

11 Normalisierung von Datenbanken

Entitäten und Schlüssel

Entität	Objekt der realen Welt (z. B. ein Kunde, eine Rechnung, ein Universitätskurs)
Attribut	Eigenschaft zur Beschreibung einer Entität
Entitätstyp	Klasse gleichartiger Entitäten
Entitätsmenge	Menge von Entitäten eines bestimmten Typs

Sei EM Entitätsmenge mit der Attributmengung $AM = \{A_1, A_2, \dots, A_n\}$. R sei Relationsschema über der Attributmengung AM . Seien X, Y und Z Attributkombinationen (Teilmengen) von EM und r sei Relation über R .

Ein Attribut A aus AM heißt **Schlüsselattribut** auf EM genau dann, wenn für alle Entitäten E, E' aus EM gilt: wenn $E \neq E'$, so auch $A(E) \neq A(E')$. Das heißt, dass keine zwei Attributwerte in der Spalte von A übereinstimmen. Die Attributwerte identifizieren somit die Entitäten eindeutig.

Eine Teilmenge $S = \{A_{s1}, \dots, A_{sk}\}$ von AM heißt **Schlüssel (Candidate Key)** genau dann, wenn a) für alle $E \neq E'$ auch $(A_{s1}(E), A_{s2}(E), \dots, A_{sk}(E)) \neq (A_{s1}(E'), A_{s2}(E'), \dots, A_{sk}(E'))$ gilt (Eindeutigkeit, S ist **Superkey**) und b) wenn keine Teilmenge von S diese Eigenschaft auch hat (Minimalität).

Im Allgemeinen kann eine Relation mehrere Schlüssel besitzen. Zum Zweck der Normalisierung einer Datenbank legt man einen davon als **Primärschlüssel (Primary Key)** fest.

Funktionale Abhängigkeiten

Y ist **funktional abhängig** von X (Schreibweise: $X \rightarrow Y$), genau dann, wenn für alle Entitäten E, E' aus r gilt: $X(E) = X(E') \Rightarrow Y(E) = Y(E')$.

Z heißt **transitiv abhängig** von X über Y genau dann, wenn $(X \rightarrow Y)$ und nicht $(Y \rightarrow X)$ und $(Y \rightarrow Z)$.

1. Normalform (1NF)

Eine Relation (Tabelle) befindet sich in 1NF, wenn im Kreuzungspunkt einer jeden Zeile und Spalte genau ein nicht zusammengesetzter (atomarer) Attributwert steht.

2. Normalform (2NF)

Eine Relation befindet sich in 2NF, wenn sie in 1NF ist und alle Nichtschlüsselattribute vom gesamten (Primär-)Schlüssel funktional abhängig (nicht aber von Teilschlüsselattributen) sind.

3. Normalform (3NF)

Eine Relation in 1NF befindet sich auch in 3NF, wenn kein Nichtschlüsselattribut von einem Schlüsselattribut transitiv abhängig ist.

3. Normalform (3NF) – äquivalente Definition

Eine Relation ist in 3NF, wenn f. a. $Y \subset AM, A \notin Y, A$ Nichtschlüsselattribut gilt:
 $Y \rightarrow A \Rightarrow Y$ ist Schlüssel der Relation.

Boyce-Codd-Normalform (BCNF)

Eine Relation ist in BCNF, wenn f. a. $Y \subset AM, A \notin Y$ (A jetzt beliebiges Attribut) gilt:
 $Y \rightarrow A \Rightarrow Y$ enthält einen Schlüssel.

Algorithmus zur Erlangung der 2NF

- Primärschlüssel für Relation r festlegen, falls dieser nur aus einem Attribut besteht, so liegt bereits 2NF vor
- Relation auf Teilschlüsselabhängigkeiten untersuchen
- Neue Relation r' bilden, die einen Teilschlüssel (dieser ist nun Primärschlüssel von r') und die allein von ihm funktional abhängigen Nichtschlüsselattribute enthält
- Streichen der Nichtschlüsselattribute aus der Relation r
- Vorgang ab dem 2. Punkt wiederholen bis alle Nichtschlüsselattribute von r vom gesamten Schlüssel funktional abhängig sind.

Algorithmus zur Erlangung der 3NF

Die Relation r liege in 1NF vor, und die Attributkombination Z sei transitiv abhängig von X über Y :

- Neue Relation r' aus Y und Z bilden mit Schlüssel Y
- Z aus r streichen
- Verfahren wiederholen bis keine transitiven Abhängigkeiten mehr vorhanden sind

Algorithmus zur Erlangung der BCNF

Die Relation r liege in 1NF aber nicht in BCNF vor:

- Funktionale Abhängigkeit $Y \rightarrow A$, die BCNF verletzt, finden
- Neue Relation r' aus Y und A bilden mit Schlüssel Y
- A aus r streichen
- Verfahren wiederholen bis alle Relationen in BCNF


Aufgabe 11.1:

Liegt folgende Relation in 1NF vor? Wie müsste im Nein-Fall die 1NF aussehen?

werkzeug					
NR	BEZ	FIRMA	STADT	LAND	MENGE
231	Säge	Zwick	Aue	Sachsen	30
		Stumpf	Celle	N-Sachs	15
		Schief	Burg	Anhalt	10
		Krach	Berlin	Berlin	10
		Schwung	Berlin	Berlin	20
232	Säge	Stumpf	Celle	N-Sachs	30
368	Beil	Rumms	Langen	Bayern	12
		Zuck	Suhl	Thür	78
		Ruck	Erfurt	Thür	117
427	Zange	Schnapp	Köln	NRW	24
		Schief	Burg	Anhalt	5
587	Bohrer	Zuck	Suhl	Thür	78
		Schwung	Berlin	Berlin	37

Aufgabe 11.2:

- Fertigen Sie von der Tabelle *werkzeug* aus der Datenbank *PRAKTIK* eine Kopie *werkkopie* an.
- Normalisieren Sie die Tabelle *werkkopie* wie folgt mit Hilfe von Access:
 - $\{NR, FIRMA\}$ ist Primärschlüssel.
 - *STADT* und *LAND* hängen funktional allein vom Teilschlüssel *FIRMA* ab.
 - **Tabellenerstellungsabfrage** durchführen, die nur die Felder *FIRMA*, *STADT* und *LAND* enthält und diese unter dem Namen *Firmen* in der aktuellen Datenbank abspeichern. Damit jeder Datensatz genau einmal in der Tabelle aufgenommen wird, wie folgt vorgehen: (bei

geöffnetem Fenster *Tabellenerstellungsabfrage*) in der Symbolleiste auf  (*Eigenschaften*) klicken und bei *Keine Duplikate* den Eintrag *ja* wählen. Jetzt die *Abfrage ausführen*.

- Die Felder *STADT* und *LAND* aus dem Entwurf der Tabelle *werkkopie* löschen.
- Welche Felder hängen allein von *NR* ab? Setzen Sie den Algorithmus selbstständig fort, bezeichnen Sie die neue Tabelle mit *Werkzeuge*.
- Benennen Sie die alte Tabelle *werkkopie* in *Bestellungen* um.

Aufgabe 11.3:

Untersuchen Sie die Ergebnis-Tabellen aus Aufgabe 11.2 auf transitive Abhängigkeiten und transformieren Sie sie gegebenenfalls in 3NF.

Als Ergebnis der Normalisierung erhält man folgende vier Tabellen:

Bestellungen		
NR	FIRMA	MENGE
231	Krach	10
231	Schief	10
231	Schwung	20
231	Stumpf	15
231	Zwick	30
232	Stumpf	30
368	Ruck	117
368	Rumms	12
368	Zuck	78
427	Schief	5
427	Schnapp	24
587	Schwung	37
587	Zuck	78

Firmen	
FIRMA	STADT
Krach	Berlin
Ruck	Erfurt
Rumms	Langen
Schief	Burg
Schnapp	Suhl
Schwung	Berlin
Stumpf	Celle
Zuck	Suhl
Zwick	Aue

Werkzeuge	
NR	BEZ
231	Säge
232	Säge
368	Beil
427	Zange
587	Bohrer

Städte	
STADT	LAND
Aue	Sachsen
Berlin	Berlin
Burg	Anhalt
Celle	N-Sachs
Erfurt	Thür
Köln	NRW
Langen	Bayern
Suhl	Thür

Aufgabe 11.4:

- Begründen Sie, warum $\{Student, LV\}$ ein Schlüssel für folgende Tabelle ist.
- Warum befindet sich die Tabelle in 3NF, aber nicht in BCNF?
- Überführen Sie die Tabelle in BCNF.

S LV L		
Student	LV	Lehrkraft
Meier	DB	Voelkel
Schmidt	DB	Linsen
Schmidt	BS	Lehmann
Schmidt	TI	Hemmerling
Walter	DB	Voelkel
Walter	BS	Schulze
König	DB	Liebscher
Köhler	DB	Linsen

Aufgabe 11.5:

Es sei die folgende Tabelle *PrüfInfo* mit Informationen über Prüfungen gegeben.

PrüfNr	Prüfer	Art	Prüfling	ImmNr	Wohnort	PLZ	Note
00815	Netto	schrl.	Bang, E.	216	Adorf	37073	3
			Unsich, R.	312	Bestadt	17961	3
			Star, K.	325	Cehausen	92500	1
			Star, K.	417	Adorf	37073	2
04711	Brutalo	mdl.	Sehrgu, T.	111	Bestadt	16961	2
			Bang, E.	216	Adorf	37073	4
30296	Dabanko	schrl.	Angs, T.	137	Deweiler	28122	3
			Star, K.	417	Adorf	37073	1
			Unsich, R.	312	Bestadt	17961	4
			Mitte, L.	621	Cehausen	92500	2
71533	Netto	mdl.	Star, K.	325	Cehausen	92500	2
			Angs, T.	137	Deweiler	28122	3

- Liegt die Tabelle schon in **erster Normalform (1NF)** vor? Begründen Sie Ihre Antwort! Wie müsste im Nein-Fall die 1NF aussehen?
- Für die 1NF ist das Attributpaar (**PrüfNr, ImmNr**) ein Schlüssel, den wir als **Primärschlüssel** auszeichnen.
 - Transformieren Sie damit die Relation in die **zweite Normalform (2NF)**.
 - Beschreiben Sie die bei dieser Transformation ausgeführten Operationen.
- Ist die als Resultat von b) entstandene Datenbank auch schon eine in der **dritten Normalform (3NF)**? Begründen Sie Ihre Antwort!
Führen Sie, falls keine vorlag, eine **Transformation in die 3NF** durch und beschreiben Sie die dabei ausgeführten Operationen.

Aufgabe 11.6:

- Kopieren Sie die Datenbank *PRUEFUNG* in den eigenen Ordner, öffnen Sie die Tabelle *Prüf-Info* und informieren Sie sich über den Tabellenentwurf.
- Normalisieren Sie die Tabelle bis zur 3NF mit Hilfe von **Access** (legen Sie auch Schlüssel fest).

Aufgabe 11.7:

Fertigen Sie von der Tabelle *rechnung* der Datenbank *VERSAND* eine Kopie an und überführen Sie die kopierte Tabelle in 3NF.