

## Inhaltsverzeichnis

<b>1</b>	<b>Normalisierung</b>	<b>1</b>
1.1	Vorbemerkungen . . . . .	1
1.2	Die 1. Normalform . . . . .	1
1.3	Die 2. Normalform . . . . .	2
1.4	Die 3. Normalform . . . . .	3
1.5	Übungsbeispiel . . . . .	4

## 1 Normalisierung

### 1.1 Vorbemerkungen

Von einer brauchbaren Datenbank fordert man auf alle Fälle die Widerspruchsfreiheit. Abfragen zu bestimmten Aspekten dürfen nicht zu sich widersprechenden Ergebnissen führen. Solche Widersprüche (oder auch **Anomalien** genannt) können sich ergeben, wenn beim Ändern, Einfügen und Löschen von Daten nicht alle erforderlichen Stellen in den entsprechenden Tabellen bearbeitet werden. Man spricht in diesem Zusammenhang von **Änderungs-, Einfüge- und Lösch-Anomalien**. Die Anomalien können bevorzugt auftreten, wenn bestimmte gleiche Informationen an verschiedenen Stellen auftreten, obwohl es ausreichen würde, sie nur an einer Stelle festzuhalten. Man spricht dann von **Redundanz** im Datenbestand.

Zur Vermeidung von Anomalien und Redundanzen in Datenbanken unterzieht man den Datenbestand einem **Normalisierungsprozess** über sogenannte **Normalformen**. Eigentlich ist ein solcher Prozess bei einem sauberen Entwurf einer Datenbank überflüssig. In der Praxis hat sich die Normalisierung aber neben dem Entwurf einer Datenbank nach dem ERM als zusätzliche Möglichkeit zur Beseitigung oder Minimierung von Redundanzen etabliert.

### 1.2 Die 1. Normalform

Eine Tabelle befindet sich in der **1. Normalform** (1. NF), wenn ihre Attribute nur einfache (atomare) Attributswerte aufweisen und keine Wiederholfelder vorkommen.

Rezept: Auslagern der nicht-atomaren Attribute in verschiedene Zeilen oder Einfügen von zusätzlichen Spalten oder zusätzliche neue Tabelle.

Als Beispiel soll die Tabelle *personen* eines Betriebes mit verschiedenen Abteilungen und Projekten dienen.

*personen*

<u>PNr</u>	PName	AbtNr	AbtName	ProjektNr	ProjektName	ProjektZeit
101	Hans	1	Physik	11, 12	A, B	60, 40
102	Rolf	2	Chemie	13	C	100
103	Urs	2	Chemie	11, 12, 13	A, B, C	20, 50, 30
104	Paul	1	Physik	11, 13	A, C	80, 20

Die Attribute ProjektNr, ProjektName und ProjektZeit sind keine einfachen Attribute, weil es Mehrfacheinträge gibt. Die Mehrfacheinträge können hier auf mehrere Tupel aufgeteilt werden.

*PersonenUndProjekte*

<u>PNr</u>	PName	AbtNr	AbtName	<u>ProjektNr</u>	ProjektName	ProjektZeit
101	Hans	1	Physik	11	A	60
101	Hans	1	Physik	12	B	40
102	Rolf	2	Chemie	13	C	100
103	Urs	2	Chemie	11	A	20
103	Urs	2	Chemie	12	B	50
103	Urs	2	Chemie	13	C	30
104	Paul	1	Physik	11	A	80
104	Paul	1	Physik	13	C	20

Diese Tabelle mit dem neuen Namen *PersonenUndProjekte* ist in der 1. Normalform. Die Attributskombination PNr und ProjektNr ist Primärschlüssel der Tabelle.

Man sieht nun sofort, dass die Tabelle Redundanzen enthält. Der Personennamen ist mit der Personennummer gekoppelt und müsste nicht für jedes Projekt wiederholt werden. Wenn der Name geändert werden müsste, dann müsste dies an mehreren Stellen geschehen, um eine Änderungsanomalie zu verhindern. In der Tabelle werden offensichtlich verschiedene Sachverhalte (Personalien, Abteilungszugehörigkeit und Projektaspekte) abgebildet, die voneinander unabhängig sind. Diese verschiedenen Sachverhalte sollten auch mit verschiedenen Tabellen dargestellt werden.

### 1.3 Die 2. Normalform

Eine Tabelle befindet sich in der **2. Normalform** (2. NF), wenn sie in der 1. Normalform ist und jedes Nichtschlüsselattribut voll funktional abhängig vom Primärschlüssel ist.

Alternativformulierung: Eine Tabelle in der 1. NF ist **nicht** in der 2. NF, wenn Attribute von einem Teil des Schlüssels eindeutig identifiziert werden.

Hinweis: Wenn der Primärschlüssel nicht zusammengesetzt ist, sondern nur aus einem Attribut besteht, dann ist die Tabelle automatisch in der 2. NF.

Rezept: Auslagern von Teilschlüsseln und zugehörigen Informationen in eigene Tabellen bzw. separate Entitätstypen mit eigenem Schlüssel finden.

Die oben dargestellte Tabelle befindet sich nicht in der 2. NF, weil z.B. das Attribut PName nicht voll funktional vom Primärschlüssel (Kombination aus PNr und ProjektNr) abhängig

ist, sondern schon durch den Teilschlüssel PNr eindeutig identifiziert werden kann. Für die 2. NF werden die Tabellen aufgespalten:

*personen*

<u>PNr</u>	PName	AbtNr	AbtName
101	Hans	1	Physik
102	Rolf	2	Chemie
103	Urs	2	Chemie
104	Paul	1	Physik

*projekte*

<u>ProjektNr</u>	ProjektName
11	A
12	B
13	C

*personenprojekte*

<u>PNr</u>	<u>ProjektNr</u>	ProjektZeit
101	11	60
101	12	40
102	13	100
103	11	20
103	12	50
103	13	30
104	11	80
104	13	20

Sind nun alle Redundanzen eliminiert? Die Tabellen *projekte* und *personenprojekte* enthalten offensichtlich keine Redundanzen mehr. Aber in der Tabelle *personen* gibt es noch Redundanz. Für jede Person ist der Abteilungsname gespeichert, obwohl dieser Name mit der Abteilungsnummer gekoppelt ist. Der Abteilungsname ist also *transitiv* (d.h. „nur über Umwege“) vom Primärschlüssel abhängig.

## 1.4 Die 3. Normalform

Eine Tabelle befindet sich in der **3. Normalform** (3. NF), wenn sie in der 2. Normalform ist und jedes Nichtschlüsselattribut nicht transitiv abhängig vom Primärschlüssel ist.

Alternativformulierung: Eine Tabelle der 2. NF ist **nicht** in der 3. NF, wenn Attribute von anderen Nichtschlüsselattributen identifiziert werden.

Rezept: Auslagern der „transitiv abhängigen“ Attribute in eigene Tabellen.

*personen*

<u>PNr</u>	PName	AbtNr
101	Hans	1
102	Rolf	2
103	Urs	2
104	Paul	1

*abteilungen*

<u>AbtNr</u>	AbtName
1	Physik
2	Chemie

*projekte*

ProjektNr	ProjektName
11	A
12	B
13	C

*personenprojekte*

PNr	ProjektNr	ProjektZeit
101	11	60
101	12	40
102	13	100
103	11	20
103	12	50
103	13	30
104	11	80
104	13	20

## 1.5 Übungsbeispiel

Die Daten eines Fahrradladens sind in der unten angegebenen Tabelle gespeichert. Bringe die Tabelle nacheinander in die 1. bis 3. Normalform.

KNr	Name	Adresse	RahmenNr	Marke	Vers	VersOrt	ReparaturDatum	Diagnose
100	Meyer	Ulm, Steinstr. 5	123, 432	Diamant, Winora	Allianz, Signal	Köln, Mainz	12.12.2011, 14.12.2011	Platten, Schleicher
101	Müller	Ulm, Urstr. 3	690	Kettler	Allianz	Köln		