

Wiederholung: Normalformen

Redundanzen im DB-Schema erzeugen Anomalien (Änderungs-, Einfüge-, Entfernungs-Anomalien)

Ziel: Vermeidung von Redundanzen und Anomalien \Rightarrow schrittweises Zerlegen des Schemas in ein äquivalentes Schema ohne Redundanzen und Anomalien = **Normalisierung**

(I) Funktionale Abhängigkeiten

Seien X, Y Attributmengen des Relationenschemas R , d.h. $X, Y \subseteq R$

Y ist von X **funktional abhängig** (oder X bestimmt Y funktional), d.h. $X \rightarrow Y \Leftrightarrow$ für alle mgl. Ausprägungen von R gilt: Zu jedem Wert in X existiert genau ein Wert in Y

Formal:

$$X \rightarrow Y \Leftrightarrow \forall r_1, r_2 \in R : r_1.X = r_2.X \Rightarrow r_1.Y = r_2.Y$$

Bsp.: Matrikelnummer \rightarrow Name

- **Triviale funktionale Abhängigkeit:** $X \rightarrow Y$, falls $Y \subseteq X$
- **Volle funktionale Abhängigkeit:** $X \rightarrow Y$, falls keine echte Teilmenge $X' \subset X$ ex. mit $X' \rightarrow Y$.
- Existiert eine solche Teilmenge, dann heißt $X \rightarrow Y$ **partielle funktionale Abhängigkeit**.
- **Transitive funktionale Abhängigkeit** $X \rightarrow Z$, falls gilt $X \rightarrow Y$ und $Y \rightarrow Z$.

(II) Schlüssel

- Teilmenge S der Attribute eines Relationenschemas R heißt **Schlüssel**, falls gilt
 - (1) Eindeutigkeit: Keine Ausprägung von R kann zwei verschiedene Tupel enthalten, die sich in allen Attributen von S gleichen
 - (2) Minimalität: Keine echte Teilmenge von S erfüllt bereits Bedingung (1)
- Ein Attribut heißt **prim**, falls es Teil eines Schlüsselkandidaten ist

(III) Normalformen

Ziel: schrittweise Beseitigung funktionaler Abhängigkeiten (außer vom gesamten Schlüssel)

- **1. Normalform**
 - Alle Attribute enthalten **atomare** Werte (String, Integer, ...), d.h. keine Tupel, Listen, ...
 - In relationalen DB sind nicht-atomare Werte nicht möglich \Rightarrow relationale DB immer in 1NF
- **2. Normalform**
 - Für jedes Attribut A gilt:
 - * A ist **prim** oder
 - * A ist **voll funktional abhängig von jedem Schlüsselkandidaten**
 - Beseitigt partielle funktionale Abhängigkeiten nicht-primere Attribute vom Schlüssel
 - 2NF kann nur verletzt werden, falls Schlüsselkandidat zusammengesetzt ist
 - Transformation in 2NF:
 - * Erstelle eine neue Relation für jeden partiellen Schlüssel mit seinen abhängigen Attributen.
 - * Attribute, die voll funktional vom Schlüssel abhängig sind, bleiben in der ursprünglichen Relation

- **3. Normalform**

- Für alle nicht-trivialen funktionalen Abhängigkeiten $X \rightarrow Y$ gilt:
 - * X **enthält Schlüsselkandidaten** oder
 - * Y **ist prim**
- Beseitigt funktionale Abhängigkeiten nicht-primärer Attribute untereinander (= transitive Abhängigkeiten)
- 3NF impliziert 2NF
- Transformation in 3NF:
 - * Erstelle eine neue Relation für alle Nicht-Schlüssel-Attribute und deren funktionalen Abhängigkeiten
 - * Attribute, die voll funktional vom ursprünglichen Schlüssel abhängig und nicht abhängig von Nicht-Schlüssel-Attributen sind, bleiben in der ursprünglichen Relation

- **Boyce-Codd Normalform**

- Für alle nicht-trivialen funktionalen Abhängigkeiten $X \rightarrow Y$ gilt: X **enthält Schlüsselkandidaten**
- Beseitigt funktionale Abhängigkeiten unter Attributen, die prim sind, aber nicht vollständig einen Schlüssel bilden
- BCNF impliziert 3NF
- Man kann nicht immer eine BCNF-Zerlegung finden, die Abhängigkeiten bewahrt

Wiederholung: Zerlegung von Relationen

Zerlegung von R in R_1, \dots, R_n ist

- **verlustlos**, falls gilt:
Jede mögliche Ausprägung r von R lässt sich durch den natürlichen Join der Ausprägungen r_1, \dots, r_n konstruieren: $r = r_1 \bowtie \dots \bowtie r_n$
- **abhängigkeitserhaltend**, falls gilt:
Alle funktionalen Abhängigkeiten F auf R bleiben in den lokalen funktionalen Abhängigkeiten F_i bewahrt: $F = F_1 \cup \dots \cup F_n$