

(3NF) Dritte Normalform

In der 3NF befindet sich die DB bereits in der letzten Form, wenn sie sich in der 2NF befindet und jede nicht Schlüsselspalte voneinander unabhängig ist. (iaw) Alle Nicht-Schlüsselspalten müssen voneinander unabhängig sein. Wir identifizieren alle Spalten in der Tabelle, die voneinander abhängig sind, und unterteilen diese Spalten in ihre eigenen, separaten Tabellen.

- REGELN .
- Die Tabelle sollte in 2NF vorliegen.
 - Es sollte keine „TRANSITIVE ABHÄNGIGKEITEN“ für nicht primäre Attribute geben.

1) Was ist eine funktionale Abhängigkeit?

2) Was ist eine transitive Abhängigkeit?

1) Was ist eine funktionale Abhängigkeit?

Wenn eine Beziehung zwischen dem Primärschlüssel und einem Nicht-Schlüssel-Attribut in einer Tabelle besteht, wird dies als funktionale Abhängigkeit bezeichnet.

$X \rightarrow Y$: Hier wird X als Determinante und Y als ..Abhängig.. bezeichnet.

Beispiel: Nehmen wir eine Angestelltentabelle
„Employee“

Employee		
EmployeeId	EmployeeName	EmployeeAddress
101	John	California
102	Mary	Oregon

Hier identifiziert „EmployeeId“ den „EmployeeName“ eindeutig, denn wenn wir die „EmployeeId“ kennen, können wir den damit verbundenen „EmployeeName“ leicht bestimmen. In diesem Fall kann die FUNKTIONALE ABHÄNGIGKEIT als

$\text{EMPLOYEEID} \rightarrow \text{EMPLOYEENAME}$

geschrieben werden, was bedeutet, dass „EmployeeName“ funktional von „EmployeeId“ abhängig ist.

2) Was ist eine transitive Abhängigkeit?

Wenn eine indirekte funktionale Abhängigkeit den Attributen besteht, nennt man sie TRANSITIVE ABHÄNGIGKEIT.

$X \rightarrow Y$ und $Y \rightarrow Z$: Wenn $X \rightarrow Y$ und $Y \rightarrow Z$ als funktionale Abhängigkeiten bestehen, dann wird

$X \rightarrow Z$ als transitive Abhangigkeit bezeichnet.

Um die 3NF zu erreichen, mussen wir die transitive Abhangigkeit eliminieren.

Beispiel.

MovieListing			
MovieID	ListingID	ListingType	DVDPrice
M08	L09	Crime	18,0
M03	L05	Drama	25,0
M05	L09	Crime	18,0

In dieser Tabelle sind $\text{MovieID} \rightarrow \text{ListingID}$ sowie $\text{ListingID} \rightarrow \text{ListingType}$ Attribute, die eine transitive Abhangigkeit aufweisen. Daher hat $\text{MovieID} \rightarrow \text{ListingType}$ eine transitive Abhangigkeit.

Wie entfernt man die transitive Abhangigkeit?

Um transitive Abhangigkeiten zu entfernen, konnen wir die Tabelle aufteilen. Im obigen Beispiel, konnen wir die Spalte ListingType aus der Tabelle entfernen und eine separate

Tabelle dafür wie folgt erstellen:

MovieTable

MovieId	ListingId	DVDPrice
---------	-----------	----------

M08	L09	18,0	L09	Crime
M03	L05	25,0	L05	Drama
M05	L09	18,0	L09	Crime

Listing Table

ListingId	listingType
-----------	-------------

Piese zwei Tabellen sind nicht mehr transitiv abhängig.

Beispiel 3NF Normierung:

client(old)

clientId	Address
1	City, State
2	City, State
3	City, State

↓ 3NF

clients(new)

ClientId	CityId
1	1
2	2
3	3

city (new)

CityId	StateId	CityName
1	1	City 1
2	2	City 2
3	2	City 1
4	1	City 1
5	1	City 2

states(new)

StateId	StateName
1	State 1
2	State 2

In der Beispieldatabelle „clients(old)“, die Adressspalte ist voneinander abhängig. Die Adresse hängt von der Stadt ab, und die Stadt hängt von dem State (Bundesland) ab u.s.w. Daher können wir die Adressspalte in eine separate state oder Staattabellle aufteilen.

Schritte: 1) Erstellen Sie die Primärschlüssel „cityid“ & „stateid“ für die Tabellen city(new) & states(new).

2) Verknüpfen Sie die Primärschlüssel mit den entsprechenden Tabellen über Fremdschlüssel.

Durch diese Änderungen wurde die Tabelle in der 3NF gebracht.

BOYCE COOD NORMALFORM (BCNF)

Es handelt sich um eine Erweiterung der 3NF und wird als 3.5NF bezeichnet.

Regeln. . Die Tabelle sollte in 3NF sein.

- Bei Abhängigkeiten wie $A \rightarrow B$ sollte A ein „SUPERSCHLÜSSEL“ sein, was bedeutet, dass A kein Nicht-Primär-

Attribut sein kann, wenn β ein Primär-Attribut ist.

Beispiel:

CollegeEnrolment

StudentId	Subject	Professor
101	Java	P. Java
101	C++	P. Cpp
102	Java	P. Java 2
103	C#	P. Chash
104	Java	P. Java

In der Tabelle „CollegeEnrolment“ kann sich ein Student für mehrere Fächer einschreiben. Für jedes Fach wird dem Studenten ein Professor zugewiesen. Es kann mehrere Profs geben, die ein Fach unterrichten. Hier bilden „Student Id“ und „Fach“ den Primärschlüssel, denn mit Hilfe von denen können wir alle Spalten der Tabelle finden.

Außerdem hängt das Attribut Fach vom Namen des Professors ab, da ein Prof. nur ein Fach unterrichtet, ein Fach aber zwei verschiedene Professoren haben kann.

Aus dem Grund entspricht die Tabelle nicht 3CNF, da hier das Fach ein primäres Attribut und der Prof. ein nicht primäres Attribut ist.

Student Table

Student

StudentID	ProfessorID
101	1
101	2
102	3
103	4
104	5

Professor Table

Professor

ProfessorID	Professor	Subject
1	P.Java	Java
2	P.Cpp	C++
3	P.Java2	Java
4	P.Chash	C#
5	P.Java	Java

Jetzt erfüllt die Beziehung BCNF.

