Normalisierung

Unter Normalisierung versteht man die schrittweise Zerlegung von Relationen (in der Datenbank: Tabellen) nach vorgegebenen Regeln (Normalisierungsregeln).
Das Ergebnis des Normalisierungsprozesses sind einzelne Relationen, die bestimmten Anforderungen entsprechen und die miteinander verknüpft sein können.
Dieses Verfahren verliert in der Praxis an Bedeutung, da modernes Datenbankdesign über Situationsanalyse->-ERM->Relationen->DB durchgeführt wird. Das aus dem ERM abgeleitete Relationenmodell ist *bei ordentlicher Durchführung* in 3. Normalform. Um dies *ordentlich* durchführen zu können, sind die Kenntnisse der Normalisierung jedoch sehr hilfreich.

ERM: Wie wir sehen konnten, ist der 1. Schritt eines modernen Datendesign die Identifizierung von Entitäten, die einem gemeinsamen Entitätstypen angehören. Der 2. Schritt ist die Suche nach Attributen, welche die gleichartigen Entitäten beschreiben. Danach wird untersucht, welche Beziehungen zwischen den Entitätstypen bestehen (3. Schritt). Aus dem entworfenen ER-Modell werden dann anschließend Relationen abgeleitet.

In der Praxis wird mit der Normalisierung ein bestehendes Relationenmodell verifiziert, d.h. es wird geprüft, ob ein bestehendes Modell den Anforderungen einer Normalform (meist der 3. Normalform) entspricht.

Normalisierungsgründe

Die Normalisierung soll Datenredundanz aufzuspüren, die einen erhöhten Speicherplatz verursacht, das Auswerten der Daten verlängert und bei der Änderung von Daten zu Inkonsistenz führen kann. Das nachfolgende Beispiel soll verdeutlichen, wie eine nicht-normalisierte Tabelle schrittweise in eine Datenbank, die der 3. Normalform entspricht, überführt werden kann.

Beispiel einer Tabelle, die nicht normalisiert ist:
Ein Bauunternehmen pflegt folgende Tabelle (z.B. mit einem Tabellenkalkulationsprogramm) zur Verwaltung seiner Aufträge samt Einsatzplanung.

ANR	AUFTRAG	BAUSTELLE	PNR	MITARBEITER	PLZ	WOHNORT	KKASSE	BEITRAG
A1	Wohnhaus	Feuerbach	P1	Klaus Mertens	75365	Calw	TKK	13,10
A2	Spielplatz	Gerlingen	P1	Klaus Mertens	75365	Calw	TKK	13,10
A3	Brücke	Eberbach	P2	Hans Keller	70187	Stuttgart	BAR	13,80
			P1	Klaus Mertens	75365	Calw	TKK	13,10
A4	Hochhaus	Heilbronn	P3	Udo Klein	70376	Stuttgart	BAR	13,80

Die oben stehende nicht normalisierte Tabelle wird jetzt schrittweise in mehrere Tabellen zerlegen. Nach jedem Zerlegungschritt befinden sich diese in so genannten *Normalformen (NF)*. Wir wollen uns hier auf die ersten (in der Praxis üblichen) drei NF beschränken.

Normalformen

ERSTE NORMALFORM (1. NF)

Eine Relation heißt „in 1. NF", wenn gilt:

1. Es gibt einen eindeutig identifizierenden (Primär-)Schlüssel. (Zusammengesetzter PS erlaubt!)
2. Es gibt keine Wiederholfelder. (Bearbeiter1, Beaberbeiter2,....)
3. Die einzelnen Zeilen und Attribute sind im Informationsgehalt unabhängig von der Reihenfolge, in der sie betrachtet werden.
4. Jedes Attribut muss „atomar" sein, d.h. es darf nicht aus Einzelteilen bestehen, die für sich betrachtet einen Informationsgehalt haben (... der gesondert ausgewertet wird.)

Das Attribute MITARBEITER ist nicht atomar.

Konsequenz: Das Attribut MITARBEITER ist aufzulösen in *NAME* und *VORNAME (VNAME)*.

PNR wäre auch nicht atomar, wenn die Bearbeiter, z.B. durch Komma getrennt, in einem Datenfeld stehen würden („P2, P1,..."). Eine Aufteilung in mehrere Zeilen führt dann aber zu „Leerstellen" und dazu, dass die Reihenfolge der Datensätze nicht unabhängig ist (s. Beispiel Zeile 3 (A3) und Zeile 4 sind nicht unabhängig). Das „Auffüllen" der 4. Zeile würde zu redundanten Daten („Brücke", „Eberbach") und zu einem Primärschlüsellkonflikt (2X "A3" bzw. 2X "P1") führen, falls ANR bzw. PNR der PS ist.
Weder die PNR, noch die ANR ist somit isoliert als Primärschlüssel brauchbar.

Konsequenz: Primärschlüsselkandidat in dieser Tabelle kann aber die **Kombination** aus ANR und PNR sein, da diese Eindeutigkeit sicherstellt.

Primärschlüssel könnte sein:

ANR	**PNR**
A1	P1
A2	P1
A3	P1...

ZWEITE NORMALFORM (2. NF)

Eine Relation heißt „in 2. NF", wenn gilt:

1. Die Relation ist in 1. NF.
2. Die Relation enthält keine Attribute, die bereits von einer Teilmenge des Schlüssels eindeutig bestimmt werden.

Die Generierung eines eindeutigen PS (1.NF) führt dazu, dass nicht alle Attribute vom gesamten Primärschlüssel, also der Kombination aus ANR und PNR abhängen.
Die Personaldaten hängen **nur** von der Personalnummer (PNR) ab und die Auftragsdaten nur von der Auftragsnummer (ANR).
Beispielsweise sind die Personaldaten von Herrn Keller alleine von seiner Personalnummer P2 (also nur einem Teil des Primärschlüssels) abhängig.

Konsequenz: Auflösung der bisherigen Tabelle in eine PERSONALTABELLE (PS: PNR), eine Auftragstabelle (ANR) und in eine Tabelle AUFTRAG_PERSONAL für die „restlichen" Attribute, die nicht nur von einem Teilschlüssel (ANR oder PNR) abhängen sind. Übrigens, eine Schlüsselredundanz, wie z.B. in Tabelle AUFTRAG_PERSONAL ist übrigens völlig in Ordnung bzw. gar nicht vermeidbar.

AUFTRAGSTABELLE

ANR	AUFTRAG	BAUSTELLE
A1	Wohnhaus	Feuerbach
A2	Spielplatz	Gerlingen
A3	Brücke	Eberbach
A4	Hochhaus	Heilbronn

Primärschlüssel: ANR

PERSONALTABELLE

PNR	NAME	VNAME	PLZ	WOHNORT	KKASSE	BEITRAG
P1	Mertens	Klaus	75365	Calw	TKK	13,10
P2	Keller	Hans	70187	Stuttgart	BAR	13,80
P3	Klein	Udo	70376	Stuttgart	BAR	13,80

Primärschlüssel: PNR

AUFTRAG_PERSONAL

ANR	PNR
A1	P1
A2	P1
A3	P2
A3	P1
A4	P3

Primärschlüssel ANR + PNR

Es ist offensichtlich, dass diese beschriebene Abhängigkeit von einem *Teil des PS* nur dann möglich ist, wenn es sich um einen zusammensetzten PS handelt. Verfügt eine „1. NF-Tabelle" über einen Primärschlüssel, der aus *nur einem* Attribut besteht, befindet sie sich automatisch in der zweiten Normalform!!!

Aufgabe: Vergleichen Sie die Datenredundanzen der 1. und 2. NF!

DRITTE NORMALFORM (3. NF)

Eine Relation heißt „in 3.. NF", wenn gilt:

1. Die Relation ist in 2. . NF.
2. Die Relation enthält keine transitiv abhängigen Attribute, d.h. keine Felder, die nicht vom Primärschlüssel, sondern von einem anderen „Nichtschlüssel"-Atribut abhängig sind.

Aufgabe: Bevor Sie weiter lesen, überlegen Sie welches Attribut der oben dargestellten Relation Mitarbeiter hängt nicht von dem PS, sonder von einem anderen Attribut dieser Relation ab!

Das Feld BEITRAG (Krankenkassenbeitrag in Prozent des Bruttolohns) ist nicht direkt abhängig vom Primärschlüssel PNR, sondern nur indirekt (transitiv) über den Umweg KKASSE (Krankenkasse). Nicht die Personalnummer bestimmt den Beitrag, sondern die Mitgliedschaft in einer Krankenkasse.

Konsequenz: Auslösung einer Tabelle KKASSE aus der bisherigen PERSONALTABELLE. Die Zuordnung bleibt durch das gemeinsame Attribut KKASSE (FS->PS) erhalten.

PNR	NAME	VNAME	PLZ	KKASSE
P1	Mertens	Klaus	75365	TKK
P2	Keller	Hans	70187	BAR
P3	Klein	Udo	70376	BAR

Primärschlüssel: PNR

KKASSE	BEITRAG
TKK	13,10
BAR	13,80

Primärschlüssel: KKASSE

Aus den gleichen Gründen können wir auch noch den WOHNORT aus der PERSONALTABELLE verschwinden lassen und die Tabelle PLZ_WOHNORT bilden[1].

Entgegen einigen Publikationen ist jedoch zu sagen, dass das Attribut "PLZ" die Eigenschaften eines Primärschlüssels nur unzureichend erfüllt.
Die funktionale Abhängigkeit des Wohnorts von der Postleitzahl ist nicht gegeben:

- Welche Postleitzahl repräsentiert Stuttgart?
- Außerdem existieren Postleitzahlen, die für verschiedene Orte gelten.
  (Die Postleitzahl 57632 bspw. gilt für 15 verschiedene Ortschaften.)

Die neue Relation besitzt daher einen „Künstlichen Primärschlüssel", z.B. willkürliche fortlaufende Nummer oder einen zusammen gesetzter Schlüssel (PLZ+"fortlaufende Nr": 710341,710342, usw.).

[1] Im Gegensatz zum ersten Beispiel (KKASSE) ist die Missachtung der 3.NF und den damit verbunden Problemen bei PLZ weniger gravierend und daher ist diese Missachtung teilweise in der Praxis sogar üblich, da die vorgeschlagene Aufspaltung zu zusätzlichem Aufwand (z.B. bei Abfragen) führt.