

Aufgabe 1

Gegeben ist die folgende Tabelle mit dem Schlüssel {**PersonNr**, **VereinsNr**}:

<u>PersonNr</u>	Name	Jahrgang	<u>VereinsNr</u>	Verein
26120	Fichte	77	5001	FC Bayern
27550	Schopenhauer	76	5001	FC Bayern
27550	Schopenhauer	76	4052	SV Werder
28106	Carnap	89	4052	SV Werder
28106	Carnap	89	5041	HSV
28106	Carnap	89	5053	VFB
28106	Carnap	89	5216	SCF
28106	Carnap	89	5259	KSC
...	

Weiterhin gelten die folgenden funktionalen Abhängigkeiten:

- x **PersonNr** → **Name**, **Jahrgang**
- x **VereinsNr** → **Verein**

Ist die Tabelle in zweiter Normalform? Falls nein, soll sie in die zweite Normalform überführt werden.

Ausführliche Lösung:

Die Definition der zweiten Normalform lautet:

Eine Tabelle befindet sich in der zweiten Normalform, wenn sie

- x in der ersten Normalform ist und
- x kein Attribut funktional abhängig von einem Teilschlüssel ist.

Da jedes Datenfeld der oben stehenden Tabelle atomar ist, ist die Tabelle in der ersten Normalform.

Um überhaupt prüfen zu können, ob ein Attribut (eine Spalte) funktional abhängig von einem Teilschlüssel ist, müssen wir natürlich zunächst alle **Teilschlüssel** bestimmen.

Da der einzige Schlüssel der Tabelle {**PersonNr**, **VereinsNr**} ein **zusammengesetzter Schlüssel aus mehreren Attributen** ist, sind die einzelnen Attribute **PersonNr** und **VereinsNr** für sich genommen die jeweiligen **Teilschlüssel** der Tabelle.

In der Tabelle gilt zu dem die funktionalen Abhängigkeit:

PersonNr → Name, Jahrgang.

Offensichtlich sind also die Attribute **Name** und **Jahrgang** vom Teilschlüssel **PersonNr** **funktional abhängig**. Somit ist die Tabelle **nicht in der zweiten Normalform**, da die zweite Bedingung verletzt ist!

Aber auch die folgende, offensichtliche funktionale Abhängigkeit

VereinsNr → Verein

verletzt die zweite Normalform, da **Verein** vom **Teilschlüssel VereinsNr** funktional abhängig ist. Jetzt geht es also darum, die Tabellen aufzuspalten. Dazu helfen genau die funktionale Abhängigkeiten, die die zweite Normalform verletzt hatten:

PersonNr → Name, Jahrgang
VereinsNr → Verein

Die Spaltung geschieht zunächst dadurch, dass **pro funktionale Abhängigkeit, die die zweite Normalform verletzt, eine separate Tabelle entsteht**, die genau aus den Attributen (Spalten) dieser funktionalen Abhängigkeit besteht.

Dieser Vorgang führt also zu den folgenden Tabellen:

Tabelle Person:

PersonNr	Name	Jahrgang
26120	Fichte	77
27550	Schopenhauer	76
28106	Carnap	89
...

Tabelle Verein:

VereinsNr	Verein
5001	FC Bayern
4052	SV Werder
5041	HSV
5053	VFB
5216	SCF
5259	KSC
...	...

Aus diesen Tabellen geht aber leider **nicht** mehr hervor, welche Person Mitglied in welchem Verein ist. Um diese Beziehung herzustellen, besitzen wir die folgenden zwei Optionen:

1. Wir nehmen den Primärschlüssel der Tabelle **Person** als Fremdschlüssel in die Tabelle **Verein** hinzu oder
2. wir nehmen den Primärschlüssel der Tabelle **Verein** als Fremdschlüssel in die Tabelle **Person** auf.

Wir entscheiden uns zunächst willkürlich für die erste Variante. Es entstehen anschließend die folgenden Tabellen:

Tabelle **Person**:

<u>PersonNr</u>	Name	Jahrgang
26120	Fichte	77
27550	Schopenhauer	76
28106	Carnap	89
...

Tabelle **Verein**:

<u>PersonNr</u>	<u>VereinsNr</u>	Verein
26120	5001	FC Bayern
27550	5001	FC Bayern
27550	4052	SV Werder
28106	4052	SV Werder
28106	5041	HSV
28106	5053	VFB
28106	5216	SCF
28106	5259	KSC
...

Die Beziehung ist nun hergestellt und die Tabellen sind aufgespalten, **aber sind die neu entstandenen Tabellen auch in der zweiten Normalform?** Die Tabelle **Person** ist nun zweifelsohne in der zweiten Normalform, weil ihr Schlüssel nur aus einem einzigen Attribut besteht und somit kein Teilschlüssel mehr existiert. Betrachten wir aber die Tabelle **Verein**, so stellen wir fest, dass die funktionale Abhängigkeit **VereinsNr → Verein** weiterhin existiert. Da aber **VereinsNr** nur ein **Teilschlüssel** dieser Tabelle ist, **verletzt** die Tabelle **Verein** weiterhin die zweite Normalform! Wir spalten die Tabelle **Verein** analog zu oben an Hand der die 2NF-verletzenden funktionalen Abhängigkeit **VereinsNr → Verein** wie folgt auf:

Tabelle **Person**:

<u>PersonNr</u>	Name	Jahrgang
26120	Fichte	77
27550	Schopenhauer	76
28106	Carnap	89
...

Tabelle **Verein**:

<u>VereinsNr</u>	Verein
5001	FC Bayern
4052	SV Werder
5041	HSV
5053	VFB
5216	SCF
5259	KSC
...	...

Tabelle **Mitglied**:

<u>PersonNr</u>	<u>VereinsNr</u>
26120	5001
27550	5001
27550	4052
28106	4052
28106	5041
28106	5053
..	..

Aufgabe 2

Überprüfen Sie, ob die folgende Tabelle in 2NF ist und überführen Sie diese in 2NF, falls nötig. Identifizieren Sie vorher sinnvolle funktionale Abhängigkeiten.

<u>KID</u>	Kunde	Str	Hnr	PLZ	Ort	<u>PID</u>	Produkte	Datum
1	Winkler	Hauptstr	23	77625	Offenburg	P1	Website	01.03.07
2	Mayer	Gartenstr	15	77933	Lahr	P2	VisitenK	10.05.07
2	Mayer	Gartenstr	15	77933	Lahr	P3	Briefe	10.05.08
2	Mayer	Gartenstr	15	77933	Lahr	P4	Logos	10.05.07
3	Schulz	Holzweg	3	77960	Seelbach	P5	Flyer	20.06.07
4	Schmitt	Hauptstr	5	77933	Lahr	P1	Website	01.09.07
4	Schmitt	Hauptstr	5	77933	Lahr	P5	Flyer	01.09.07
1	Winkler	Hauptstr	23	77625	Offenburg	P2	VisitenK	01.10.07
3	Schulz	Holzweg	3	77960	Seelbach	P3	Briefe	01.10.07

Lösung:

Die oben stehende Tabelle enthält Informationen über Kunden und ihre erworbenen Produkte. Der Schlüssel der Tabelle ist {**KID**, **PID**}, weil diese Kombination von Attributen die Zeilen der Tabelle eindeutig identifiziert.

Das Attribut **KID** identifiziert eindeutig einen Kunden. Außerdem identifiziert **PID** eindeutig ein Produkt. Es gelten somit die folgende funktionale Abhängigkeit

KID → **Kunde**, **Str**, **Hnr**, **PLZ**, **Ort**
PID → **Produkte**

Weil aber **KID** und **PID** nur **Teilschlüssel** sind, **verletzen sie zweite Normalform**. Wir zerlegen die Tabelle zunächst nur anhand **KID** → **Kunde**, **Str**, **Hnr**, **PLZ**, **Ort** wie folgt:

Kunden (**KID**, **Kunde**, **Str**, **PLZ**, **Ort**)

Es bleiben dann nur die folgenden Spalten in der ursprünglichen Tabelle - nennen wir es **Resttabelle** - übrig:

Resttabelle (**Datum**, **PID**, **Produkte**)

Um die Beziehung zwischen Produkte und Kunden herzustellen könnten wir theoretisch die Spalte **KID** aus der Tabelle **Kunden** in die **Resttabelle** – nennen wir es doch lieber **Bestellung** - als Fremdschlüssel aufnehmen:

Bestellung (**KID**, **Datum**, **PID**, **Produkte**)

Aber besteht der Primärschlüssel der Tabelle Bestellung wirklich nur aus dem Attribut **PID**? Um diese Frage am sichersten zu beantworten füllen wir doch die Tabelle mit einigen Datensätzen:

Tabelle **Bestellung**:

KID	PID	Produkte	Datum
1	P1	Website	01.03.07
2	P2	VisitenK	10.05.07
2	P3	Briefe	10.05.08
2	P4	Logos	10.05.07
3	P5	Flyer	20.06.07
4	P1	Website	01.09.07
4	P5	Flyer	01.09.07
1	P2	VisitenK	01.10.07
3	P3	Briefe	01.10.07

Offensichtlich kommt der **P1** und auch andere **PID**-Werte mehrfach vor. Es ist also klar, dass die Spalte **PID** allein kein Primärschlüssel sein darf.

Wenn man einen strengen Blick auf diese Tabelle wirft :-)) dann erkennt man, dass sie natürlich dieselbe ist wie die ursprüngliche Tabelle aus dieser Aufgabe, nur eben dass einige Infos zum Kunden wie Strasse oder PLZ nicht mehr aufgelistet sind. Ein Kunde ist hier aber weiterhin repräsentiert, nämlich durch seine Kundennummer **KID**.

Da in der ursprünglichen Tabelle {**KID**, **PID**} den (Primär-) Schlüssel bildeten müssen sie es also hier genauso tun!

D.h. Die Tabelle Bestellung hat also das folgende Format:

Bestellung (***KID***, **Datum**, ***PID***, **Produkte**)

Die Spalte ***KID*** ist deshalb kursiv markiert, weil sie zusätzlich ein Fremdschlüssel ist...

Sind wir jetzt fertig? Fast...

Gilt in der Tabelle Bestellung nicht immer noch **PID** → **Produkte** ?

Na klar gilt die und die verletzt wieder die zweite Normalform (warum?). Die Aufteilung ergibt dann letztendlich drei Tabellen. Kursive Spalten sind zusätzlich Fremdschlüssel.

Kunden (***KID***, **Kunde**, **Str**, **PLZ**, **Ort**)

Produkte (***PID***, **Produkte**)

Bestellung (***KID***, ***PID***, **Datum**)

Der Primärschlüssel der Tabelle Bestellung setzt sich dann in dem Fall aus den zwei Fremdschlüsseln **KID** und **PID** zusammen.

Aufgabe 3

Überprüfen Sie, ob die folgende Tabelle in 2NF ist und überführen Sie diese in 2NF, falls nötig. Identifizieren Sie vorher sinnvolle funktionale Abhängigkeiten.

<u>ISBN</u>	Titel	Verlag	Ort	<u>Anr</u>	Autor
3-8266-0126-2	Datenbanktheorie	Thomsen	Bonn	1	Vossen
3-343-00892-3	Taschenbuch Inft.	FBV L.	Leipzig	2	Werner
3-540-62477-5	Einführung Wirt.-Infor.	Springer	Berlin	3	Stahlknecht
3-540-62477-5	Einführung Wirt.-Infor.	Springer	Berlin	2	Werner
3-612-28098-8	VBA-Progr. in Excel 97	Econ	Düsseldorf	5	Cuber
3-8252-0802-8	Wirtschaftsinformatik	UTB	Stuttgart	6	Hansen
3-8274-0042-2	Software Technik	Spektrum	Heidelberg	7	Balzert
3-441-31055-1	BWL mit Rewe	BV Eins	Troisdorf	8	Blank
3-441-31055-1	BWL mit Rewe	BV eins	Troisdorf	9	Hagel
3-441-31055-1	BWL mit Rewe	BV Eins	Troisdorf	1	Vossen
3-441-31055-1	BWL mit Rewe	BV Eins	Troisdorf	3	Stahlknecht
3-441-31055-1	BWL mit Rewe	BV Eins	Troisdorf	12	Meyer
3-8237-3517-9	Wirtschaftsrechnen	BV Eins	Troisdorf	13	Dax

Lösung:

Die Tabelle speichert Informationen über Bücher und ihre Autoren. Die ISBN-Nummer identifiziert eindeutig ein Buch und die Autor-Nummer (**Anr**) einen Autor. Es gelten somit die folgenden funktionalen Abhängigkeiten:

ISBN, Anr → Titel, Autor, Verlag, Ort,
ISBN → Titel, Verlag, Ort,
Anr → Autor,
Verlag → Ort.

Weil ein Buch von mehreren Autoren verfasst werden kann und ein Autor mehrere Bücher verfassen kann, identifiziert nur die Kombination von ISBN-Nummer und Autor-Nummer einen Datensatz eindeutig. Somit verletzen aber die funktionalen Abhängigkeiten **ISBN → Titel, Verlag, Ort** und **Anr → Autor** die zweite Normalform, **weil ISBN und auch Anr jeweils nur Teilschlüssel sind**. Wir zerlegen die Tabelle demnach wie folgt:

Bücher (ISBN, Titel, Verlag, Ort)

Autoren (Anr, Autor)

AutorenUndIhreBücher (Anr, ISBN)

Begründen Sie, warum die Zusatzrelation **AutorenUndIhreBuecher** hier notwendig ist!