

# Normalisierungsformen (NF1 - NF3)

## Einleitung

Wenn wir Datenbanken einrichten, wollen wir die Daten haben, die zu keinem Problem führen. Wenn wir eine Datenbank erstellen, deren Tabellen alle 3 Normalisierungsformen erfüllen, haben wir Tabellen, die uns die höchste Performance und Zuverlässigkeit bietet. Daher sollte man versuchen, die Datenbanktabellen nach diesen Regeln anzulegen.

Die Regeln müssen nach und nach erfüllt werden, das bedeutet es muss erst die **NF1**, dann die **NF2** und zum Schluss die **NF3** erfüllt sein. Also eine Tabelle, die sich in der **Normalisierungsform 2** befindet, hat automatisch die Anforderungen der **Normalisierungsform 1** erfüllt.

## Normalform 0 (0NF)

Eine Tabelle liegt in der "**Nullten Normalform**" vor, wenn die Daten sich einfach in der Tabelle befinden. Das bedeutet, die Daten sind nicht atomar. Das bedeutet wir haben **Redundanzen** in der Datenbank ohne **signifikanten Informationsgewinn**. Dadurch ist unsere Tabelle durch sogenannte Anomalien gefährdet.

In dem ersten Beispiel haben wir eine Tabelle mit Kundendaten mit den Bestellungen, die diese getätigt haben. Es gibt in der Tabelle Redundanzen, z.B. **DHL => Deutsche Post DHL Group** als auch im Merkmal / Attribut **Adresse** bei der Postleitzahl mit der Ortsangabe. Im Weiteren haben wir Probleme, wenn wir z.B. Kunden nach Nachnamen sortieren wollen. Dies ist laut unserer aktuellen Tabelle nicht möglich.

[Normalform0.png](#)

## Normalform 1 (1NF)

Die Tabelle in der **ersten Normalform** existiert, wenn folgende Bedingungen erfüllt sind:

1. Jedes Merkmal atomar ist
2. Nicht weiter unterteilbar ist
3. Listen aufgelöst sind

Es befinden sich in der Tabelle dennoch weiterhin Redundanzen, da diese noch nicht entfernt werden. In der 1. Normalisierungsform werden die Daten nur aufgeteilt, dass wir eine reine Tabelle erhalten, in dem wir die entsprechenden Daten in einzelnen Merkmalen aufgeteilt haben.

Ab wann ein Wert **atomar** ist, hängt vom Nutzungskontext ab. Wenn man gewisse Teilungen nicht benötigt, ist dies nicht unbedingt nötig. In der Praxis empfiehlt sich trotzdem alles möglichst klein aufzuteilen.

[Normalform1.png](#)

## Normalform 2 (2NF)

In der zweiten Normalform muss die Tabelle in der ersten Normalform vorliegen, und alle Nichtschlüsselmerkmale voll funktional vom Primärschlüssel abhängen. Unterm Strich bedeutet dies, dass jedes Nichtschlüsselmerkmal aus dem Primärschlüssel ableitbar ist. Merkmale, die von einem Teilschlüssel abhängig sind, müssen in eine eigene Tabelle geschrieben werden. Hier teilen wir also die Tabellen in einzelne Tabellen auf.

Um jetzt eine Tabelle in die zweite Normalform zu übernehmen, müssen wir folgende Schritte durchführen:

1. Alle Nichtschlüsselmerkmale, die von einem Teilschlüssel funktional abhängig sind, bestimmen.
2. Aus den Teilschlüsseln mit allen funktional abhängigen Nichtschlüsselmerkmalen eigene Tabellen bilden.
3. Im letzten Schritt entfernen wir alle nicht voll funktional abhängigen Nichtschlüsselmerkmale.

[Normalform2.png](#)

## Normalform 3 (3NF)

Im letzten Schritt entfernen wir die letzten Redundanzen. Laut der Definition ist eine Tabelle in der Normalform 3, wenn jedes Schlüsselmerkmal nicht transitiv vom Primärschlüssel abhängig ist.

Im Weiteren bedeutet **Transitiv**, dass ein Merkmal einen Umweg nutzen kann, um funktional abhängig von einem anderen Merkmal abhängig ist. Unterm Strich sollen alle Merkmale nur von einem Primärschlüssel abhängig sein.

Um eine Tabelle in die dritte Normalform zu übernehmen, verwenden wir folgende Schritte:

1. Alle Nichtschlüsselmerkmale, die transitiv vom Schlüssel abhängen, bestimmen.
2. Im nächsten Schritt sollen aus diesen transitiv abhängigen Nichtschlüsselmerkmalen und den Nichtschlüsselmerkmalen, von denen diese funktional abhängig sind, eigene Tabellen bilden.
3. Im letzten Schritt entfernen wir alle transitiv abhängigen Nichtschlüsselmerkmale aus der Ursprungstabelle. [Normalform3.png](#)

Updated 14 May 2024 07:26:03 by Phillip U.