

# Datenbanksysteme

## Normalformen

Burkhardt Renz

Fachbereich MNI  
Technische Hochschule Mittelhessen

Sommersemester 2019

# Inhalt

- Motivation und 1NF
  - Folgen von Redundanz
  - Hierarchie der Normalformen
  - Erste Normalform (1NF)
- 2NF, 3NF und BCNF
  - Funktionale Abhängigkeiten
  - Zweite Normalform (2NF)
  - Dritte Normalform (3NF)
  - Boyce-Codd-Normalform (BCNF)
- 4NF und 5NF
  - Mehrwertige Abhängigkeiten und 4NF
  - Verbundabhängigkeiten und 5NF
  - Literatur

# Systematisches Daten-Design

- Ziel: Reduzieren der Redundanz; nur notwendige Redundanz
- Mittel: Erkennen von Abhängigkeiten von Werten
- Ergebnis: Geeignete Strukturierung der Daten in RelVars sowie Integritätsbedingungen für diese RelVars
- Systematisches Design  $\hat{=}$  Bedeutung der Daten und ihrer Abhängigkeiten erfassen

## Beispiel „Suppliers“

SNo	SName	Status	City
S1	Smith	20	London
S2	Jones	30	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	10	Athens

Funktionale Abhängigkeit?

$\{City\} \rightarrow \{Status\}$

oder hängt der Status nicht von der City ab?

Wir gehen für das Folgende davon aus, dass  $\{City\} \rightarrow \{Status\}$  gilt.

# Anomalien

**Einfüge-Anomalie** Wir können nicht verzeichnen, dass die City „Rome“ den Status „15“ hat, solange es keinen Lieferanten aus Rom gibt.

**Lösch-Anomalie** Wenn wir den Lieferanten aus Athen löschen, verlieren wir die Information, dass Athen den Status „10“ hat.

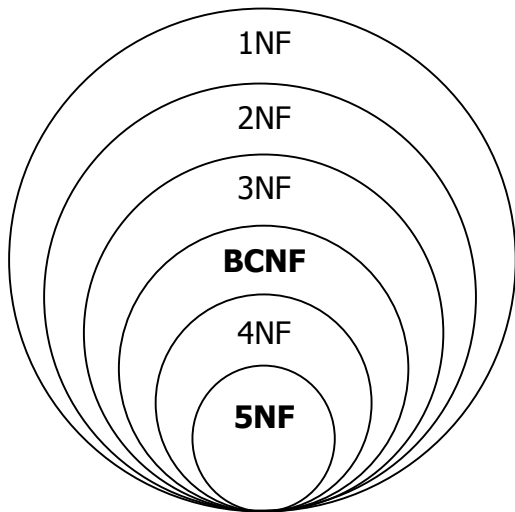
**Änderungs-Anomalie** Wenn wir die City eines Lieferanten ändern, müssen wir auch den Status entsprechend „nachziehen“.

## Vermeidung der Anomalien

SNo	SName	City
S1	Smith	London
S2	Jones	Paris
S3	Blake	Paris
S4	Clark	London
S5	Adams	Athens

City	Status
London	20
Paris	30
Athens	10
Rome	15

# Hierarchie der Normalformen



# Erste Normalform (1NF)

## Definition

Sei  $R_S$  das Relationschema von  $R$ .

$R$  ist in der **ersten Normalform (1NF)** genau dann, wenn in allen Tupeln von  $R$  die Werte den Datentyp des entsprechenden Attributs haben.



# Diskussion

Andere Definition in Lehrbüchern:

Eine Relation ist in der ersten Normalform, wenn alle Attribute nur **atomare Werte** enthalten können.

Technisch gesehen kann man in SQL92 die 1NF nicht verletzen. Erst objekt-relationale Erweiterungen sehen zusammengesetzte Datentypen wie Arrays u.ä. vor.

# Diskussion 1NF

Wie sind folgende Entwurfsentscheidungen zu beurteilen?

- Tabelle Person mit folgendem Schema:  
(Name, Vorname, Adresse)
- Tabelle Person mit folgendem Schema:  
(Name, Vorname, Telefon)
- Tabelle Person mit folgendem Schema:  
(Name, Vorname, Tel1, Tel2, Tel3)
- Tabelle Person mit folgendem Schema:  
(Name, Vorname, TelDienst, TelPrivat, TelMobil)

# Inhalt

- Motivation und 1NF
  - Folgen von Redundanz
  - Hierarchie der Normalformen
  - Erste Normalform (1NF)
- 2NF, 3NF und BCNF
  - Funktionale Abhängigkeiten
  - Zweite Normalform (2NF)
  - Dritte Normalform (3NF)
  - Boyce-Codd-Normalform (BCNF)
- 4NF und 5NF
  - Mehrwertige Abhängigkeiten und 4NF
  - Verbundabhängigkeiten und 5NF
  - Literatur

# Funktionale Abhängigkeiten

## Definition

Seien  $X$  und  $Y$  Teilmengen des Schemas einer RelVar.

Dann besteht die **funktionale Abhängigkeit (FD)**  $X \rightarrow Y$  genau dann, wenn zwei Tupel, die an den Attributen in  $X$  übereinstimmen, auch an den Attributen in  $Y$  übereinstimmen.

# Triviale funktionale Abhängigkeiten

## Definition

Eine funktionale Abhängigkeit  $X \rightarrow Y$  heißt **trivial**, wenn sie automatisch erfüllt ist und niemals verletzt werden kann.

## Beispiel

$\{\text{City}, \text{Status}\} \rightarrow \{\text{City}\}$

$\{\text{City}, \text{Status}\} \rightarrow \{\text{Status}\}$

$\{\text{City}\} \rightarrow \{\text{City}\}$

$\{\text{City}\} \rightarrow \{\}$

# Irreduzible funktionale Abhängigkeiten

## Definition

Eine funktionale Abhängigkeit  $X \rightarrow Y$  heißt **irreduzibel** genau dann, wenn für jede Teilmenge  $X' \subset X$  *nicht*  $X' \rightarrow Y$  gilt.

## Beispiel

In der Beispieldatenbank SAP aus den Übungen ist  $\{\text{SNo}, \text{PNo}\} \rightarrow \{\text{Qty}\}$  in der RelVar SP irreduzibel.

# Superschlüssel

## Definition

Ein **Superschlüssel** einer RelVar ist eine Menge  $X$  von Attributen, für die gilt:

Tupel werden durch die Werte an diesen Attributen eindeutig bestimmt.

## Beispiel

{SNo, City, Status} ist ein Superschlüssel

# Schlüssel

## Definition

Ein **Schlüssel** einer RelVar ist ein minimaler Superschlüssel  $X$ .  
D.h. für jede (echte) Teilmenge  $X' \subset X$  gilt:  $X'$  ist kein Superschlüssel für die RelVar.

## Beispiel

$\{SNo, PNo, Qty\}$  ist ein Superschlüssel für SP

$\{SNo, PNo\}$  ist ein Schlüssel für SP

$\{SNo\}$  und  $\{Pno\}$  sind nicht Schlüssel für SP



# Subschlüssel

## Definition

Ein **Subschlüssel** einer RelVar ist eine Menge von Attributen  $X'$ , die Teilmenge eines Schlüssels der RelVar ist.

## Beispiel

$\{SNo, PNo, Qty\}$  ist kein Subschlüssel für SP

$\{SNo, PNo\}$  ist ein Subschlüssel für SP

$\{SNo\}$  und  $\{Pno\}$  sind echte Subschlüssel für SP

$\{Qty\}$  ist kein Subschlüssel für SP

# Schlüsselattribut

## Definition

Ein **Schlüsselattribut** einer RelVar ist ein Attribut, das in wenigstens einem Schlüssel vorkommt.

## Definition

Ein **Nicht-Schlüsselattribut** einer RelVar ist ein Attribut, das in keinem Schlüssel vorkommt.

## Beispiel

{SNo} ist ein Schlüsselattribut von SP

{Qty} ist ein Nicht-Schlüsselattribut von SP

# Zweite Normalform (2NF)

## Definition

Eine RelVar  $R$  ist in der **zweiten Normalform (2NF)** genau dann, wenn gilt:

Für jeden Schlüssel  $K$  von  $R$  und jedes Nicht-Schlüsselattribut  $A$  ist  $K \rightarrow \{A\}$  irreduzibel.

## Informell

2NF ist verletzt, wenn ein Attribut **nur von einem Teil** eines Schlüssels abhängt.

## Beispiel

<u>SNo</u>	<u>PNo</u>	SName	Qty
S1	P1	Smith	300
S1	P2	Smith	600
S2	P1	Jones	100
...	...	...	...

Diese RelVar ist nicht in 2NF:

{SNo, PNo} ist Schlüssel

{SNo, PNo}  $\rightarrow$  {SName} gilt

**aber:** {SNo, PNo}  $\rightarrow$  {SName} ist nicht irreduzibel, weil

{SNo}  $\rightarrow$  {SName} gilt

## Zweite Normalform (2NF)

### Definition

Eine RelVar  $R$  ist in der **zweiten Normalform (2NF)** genau dann, wenn für jede nicht-triviale funktionale Abhängigkeit  $X \rightarrow Y$  eine der folgenden Aussagen gilt:

- (a)  $X$  ist ein Superschlüssel
- (b)  $Y$  ist ein Subschlüssel
- (c)  $X$  ist kein Subschlüssel

Diese Definition ist äquivalent zur ersten Definition.

# Dritte Normalform (3NF)

## Definition

Eine RelVar  $R$  ist in der **dritten Normalform (3NF)** genau dann, wenn für jede nicht-triviale funktionale Abhängigkeit  $X \rightarrow Y$  eine der folgenden Aussagen gilt:

- (a)  $X$  ist ein Superschlüssel
- (b)  $Y$  ist ein Subschlüssel

## Informell

3NF ist verletzt, wenn ein Attribut **nicht nur** von einem Schlüssel abhängt.

## Beispiel

SNo	SName	Status	City
S1	Smith	20	London
S2	Jones	30	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	10	Athens

Es gilt die funktionale Abhängigkeit:  $\{City\} \rightarrow \{Status\}$

Dann ist 3NF nicht erfüllt, denn

- (a)  $\{City\}$  ist *nicht* Superschlüssel und
- (b)  $\{Status\}$  ist *nicht* Subschlüssel

# Boyce-Codd-Normalform (BCNF)

## Definition

Eine RelVar  $R$  ist in der **Boyce-Codd-Normalform (BCNF)** genau dann, wenn für jede nicht-triviale funktionale Abhängigkeit  $X \rightarrow Y$  gilt:

- (a)  $X$  ist ein Superschlüssel

## Informell

Jedes Attribut hängt von einem Schlüssel und **nur** von einem Schlüssel ab.



## Beispiel

In folgendem Beispiel soll gelten: Es gibt keine zwei Lieferanten mit demselben SName.

SNo	PNo	SName	Qty
S1	P1	Smith	300
S1	P2	Smith	600
S2	P1	Jones	100
...	...	...	...

Diese RelVar ist nicht in BCNF:

$\{SNo, PNo\}$  und  $\{SName, Pno\}$  sind Schlüssel

$\{SNo\} \rightarrow \{SName\}$  gilt und ist nicht-trivial

**aber:**  $\{SNo\}$  ist kein Schlüssel

## Fazit soweit

*„The key, the whole key and nothing but the key  
– so help me Codd“ [Wikipedia]*

# Inhalt

- Motivation und 1NF
  - Folgen von Redundanz
  - Hierarchie der Normalformen
  - Erste Normalform (1NF)
- 2NF, 3NF und BCNF
  - Funktionale Abhängigkeiten
  - Zweite Normalform (2NF)
  - Dritte Normalform (3NF)
  - Boyce-Codd-Normalform (BCNF)
- 4NF und 5NF
  - Mehrwertige Abhängigkeiten und 4NF
  - Verbundabhängigkeiten und 5NF
  - Literatur

## Mehrwertige Abhängigkeiten – Beispiel I

Prädikat für folgende Relation:

„Modul <MNo> wird angeboten von Dozent <DNo> und hat Lehrbuch <LNo> als Textgrundlage.“

<u>MNo</u>	<u>DNo</u>	<u>LNo</u>
M1	D1	L1
M1	D1	L2
M1	D2	L1
M1	D2	L2

Inwiefern enthält diese Relation Redundanz?

## Mehrwertige Abhängigkeiten – Beispiel II

- Die zugeordneten Dozenten verwenden alle dieselben Lehrbücher, d.h. die Zuordnung der Lehrbücher zum Modul hängt gar nicht vom Dozenten ab.
- Die zugeordneten Lehrbücher sind bei allen Dozenten gleich, hängen also nur vom Modul ab, nicht vom Dozenten.
- Die Kombinationen von Dozenten und Lehrbüchern sind eine Art Sub-Kreuzprodukt.

## Mehrwertige Abhängigkeiten – Beispiel III

- Ein Modul bestimmt nicht eindeutig *einen* Dozenten, sondern eine *Menge* von Dozenten.
- Genauso bestimmt ein Modul eine *Menge* von Lehrbüchern.
- Und diese beiden Abhängigkeiten sind unabhängig voneinander.

⇒ mehrwertige Abhängigkeiten

# Mehrwertige Abhängigkeit

## Definition

Sei  $R_S$  ein Relationenschema.

Eine **mehrwertige Abhängigkeit**  $X \twoheadrightarrow Y$  bedeutet, dass die Werte von  $X$  eine Menge von Werten in  $Y$  bestimmen, unabhängig von den Werten der restlichen Attribute von  $R_S$ .

## Beispiel

$$\{MNo\} \twoheadrightarrow \{DNo\}$$

$$\{MNo\} \twoheadrightarrow \{LNo\}$$

# Triviale mehrwertige Abhängigkeiten

## Definition

Eine mehrwertige Abhängigkeit  $X \twoheadrightarrow Y$  über einem Relationenschema  $R_S$  heißt **trivial**, wenn sie von jeder Relation mit diesem Schema erfüllt wird.

## Satz

*Eine mehrwertige Abhängigkeit ist genau dann trivial, wenn gilt:  
 $Y \subseteq X$  oder  $X \cup Y = R_S$*



# Vierte Normalform (4NF)

## Definition

Eine RelVar  $R$  ist in der **vierten Normalform (4NF)** genau dann, wenn ihr Schema  $R_S$  nur triviale mehrwertige Abhängigkeiten hat.

## Beispiel

Wir zerlegen unser Beispiel:

<u>MNo</u>	<u>DNo</u>
M1	D1
M1	D2

<u>MNo</u>	<u>LNo</u>
M1	L1
M1	L2

# Verbundabhängigkeit

## Definition

Eine Verbundabhängigkeit  $\bowtie \{X_1, \dots, X_n\}$  eines Relationenschemas  $R_S$  ist eine Menge von Teilmengen von  $R_S$ , deren Vereinigung  $R_S$  ergibt.

## Beispiel

- $\bowtie \{\{SNo, SName, City\}, \{City, Status\}\}$
- $\bowtie \{\{SNo, SName\}, \{SNo, Status\}, \{SName, City\}\}$

# Verbundabhängigkeit

## Definition

Eine Relation  $R$  mit dem Relationsschema  $R_S$  erfüllt die **Verbundabhängigkeit**  $\bowtie \{X_1, \dots, X_n\}$ , wenn  $R$  gerade der Verbund der Projektionen auf die  $X_i$  ist.

## Definition

In einer RelVar mit Relationsschema  $R_S$  gilt die **Verbundabhängigkeit**  $\bowtie \{X_1, \dots, X_n\}$ , wenn jede Relation, die der RelVar zugewiesen werden kann, sie erfüllt.

## Beispiel

- $\bowtie \{\{SNo, SName, City\}, \{City, Status\}\}$  gilt
- $\bowtie \{\{SNo, SName\}, \{SNo, Status\}, \{SName, City\}\}$  gilt nicht

# Triviale Verbundabhängigkeiten

## Definition

Eine Verbundabhängigkeit  $\bowtie \{X_1, \dots, X_n\}$  über einem Relationenschema  $R_S$  heißt **trivial**, wenn sie von jeder Relation mit diesem Schema erfüllt wird.

## Satz

*Eine Verbundabhängigkeit ist genau dann trivial, wenn gilt:  
Es gibt ein  $i$  mit  $X_i = R_S$*

# Fünfte Normalform (5NF)

## Definition

Eine RelVar  $R$  ist in der **fünften Normalform (5NF)** genau dann, wenn jede Relation, die die Schlüssel-Bedingungen erfüllt, auch alle Verbundabhängigkeiten erfüllt.

## Beispiel

Wir haben gesehen, dass in der RelVar  $S$  folgende Verbundabhängigkeit gilt:

$$\bowtie \{ \{ SNo, SName, City \}, \{ City, Status \} \}$$

Da der Schlüssel aber nur  $SNo$  ist, ergibt sich diese Verbundabhängigkeit *nicht* aus den Schlüssel-Eigenschaften.

Man muss das Schema also zerlegen, um 5NF zu erreichen.

# BCNF und 5NF

## Satz




*Sei  $R$  eine RelVar in BCNF und  $R$  habe keine zusammengesetzten Schlüssel.*

*Dann ist  $R$  auch in 5NF.*

# Fazit

- Normalisierung dient der Vermeidung von Redundanz
- Normalisierung setzt voraus, dass man die funktionalen, mehrwertigen und Verbund-Abhängigkeiten erkennt – sich also gut im Anwendungsgebiet auskennt
- Je mehr man über die Informationen Bescheid weiß, desto besser kann man das Datenbankschema machen

# Literatur

-  C.J. Date: *Database Design & Relational Theory: Normal Forms & All That Jazz*, O'Reilly 2012
-  G. Saake, K.-U. Sattler, A. Heuer: *Datenbanken: Konzepte und Sprachen* Kapitel 6, mitp 2010
-  W. Kent: *A Simple Guide to Five Normal Forms in Relational Database Theory*, Communications of the ACM, vol. 26 (1983), pp. 120-125