

# Normalisierung

Normalformen dienen als Entscheidungsgrundlage für die Verwendung von Relationenschemata. Sie ermöglichen einen objektiven Vergleich relationaler DB-Schemata.

## Funktionale Abhängigkeit

Bestimmen einige Attribute eindeutig die Werte anderer Attribute, so sind die anderen Attribute funktional abhängig. Daher sind Attribute immer abhängig von Schlüsselattributen. Einige Abhängigkeiten können sich auch transitiv aus anderen ergeben.

## Anomalien

Eine Datenbank sollte so entworfen sein, dass keine Einfüge-, Lösch oder Update-Anomalien auftreten.

**Einfügen** sollte möglich sein, ohne viele NULL-Werte einfügen zu müssen, da Daten fehlen.

**Löschen** sollte möglich sein, ohne zu viele Informationen zu löschen.

**Update** sollte ohne großen Aufwand (z.B. Redundanzen) möglich sein.

Durch Normalisierung soll möglichst Redundanzfreiheit und Eliminierung der Anomalien geschaffen werden.

## Normalformen

### 1. Normalform

Alle Attribute enthalten nur atomare Werte.

Wenn nicht: Ein neues Relationenschema für mehrwertige Attribute erstellen.

#### Probleme

- unnötige Redundanzen
- Anomalien

### 2. Normalform

Zusätzlich zu 1. NF sind keine Attribute nur von einem Teil des Primärschlüssels abhängig.

Wenn doch: Neues Relationenschema für Teilabhängigkeiten

#### Probleme

- Redundanzen



- Anomalien

### 3. Normalform

Zusätzlich zur 2. NF hängt kein Attribut transitiv vom Primärschlüssel ab.  
Wenn doch: Neues Relationenschema.

#### Probleme

- Änderungsaufwand
  - Unvollständiges Einfügen
- es können bei der Zerlegung Informationen verloren gehen.

### BC-Normalform

Für jede funktionale Abhängigkeit gilt:  $X \rightarrow A$ ;  $X$  ist ein Superschlüssel von  $R$ . So sind keine Teile zweier zusammengesetzter Schlüsselkandidaten voneinander abhängig.

#### Probleme

- Die Überführung in BCNF ist nicht immer abhängigkeitswährend
  - Die BCNF kann nicht immer erreicht werden!
- Die Eigenschaften der Abhängigkeitswahrung und der informationserhaltenden Zerlegung sind ebenfalls von Bedeutung für ein insgesamt gutes Relationsschema.

## Abhängigkeitswahrung

### Attributserhaltende Zerlegung

$R$  ist nun ein Relationsschema, das alle Attribute einer DB enthält.  $R$  soll nun so zerlegt werden, dass jedes Attribut in einem  $R_i$  erscheint.

Nun soll jede funktionale Abhängigkeit aus  $R$  auch in  $R_i$  erscheinen oder abgeleitet werden können. Dies ist die Abhängigkeitswahrung.

### Algorithmus zur abhängigkeitswährenden Zerlegung

1. Es muss die minimale Hülle von  $G$  für  $F$  gefunden werden
  1.  $n$  funktionale Abhängigkeiten schaffen.
  2. Abhängigkeiten mit mehreren Attributen als Quelle auf Notwendige reduzieren.
  3. überflüssige Abhängigkeiten streichen.
2. Erzeuge Relationsschemata für funktionale Abhängigkeiten.
3. Bei fehlendem Schlüssel von  $R$  muss ein solches gebildet werden.

## NICHT-additiver JOIN

Wenn der JOIN zweier Relationen zu unechten Tupeln führt, also dessen Informationen nicht




korrekt sind, ist die Zerlegung nicht korrekt.

	A1	A2	A3	A4	...
R1	b11	b12	b13	b14	...
R2	b21	b22	b23	b24	...
R3	b31	b32	b33	b34	...
...	...	...	...	...	...

$$R_1(A_1, A_2) \quad R_2(A_2, A_3) \quad R_3(A_1, A_2, A_3, A_4)$$

1. Setze  $R_i - A_i = b_{ij}$
2. Wenn das Attribut Teil von  $R_i$  ist, setze  $b_{ij} = a_j$

	A1	A2	A3	A4
R1	 a1	a2	b13	b14
R2	b21	a2	a3	b24
R3	a1	a2	a3	a4

3. Nun werden funktionale Abhängigkeiten behandelt  $X \rightarrow Y$ 
  1. Wenn eine Zeile  $a$  enthält aus einer Attributsspalte  $x$ , wird in dieser Zeile von Spalte  $Y$   $Y = a$  gesetzt.  
z.B.  $A_1 \rightarrow A_3$ , dann ist  $b_{13} = a_3$ , aber nur weil  $R3 \times A3 = a_3$
  2. Existiert kein  $a$  in  $Y$  wo  $a$  in  $x$  ist, werden diese Werte auf  $b$  gesetzt.
4. Ist nun eine Zeile vollständig mit  $a$  gefüllt ( $R3$ ), dann hat die Zerlegung die Eigenschaft des nicht-additiven JOIN.

## Datenqualität

### Probleme

- Datenspeicherung
- Datenproduktion
- Datennutzung

- Vollständigkeit
- Genauigkeit
- Zeitnähe



- Relevanz
- Kosten
- Verständlichkeit
- Konsistenz
- Verfügbarkeit
- Glaubwürdigkeit

---

Revision #4

Created 11 August 2022 08:49:02 by Martin Tienken

Updated 4 September 2022 20:41:17 by Martin Tienken