehen
- Durchsetzen von Regeln (Business Rules) als Schlüsselbedingung (Key Constraint) (SB)
 - ▶ Wichtigste Funktion von Schlüsseln
 - ▶ Durch entsprechendes Setzen von (Mehrfelder-)Schlüsseln werden gewisse Wertkombinationen erlaubt oder verboten
 - ▶ Beinhaltet eigentliche Regellogik des Datenmodells, die Restriktionen der abzubildenden Realität nachformt
 - ▶ Sicherstellung der **existentiellen Integrität**
- In Form von Fremdschlüsseln (Foreign Keys) (FS)
 - ▶ Abbilden der Tabellenbeziehungen
 - ▶ Überwachung der **referentiellen Integrität**

- Machen Client-Programmierung größtenteils überflüssig

Noch ein Wort zur Regelmodellierung durch eindeutige Indizes (UI): Die (leider) oft verwendete Alternative, Regeln über Programm-Code im Client durchzusetzen, entspricht in ihrem Vorgehen den überholten Textdatenbanken. Wenn man eine relationale DB modellieren will, sollte man die definitiven Forderungen der relationalen Theorie auch umsetzen. Man würde sich auch kein Auto kaufen, um es dann von Pferden ziehen zu lassen.

Beispiele:

UI (Student; Fach): Student darf Prüfung nur einmal machen

UI (Student; Fach; Semester): Student darf Prüfung einmal jedes Semester wiederholen

In Stammdaten („natürlichen Daten“) sind Mehrfelderschlüssel selten.

Sie treten meist in Beziehungstabellen oder in Form von Constraints auf

2. Normalform

Tableauschema, falsch

| MNr | VNr | Thema | StTel | Note | Fehltage |
|-----|-----|------------|-------|------|----------|
| 112 | 17 | Informatik | 12345 | 1 | 3 |
| 112 | 23 | Atomphysik | 12345 | 2 | 0 |
| 98 | 17 | Informatik | 45678 | 4 | 5 |
| 105 | 23 | Atomphysik | 91234 | 3 | 2 |

Michael Zimmermann: DB-Normalisierung

MNr: Matrikelnummer, Identifikator für Student

VNr: Veranstaltungsnummer, Identifikator für Vorlesung/Seminar

(MNr, VNr) farbig hinterlegt: Kandidatschlüsselfelder (SF)

Andere Felder grau hinterlegt: gewöhnliche Datenfelder (NSF)

Thema der Veranstaltung

StTel: Telephonnummer, Student

Note dieses Studenten in dieser Veranstaltung

Fehltage dieses Studenten in dieser Veranstaltung

Probleme:

Note und Fehltage hängen vom ganzen Schlüssel (MNr, VNr) ab. Thema hängt jedoch schon vom Schlüsselteil VNr ab, ebenso StTel von MNr

In der Folge gibt es bei Thema Redundanzen („Wiederholungsgruppen“), z. B. zweimal Informatik, da die Veranstaltung von zwei Studenten belegt wird.

Der Begriff der Wiederholungsgruppe ist beim Entwurf allerdings nicht hilfreich: In der Entwurfsphase gibt es noch gar keine Daten und damit auch keine Wiederholungen. Außerdem kann es sein, daß zunächst für jede Veranstaltung nur ein Student angemeldet ist. Obwohl der Entwurf fehlerhaft ist, würden sich dennoch zunächst keine Wiederholungen zeigen. Obendrein ist eine vorhandene Wiederholung z. B. gleicher Nachnamen in aller Regel (außer in der DKNF) kein Fehler.

An der Beschäftigung mit Kandidatschlüsseln und Abhängigkeiten auf rein logischer Ebene führt also kein Weg vorbei.

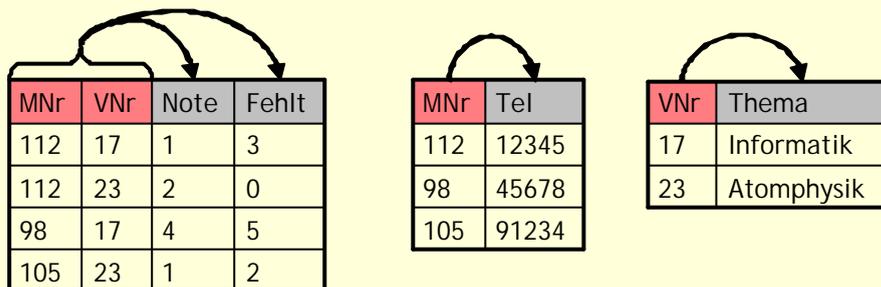
Anomalien dieses Entwurfs unter anderem:

- Vorlesung, zu der noch keine Teilnehmer erfaßt sind, kann nicht angelegt werden
- Wenn alle Teilnehmer einer Vorlesung gelöscht werden, geht auch die Information über die Vorlesung verloren
- Wenn zu VNr 17 fälschlich einmal Informatik und einmal Chemie eingegeben wird, weiß keiner, welche Information richtig ist

Das Schema drückt die Abhängigkeiten $MNr \rightarrow StTel$ und $VNr \rightarrow Thema$ in keinsten Weise aus: Wir wissen, daß Thema allein von Vorlesung (entsprechend StTel von Student) und nicht von der Kombination Vorlesung/Student abhängt. Aber: Wo kommt das im Tabellenschema zum Ausdruck? Wie wird verhindert, daß zu einer VNr zwei verschiedene Themen eingegeben werden?

Folge: Das Tabellenschema entspricht nicht der logischen Struktur der Realität, ist also fehlerhaft.

Tableauschema, richtig



Michael Zimmermann: DB-Normalisierung

Nach der Zerlegung treten nur noch voll funktionale Abhängigkeiten von Kandidaten auf, alle Anomalien sind getilgt, alle Restriktionen durchgesetzt.

Die einzige (erlaubte) Redundanz: ist die **Fremdschlüsselwiederholung**, z. B. VNr 17, VNr 17, die durch referentielle Integrität abgesichert ist.

Attributschemata

Datenbeispiele sind gut und schön, aber um allen Möglichkeiten, die dem DB-Entwickler begegnen können abzudecken, muß man sich abstrahiert von allen konkreten Daten die allgemeinen logischen Strukturen klarmachen, ebenso wie man die binomischen Formeln nicht durch viele Beispiele sondern durch allgemeine Regeln und Variablen begriffen hat.

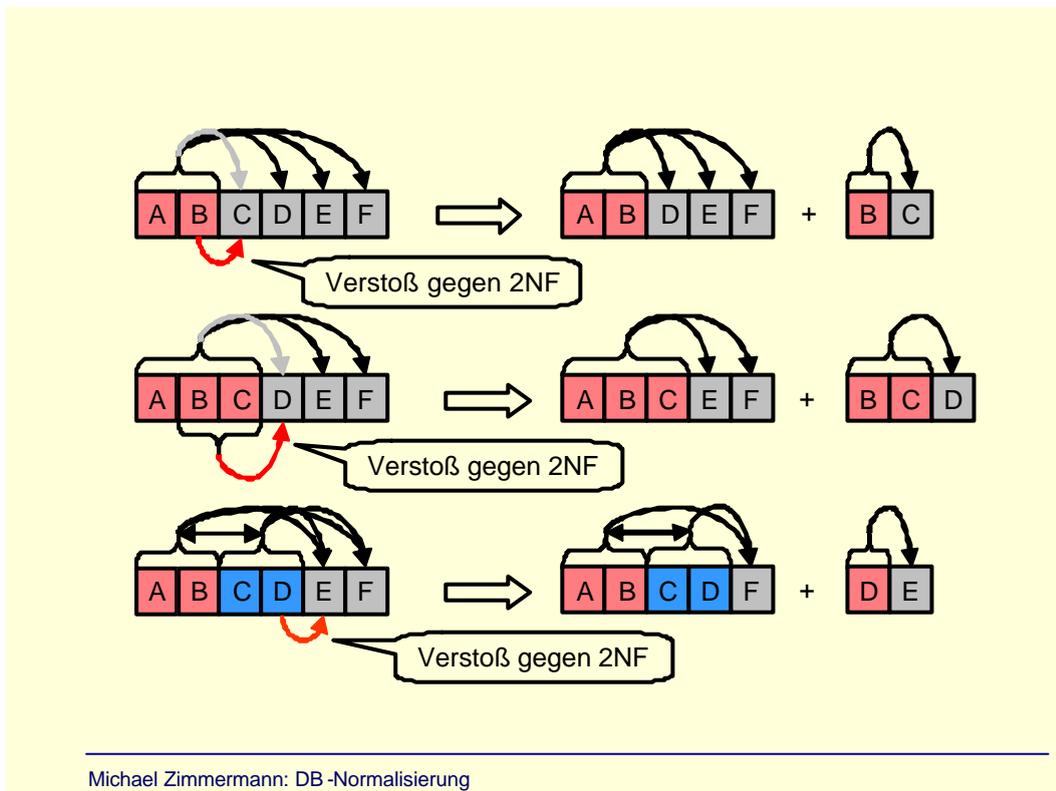
Daher habe ich wie oben abstrahierte Attributschemata entworfen, die den Sachverhalt aus dem Beispiel in logischer Reinheit verdeutlichen sollen.

Es sind im folgenden immer ABCDEF Felder (Spalten) im Tabellenentwurf

Rote/blau/gelbe Hinterlegung kennzeichnet Kandidatschlüsselfelder (Unique Index ohne NULL), graue Hinterlegung normale Daten. Geschweifte Klammern fassen Felder zu Feldgruppen zusammen.

Die als Pfeile dargestellten Abhängigkeiten sind **logisch**; sie ergeben sich aus der Realität (Restriktionen). Rote Pfeile bedeuten, daß diese logische Abhängigkeit in der Tabelle eben nicht realisiert ist, schwarze Pfeile bedeuten, daß diese Abhängigkeit durch Indizes in der Tabelle auch ausgedrückt wird.

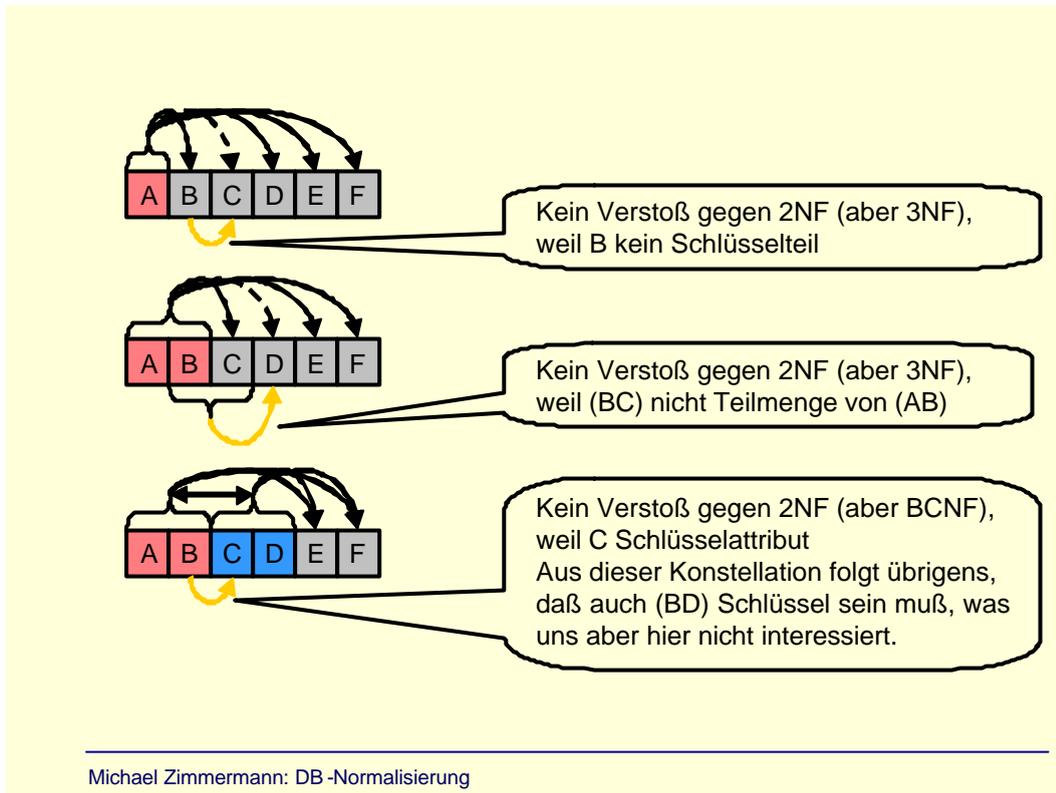
Attributschema, Verstoß



1. $B \rightarrow C$ verboten, weil B Schlüsselteil (partielle FA)
2. $(BC) \rightarrow D$ verboten, weil (BC) Schlüsselteil (partielle FA)
3. Gilt auch, wenn partielle FA von **Sekundärschlüssel** (Index) ausgeht!

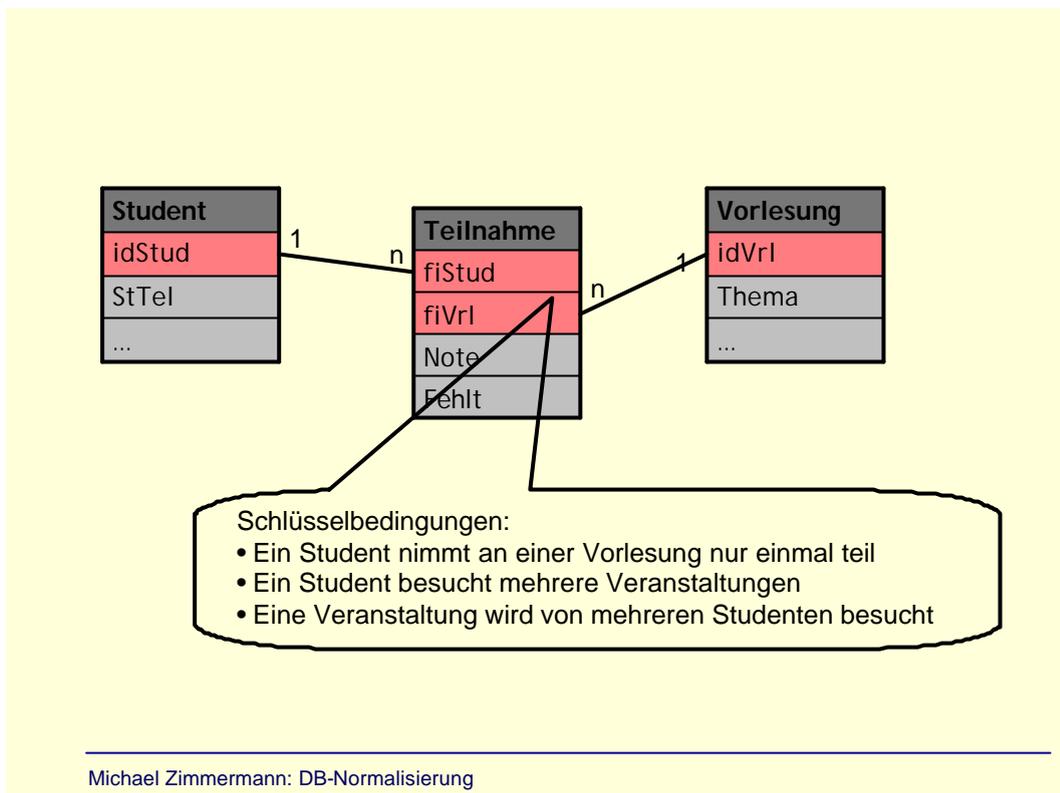
Lösung: Partielle FA in eigene Tabelle auslagern

Attributschema, kein Verstoß



Diese Beispiele sollen nur verdeutlichen, was genau die 2 NF fordert. Die oben gelb gefärbten Abhängigkeiten verstoßen gegen höhere NF und können selbstverständlich nicht belassen werden.

Beziehungsschema



Das Attributschema noch einmal mit dem Ausgangsbeispiel in vertrauter Ansicht als Beziehungsschema bzw. SQL-Server-Diagramm. Nur durch Index/Primärschlüssel über Student(fiStud)/Vorlesung(fiVrl) sind alle Regeln durchgesetzt.

Definition

- **Jedes** Nichtschlüsselfeld hängt voll funktional von **jedem** Kandidatschlüssel ab

Bedeutung

- Es darf nicht sein, daß ein Datenfeld von einem Teilfeld eines Mehrfelderkandidaten abhängt
- Da die Werte in einem Kandidatenteilfeld nicht eindeutig sein müssen, könnten sich sonst systematische Wiederholungsgruppen im abhängigen Feld bilden

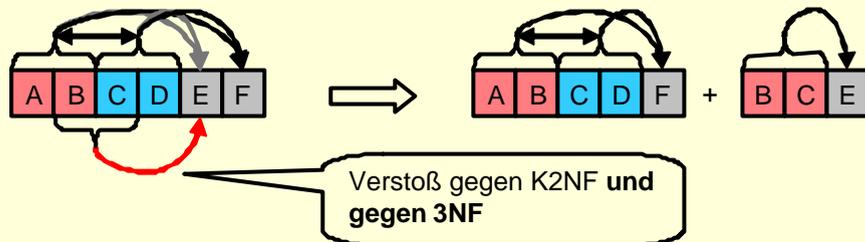
Hinweise

- Es genügt **nicht**, diese Forderung nur für den Primärschlüssel zu erfüllen, wie es oft falsch in Büchern angegeben wird
- Abhängigkeiten von 1-Feld-Kandidaten sind immer voll, 2 NF automatisch erfüllt

Kentsche 2. Normalform

Definition

- **Jedes Feld**, hängt voll funktional von **jedem** Kandidatschlüssel ab, sofern es nicht Teilmenge desselben ist.



Michael Zimmermann: DB-Normalisierung

- Nur der Vollständigkeit halber erwähnt
- 2 NF wird von Nichtschlüselfeld auf Feld (also auch Schlüsselfelder) verschärft
- Praktisch bedeutungslos, da in 3 NF enthalten

3. Normalform

Tableauschema, falsch

| idAuf | idKd | KdTel | KdName | AufDatum |
|-------|------|-------|-------------|------------|
| 112 | 37 | 12345 | Meier AG | 10.05.2003 |
| 117 | 48 | 45678 | Schmitt OHG | 17.07.2003 |
| 123 | 37 | 12345 | Meier AG | 14.11.2003 |
| 124 | 37 | 12345 | Meier AG | 20.12.2003 |

Michael Zimmermann: DB-Normalisierung

idAuf: Auftragsnummer

idKd: Kundennummer

KdTel: Kunde, Telephon

KdName: Kunde, Name

AufDatum: Auftragsdatum

Probleme wie bei 2 NF:

Das Schema drückt die Abhängigkeiten $KdNr \rightarrow KdTel, KdName$ nicht aus.

Da $idAuf$ alleine Schlüssel (Kandidat) ist, liegt aber kein 2NF-Verstoß vor.

Trotzdem vergleichbare Probleme, zum Beispiel Wiederholungsgruppen $KdTel, KdName$

Die funktionalen Abhängigkeiten (FA)

- $idAuf \rightarrow KdName$
- $idAuf \rightarrow KdTel$

bestehen nur indirekt über $idAuf \rightarrow idKd + idKd \rightarrow KdName$ etc.

Tableauschema, richtig

| idAuf | fiKd | AufDatum |
|-------|------|------------|
| 112 | 37 | 10.05.2003 |
| 117 | 48 | 17.07.2003 |
| 123 | 37 | 14.11.2003 |
| 124 | 37 | 20.12.2003 |

| idKd | KdName | KdTel |
|------|-------------|-------|
| 37 | Meier AG | 12345 |
| 48 | Schmitt OHG | 45678 |

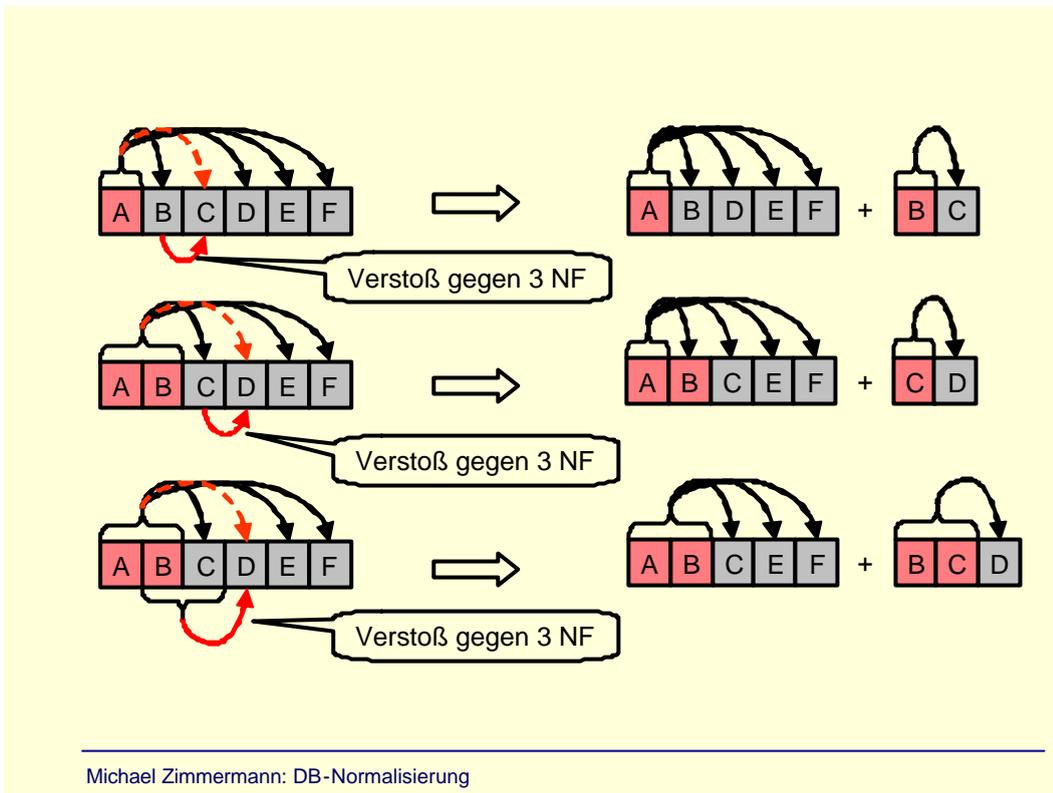
Michael Zimmermann: DB-Normalisierung

Nur noch direkte voll funktionale Abhängigkeiten von Kandidaten

Alle Anomalien getilgt

Nur noch Fremdschlüsselwiederholung fiKd

Attributschema, Verstoß

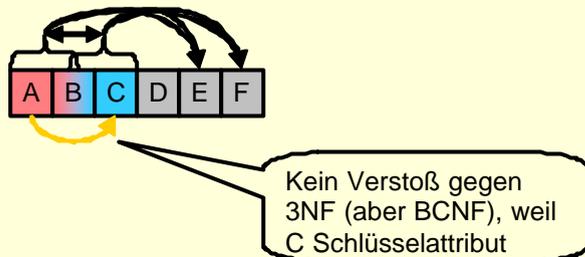


1. $B \rightarrow C$ verboten, weil B kein Kandidat
2. $C \rightarrow D$ verboten, weil C kein Kandidat
3. $(BC) \rightarrow D$ verboten, weil BC kein Kandidat

Lösung

Indirekte FA in eigene Tabelle auslagern

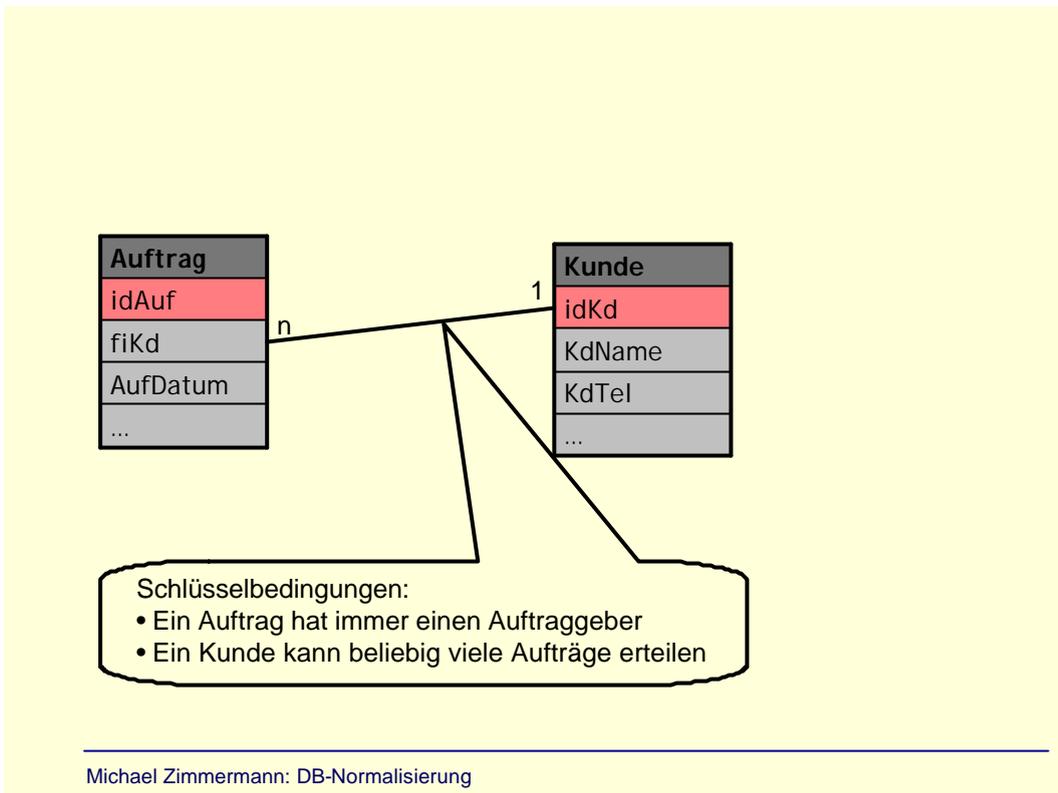
Attributschema, kein Verstoß



Michael Zimmermann: DB-Normalisierung

Das Beispiel soll wieder nur klären, was die 3 NF verbietet und was nicht. Hier läge zwar kein 3-NF-Verstoß vor, weil C ein Schlüsselfeld ist. Dennoch kann das nicht so bleiben und wird dann auch prompt in der BCNF verboten.

Beziehungsschema



Nur durch Tabellenstruktur/Indexverteilung Regel durchgesetzt:

1 Kunde: n Aufträge

1 Auftrag: 1 Auftraggeber

Definition

- **Jedes** Nichtschlüselfeld hängt direkt (nicht transitiv) von **jedem** Kandidatschlüssel ab

oder

- Kein **Nichtschlüselfeld** ist funktional abhängig von einer Feldgruppe ohne Oberschlüsseleigenschaft. d. h. diese muß einen vollständigen Schlüsselkandidaten beinhalten.

Bedeutung

- Was bei der 2 NF für Schlüsselteile verboten wurde, wird auch für Schlüsselteile, ergänzt um weitere Felder, verboten
- Oft hängt ein NSF von einem anderen NSF und somit nur indirekt (transitiv) von Kandidatschlüsseln ab

- Es gibt aber auch den Fall (Kent), daß die transitive Abhängigkeit ohne Beteiligung von NSF entsteht
- In der Folge bilden sich in beiden Fällen systematische Wiederholungsgruppen im abhängigen Feld

Hinweise

- Es genügt **nicht**, diese Forderung nur für den Primärschlüssel zu erfüllen, wie es oft falsch in Büchern angegeben wird
- Es stimmt nicht, wie gelegentlich zu lesen, daß es nur um Abhängigkeit von NSF ginge

Boyce-Codd-Normalform 1

Tableauschema, falsch

| Prüfer | Student | Fach | Note |
|--------|---------|------------|------|
| 117 | 22 | Mathe | 1 |
| 117 | 37 | Mathe | 2 |
| 117 | 56 | Mathe | 3 |
| 83 | 22 | Physik | 3 |
| 83 | 56 | Physik | 2 |
| 15 | 22 | Informatik | 2 |

Michael Zimmermann: DB-Normalisierung

- Folgende Bedingungen sollen modelliert werden:

1. Ein Student wird in mehreren Fächern geprüft
2. ... in einem Fach aber nur einmal
3. Ein Prüfer prüft viele Studenten
4. Ein Student hat mehrere Prüfer
5. Jeder Prüfer prüft genau ein Fach
6. Daher treffen jeder Prüfer und jeder Student nur einmal zusammen.

7. Jedes Fach hat nur genau einen Prüfer

Unique Index (Student, Fach) modelliert 1, 2

Unique Index (Student, Prüfer) modelliert 3, 4, 6

Probleme

Die Abhängigkeit $P \leftrightarrow F$ (5 und 7) ist nicht modelliert; es wäre z. B. möglich, einem P mehrere F zuzuordnen

Fächer können nur in den Katalog aufgenommen werden, wenn sie wenigstens einmal geprüft wurden

Prüferkompetenzen können nur gespeichert werden, wenn eine entsprechende Prüfung existiert

Gleiche Probleme wie bei 3NF: BCNF darf nicht ignoriert werden

Tableauschema, richtig

| Student | Fach | Note |
|---------|------------|------|
| 22 | Mathe | 1 |
| 37 | Mathe | 2 |
| 56 | Mathe | 3 |
| 22 | Physik | 3 |
| 56 | Physik | 2 |
| 22 | Informatik | 2 |

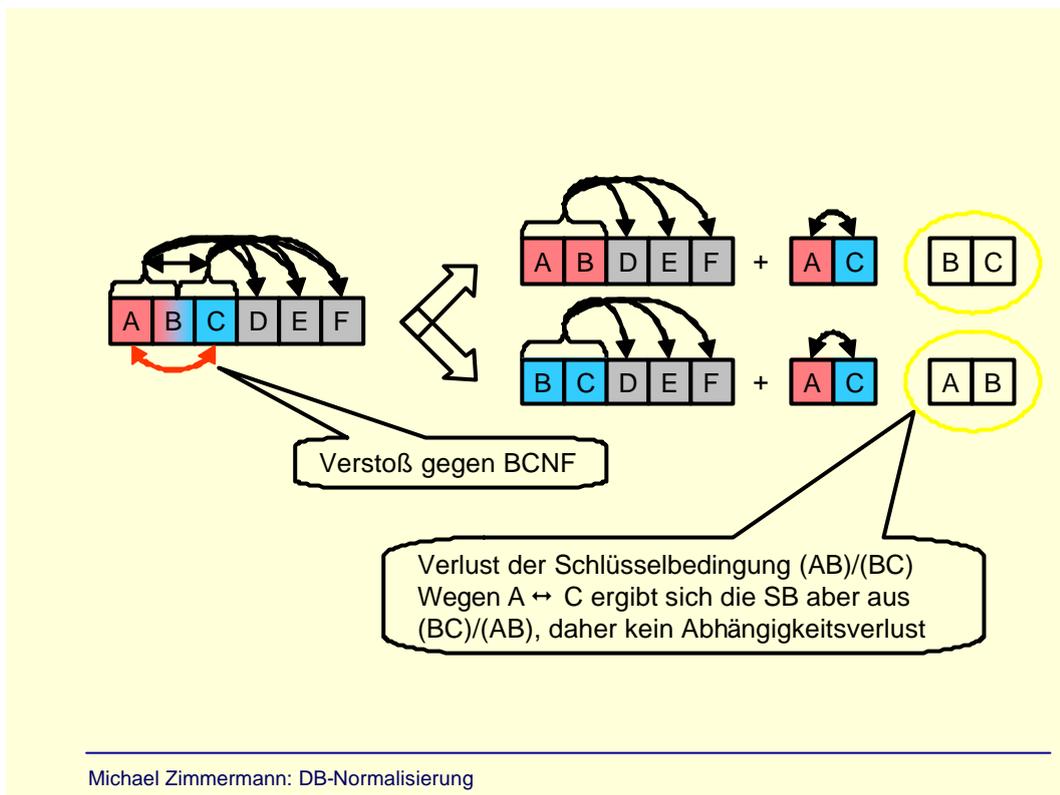
| Prüfer | Fach |
|--------|------------|
| 117 | Mathe |
| 83 | Physik |
| 15 | Informatik |

Michael Zimmermann: DB-Normalisierung

Alle Redundanzen und Folgeprobleme sind beseitigt

Die Schlüsselbedingung (Student, Prüfer) ist **indirekt sichergestellt**, die $FA P \leftrightarrow F$ ist modelliert, da Student-Fach eindeutig und Prüfer-Fach 1:1 ist, ist auch Student-Prüfer eindeutig

Attributschema, der harmlose Verstoß

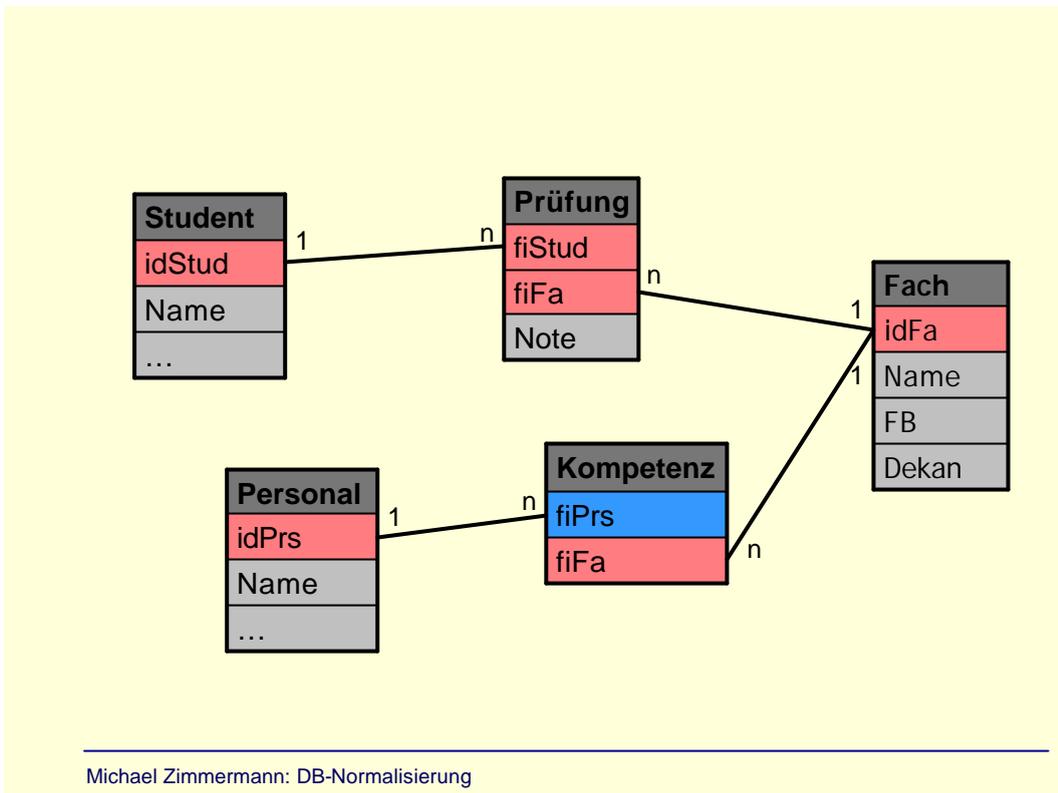


Die Aufteilung kann wegen der Symmetrie $A \leftrightarrow C$ beliebig nach dem ersten oder zweiten Schema erfolgen.

Man bevorzugt die Variante mit besserer Semantik, also die anschaulicher eherden Gegebenheiten entspricht.

Im Beispiel (SF) und $F \leftrightarrow P$ statt (SP) und $P \leftrightarrow F$, da man eher sagen würde, ein Student wird in einem Fach geprüft, und der Prüfer ergibt sich. Die logisch gleichwertige Variante: Ein Student hat einen bestimmten Prüfer, das Fach ergibt sich dann schon daraus, bürstet zumindest den Alltagsverstand gegen den Strich.

Beziehungsschema



Die zunächst zusammengelegten Tabellen waren Prüfung-Kompetenz in einer Tabelle.

Wir sehen: BC-Verstöße scheinen vorwiegend Referenz-(Beziehungs-)tabellen zu betreffen und die zunächst falsch modellierte ternäre Beziehung zwischen den drei Entitäten (Objekten) Student, Fach und Prüfer erweist sich als zwei binäre Beziehungen zwischen Student, Fach einerseits und Fach, Prüfer andererseits.

Boyce-Codd-Normalform 2

Tableauschema, falsch

| Prüfer | Student | Fach | Note |
|--------|---------|------------|------|
| 117 | 22 | Mathe | 1 |
| 113 | 37 | Mathe | 2 |
| 117 | 56 | Mathe | 3 |
| 83 | 22 | Physik | 3 |
| 87 | 56 | Physik | 2 |
| 15 | 22 | Informatik | 2 |

Michael Zimmermann: DB-Normalisierung

- Folgende Bedingungen sollen modelliert werden:

1. Ein Student wird in mehreren Fächern geprüft
2. ... in einem Fach aber nur einmal
3. Ein Prüfer prüft viele Studenten
4. Ein Student hat mehrere Prüfer
5. Jeder Prüfer prüft genau ein Fach
6. Daher treffen jeder Prüfer und jeder Student nur einmal zusammen.

7. Fächer haben mehrere Prüfer

Man beachte 7; das ist der einzige – aber entscheidende – Unterschied zum ersten Beispiel

Probleme:

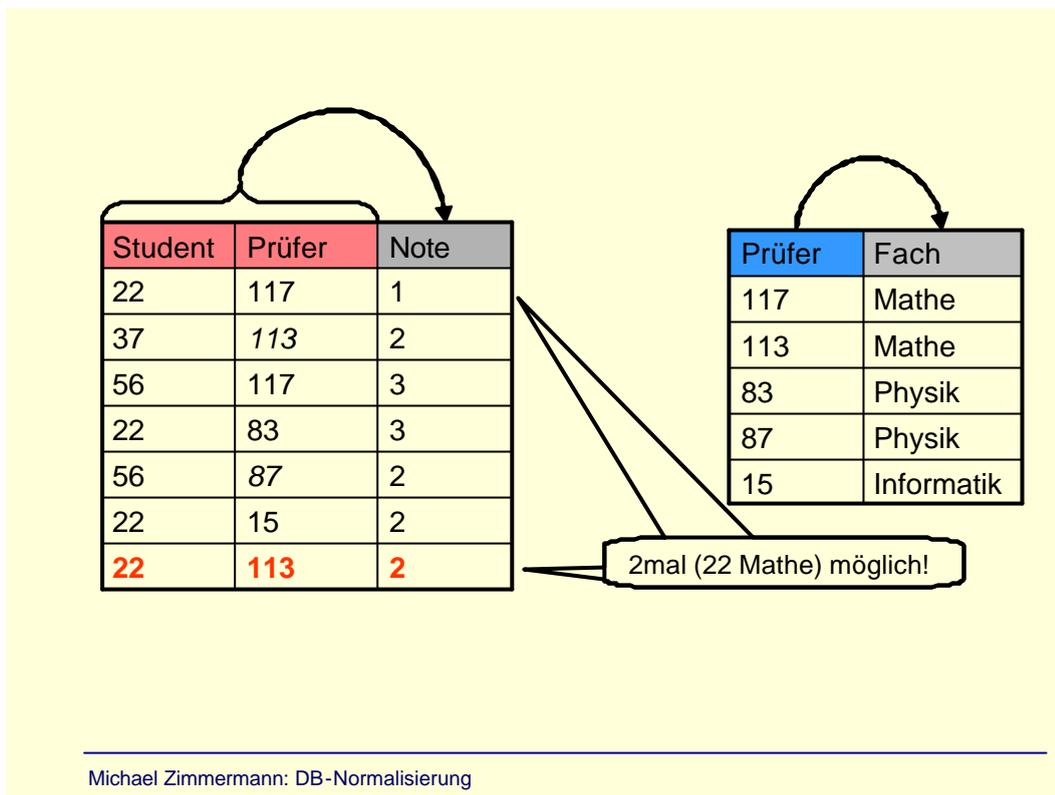
Die Abhängigkeit $P \rightarrow F$ ist nicht modelliert; es wäre z. B. möglich, einem F mehrere P zuzuordnen.

Fächer können nur in den Katalog aufgenommen werden, wenn sie wenigstens einmal geprüft wurden.

Prüferkompetenzen können nur gespeichert werden, wenn eine entsprechende Prüfung existiert.

Also zunächst gleiche Probleme wie bei Fall 1.

Tableauschema, richtig



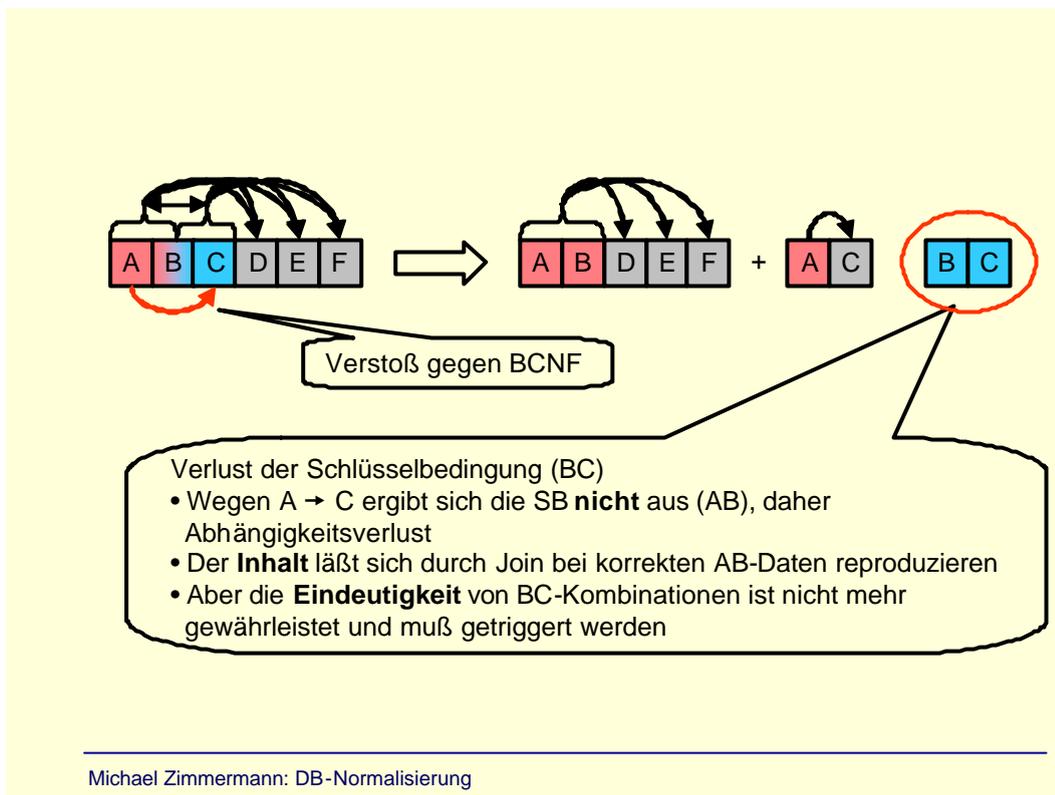
Alle Redundanzen und Folgeprobleme sind beseitigt.

Alle Student-Fach-Kombinationen lassen sich über den Prüfer erschließen, die FA $P \rightarrow F$ ist modelliert.

Aber es taucht ein **neues Problem** auf: Die Schlüsselbedingung (Unique Index) Student-Fach ist nicht mehr gesichert.

Es könnte in der obigen Konstellation der Student mit der Matrikelnummer 22 die Prüfer mit der Personalnummer 117 und 113 bekommen. Da beide Mathematiker sind, würde die Tabelle zulassen, daß 22 die Mathe-Prüfung zweimal ablegt. Wir haben sofort die Inkonsistenz (logischer Widerspruch), daß wir nicht wissen, ob die Mathe-Note von 22 jetzt 1 oder 2 beträgt.

Attributschema, der schlimme Verstoß



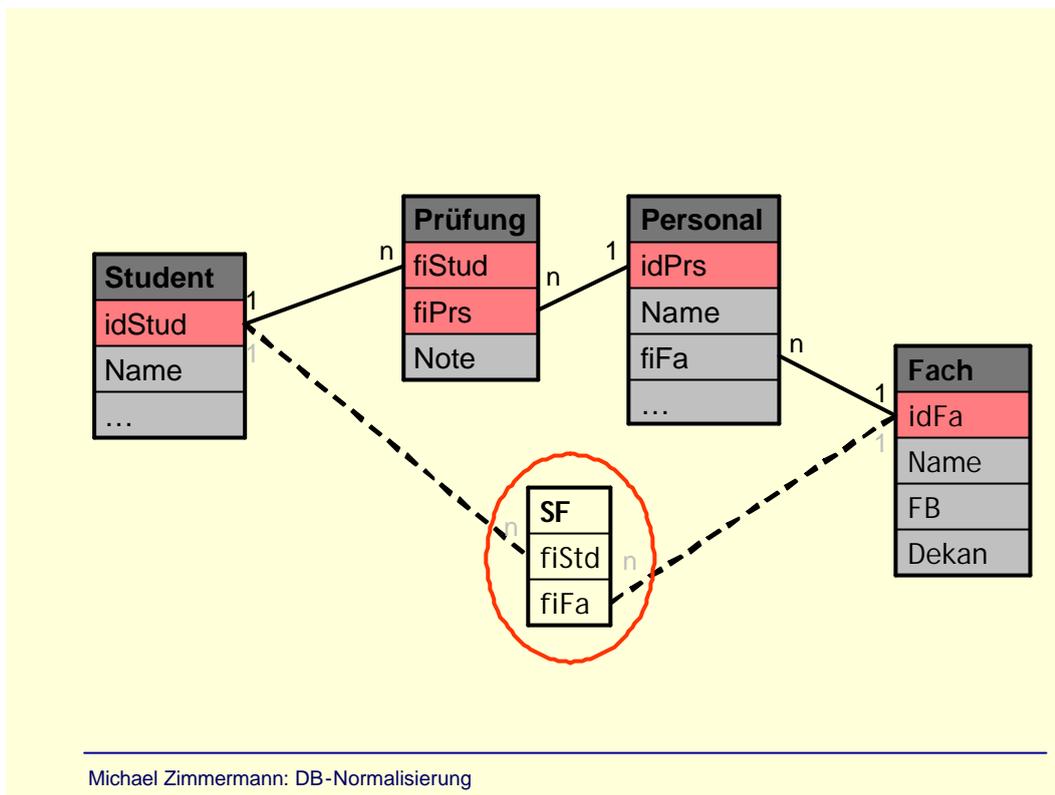
Der Verstoß tritt auf, wenn eine FA in **einer Richtung** zwischen den äußeren Feldern zweier überlappender Kandidatschlüssel vorliegt.

Man zerlegt so, daß Zielfeld der FA – hier C – aus der Tabelle entfernt wird, und legt eine zweite Tabelle mit der BC-Abhängigkeit – hier $A \rightarrow C$ – an.

Der Verlust der Schlüsselbedingung (BC) wird notgedrungen vorläufig in Kauf genommen.

Natürlich muß die Einhaltung anderweitig sichergestellt werden.

Beziehungsschema



- SF prüfen

- ▶ Per VBA-Programmierung
- ▶ Tabelle SF mit Index führen und per VBA/SQL beschreiben (unschöne Lösung, da Abhängigkeit doppelt modelliert)
- ▶ Mit Trigger
- ▶ Tabelle SF mit Index führen und per Trigger beschreiben (unschöne Lösung, da Abhängigkeit doppelt modelliert)
- ▶ Über


```
SELECT SP.S, Fächer.F FROM Student INNER JOIN ((Fächer INNER JOIN
Personal ON Fächer.F = Personal.F) INNER JOIN SP ON Personal.P = SP.P) ON
Student.S = SP.S;
```

 einen View-Index legen (SQL-Server ab 2000)
- ▶ Abfrage `SELECT COUNT(*) FROM (SELECT SP.S, Fächer.F FROM Student INNER JOIN ((Fächer INNER JOIN Personal ON Fächer.F = Personal.F) INNER JOIN SP ON Personal.P = SP.P) ON Student.S = SP.S;) GROUP BY SP.S, Fächer.F;` muß bei Neuanlage DS mit Parameter fiStud (S), fiFa (F) 0 (Zero) ergeben

Boyce-Codd-Normalform

Definition

- Jedes Feld hängt direkt (nicht transitiv) von jedem Kandidatschlüssel ab, falls es nicht selbst Bestandteil des Kandidaten ist

oder

- Kein Feld ist funktional abhängig von einer Feldgruppe ohne Oberschlüsseleigenschaft. d. h. diese muß einen vollständigen Schlüsselkandidaten beinhalten.

Hinweis: Bei der 3 NF hieß es: Kein **Nichtschlüsselfeld**

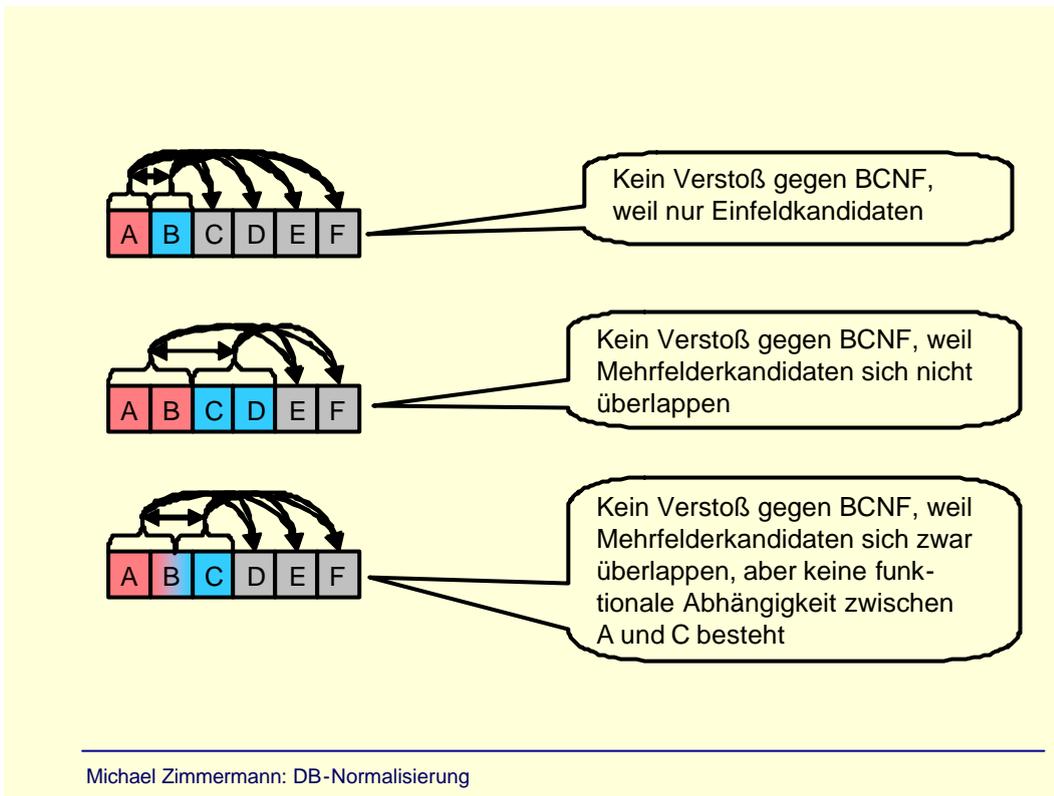
Bedeutung

- 3 NF wird auf alle Felder verschärft – auch Schlüsselfelder

Hinweis

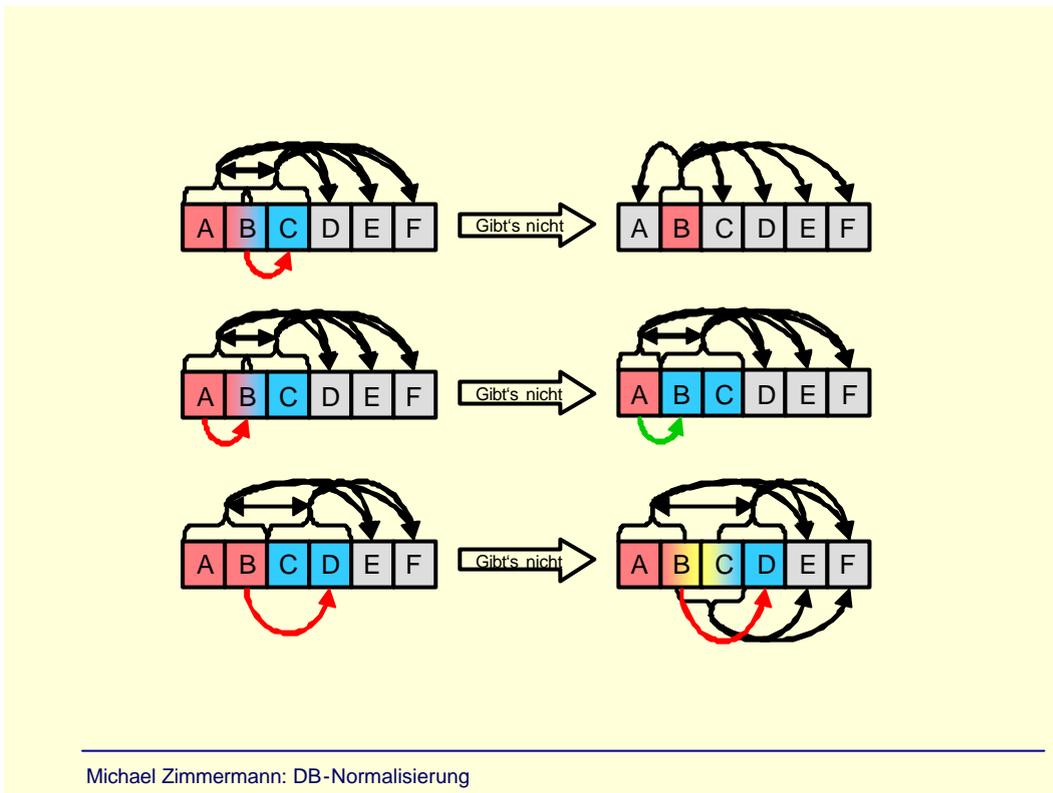
- Eine BCNF-Verletzung kann nur unter folgenden Bedingungen eintreten: Es gibt ...
 - ▶ wenigstens zwei Kandidatschlüssel ...
 - ▶ mit je wenigstens zwei Feldern, ...
 - ▶ die sich auch in wenigstens einem Feld überlappen ...
 - ▶ mit funktionaler Abhängigkeit zwischen äußeren Feldern
 - ▶ In allen anderen Fällen ist die 3 NF automatisch auch BCNF
- Was zunächst wie eine einfache kleine Definitionserweiterung und von abstrusem Seltenheitswert erscheint,
 - ▶ hat aber weitreichende Folgen und
 - ▶ tritt (unerkannt) öfter auf, als einem lieb sein kann

Attributschema, kein Verstoß möglich



Einige Beispiele, die nicht gegen BCNF verstoßen können, also 3 NF = BCNF

Einige Verwirrspiele und ihre Auflösung



Nach der Definition müßten obige Beispiele eigentlich gegen BCNF verstoßen

Aber mit **Inferenzregularithmetik** läßt sich ausrechnen, daß diese Konstellationen so nicht existieren können

$$1. (AB) \rightarrow T; [(BC) \rightarrow T; B \rightarrow C] \Leftrightarrow B \rightarrow T \Leftrightarrow \neg[(AB) \rightarrow T]$$

Wenn BC die ganze Tabelle bestimmt und B C bestimmt, dann bestimmt B alleine schon die ganze Tabelle. Wenn B alleine aber schon Kandidat ist, ist AB nur Oberschlüssel und kein Kandidat.

$$2. (AB) \rightarrow T; A \rightarrow B \Leftrightarrow A \rightarrow T \Leftrightarrow \neg[(AB) \rightarrow T]$$

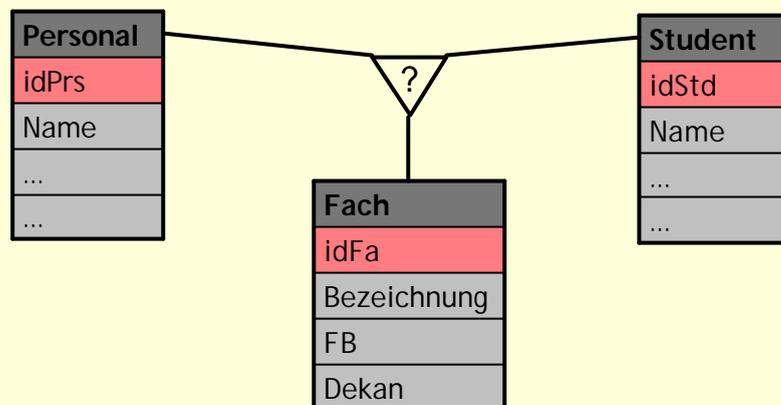
Wenn AB die ganze Tabelle bestimmt und A B bestimmt, dann bestimmt A alleine schon die ganze Tabelle. Wenn A alleine aber schon Kandidat ist, ist AB nur Oberschlüssel und kein Kandidat.

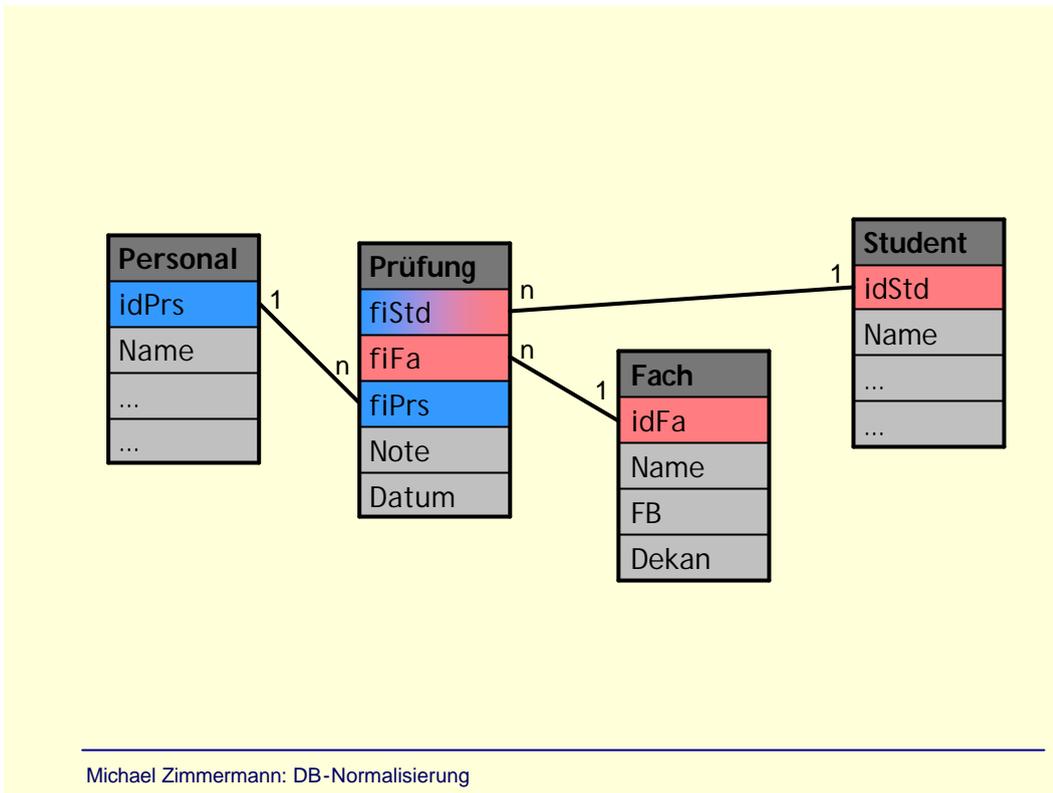
$$3. (CD) \rightarrow T; B \rightarrow D \Leftrightarrow (BC) \rightarrow T$$

Wenn CD die ganze Tabelle bestimmt und B D bestimmt, dann bestimmt auch BC die ganze Tabelle, ist also Kandidat. Daher liegen doch überlappende Schlüssel vor – sogar gleich drei.

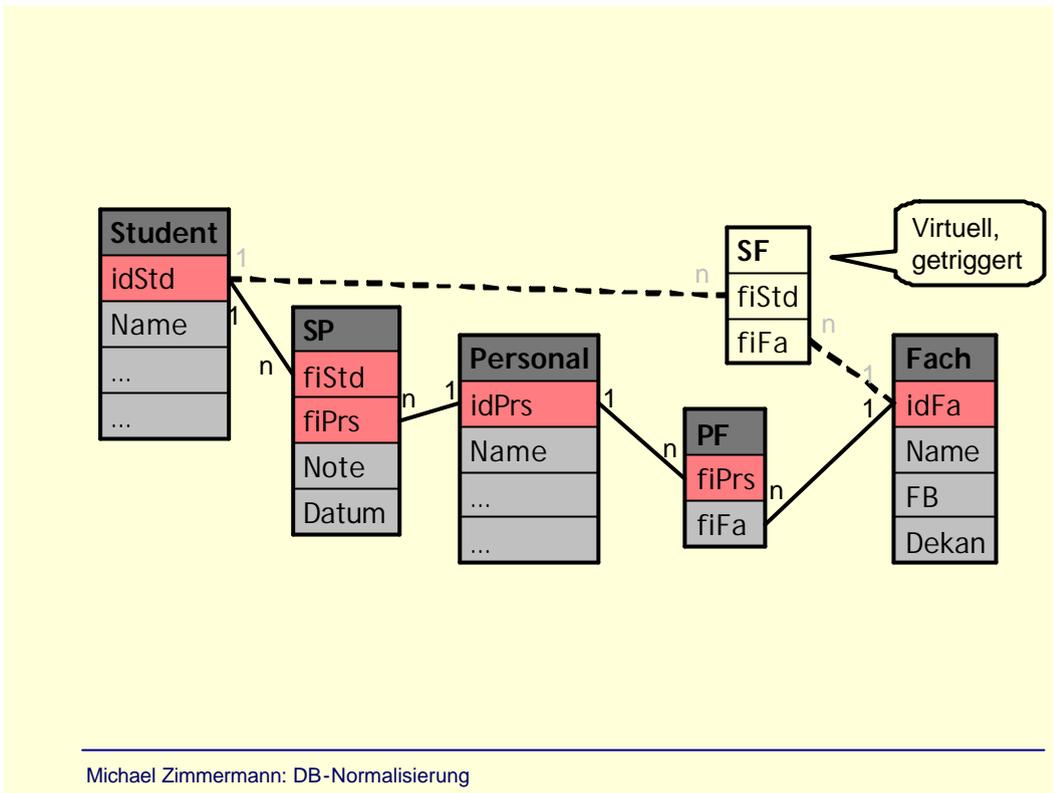
Auftreten von BCNF-Verletzungen

- Sind mehrere überlappende Mehrfelderschlüssel realistisch?
- Ja: typisch für komplexe Beziehungen, die fälschlich als ternär (dreiwertig) oder n-är (mehrwertig) angesehen werden
- Wie hängen Personal, Studenten und Fächer zusammen?





- Bedingungen wie vorhin, aber jeder kann alles prüfen, d. h. Prüfer – Fach sind unabhängig voneinander
- Folgende Bedingungen sollen modelliert werden:
 1. Ein Student wird in mehreren Fächern geprüft
 2. ... in einem Fach aber nur einmal
 3. Ein Prüfer prüft viele Studenten
 4. Ein Student hat mehrere Prüfer
 5. **Jeder Prüfer prüft alles**
 6. Prüfer und Student dürfen nur jeweils einmal zusammentreffen, um Voreingenommenheiten auszuschließen
 7. Ein Fach kann von jedem geprüft werden
- Die Einmaligkeit von (SP) ist jetzt keine Folge mehr aus (PF), sondern eine eigenständige Forderung
- Prüfung ist hier eine ternäre Beziehung (Wer ist von wem in was geprüft worden), der Aufbau völlig okay. Der Grund hierfür ist die in diesem Beispiel geforderte Regel 5



- Im Beispiel, das BCNF verletzt, liegen aber drei binäre Fakten vor
 - ▶ Welcher Student wird in welchem Fach geprüft (SF)
 - ▶ Welcher Student ist bei welchem Prüfer (SP)
 - ▶ Welcher Prüfer ist für welche Fächer kompetent (PF)

Das ist nicht identisch mit „Wer ist bei wem in welchem Fach geprüft worden“.

Rechne mit BCNF-Verstößen in falsch modellierten 3- oder Mehrfach-**beziehungen!**

4. Normalform

Tableauschema, falsch

| PrsNr | Sprachen | Sport |
|-------|-------------|--------|
| 5 | Englisch | Golf |
| 5 | Französisch | Golf |
| 5 | Englisch | Reiten |
| 5 | Französisch | Reiten |
| 7 | Französisch | Tennis |
| 7 | Latein | Tennis |
| 7 | Italienisch | Tennis |
| 7 | Französisch | Golf |
| 7 | Latein | Golf |
| 7 | Italienisch | Golf |

| PrsNr | FilialNr | TNr |
|-------|----------|-----|
| 112 | 3 | 12 |
| 112 | 3 | 17 |
| 112 | 7 | 12 |
| 112 | 7 | 17 |
| 205 | 5 | 12 |
| 205 | 5 | 23 |
| 205 | 5 | 37 |
| 205 | 7 | 12 |
| 205 | 7 | 23 |
| 205 | 7 | 37 |

Michael Zimmermann: DB-Normalisierung

Die beiden Beispiele Personalnummer, Sprachen, Sport sowie Personalnummer, Filialnummer, Tätigkeitsnummer sind völlig unabhängig aufeinander und nicht aufeinander bezogen. Es ist daher Absicht, daß verschiedene PrsNr'n verwendet wurden.

Probleme:

- Für jede neue Information müssen mehrere Datensätze angelegt werden
- Anzahl der DS wächst potentiell statt linear
- Sprachkenntnisse von Leuten, die keinen Sport treiben, können nicht eingetragen werden

Das Beispiel Sprache /Sport ist zwar **einsichtig**, aber **unsinnig**, da die Eigenschaften nicht nur unabhängig sondern völlig **zusammenhanglos** sind.

Da derartige Beispiele bei der 4 NF oft verwendet werden, wollte ich darauf zumindest hinweisen.

Ein weit besseres Beispiel ist Personal-Filiale -Tätigkeit mit

- $P \leftrightarrow F$: Personal – Filiale wechselseitig mehrwertig
- $P \leftrightarrow T$: Personal – Tätigkeit wechselseitig mehrwertig
- **F ••• T**: Filiale – Tätigkeit unabhängig

..., da jeder alle seine zugelassenen Tätigkeiten in allen ihm zugeordneten Filialen ausüben kann. Da aber durchaus ein semantischer Zusammenhang besteht („Der macht das da“), könnte man auf die falsche Idee kommen, die Informationen in eine Tabelle zu packen.

Tableauschema, richtig

| PrsNr | Sprachen |
|-------|-------------|
| 5 | Englisch |
| 5 | Französisch |
| 7 | Französisch |
| 7 | Latein |
| 7 | Italienisch |

| PrsNr | Sport |
|-------|--------|
| 5 | Golf |
| 5 | Reiten |
| 7 | Tennis |
| 7 | Golf |

| PrsNr | FilialNr |
|-------|----------|
| 112 | 3 |
| 112 | 7 |
| 205 | 5 |
| 205 | 7 |

| PrsNr | TNr |
|-------|-----|
| 112 | 12 |
| 112 | 17 |
| 205 | 12 |
| 205 | 23 |
| 205 | 37 |

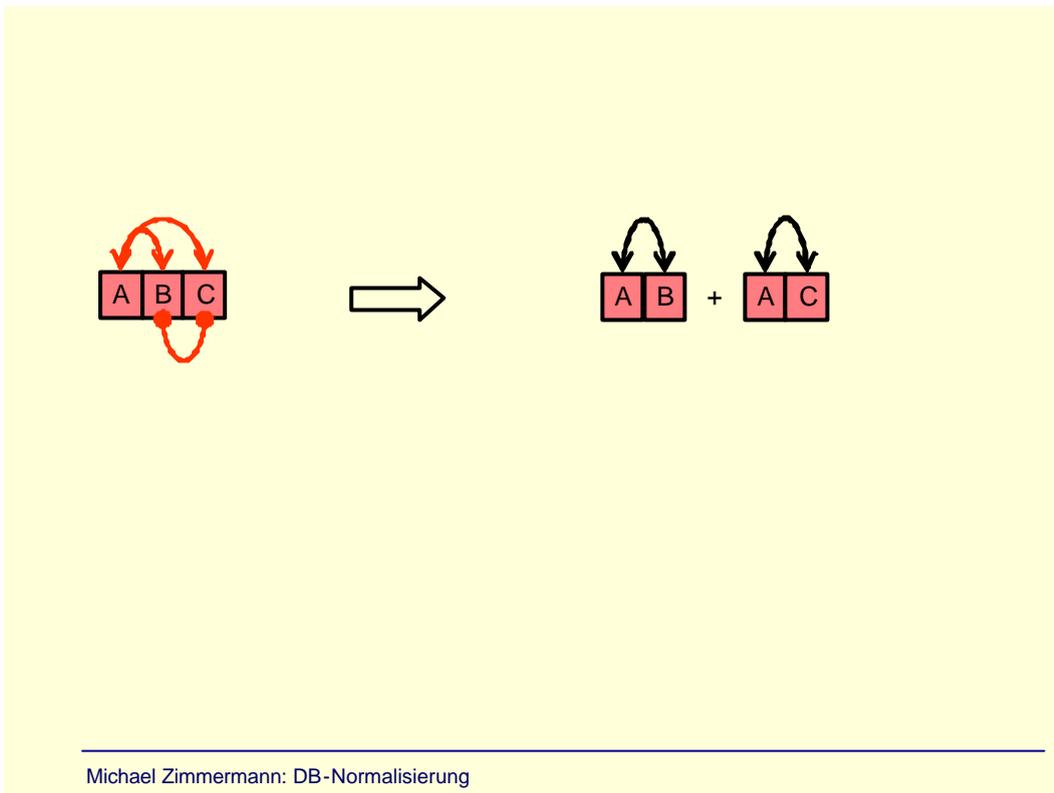
Michael Zimmermann: DB-Normalisierung

Redundanzen beseitigt

Keine Lös-, Einfügeanomalien mehr

Eine Änderung an einer Stelle

Attributschema, Verstoß

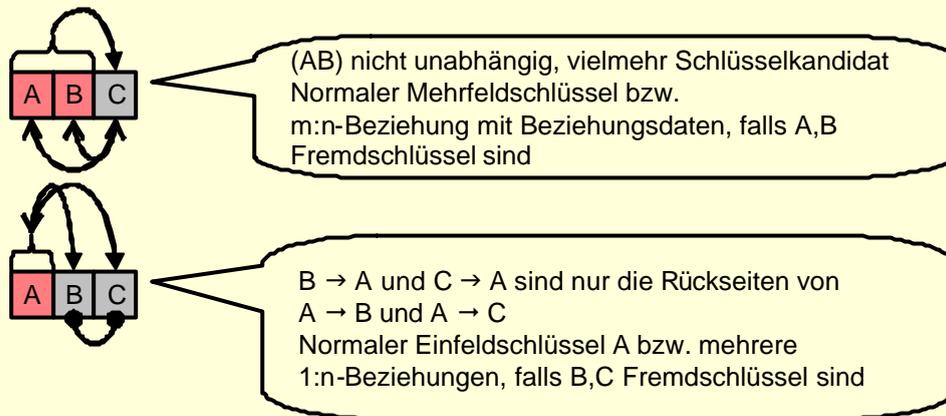


Als allgemeines Attributschema:

Zwei unabhängige m:n-Beziehungen, d.h. die beiden m:n-Beziehungen überlappen sich in einem gemeinsamen Feld, und die beiden nicht überlappenden Felder sind untereinander unabhängig.

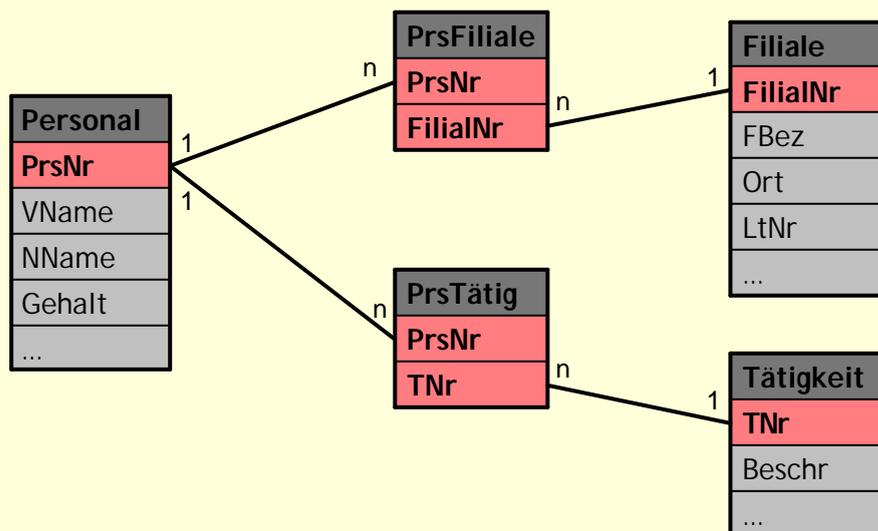
Der mögliche Fall mit 1:n statt m:n ist durch die 2 NF bereits wegnormalisiert, wäre aber natürlich auch durch die 4 NF verboten. An diesem Beispiel sieht man sehr schön, daß die höheren NF die darunterliegenden beinhalten.

Attributschema, kein Verstoß



Michael Zimmermann: DB-Normalisierung

Beziehungsschema



Michael Zimmermann: DB-Normalisierung

Die vorher fälschlich in einer Tabelle zusammengefaßten Beziehungen PrsNr-FilialNr und PrsNr-TNr sind ordnungsgemäß in je einer eigenen Referenztable modelliert.

Definition

- Jede MVD ist entweder trivial oder geht von einem Oberschlüssel aus

Bedeutung

- Trivialität heißt wie erinnerlich, daß
 - ▶ Ausgangs- und Zielgruppe der MWA zusammen die gesamte Tabelle bilden – es also nur eine MWA gibt oder
 - ▶ Die Zielgruppe in der Ausgangsgruppe enthalten ist – solche MWAs können zu mehreren auftreten, da sie nicht schaden
- Ausgangsgruppe ist Oberschlüssel heißt schlicht, daß die MWA eigentlich eine FA ist
- Verboten ist also letztlich das Auftreten mehrerer untereinander unabhängiger echter MWAs
- Wenn man die 2 NF voraussetzt, kann ein 4NF-Verstoß nur auftreten, wenn ...
 - ▶ zwei wechselseitig mehrwertige aber untereinander unabhängige Abhängigkeiten auftreten und
 - ▶ der Kandidatschlüssel über die gesamte Tabelle geht

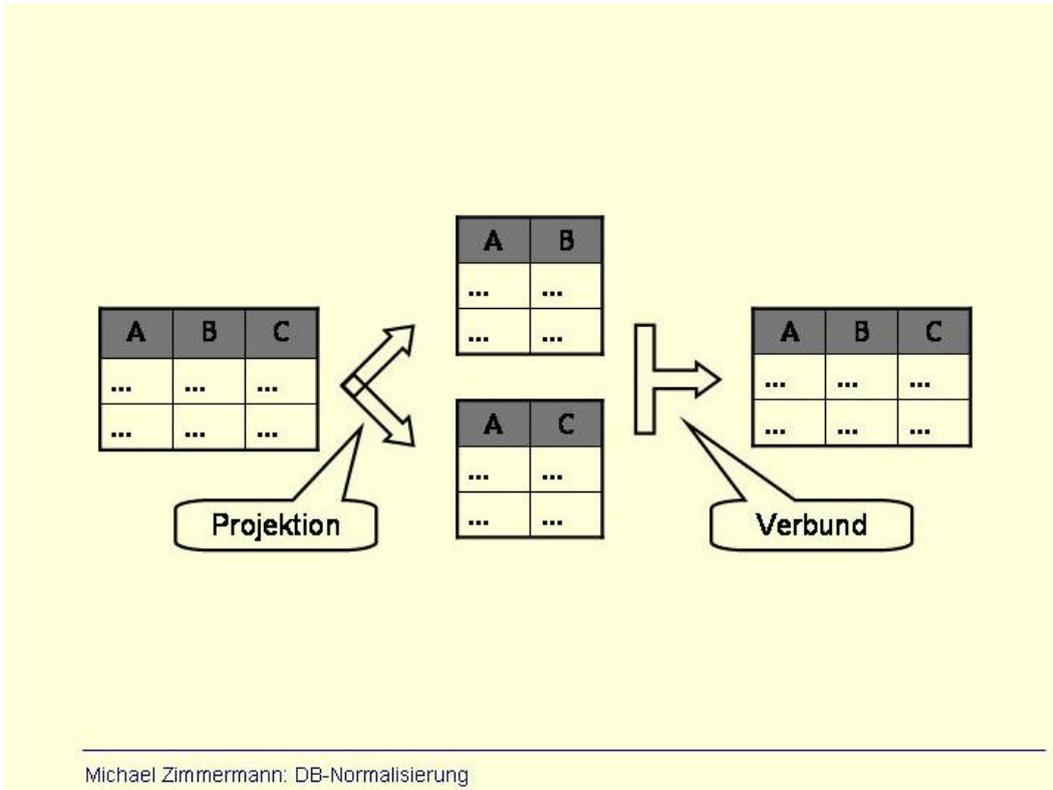
Auftreten von 4NF-Verletzungen

- Ähnlich BCNF-Verletzungen treten 4NF-Verletzungen vorwiegend in Beziehungstabellen auf
- Insbesondere, wenn zwei (oder mehr) unabhängige m:n-Beziehungen wegen thematischer Ähnlichkeit fälschlich gehalten werden für
 - ▶ eine ternäre (n-äre) Beziehung
 - ▶ ein eigenständiges Objekt
- 4NF-Verstöße relativ leicht erkennbar – auffälliger als BCNF
- Unabhängigkeit heißt nicht, daß kein thematischer Zusammenhang vorliegt auch wenn unsinnige Standardbeispiele wie Sprache/Hobby das nahelegen und so den Ernst der 4 NF verschleiern

Wie bei der BCNF sind 4-NF-Verstöße ein typisches Beziehungsproblem. Einfach gesagt: Man packe nicht mehrere m:n-Beziehungen in eine Tabelle.

Noch einmal: Abhängigkeitstheorie 2

Was ist eine Projektions-Verbund-Abhängigkeit (Projection Join Dependency, JD)?



- Projektion: Zerlegen einer Tabelle in Teile mit einem überlappenden Feld (Spaltenauswahl, SELECT in Abfragen)
- Verbund: Zusammenführen mehrerer Tabellen/Abfragen zu einem Verbundschemata (INNER JOIN in Abfragen)
- Wenn man eine Tabelle in Teile zerlegt und diese wieder zusammenfügt, kann das Ergebnis ...
 - ▶ 1. mit der Ursprungstabelle identisch sein:
Die Tabelle erfüllt die JD bezüglich der Zerlegung
 - ▶ 2. zusätzliche oder fehlende Datensätze aufweisen:
Es liegt bezüglich der Zerlegung keine JD vor
- Eine Tabelle ist JD-frei, wenn es gar keine Zerlegung gibt, die eine Rekonstruktion der Tabelle gemäß 1 erlaubt
- Eine JD ist trivial, wenn sie sich auf die ganze Tabelle bezieht

PV-Abhängigkeit

Was bedeutet JD anschaulich?

- Das geschilderte Verhalten kann auftreten, wenn mehrere sich ringförmig in Feldern überlappende mehrwertige (mehr-mehr und mehr-ein) Abhängigkeiten auftreten
- Wenn eine JD in einer Tabelle ABC auftritt, bedeutet das,
 - ▶ ... daß die Dreiergruppen in den Datensätzen sich logisch folgern lassen aus den Zweiergruppen AB, BC, AC
 - ▶ ... daß in den Dreiergruppierungen keine echte Information steckt, die sich nicht auch aus einfacheren Strukturen erschließen ließe
- Wenn keine JD in einer Tabelle ABC auftritt, bedeutet das,
 - ▶ ... daß in der Zusammenstellung ABC echte Information steckt, die sich nicht aus AB, BC, AC erzeugen läßt,
 - ▶ ... daß tatsächlich eine mehrstellige Beziehung vorliegt

Um das ganz klar zu sagen: Es ist **schlecht**, wenn eine Tabelle sich aus ihren Teilen rekonstruieren läßt, es sei denn, es sind nur Rekonstruktionen möglich, die auf funktionalen Abhängigkeiten beruhen.

5. Normalform

Tableauschema, falsch

| Sparte | Vertreter | VGes |
|-------------|-----------|------|
| Haftpflicht | Bäcker | HUK |
| Haftpflicht | Müller | A&M |
| Haftpflicht | Müller | HDI |
| Leben | Bäcker | DEVK |
| Leben | Bäcker | HUK |
| Leben | Meier | DEVK |
| Leben | Meier | HUK |
| Leben | Müller | A&M |
| Leben | Schmidt | A&M |
| Leben | Schmidt | DEVK |
| Rente | Bäcker | DEVK |
| Rente | Meier | DEVK |

Michael Zimmermann: DB-Normalisierung

Thema: Wer kann für welche Gesellschaft welche Tätigkeit ausüben?

- Vertreter sind kompetent für mehrere Sparten
- Gesellschaften sind in mehreren Sparten tätig
- Vertreter sind freie Agenten, d.h. Vertreter - Gesellschaft m:n

Probleme

Ähnlich wie beim 4-NF-Problem

Aber: Alle über Sparte gebildeten Agent-Gesellschaft-Kombinationen sind zusätzlich eingeschränkt, durch die Bedingung: Welcher Agent arbeitet für welche Gesellschaft?

Wenn ein Agent (vorübergehend) alle Mitarbeiterverträge auflöst, können seine Kompetenzen nicht mehr gespeichert werden

Wenn ein Agent/eine Gesellschaft neue Sparten übernimmt, müssen für eine Information mehrere Datensätze erzeugt werden

Tableauschema, richtig

| Sparte | Vertreter |
|-------------|-----------|
| Haftpflicht | Bäcker |
| Haftpflicht | Müller |
| Leben | Bäcker |
| Leben | Meier |
| Leben | Müller |
| Leben | Schmidt |
| Rente | Bäcker |
| Rente | Meier |

| Vertreter | VGes |
|-----------|------|
| Bäcker | HUK |
| Bäcker | DEVK |
| Müller | A&M |
| Müller | HDI |
| Meier | DEVK |
| Meier | HUK |
| Schmidt | A&M |
| Schmidt | DEVK |

| Sparte | VGes |
|-------------|------|
| Haftpflicht | HUK |
| Haftpflicht | A&M |
| Haftpflicht | HDI |
| Leben | DEVK |
| Leben | HUK |
| Leben | A&M |
| Rente | DEVK |

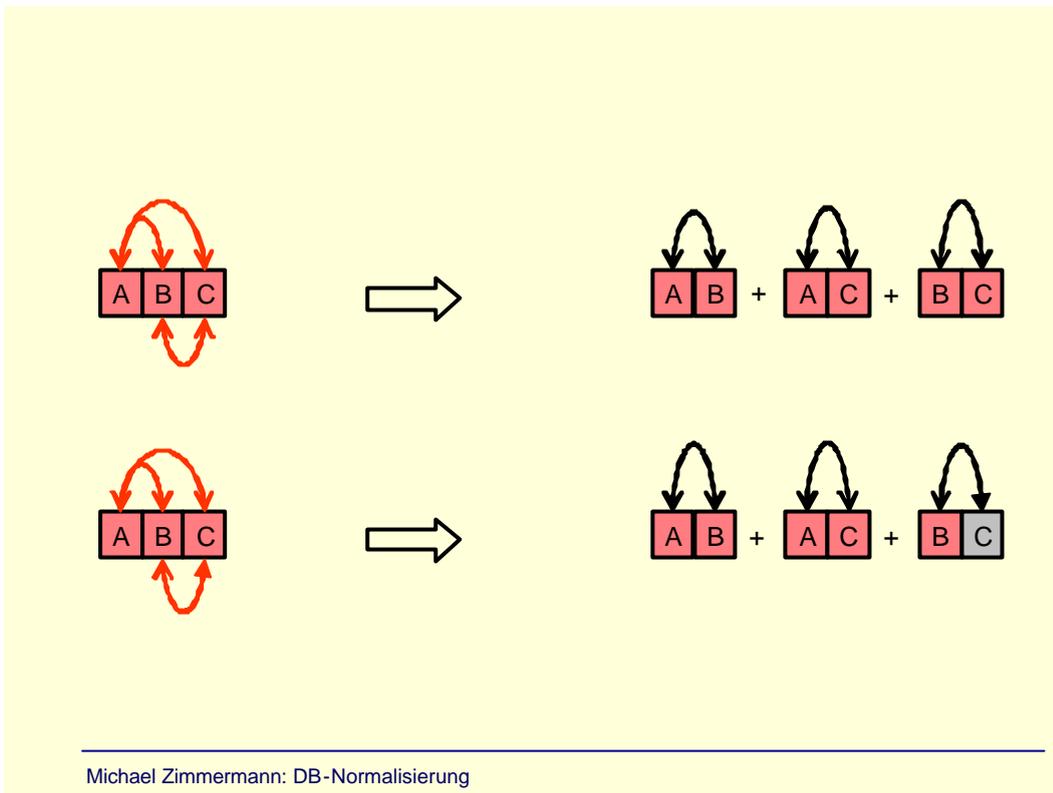
Michael Zimmermann: DB-Normalisierung

Jede der drei vorher vermengten Informationen:

- Wer kann was?
- Wer arbeitet für welche Gesellschaft?
- Welche Gesellschaft macht was?

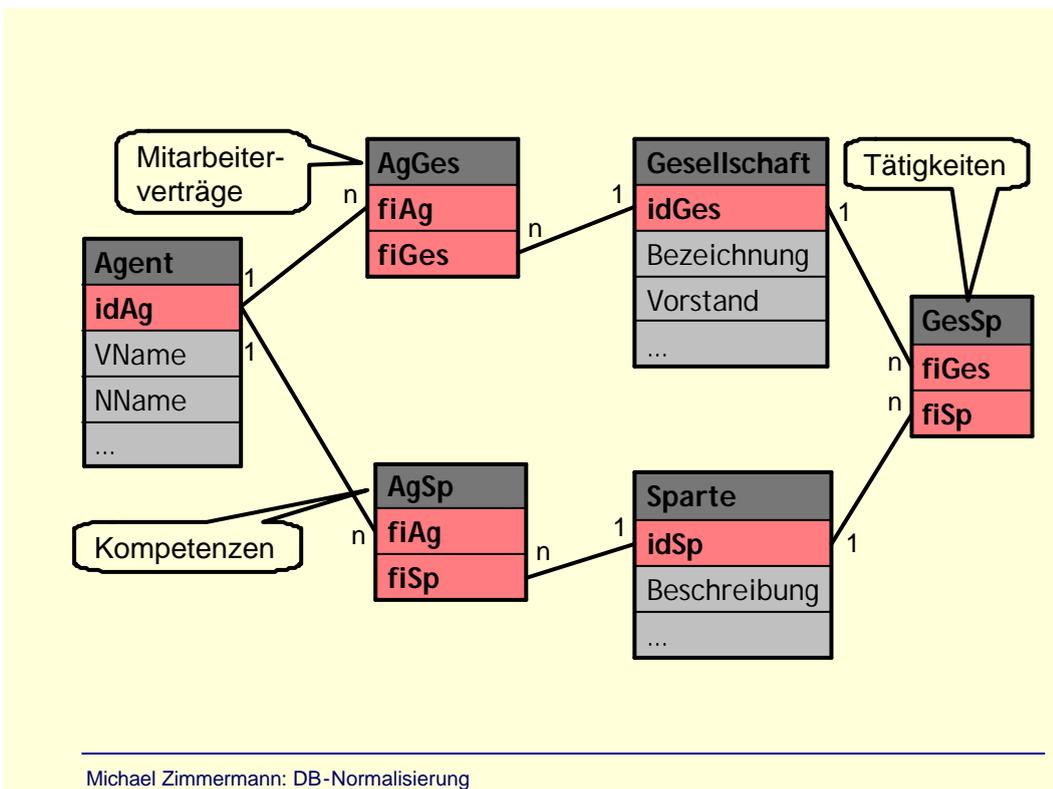
ist ordnungsgemäß getrennt modelliert

Attributschema

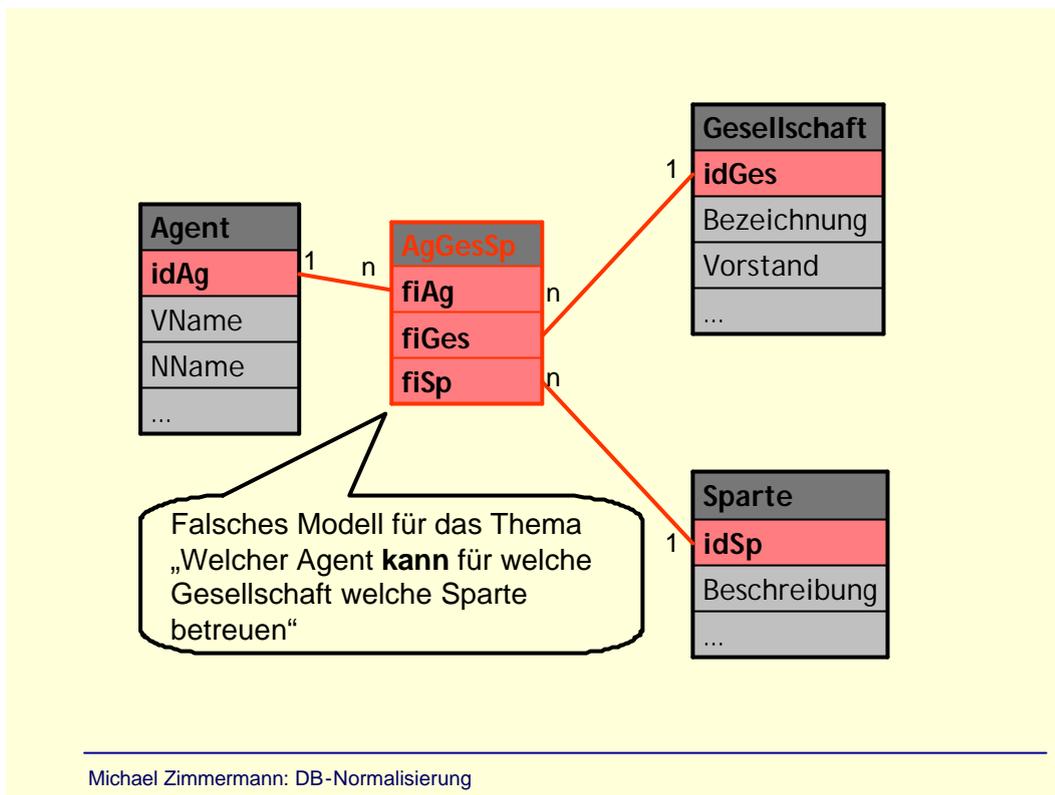


Das Attributschema alleine ist bei 5-NF-Problematiken noch nicht aussagekräftig. Entscheidend ist die **logische** Frage: Ist (ABC) ternär oder (AB), (BC), (AC) jeweils binär?

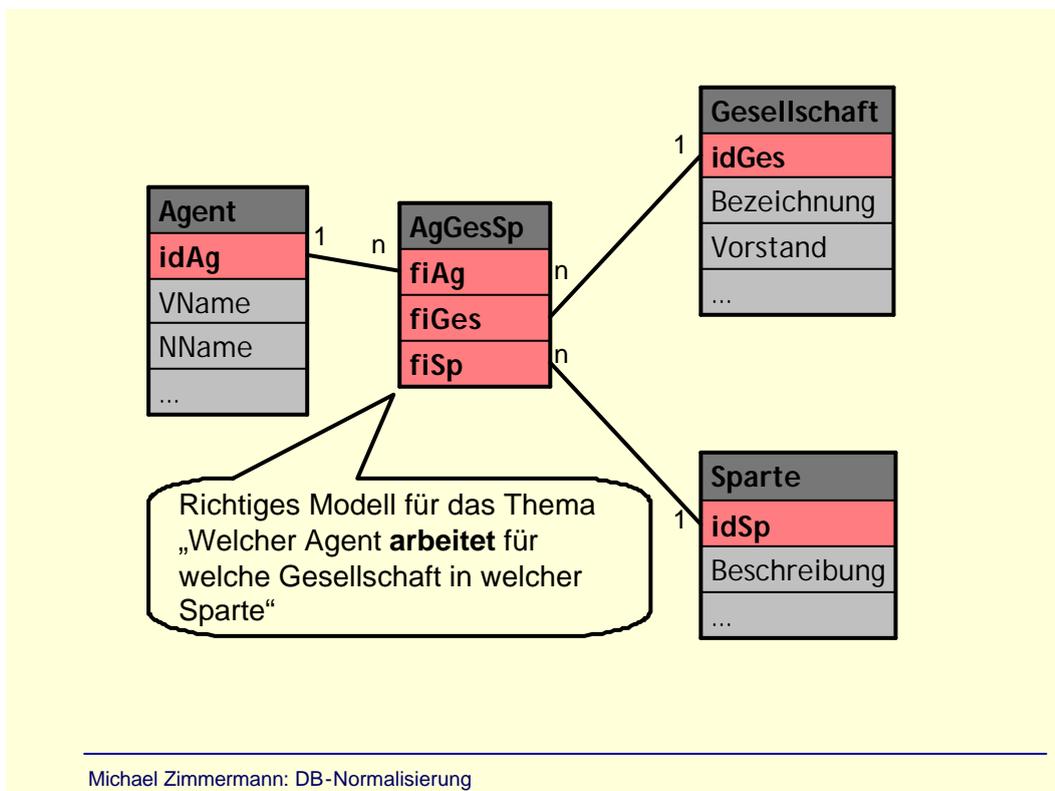
Beziehungsschema, richtig



Beziehungsschema, falsch



Beziehungsschema, richtig(!)



Man mache sich klar, daß die beiden letzten Beispiele völlig identisch strukturiert sind. Im ersten Fall wird das Schema aber auf einen logischen Sachverhalt angewandt, den es nicht beschreibt; genau darin liegt der Fehler.

Im zweiten Fall wird dasselbe Modell auf einen logischen Sachverhalt angewandt, auf den es paßt, daher ist es dort richtig.

Definition

- Jedes Teilschema aller Zerlegungen einer Tabelle ist ein Oberschlüssel der Tabelle

oder

- Alle vorkommenden JDs sind trivial oder funktional

Bedeutung

- Wie bei der MWA (4 NF), die ein Spezialfall einer JD ist, kann man sagen, daß echte JDs (also nichttriviale und nichtfunktionale) schlicht verboten sind
- Anders gesagt: Wenn es möglich ist, eine Tabelle so in mehrere Teile zu zerlegen, daß die ursprüngliche Tabelle rekonstruierbar bleibt, ist die Zerlegung vorzunehmen, es sei denn, daß die Zerlegung ausschließlich auf vollen funktionalen Abhängigkeiten sowie Unabhängigkeiten beruht. Dann kann auf die Zerlegung verzichtet werden

Hinweis

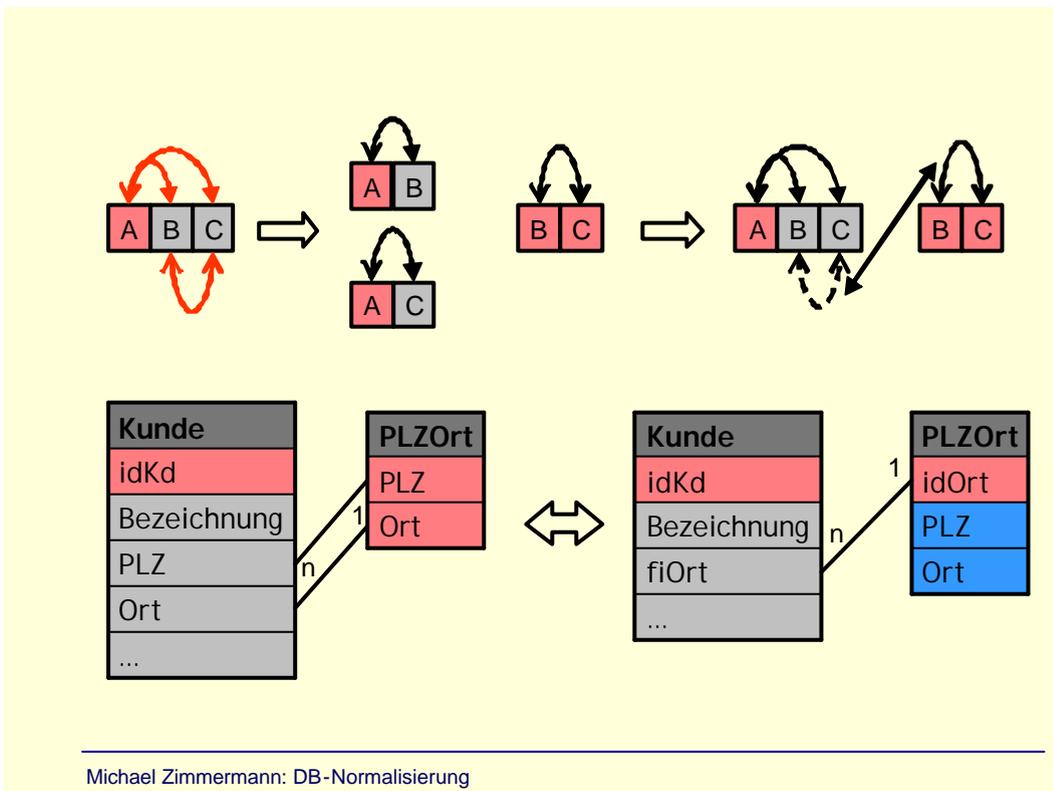
- JDs in einer Tabelle mit nur zwei Feldern sind immer trivial
- Eine Tabelle, in der ausschließlich funktionale Abhängigkeiten von Kandidatschlüsseln vorkommen und Unabhängigkeit zwischen allen Nichtschlüselfeldern besteht, ist JD-frei

Auftreten von 5NF-Verletzungen

- Ähnlich BCNF-Verletzungen treten auch 5NF-Verletzungen vorwiegend in Beziehungstabellen auf
- 5NF-Verletzungen sind sehr schwer zu erkennen
- 5NF-Verletzungen und echte ternäre (oder höhere) Beziehungen können sehr leicht verwechselt werden
- Strukturell identische Tabellenschemata können abhängig von ihrer Semantik die 5 NF verletzen oder erfüllen

- Typisch ist:
 - ▶ **Potentielle** Drei(mehr)fachbeziehungen
Wer **kann** was wo tun?
 - ▶ sind häufig verkappte Zweier
Wer kann wo?
Wer kann was?
Was passiert wo?
 - ▶ **Reale** Drei(mehr)fachbeziehungen
Wer **macht** wo was?
 - ▶ sind meistens echte Drei- oder Mehrfachbeziehungen

Ein abschließendes Beispiel



Sei A Kundennummer B Postleitzahl, C Ort. Wenn man daraus 3 Tabellen macht: Kunde PLZ; Kunde Ort; PLZ Ort, läßt sich daraus die ursprüngliche Tabelle joinen, d. h. JD liegt vor, also zerlegen. Andererseits dürfen AB, AC wieder zusammengefaßt werden, da beide funktional sind. In der Folge bleibt eine zusätzliche zweite Tabelle PLZ Ort, die die zulässigen PLZ Ort-Kombinationen repräsentiert und als Fremdschlüssel in Kunde einbezogen wird. Man kann das über eine Zweifelderbeziehung oder auch einen einfeldigen Surrogatschlüssel machen.

- Mehrwertige Abhängigkeit unter Nichtschlüselfeldern
 - ▶ ... nach obigem Schema ausnormalisieren
 - ▶ eine der wenigen JDs, die rein strukturell erkennbar ist

Domain-Key-Normalform

Informelle Erläuterung

- DKNF geht davon aus, daß bei Angabe der gesamten Definitionsbereiche (Domänen) aller Felder und Identifikation aller Werte über eindeutige Schlüssel keine Anomalien mehr möglich sind
- Führt praktisch zu erheblichen bis unüberwindlichen Problemen
- Aber sinnvoll bei kleinen Definitionsbereichen mit wenigen (einem) Feldern; dadurch nahezu fehlerresistente Eingabe und Suche z. B. Kategorien, Kataloge etc.

Anschauliche Schilderung des DKNF-Problems

Unterscheide:

- ▶ Ähnlichkeit
- ▶ Kongruenz
- ▶ Identität

| idPrs | Nachname | Wohnort | ... |
|-------|----------|-----------|-----|
| 112 | Müller | Mainz | ... |
| 117 | Meier | Mainz | ... |
| 120 | Müller | Wiesbaden | ... |
| 135 | Müller | Bingen | ... |
| 143 | Schmidt | Mainz | ... |
| 144 | Maier | Wiesbaden | ... |
| 152 | Schnitt | Mainz | ... |
| 160 | Paulsen | Alzey | ... |

Michael Zimmermann: DB-Normalisierung

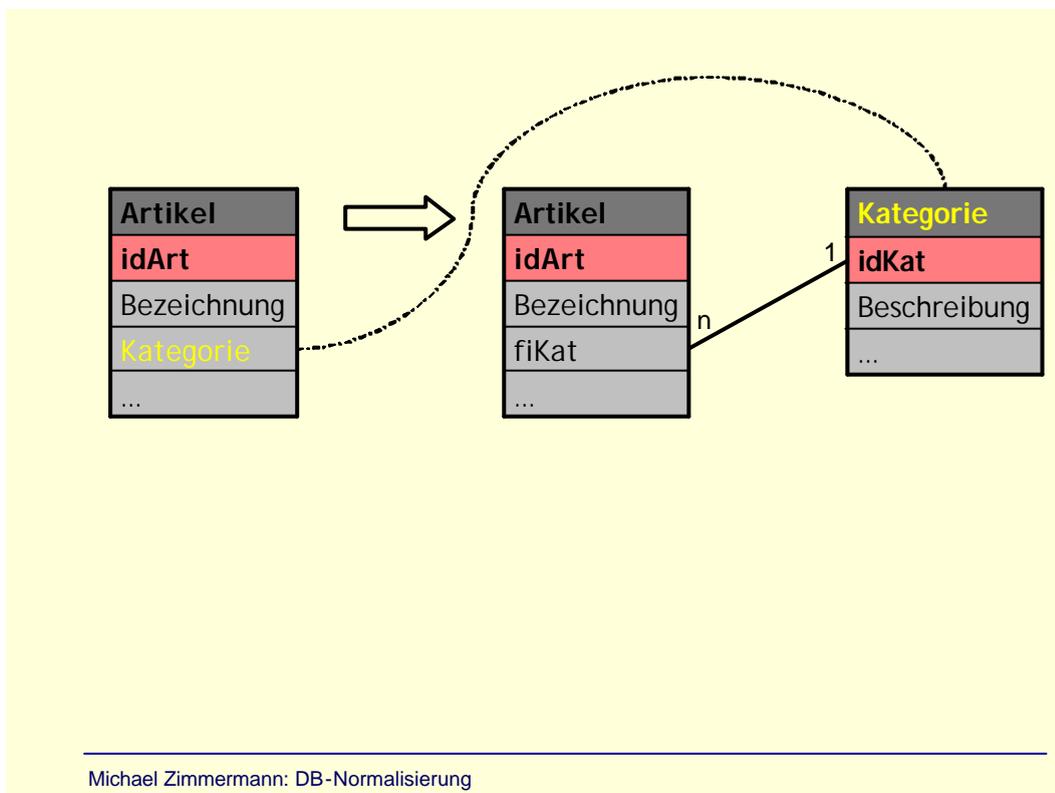
Die Namen Schmidt und Schnitt ähneln sich sehr. Die Namen Müller, Müller und Müller ähneln sich bis zur Ununterscheidbarkeit; sie sind kongruent.

Anhang

Dennoch haben die drei Müllers nur den gleichen und nicht denselben Namen. Es sind drei Personen, und eine Änderung bei einer Person betrifft die anderen nicht. Die Orte Mainz, Mainz, Mainz, Mainz haben aber nicht zufällig gleiche Namen, sondern bezeichnen denselben Ort, d.h. sie sind tatsächlich identisch; es ist nur ein Ort.

- Die Identitätswiederholungen sorgen für Anomalien (Änderung, Tippfehler, etc.)
- Eigentlich liegt ein selbstständiges Objekt (Ort) vor
- Aber: Da Objekt nur aus einem Feld besteht, kein Verstoß bis einschließlich 5 NF

Ein sinnvolles Beispiel



- Vorteil gegenüber Feldgültigkeitsregel:
 - Liste durch Benutzer erweiterbar/änderbar
 - Wenn Daten und unsemantischer (gehaltloser) Schlüssel streng getrennt gehalten werden, sind Änderungsanomalien ausgeschlossen
- Methode: Alle Attribute (Felder) werden als Objekte (Tabellen) aufgefaßt
- Das Vorgehen ist in der DKNF für alle Felder Vorschrift, was aber praktisch kaum umsetzbar ist

Anhang

Einzelne Tabellen in DKNF zu bringen, ist vorteilhaft, wenn:

- ... die Tabelle wenige (ein) Attribute (Felder) hat.
- ... die Definitionsbereiche der Attribute überschaubar klein sind.

Eine Auffassung des Datenfeldes Kategorie als eigenständiges Objekt (Tabelle) Kategorie bringt nicht nur Fehlersicherheit sondern auch Performance-Vorteile, da Sortierung/Gruppierung nach Kategorien über ein Integerfeld statt über Strings laufen kann.

Entwurfsempfehlungen

- In **Datentabellen** ausschließlich Kandidatschlüssel und davon funktional abhängige Nichtschlüselfelder, die untereinander unabhängig sind
- **Mehrwertige** Abhängigkeiten zwischen NSF auslagern
- **Parallelschlüssel** sind erlaubt, ibs. darf jederzeit ein AutoWert-Feld (Surrogatschlüssel) eingefügt werden
- Einzelne Felder, die Objekte repräsentieren, als **Katalogtabellen** auslagern, wenn Domäne klein bzw. dynamisch (DKNF)
- **Beziehungstabellen**, die mehr als zwei Objekte verknüpfen, peinlich genau auf BCNF, 4NF, 5NF-Verstöße untersuchen – sicherstellen, daß die Information nicht auch mit mehreren 2er-Tabellen darstellbar ist. **Echte Drei-oder Mehrfachbeziehungen sind selten**
- Nicht zu sehr auf **Primärschlüssel** fixieren – die gerne vergessenen **Sekundärschlüssel** (Unique Indizes) sind genauso wichtig
- Wenn Validitätsprüfungen in Formularen ausprogrammiert werden, liegt immer der Verdacht auf ein fehlerhaftes Datenmodell nahe

Glossar

Einige Begriffsentsprechungen

| Relationale Algebra | Entity-Relationship-Modell | Datenbanksprache |
|---------------------|----------------------------|------------------------|
| Relation | Entität (Objekt) | Tabelle |
| Referenz | Relation (Beziehung) | Fremdschlüsselfeld |
| Tupel | Instanz | Datensatz (Zeile) |
| Attribut | Attribut | Feld (Spalte) |
| Attributwert | Ausprägung | Feldwert |
| Domäne | Domäne | Datentyp + Restriktion |
| Schlüsselkandidat | - | Unique Index |
| - | - | Primärschlüssel |

Einige Abkürzungen und englische Fachbegriffe

- **Funktionale Abhängigkeit**, Functional Dependency, **FA**, FD
- **Volle Funktionale Abhängigkeit**, Full Functional Dependency, **VFA**, FFD
- **Mehrwertige Abhängigkeit**, Multi Value Dependency, **MWA**, MVD
- (Projektions-) **Verbund-Abhängigkeit**, Join Dependency, PVA, **JD**
- **Oberschlüssel**, Superschlüssel, Schlüssel, Super Key, **OS**, SS, SK
- **Kandidatschlüssel**, Schlüsselkandidat, Schlüssel, Minimalschlüssel, Candidate Key, **KS**, CK

Symbole

- $\cap \cup$ Schnittmenge, Vereinigungsmenge
- $\supseteq \supset \subset \subseteq$ Obermenge, Teilmenge
- $\bullet \cdots$ Unabhängigkeit
- $\Leftrightarrow \rightarrow$ Mehrwertig abhängig
- $\leftrightarrow \rightarrow$ Funktional abhängig
- $\Leftrightarrow \Rightarrow$ Logische Folgerung