



Normalisierung von Datenbanken



Unter **Normalisierung** eines relationalen Datenbankmodells versteht man die **Aufteilung** von **Attributen** in mehrere **Relationen** (Tabellen) mithilfe von Normalisierungsregeln und deren Normalformen, sodass eine Form entsteht, die keine vermeidbaren Redundanzen mehr enthält.

Ziel der Normalisierung ist somit eine **redundanzfreie Datenspeicherung** zu erstellen. Redundanzfrei bedeutet, dass Daten entfernt werden können, ohne dass es zu Informationsverlusten kommt. Weiterhin soll die Normalisierung Anomalien entfernen.

Im Normalisierungsprozess gibt es **fünf Normalformen**. In der **Datenbankentwicklung** ist die **dritte Normalform** oft **ausreichend**, um die perfekte Balance aus Redundanz, Performance und Flexibilität für eine Datenbank zu gewährleisten. Natürlich gibt es auch Sonderfälle, z.B. im wissenschaftlichen Bereich, wo eine Datenbank bis zur 5. Normalform normalisiert werden kann bzw. muss.

Normalformen

Erste Normalform (1. NF)

- Eine Relation befindet sich in der **ersten Normalform (1. NF)**, wenn alle **Attributwerte atomar** sind.

Das bedeutet, dass **jede Information** innerhalb einer Tabelle eine **eigene Tabellenspalte** bekommt und zusammenhängende Informationen, wie zum Beispiel die Postleitzahl (PLZ) und der Ort, nicht in einer Tabellenspalte vorliegen dürfen. Informationen, die vorher unstrukturiert und unsortiert vorlagen, werden nun einheitlich und klar strukturiert.

Beispiel: Verkauf von Artikeln

Daten in der nullten Normalform (0. NF):

Alle Datenelemente der realen Welt sind in einer Tabelle zusammengefasst (unnormalisiert).

Rnr.	Datum	Name	Straße	Ort	Artikel	Anzahl	Preis
100	14.07.2021	Edi Bauer	Bergstraße 1	94032 Passau	Reifen	4	200 €

Nach Anwendung der ersten Normalform (1. NF):

Rnr.	Datum	Name	Vorname	Straße	HNr.	PLZ	Ort	Artikel	Anzahl	Preis	Währung
100	14.07.2021	Bauer	Edi	Bergstraße	1	94032	Passau	Reifen	4	200	Euro

Zweite Normalform (2. NF)

- Eine Relation befindet sich in der **zweiten Normalform (2. NF)**, wenn sie sich in 1.NF befindet und **jedes Attribut** vom **gesamten Primärschlüssel** (und nicht nur von Teilen des Primärschlüssels) **voll funktional abhängig** ist. Man spricht von voll funktionaler Abhängigkeit, wenn der Wert eines einzigen Attributs eindeutig die Werte der anderen Attribute bestimmt. Relationen in 1.NF sind automatisch in 2.NF, wenn ihr Primärschlüssel nicht zusammengesetzt ist.

Die **zweite Normalform** wird meistens schon **indirekt erreicht**, wenn der Datenbankentwickler mit der **Erstellung** eines **ER-Diagramms** beschäftigt ist. Die logische Aufspaltung von komplexen Sachverhalten zwingt den Datenbankentwickler Geschäftsprozesse in Relationen abzubilden.

Fortsetzung Beispiel**Nach Anwendung der zweiten Normalform (2. NF):**

<u>Rechnung</u>		
Rnr.	Datum	Knr.
100	14.07.2021	20

<u>Kunden</u>						
Knr.	Name	Vorname	Straße	Hnr.	PLZ	Ort
20	Bauer	Edi	Bergstraße	1	94032	Passau

<u>Rechnungsposition</u>			
Rpnr.	Rnr.	Artnr.	Anzahl
1	100	7	4

<u>Artikel</u>		
Artnr.	Artikel	Preis
7	Reifen	200

Dritte Normalform (3. NF)

- Ein Relationenschema befindet sich in der **dritten Normalform (3.NF)**, wenn es sich in 2.NF befindet und zudem **keine transitiven Abhängigkeiten** aufweist. Unter transitiver Abhängigkeit versteht man die **funktionale Abhängigkeit** eines Nicht-Schlüsselattributes einer Relation R von einem anderen Nicht-Schlüsselattribut in R.

Eine **Funktionale Abhängigkeit** zwischen Attribut Y und Attribut X liegt dann vor, wenn es zu **jedem X genau ein Y** gibt.

Eine **transitive Abhängigkeit** liegt dann vor, wenn Y von X funktional abhängig und Z von Y, so ist Z von X funktional abhängig.

Fortsetzung Beispiel

Kundeninformationen liegen in der zweiten Normalform (2. NF) vor:

<u>Kunden</u>						
Knr.	Name	Vorname	Straße	Hnr.	PLZ	Ort
20	Bauer	Edi	Bergstraße	1	94032	Passau

Nach Anwendung der dritten Normalform (3. NF):

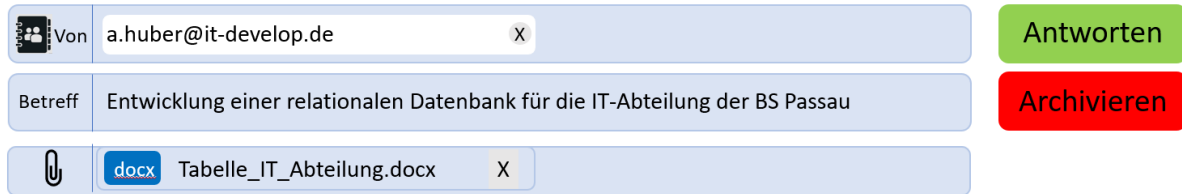
<u>Kunden</u>					
Knr.	Name	Vorname	Straße	Hnr.	PLZ
20	Bauer	Edi	Bergstraße	1	94032

<u>Postleitzahl</u>	
PLZ	Ort
94032	Passau

In der Tabelle Kunden sind die Attribute *Vorname*, *Straße* und *PLZ* abhängig vom Attribut *Name*, nicht vom Primärschlüssel *Knr.* Außerdem ist das Attribut *Ort* abhängig vom Attribut *PLZ*.

Die **transitiv abhängigen** Spalten werden in eine weitere **Untertabelle** ausgelagert, da sie nicht direkt vom Schlüsselkandidaten abhängen, sondern nur indirekt.

Anhang: E-Mail von Herrn Huber



Hallo IT'ler,

die IT-Abteilung der Berufsschule Passau möchte zukünftig eine relationale Datenbank für die Verwaltung der Abteilung der IT-Berufe einführen. Bevor wir jedoch mit der physischen Umsetzung der Datenbank beginnen können, müssen wir zunächst ein **relationales Datenmodell in dritter Normalform** entwerfen.

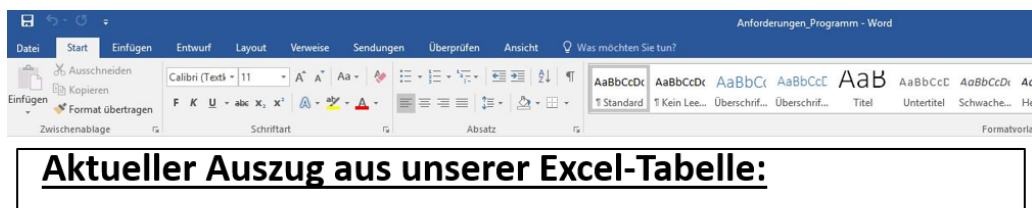
Wir möchten morgen dem Abteilungsleiter bereits einen Lösungsvorschlag präsentieren.

Bitte **entwickeln** Sie mir deshalb ein **relationales Datenmodell in dritter Normalform**!

Infos zu den benötigten **Daten** entnehmen Sie bitte der angefügten Word-Datei.

Beste Grüße
A. Huber

Anhang: Anforderungen für den Meilenstein-Termin 1



SchülerNr	Name	Klasse	Fach	Klassenlehrer
1	Jürgens Ina Bergstraße 17 94032 Passau	IT11a	VS (Neuhauser) ITS (Böhm) AP (Wagner) En (Waldvogel) <u>PuG</u> (Kaiser)	Wagner
2	Schmidt Tom Talstraße 12 94474 Vilshofen	IT12a	VS (Gerg) AP (Reckziegel) En (Waldvogel) <u>Rel</u> (Kaiser) ITAG (Wagner)	Reckziegel
3	Jäger Fritz Langgasse 3 94078 Freyung	IT10a	ITT (Wagner) AP (Schober) ITS (Neuhauser) <u>Sp</u> (Reiser) BWP (Weinmann)	Böhm
...



Arbeitsauftrag

- Erstellen Sie ein **relationales Datenmodell in dritter Normalform (3NF)**!

