



B	Lehrgang: Datenbanken	Arbeitsblatt Nr. 14
S	Thema: Normalisierung	Datum:
G	Name:	Seite 1 von 4

Normalisierung von Relationen

Nach der Umsetzung eines ER-Diagramms in das relationale Modell können (aber müssen nicht!) noch Redundanzen in den Tabellen enthalten sein, die Probleme verursachen -abgesehen von dem zusätzlich erforderlichen Speicherplatz für die redundanten Daten.

An einer Beruflichen Schule werden Fächer von Lehrern unterrichtet. Die Lehrer gehören zu einer Abteilung, welche von einem Abteilungsleiter geführt wird. Ein Eintrag in der Datenbank könnte wie folgt aussehen:

LEHRER_ID	ABTEILUNG	ABTLTR_ID	KLASSE	FACH	ANZ_SCHUELER
1238	METALL	1050	13 BG M1	TW	23
1239	ELEKTRO	1438	12 EL O2	BTH	19
1237	METALL	1050	13 BG M1	TECH	23
1240	HAUSWIRT	1349	11 BF H3	EK	25

Anomalien

Fehler im Datenbestand -sogenannte Anomalien- können durch die Pflege des Datenbestandes entstehen. Es gibt drei Arten von Anomalien:

1. Löschanomalie ↗ würde der Lehrer mit der ID 1237 in der Tabelle, in der die zugehörigen Personendaten enthalten sind, wegen z.B. seiner Pensionierung gelöscht, hätte die Klasse 13 BG M1 keinen Technologieunterricht.
2. Einfügeanomalie ↗ würde für die Klasse 13 BG M1 neuerdings Sport angeboten und gäbe es z.Z. keinen Sportlehrer an der Schule, enthielten die ersten Spalten NULL-Werte, die bei Abfragen wiederum zu Problemen führen würden.
3. Änderungsanomalie ↗ würde sich die Schülerzahl in der Klasse 13 BG M1 ändern, müssten zwei Datensätze verändert werden. Sollte einer der beiden nicht geändert werden, wäre der Datenbestand inkonsistent; d.h. es gäbe widersprüchliche Aussagen über die Schülerzahl, je nach dem welcher Datensatz zu Rate gezogen würde

Zur Vermeidung von solchen Inkonsistenzen und Anomalien sind Tabellen in eine wohldefinierte Form zu bringen.

Normalformen

Insgesamt existieren fünf Normalformen und die sogenannte Boyce-Codd-Normalform. In der Praxis werden Datenbanken in die dritte Normalform, möglicherweise noch in die Boyce-Codd-Normalform überführt. Die vierte und fünfte Normalform erfordern einen zumeist unangemessenen Aufwand. Für die Normalformen gilt: Befindet sich eine Datenbank in einer bestimmten Normalform, sind bereits alle vorigen Normalformen erfüllt.



B S G G	Lehrgang: Datenbanken	Arbeitsblatt Nr. 14
	Thema: Normalisierung	Datum:
	Name:	Seite 2 von 4

Unnormalisierte Daten

Als Beispiel soll folgende Tabelle einer Adresssammlung dienen:

Vorname	Name	Adresse	Geburtstag	Alter	Telefon
Fred	Feuerstein	12345 Steintal, Am Fels 23	23.04.54	50	763458
Roman	Koslowski	56234 Mainz, Lindweg 3	31.03.47	57	21345
Fred	Feuerstein	76423 Ulm, Geröllweg 5	12.08.63	41	87621

Tabelle ADRESSEN

1. Normalform

Eine Relation befindet sich in der 1. Normalform, wenn alle zugrunde liegenden Wertebereiche nur atomare Werte enthalten.

In der obigen Tabelle wird die erste Normalform verletzt. So sind die Adressen nicht in atomarer Weise gespeichert.

Vorname	Name	Plz	Stadt	Straße	Geburtstag	Alter	Telefon
Fred	Feuerstein	12345	Steintal	Am Fels 23	23.04.54	50	763458
Roman	Koslowski	56234	Mainz	Lindweg 3	31.03.47	57	21345
Fred	Feuerstein	76423	Ulm	Geröllweg 5	12.08.63	41	87621

Tabelle ADRESSEN

Zur anschließenden Überführung in die zweite Normalform, ist die Wahl eines Primärschlüssels erforderlich. Für diese Tabelle existieren mehrere Schlüsselkandidaten; u.a. die Attributkombination **Name** und **Plz**. Als Primärschlüssel wird daher diese Attributkombination gewählt.

2. Normalform

Eine Relation befindet sich in der 2. Normalform, wenn

1. sie die erste Normalform erfüllt, und
2. wenn jedes Nichtschlüsselattribut voll funktional abhängig ist vom Primärschlüssel.

Ein Nichtschlüsselattribut darf also nicht vollständig von einem Teil des Primärschlüssels abhängen! Eine Relation, **die keinen zusammengesetzten Schlüssel** hat und sich in der ersten Normalform befindet, ist automatisch in der zweiten Normalform.

In diesem Beispiel ist der Primärschlüssel jedoch zusammen gesetzt. Er besteht aus dem Namen und aus der Plz. Nun ist jedoch die Stadt (ein Nichtschlüsselattribut) funktional abhängig von der Plz (Teil des Primärschlüssels).

Um diese Tabelle ADRESSEN in die zweite Normalform zu überführen, muss diese funktionale Abhängigkeit beseitigt werden.

Hierzu wird das Teilschlüsselattribut (hier Plz) mit den voll funktional abhängigen Nichtschlüsselattributen (hier Stadt) in eine weitere Tabelle PLZ ausgelagert. Dort bildet dieser Teilschlüssel den neuen Primärschlüssel (hier Plz).



Vorname	Name	Plz	Straße	Geburtstag	Alter	Telefon
Fred	Feuerstein	12345	Am Fels 23	23.04.54	50	763458
Roman	Koslowski	56234	Lindweg 3	31.03.47	57	21345
Fred	Feuerstein	76423	Geröllweg 5	12.08.63	41	87621

Tabelle ADRESSEN

Plz	Stadt
12345	Steintal
56234	Mainz
76423	Ulm

Tabelle PLZ

3. Normalform

Eine Relation befindet sich in der 3. Normalform, wenn

1. sie die erste und zweite Normalform erfüllt, und
2. wenn alle Nichtschlüsselattribute gegenseitig unabhängig, aber voll funktional abhängig vom gesamten Primärschlüssel sind (keine transitiven Abhängigkeiten)

Mathematisch lässt sich solch eine transitive Abhängigkeit wie folgt darstellen:

$X \rightarrow Y, Y \rightarrow Z$, aber es gilt nicht: $Y \rightarrow X$

Aus dem Attribut X folgt das Attribut Y und aus dem Attribut Y folgt das Attribut Z. Aber aus Y folgt nicht X. Eine transitive Abhängigkeit existiert in der Tabelle, die sich momentan in der zweiten Normalform befindet:

Aus dem zusammengesetzten Schlüssel (Name+Plz; dies wäre X) folgt der Geburtstag als Nichtschlüsselattribut (dies wäre Y). Umgekehrt kann vom Geburtstag nicht auf den zusammengesetzten Schlüssel gefolgert werden! Das Nichtschlüsselattribut Alter (dies wäre Z) ist jedoch voll funktional vom Attribut Geburtstag (Attribut Y) abhängig. Das Alter einer Person kann aus dem Geburtstag und dem jeweils aktuellen Datum berechnet werden.

Wenn also zwei oder mehr Attribute einer Tabelle, die alle nicht zum Schlüssel gehören, voneinander funktional abhängig sind, muss diese sogenannte transitive Abhängigkeit aufgelöst werden. Eine solche Auflösung erfolgt unterschiedlich. Im oben dargestellten Fall kann das Attribut Alter vollständig entfallen!

Sehen sie sich einen anderen Fall an, bei dem ebenfalls die dritte Normalform verletzt ist. Hierzu sei die obige Tabelle um die Attribute Abtnr (für Abteilungsnummer) und Abtbez (für Abteilungsbezeichnung) ergänzt (die Tabelle PLZ sei für die nachfolgende Betrachtung irrelevant):

Vorname	Name	Plz	Straße	Geburtstag	Telefon	Abtnr	Abtbez
Fred	Feuerstein	12345	Am Fels 23	23.04.54	763458	3	Personal
Roman	Koslowski	56234	Lindweg 3	31.03.47	21345	7	Entwicklung
Fred	Feuerstein	76423	Geröllweg 5	12.08.63	87621	5	Haustechnik

Tabelle ADRESSEN

Übertragen Sie die oben geschilderten Zusammenhänge auf diese Tabelle. Wiederum soll der zusammengesetzte Schlüssel mit X bezeichnet sein. Das Nichtschlüsselattribut Abtnr sei mit Y und das Nichtschlüsselattribut Abtbez mit Z bezeichnet.



Somit folgt aus $X=\{\text{Name, Plz}\}$ das Attribut $Y=\{\text{Abtnr}\}$; allerdings nicht umgekehrt! Und aus dem Attribut Y folgt das Attribut $Z=\{\text{Abtbez}\}$. Diese nun vorliegende transitive Abhängigkeit wird nun durch Auslagern der transitiv abhängigen Attribute in eine eigene Tabelle, z.B. ABTEILUNG aufgelöst. Es ergibt sich die neue nachfolgende Struktur:

Vorname	Name	Plz	Straße	Geburtstag	Telefon	Abtnr
Fred	Feuerstein	12345	Am Fels 23	23.04.54	763458	3
Roman	Koslowski	56234	Lindweg 3	31.03.47	21345	7
Fred	Feuerstein	76423	Geröllweg 5	12.08.63	87621	5

Abtnr	Abtbez
3	Personal
5	Haustechnik
7	Entwicklung

Tabelle ADRESSEN

Tabelle ABTEILUNG

Hiermit soll die Betrachtung der Normalformen mit dem Hinweis, dass noch insgesamt drei weitere Normalformen existieren, abgeschlossen werden.

- Eine etwas stärkere Form der dritten Normalform stellt die sogenannte **Boyce-Codd-Normalform** (abgekürzt: BCNF) dar.
- mit mehrwertigen funktionalen Abhängigkeiten beschäftigt sich die **vierte Normalform**.
- In der **fünften Normalform** geht es um die Zusammenfassung von mehreren Tabellen durch sogenannte "Joins".

Weitere Informationen und andere Beispiele finden sich unter der URL

<http://v.hdm-stuttgart.de/~riekert/lehre/db-kelz/chap4.htm>

(Inhalt: <http://v.hdm-stuttgart.de/~riekert/lehre/db-kelz/index.htm>)

In der Praxis genügt es im allgemeinen eine Normalisierung bis zur dritten Normalform durchzuführen!

Übungen

Die Daten eines Busreiseunternehmens sollen in einer Datenbank gespeichert werden. Die folgende Abbildung zeigt die Daten in einer nicht normalisierten Form:

Attribut	Beschreibung	Attribut	Beschreibung	Attribut	Beschreibung
FahrtNr	Nummer der Fahrt	BusNr	Nummer des Busses	FDatum	Datum einer Fahrt
Hersteller	Herstellerfirma des Busses	AnzPlatz	Anzahl der Plätze im Bus	FZeit	Dauer der Fahrt
PolKz	Polizeiliches Kennzeichen Bus	EZul	Erstzulassung des Busses	FVname	Vorname des Busfahrers
APreis	Anschaffungspreis des Busses	gew_am	Datum der Wartung	FPLZ	Postleitzahl des Busfahrers
gew_von	Name der Wartungsfirma	Maengel	festgestellte Mängel bei Wart.	FTelefon	Telefonnummer Busfahrer
Kosten	Kosten für Wartung/Reparatur	FZiel	Ort, an den der Bus fahren soll	FStrasse	Strasse, in der der Fahrer wohnt
g_Platz	gebuchte Plätze einer Fahrt	AZeit	Zeit der Busabfahrt	FOrt	Wohnort des Busfahrers
		FName	Name des Busfahrers	FLohn	Gehalt des Fahrers

1. Erstellen Sie für jede Entitätsklasse eine Tabelle mit den Informationen:

Attribut	Datentyp	Schlüssel	Referenztablelle
-----------------	-----------------	------------------	-------------------------
2. Entwerfen Sie ein ER-Modell und überführen Sie dieses in das relationale Modell
3. Normalisieren Sie das Ergebnis bis hin zur dritten Normalform
4. Erzeugen Sie die Datenbank in Access und protokollieren Sie die Befehle