2. Normalform (2NF)

Ein Relationstyp (Tabelle) befindet sich in der zweiten Normalform (2NF), wenn die erste Normalform erfüllt ist und für jeden Primärschlüssel eindeutige Attributwerte vorhanden sind.

Die 2. Normalform entfernt Redundanzen:

id	produkt	preis	nummer	anzahl	name	vorname	telefon	fax	strasse	wohnort	plz	firma	f_strasse	f_ort	f_plz
1	Schlagbohrer	199.95	1000-1	1	Blutgesicht	Hans	123-550	123-551	Kariesweg 1	Zahnstein	12345	Hilti	Bohrmaschinenstr. 58a	Hilterhausen	88887
2	Zement	5.95	1000-2	5	Quälstein	Bertha	456-777	456-778	Lochpfad 23	Schmerzstadt	23458	Hoch & Tief	Am Höhenzug 22	Tiefenbau	12345
3	Kneifzange	19.95	1000-3	3	Eisenfaust	Viktoria	789-250	789-251	Zahnwurzel 3	Dolomostadt	87454	Eisen-Karl	Karlstrasse 5	Eisenhafen	76542
4	Brecheisen	49.95	1000-4	7	Rostzange	Eberhart	234-100	234-101	Peinweg 5	Reissheim	74512	Eisen-Karl	Karlstrasse 5	Eisenhafen	76542
5	Hammer	19.95	1000-5	4	Frankenstein	Emilia	567-200	567-201	Am Dom 5	Köln	50670	Stahl AG	Blechweg 102b	Alustetten	89452
6	Zement	5.95	1000-2	9	Eisenfaust	Viktoria	789-250	789-251	Zahnwurzel 3	Dolomostadt	87454	Hoch & Tief	Am Höhenzug 22	Tiefenbau	12345
7	Brecheisen	49.95	1000-4	2	Blutgesicht	Hans	123-550	123-551	Kariesweg 1	Zahnstein	12345	Eisen-Karl	Karlstrasse 5	Eisenhafen	76542

In der Tabelle in der 1NF tauchen zahlreiche Attributwerte mehrfach auf, z.B. Die Namen der Doktoren oder Hersteller.

Grundsatz: Eine Entität - eine Tabelle

Wenn man den **Grundsatz** – den unser Sekretär mit seinem an die Tabellenkalkulation angelehnten Vorgehen bereits verletzt hat, den wir aber eigentlich kennen – befolgt: **Jede Entität bekommt ihre eigene Relation, die Attribute sind die Tabellenspalten**[1)] und die Attribute entsprechend der Miniwelt atomar wählt erhält man relativ direkt die 2NF.

Hier sieht man nochmal, dass die **Normalisierung beim Design der Datenbank stattfinden sollte**, nicht wie wir das hier machen, nachdem man bereits ein schlechtes Design implementiert hat.

(A1)

- Überführe die Inhalte der "Universaltabelle" in der 1NF in drei Tabellen: `doktoren`, `hersteller`, `produkte`. Verteile die Attribute auf die Tabellen der Entitäten. Gehe wie folgt vor:
  1. Lege die drei Tabellen mit den entsprechenden Attributen an, jedoch zunächst ohne Surrogatschlüssel (keine `id` Spalte!
  2. Überführe die Inhalte aus der Universaltabelle in die jeweilige Tabelle, indem du das Ergebnis einer Abfrage direkt an eine INSERT INTO Statement weitergibst: `INSERT INTO doktoren SELECT DISTINCT name, vorname, strasse, plz, wohnort, telefon, fax FROM `zahnarztbedarf``. Du musst auf die Reihenfolge der Attribute

bei der Abfrage achten, damit die Werte in die richtigen Felder eingefügt werden. Warum benötigt man das Schlüsselwort `DISTINCT` bei der Abfrage?

3. Wenn alle Werte überführt sind, fügst du jeder Tabelle noch eine Feld `id` als Surrogatschlüssel hinzu (Primary-Key, Auto-Increment).

- Zwei Tabellenspalten der Universaltabelle können den Entitätstabellen nicht sinnvoll zugeordet werden - bei einer geht keine wesentliche Information verloren, wenn man sie weglässt, bei der anderen schon. Welche Attribute sind das? Lasse beide Attribute vorerst einfach aus - wir kommen später darauf zurück.
- Fügen nun den Datensatz für Frau Bohrgut ein aus den Vorüberlegungen ein - du siehst, nun ist ganz klar, welche Informationen in welche Tabelle eingefügt werden müssen.

Lösungshinweise Tabellenstrukturen:

```
DESCRIBE doktoren
```

+ Optionen

Field	Type	Null	Key	Default	Extra
id	int(10)	NO	PRI	NULL	auto_increment
name	varchar(200)	NO		NULL	
vorname	varchar(200)	NO		NULL	
strasse	varchar(200)	NO		NULL	
plz	varchar(20)	NO		NULL	
wohnort	varchar(200)	NO		NULL	
telefon	varchar(20)	NO		NULL	
fax	varchar(20)	NO		NULL	

```
DESCRIBE hersteller
```

+ Optionen

Field	Type	Null	Key	Default	Extra
id	int(10)	NO	PRI	NULL	auto_increment
firma	varchar(200)	NO		NULL	
strasse	varchar(200)	NO		NULL	
ort	varchar(200)	NO		NULL	
plz	varchar(200)	NO		NULL	

```
DESCRIBE produkte
```

+ Optionen

Field	Type	Null	Key	Default	Extra
id	int(10)	NO	PRI	NULL	auto_increment
produkt	varchar(800)	NO		NULL	
preis	float	NO		NULL	
nummer	varchar(50)	NO		NULL	

SQL Befehle:

```
INSERT INTO doktoren
SELECT DISTINCT name, vorname, strasse, plz, wohnort, telefon, fax
FROM `zahnarztbedarf`

INSERT INTO hersteller
SELECT DISTINCT firma, f_strasse, f_ort, f_plz
FROM `zahnarztbedarf`

INSERT INTO produkte
SELECT DISTINCT produkt, preis, nummer
FROM `zahnarztbedarf`
```

Ergebnis:

SELECT * FROm doktoren

	id	name	vorname	strasse	plz	wohnort	telefon	fax
Bearbeiten Kopieren Löschen	1	Blutgesicht	Hans	Kariesweg 1	12345	Zahnstein	123-550	123-551
Bearbeiten Kopieren Löschen	2	Quälstein	Bertha	Lochpfad 23	23458	Schmerzstadt	456-777	456-778
Bearbeiten Kopieren Löschen	3	Eisenfaust	Viktoria	Zahnwurzel 3	87454	Dolomostadt	789-250	789-251
Bearbeiten Kopieren Löschen	4	Rostzange	Eberhart	Peinweg 5	74512	Reissheim	234-100	234-101
Bearbeiten Kopieren Löschen	5	Frankenstein	Emilia	Am Dom 5	50670	Köln	567-200	567-201

SELECT * FROM hersteller

	id	firma	strasse	ort	plz
Bearbeiten Kopieren Löschen	1	Hilti	Bohrmaschinenstr. 58a	Hilterhausen	88887
Bearbeiten Kopieren Löschen	2	Hoch & Tief	Am Höhenzug 22	Tiefenbau	12345
Bearbeiten Kopieren Löschen	3	Eisen-Karl	Karlstrasse 5	Eisenhafen	76542
Bearbeiten Kopieren Löschen	4	Stahl AG	Blechweg 102b	Alustetten	89452

SELECT * FROM `produkte`

	id	produkt	preis	nummer
Bearbeiten Kopieren Löschen	1	Schlagbohrer	199.95	1000-1
Bearbeiten Kopieren Löschen	2	Zement	5.95	1000-2
Bearbeiten Kopieren Löschen	3	Kneifzange	19.95	1000-3
Bearbeiten Kopieren Löschen	4	Brecheisen	49.95	1000-4
Bearbeiten Kopieren Löschen	5	Hammer	19.95	1000-5

(A2)

Was bei der Normalisierung nun dummerweise verloren gegangen ist, ist die Zuordnung der Bestellungen zu unseren Kunden, den Doktoren - deswegen hatte der Sekretär die Tabelle ja mal angelegt.

Um die Verknüpfung zwischen den Entitäten nun wieder herzustellen, wird eine weitere Tabelle angelegt, die lediglich die Verknüpfung zwischen den anderen Entitäten herstellt.

Erstelle eine Tabelle `bestellungen` mit den Feldern `id` (Primary-Key, Auto-Increment), `produkt_id`, `doktor_id`, `hersteller_id` und `anzahl`.

Nun schauen wir in unserer Universaltabelle nach den Bestellungen:

id	produkt	preis	nummer	anzahl	doktor	telefon_fax	adresse	hersteller
1	Schlagbohrer	199.95	1000-1	1	Blutgesicht, Hans	123-550,123-551	Kariesweg 1, 12345 Zahnstein	Hilti, Bohrmaschinenstr. 58a, 88887 Hilterhausen
2	Zement	5.95	1000-2	5	Quälstein, Bertha	456-777,456-778	Lochpfad 23, 23458 Schmerzstadt	Hoch & Tief, Am Höhenzug 22, 12345 Tiefenbau
3	Kneifzange	19.95	1000-3	3	Eisenfaust, Viktoria	789-250,789-251	Zahnwurzel 3, 87454 Dolomostadt	Eisen-Karl, Karlstrasse 5, 76542 Eisenhafen
4	Brecheisen	49.95	1000-4	7	Rostzange, Eberhart	234-100,234-101	Peinweg 5, 74512 Reissheim	Eisen-Karl, Karlstrasse 5, 76542 Eisenhafen
5	Hammer	19.95	1000-5	4	Frankenstein, Emilia	567-200,567-201	Am Dom 5, 50670 Köln	Stahl AG, Blechweg 102b,89452 Alustetten
6	Zement	5.95	1000-2	9	Eisenfaust, Viktoria	789-250,789-251	Zahnwurzel 3, 87454 Dolomostadt	Hoch & Tief, Am Höhenzug 22, 12345 Tiefenbau
7	Brecheisen	49.95	1000-4	2	Blutgesicht, Hans	123-550,123-551	Kariesweg 1, 12345 Zahnstein	Eisen-Karl, Karlstrasse 5, 76542 Eisenhafen

Die erste Zeile sagt uns, dass Hans Blutgesicht einen Schlagborer bestellt hat.

Um das in unserer bestellungen-Tabelle abzubilden, müssen wir folgendes eintragen:

```
| id | anzahl | produkt_id | doktor_id | hersteller_id |
| 1  | 1      | 1
```

1)

Erinnerung: Relation ist nur ein anderer Name für Tabelle...

From:
https://info-bw.de/ -

Permanent link:
https://info-bw.de/faecher:informatik:oberstufe:datenbanken:normalisierung:2_normalform:start?rev=1606330464

Last update: **25.11.2020 18:54**