

Was ist Normalisierung?

- Vermeidung redundanter Daten im Relationenmodell mittels Normalisierung
- Normalisierung führt zu Vorgaben, die von Tabellen erfüllt werden müssen
- Normalformen
- Normalisierung ist weitere Technik zur Erstellung relationaler Datenbanken

Wozu Normalisierung?

Tabellen (Relationen) sollten letztlich so geplant werden, dass

- logische Widersprüche (Inkonsistenzen, Anomalien) in der Datenbasis und
- Datenredundanz (Mehrfachspeicherung gleicher Daten) vermieden sowie
- eine höchstmögliche Flexibilität und schneller Zugriff gewährleistet werden.

Normalformenlehre

Nutzung und Modellierung von Datenbanken



PL RLP, E-Tutoren



1. Normalform

Beispiel 1

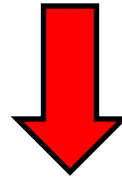
Mitarbeiter			
<u>MNR</u>	Name	Vorname	Adresse



1. Normalform

Beispiel 1

Mitarbeiter			
<u>MNR</u>	Name	Vorname	Adresse



Mitarbeiter					
<u>MNR</u>	Name	Vorname	Straße	PLZ	Ort

1. Normalform:

Die erste Normalform fordert, dass alle Attributwerte *atomar* sind, d.h. nicht weiter sinnvoll untergliedert werden können.





1. Normalform

Beispiel 2

Mitarbeiter			
<u>MNR</u>	Name	Ort	<i>Vorgesetzte</i>
122	Maier	Berlin	145,287
134	Müller	Bonn	145, 299, 302

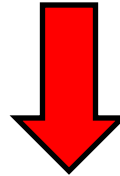




1. Normalform

Beispiel 2

Mitarbeiter			
<u>MNR</u>	Name	Ort	Vorgesetzte
122	Maier	Berlin	145,287
134	Müller	Bonn	145, 299, 302



MitarbeiterVorgesetzte			
<u>MNR</u>	<u>Vorgesetzte</u>	Name	GebDat
122	145	Maier	03.12.1971
122	287	Maier	03.12.1971
134	145	Müller	23.05.1982
134	299	Müller	23.05.1982
134	302	Müller	23.05.1982



1. Normalform

Beispiel 2

Mitarbeiter			
<u>MNR</u>	Name	Ort	Vorgesetzte
122	Maier	Berlin	145,287
	Müller	Bonn	145, 299, 302

Vorgriff:

Die Tabelle *MitarbeiterVorgesetzte* erfüllt nun zwar die erste Normalform, verstößt jedoch gegen die 2. Normalform (siehe folgende Folien) und kann somit keinesfalls so realisiert werden.

Mitarbeiter			
<u>MNR</u>	<u>Vorgesetzte</u>	Name	GEB
122	145	Maier	03.12.1971
122	287	Maier	03.12.1971
134	145	Müller	23.05.1982
134	299	Müller	23.05.1982
134	302	Müller	23.05.1982



2. Normalform

ProjektMitarbeiter			
<u>MNR</u>	<u>ProjNr</u>	Name	Ort
319	27	Schmidt	Berlin
319	45	Schmidt	Berlin
102	19	Schulz	Bonn
102	81	Schulz	Bonn
102	27	Schulz	Bonn

ProjektMitarbeiter	
<u>MNR</u>	<u>ProjNr</u>
319	27
319	45
102	19
102	81
102	27

Mitarbeiter		
<u>MNR</u>	Name	Ort
319	Schmidt	Berlin
102	Schulz	Bonn



2. Normalform

2. Normalform:

Die zweite Normalform fordert, dass in einer Tabelle mit zusammengesetzten Primärschlüsseln, alle Nichtschlüsselattribute voll funktional abhängig vom Primärschlüssel sind.

ProjektMitarbeiter			
ProjNr	MNR	Name	Ort
27	Schulz	Bonn	
	Schulz	Bonn	

Erläuterung:

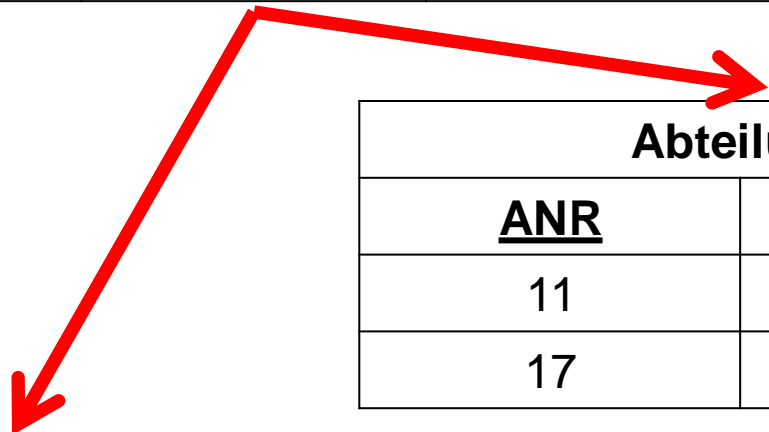
In der Tabelle *ProjektMitarbeiter* hängen sowohl das Attribut *Name* als auch *Ort* ausschließlich vom Schlüssel *MNR*, jedoch nicht vom Schlüssel *ProjNr* ab. (Sie hängen also nicht **voll funktional** vom zusammengesetzten Primärschlüssel ab!) Aus diesem Grund verstößt diese Tabelle gegen die 2. Normalform.

102	81
102	27



3. Normalform

Mitarbeiter				
<u>MNR</u>	Name	Ort	Abteilung	Bezeichnung
102	Schulz	Bonn	11	Personal
122	Maier	Hannover	17	Einkauf
134	Müller	Bremen	11	Personal



Abteilung	
<u>ANR</u>	Bezeichnung
11	Personal
17	Einkauf

Mitarbeiter			
<u>MNR</u>	Name	Ort	<i>Abteilung</i>
102	Schulz	Bonn	11
122	Maier	Hannover	17
134	Müller	Bremen	11





3. Normalform

3. Normalform:

Die dritte Normalform fordert, dass in einer Tabelle mit einem „einfachen“ Primärschlüssel, keine Nichtschlüsselattribute existieren, die von einem anderen Attribut als dem Primärschlüssel abhängen. Es dürfen demnach keine **transitiven Abhängigkeiten** bestehen.

Erläuterung:

In der Tabelle *Mitarbeiter* (oben) hängt das Nichtschlüsselattribut *Bezeichnung* vom Attribut *Abteilung* ab, welches jedoch nicht den Primärschlüssel darstellt. Zwischen dem Attribut *Bezeichnung* und dem Primärschlüssel *MNR* besteht also eine **transitive** (d.h. eine über das Attribut *Abteilung* mittelbare) **Abhängigkeit**. Demnach verstößt die obige Tabelle gegen die dritte Normalform.

Abteilung	
<u>ANR</u>	Bezeichnung

122	Mare		
134	Müller	Bremen	11



Bemerkungen

1. Streng genommen ist die n-te Normalform erst dann erfüllt, wenn zum einen die entsprechende Bedingung erfüllt ist und zum anderen auch die (n-1)-te Normalform erfüllt ist.
Beispiel: Die 3. Normalform ist erfüllt, wenn in einer Tabelle mit einem „einfachen“ Primärschlüssel keine Nichtschlüsselattribute existieren, die von einem anderen Attribut als dem Primärschlüssel abhängen und die 2. Normalform (und damit auch die erste Normalform) erfüllt ist.
2. Die Normalformen verhindern Redundanzen und damit auch Dateninkonsistenzen.
3. In Ausnahmefällen darf man auch gegen eine oder mehrere Normalformen verstoßen, sofern triftige Gründe vorliegen. Man spricht dann von *Denormalisierung*. Allerdings sollte man im Relationenmodell anmerken, warum man (absichtlich) gegen eine Normalform verstößt.
4. Die Normalisierung wird zwischen den Entwurfsschritten *Ableiten des Relationenmodells* und *Implementierung* durchgeführt.

