5. Relationaler Datenbank-Entwurf

- zum Verfeinern des logischen Entwurfs
- Problem: Wann ist der logische Entwurf gut?
- Ziel: Vermeidung von Redundanzen durch Aufspaltung von Relationenschemata
  - ohne semantische Information zu verlieren (Abhängigkeitstreue)
  - ohne Möglichkeit zur Rekonstruktion zu verlieren (Verbundtreue)
- Hilfsmittel: Normalformen

5.1 Normalformen

- verhindern Redundanz
- sichern guten DB-Schema-Entwurf zu
- **1. Normalform** (1NF):
  - alle Attributwerte sind atomar
  - Bsp.:

LName	LAdresse	Artikelnrs
Meier	Bochum	3,7,19
Müller	Essen	3,4
...	...	...

    nicht erlaubt, da Artikelnrs strukturiert

  - Bsp. in erster Normalform:

LName	LAdresse	Artikelnrs
Meier	Bochum	3
Meier	Bochum	7
Meier	Bochum	19
Müller	Essen	3
Müller	Essen	4
...	...	...

Normalformen

- **2. Normalform** (2NF):
  - Relation R ist in 1 NF und jedes Nicht-Primattribut von R hängt voll von jedem Schlüssel ab
  - = keine partiellen Abhängigkeiten zwischen Schlüssel und weiteren Attributen
  - Definition: *Funktionale Abhängigkeit* (FD) zwischen Attributmenge X und Attributmenge Y (X→Y): falls Tupel in den Spalten X übereinstimmen, so stimmen sie auch in den Spalten Y überein
  - Bsp.:

Auftr.nr	Datum	KName	LName	Artikel	Menge
4711	4.6.96	Schulze	Meier	3	10
4711	4.6.96	Schulze	Meier	7	2
4712	4.6.96	Schnell	Müller	4	6
4713	7.6.96	Schmidt	Mauser	3	1
4713	7.6.96	Schmidt	Mauser	122	12

    FDs bspw. Auftr.nr → Datum,KName,LName
    Auftr.nr,Artikel → Datum,KName, Menge
    Auftr.nr,Datum → Auftr.nr.,Datum,LName

  - steht rechts von → ganzes Relationenschema und ist linke Seite minimal: linke Seite Schlüssel
  - partielle Abhängigkeit: ein Attribut hängt schon funktional von einem TEIL des Schlüssels ab
  - im Bsp.: Schlüssel ist Auftr.nr,Artikel; Datum hängt aber nur von Auftr.nr ab

Normalformen

- – Erreichen der 2NF durch Elimination der rechten Seite der part. Abhängigkeit, linke Seite wird kopiert
  – im Bsp.:

Auftr.nr.	Artikel	Menge
4711	3	10
4711	7	2
4712	4	6
4713	3	1
4713	122	12

Auftr.nr.	Datum	KName	LName
4711	4.6.96	Schluze	Meier
4712	4.6.96	Schnell	Müller
4713	7.6.96	Schmidt	Mauser

FDs formal

Definition 1: Funktionale Anhängigkeit (FD)
Sei R $\subseteq$ U, r(R), X,Y $\in$ R; R genügt der funktionalen Abhängigkeit X$\rightarrow$Y :$\Leftrightarrow$
$\forall$ X-Werte x gilt: $\mid \pi_y(\sigma_{X=x}(r)) \mid \leq 1 \leftrightarrow$
$\forall t_1, t_2 \in r : t_1(X) = t_2(X) \Rightarrow t_1(Y) = t_2(Y)$

Bemerkung: eine Menge von FDs F kann als Menge von lokalen Integritätsbedingungen aufgefaßt werden

Beispiel:

r:

A	B	C
a1	b1	c1
a2	b1	c1
a3	b2	c1
a4	b2	c1

r genügt den FDs A$\rightarrow$B, B$\rightarrow$C, also auch A$\rightarrow$C, aber nicht C$\rightarrow$A oder C$\rightarrow$B

Definition 2: Die Hülle von F ist $F^+ := \{$f$\mid$ F impliziert f (F$\models$f)$\}$

Bemerkung: $\models$ kann über Ableitungsregeln simuliert werden; folgende Ableitungsregeln sind gültig, vollständig und unabhängig:

1. Reflexivität: $\{\} \Rightarrow$ r genügt X$\rightarrow$X

2. Akkumulation: r genügt $\{$X$\rightarrow$YZ, Z$\rightarrow$VW$\}$ $\Rightarrow$ r genügt x$\rightarrow$YZV

3. Projektivität: r genügt X$\rightarrow$YZ $\Rightarrow$ r genügt X$\rightarrow$Y

Beispiel: $\{A\to B,\ B\to C\} \models A\to BC \models A\to C$

Definition 3: Die Hülle einer Attributmenge X bzgl. F ist $X_F^* = \{A \mid X\to A \in F^+\}$

Membership-Problem: '$X\to Y \in F^+$?' lösen durch '$Y \subseteq X_F^*$?'

Vorgehen:

1. bestimme X, setze $X^* := X$

2. gibt es FD $f_1 := X_1 \to Y_1 \in F$ mit $X_1 \subseteq X^*$?

3. Ja: dann $X^* := X^* \cup Y_1$

4. führe 2.,3. solange aus, bis X^* stabil; ist $Y \subseteq X^*$, dann ist $x\to Y \in F^+$

Beispiel: $A\to C \in \{A\to B,\ B\to C\}$?
1. $A^* := A$
2.,3. für $A\to B$ gilt: $A \in A^*$, also $A^* := A \cup B = AB$
für $B\to C$ gilt: $B \in AB$, also $A^* := AB \cup C = ABC$
4. $C \in ABC$, also $A\to C \in \{A\to B,\ B\to C\}$

Definition 4: Sei R Relationenschema; eine FD $X\to R$ heißt Schlüsselabhängigkeit, X ist dann Schlüssel von R

Definition 5: Seien F,G FD-Mengen über U; F heißt äquivalent zu G (F Überdeckung von G), kurz $F \equiv G$, $:\Leftrightarrow F^+ = G^+ \Leftrightarrow$
$F \models \{g \mid g\in G\} \land G \models \{f \mid f\in F\} \Leftrightarrow$
$\forall\, g\in G: g\in F^+ \land \forall\, f\in F: f\in G^+$

Normalformen

- **3. Normalform** (3NF):
  - Relation R ist in 1NF und kein Nicht-Primattribut hängt transitiv von einem Schlüssel ab
  - = keine transitiven Abhängigkeiten zwischen Schlüssel und weiteren Attributen
  - Bsp.:

<u>Auftr.nr</u>	LName	LAdresse
4711	Meier	Bochum
4712	Müller	Essen
4713	Mauser	Essen
4714	Meier	Bochum

FDs: Auftr.nr $\rightarrow$ LName, LName $\rightarrow$ LAdresse

  - Transitive Abhängigkeit: Attributmenge X hängt vom Schlüssel ab, ein anderes Attribut A hängt von X ab
  - im Bsp.: Schlüssel ist Auftr.nr., LName hängt von Auftr.nr ab, aber LAdresse hängt allein von LName ab
  - Erreichen der 3NF durch Elimination von A, X wird kopiert
  - im Bsp.:

Auftr.nr	LName
4711	Meier
4712	Müller
4713	Mauser
4714	Meier

LName	LAdresse
Meier	Bochum
Müller	Essen
Mauser	Essen

- **4. Normalform** (4NF):
  - Keine zwei mehrwertigen Abhängigkeiten zwischen Attributen
  - Definition: mehrwertige Abhängigkeit (MD) zwischen Attributmenge X und Attributmenge Y ($X \rightarrow\rightarrow Y$), falls einem Wert in den Spalten X mehrere Werte in den Spalten Y zugeordnet werden
  - Bsp.:

Name	Kind	Hobby
Willy Winzig	Hugo	Wandern
Willy Winzig	Egon	Wandern
Willy Winzig	Hugo	Wasserball
Willy Winzig	Egon	Wasserball
Willy Winzig	Hugo	Wirtshäuser
Willy Winzig	Egon	Wirtshäuser

mehrwertige Abhängigkeiten: Name $\rightarrow\rightarrow$Kind, Name$\rightarrow\rightarrow$Hobby

  - Erreichen der 4NF durch Elimination der rechten Seite einer der beiden mehrwertigen Abhängigkeiten, linke Seite wird kopiert
  - im Bsp.:

Name	Kind
Willi Winzig	Hugo
Willi Winzig	Egon

Name	Hobby
Willi Winzig	Wandern
Willi Winzig	Wasserball
Willi Winzig	Wirtshäuser

5.2 Datenbankschemaeigenschaften

DB-Entwurfsproblem:

- aus gegebenem U und FD-Menge F soll lokal erweitertes DB-Schema $S = \{R_1,...,R_p\}$ mit R_i=(R_i,K_i) erzeugt werden, wobei K_i Menge von Schlüsseln ist, denen die Basisrelationen r(R_i) genügen
- es sollen nur noch (unvermeidbare) Redundanzen bei Schlüsseln auftreten
- alle FDs sollen durch Schlüssel dargestellt werden können
- urspr. Relation soll immer durch Anfrageoperationen wiederhergestellt werden können
- man produziere dabei nicht unnötig viele Tabellen

Datenbankschemaeigenschaften

Datenbankschemaeigenschaften:

- **N1**: Abhängigkeitstreue

  S charakterisiert vollständig F $\Leftrightarrow$ F $\equiv \{K \rightarrow R \mid \mathcal{R}=(R,\mathcal{K}) \in S,\ K \in \mathcal{K}\}$, d.h. Hülle der Schlüsselabhängigkeiten ist gleich Hülle von F

- **N2**: Normalformen
  - man fordert meist 3NF für guten Entwurf
  - es ex. Boyce-Codd-NF (BCNF; R ist in 1NF und kein Attribut hängt transitiv von einem Schlüssel ab)
  - BCNF $\Rightarrow$ 3NF $\Rightarrow$ 2NF, aber BCNF und N1 i.A. unvereinbar

- **N3**: Minimalität
  - zur Vermeidung globaler Redundanzen
  - $\nexists$ S': $|S'|<|S|$, S' erfüllt N1, N2, N3, N4

Datenbankschemaeigenschaften

- **N4**: Verbundtreue
  - für die korrekte Informationswiedergewinnung
  - Forderung: die Dekomposition von U in $R_1,..,R_p$ ist bzgl. F $\pi \bowtie$-treu (verbundtreu)
  - Definition: die Dekomposition einer Attributmenge X in $X_1,..,X_p$ mit $X_1 \cup ... \cup X_p = X$ heißt $\pi \bowtie$-treu bzgl. einer FD-Menge F über X $:\Leftrightarrow$ $\pi_{X_1}(r) \bowtie ... \bowtie \pi_{X_p}(r) = r$
  - Bsp.:

    a) Dekomposition von ABC in AB, BC ist nicht $\pi \bowtie$-treu bzgl. $\{A \rightarrow B,\ C \rightarrow B\}$
    Quicky: warum?

    b) Dekomposition von ABC in AB, BC ist $\pi \bowtie$-treu bzgl. $\{A \rightarrow B,\ B \rightarrow C\}$

- formale Algorithmen zum Erreichen von N1 bis N4:
  - Dekomposition: liefert N2, N4
  - Synthese: liefert N1 bis N4