Datenbanken und SQL

(Woche 1 - Tag 5)

Agenda

Normalisierung – Qualitätsstandard für Datenbanken

- 1. Normalform (1. NF)
  - Definition
  - Motivation
  - Beispiel
- 2. Normalform (2. NF)
  - Definition
  - Motivation
  - Beispiel
- 3. Normalform (3. NF)
  - Definition
  - Motivation
  - Beispiel

1. Normalform

1. Normalform -> Definition

- Eine Relationale Datenbank erfüllt die **1. Normalform**, wenn kein Attribut einer Tabelle existiert, dessen Informationsgehalt in eigenständige Teil-Informationen zerlegt werden kann.
- Man spricht in diesem Fall davon, dass alle Attribute **„atomar“** (also „unteilbar“) sind.
- Entsprechend wird der Bearbeitungs-Prozess einer Datenbank, mit dessen Hilfe die 1. Normalform erreicht werden soll, auch als **„Atomisierung“** bezeichnet.

1. Normalform -> Motivation

- Zu den zentralen Aufgaben einer Datenbank zählt es, **Informationen** bereitzuhalten, um diese bei entsprechenden Abfragen zur Verfügung stellen zu können.
- Solche Abfragen können aber **erschwert** (oder gegebenenfalls unmöglich) werden falls mehrere Informationen innerhalb eines einzigen Attributes „vermengt“ wurden.
- Mittels der **Atomisierung** wird eine solche „Vermengung“ rückgängig gemacht.

1. Normalform -> Beispiel

Person_ID	Name	Adresse	Telefonnummer
1	Petra Bayram	20095 Hamburg (Kölner Straße 1)	040 / 123456
2	Schmidt, Claudia	Baumweg 2, 10115 Berlin	030 - 654321
3	Hans Dieter Hüsch	Dorfweg 7, 67433 Neustadt	09161 1234

Atomisierung

Atomisierung

Person_ID	Vorname	Nachname	Straße	Hausnummer	PLZ	Stadt	Vorwahl	Rufnummer
1	Petra	Bayram	Kölner Straße	1	20095	Hamburg	040	123456
2	Claudia	Schmidt	Baumweg	2	10115	Berlin	030	654321
3	Hans Dieter	Hüsch	Dorfweg	7	67433	Neustadt	09161	1234

2. Normalform

2. Normalform -> Definition

- Die **formale Definition** der 2. Normalform klingt zunächst recht abschreckend:
- Eine Relationale Datenbank erfüllt die 2. Normalform, wenn sie zum einen die 1. NF erfüllt und darüber hinaus keine Tabelle besitzt, in der ein Nicht-Schlüssel-Attribut **eine funktionale Abhängigkeit zu einer echten Teilmenge des Primärschlüssels** aufweist.
- Wir werden aber an Hand eines **anschaulichen Beispiels** Licht ins Dunkel bringen.
- Ferner gilt, dass es eine nahezu **amüsante Methode** geben wird, diese hochkompliziert erscheinende Problematik mühelos zu umgehen … kurzer Cliffhanger ;-)

2. Normalform -> Motivation

- Verstöße gegen die 2. Normalform sind Hinweise darauf, dass eine Tabelle nicht **„monothematisch“** ist. Dieses ist aber ein zentrales Ziel der Normalisierung, da auf diese Weise „Redundanz“ vermieden wird.
- Das Ziel der **Redundanz-Vermeidung** kann auf zweierlei Weise motiviert werden:

  a) Die Vermeidung des mehrfachen Speicherns der selben Informationen **spart Speicherplatz**.
  b) Die Vermeidung von Informations-Dubletten erleichtert die **konsistente Pflege** der Datenbank.

2. Normalform -> Beispiel

Vorbemerkung:
Das folgende Beispiel verzichtet zunächst auf ein „ID-Attribut“ und bemüht sich stattdessen mit Hilfe ohnehin vorhandener Attribute einen (eindeutigen) **zusammengesetzten Primärschlüssel** zu bilden. Dem Beispiel geschuldet wollen wir dabei ignorieren, dass eine Person über Geburtsdatum und Postleitzahl üblicherweise nicht eindeutig beschrieben ist:

Geburtsdatum	PLZ	Vorname	Nachname	Vorwahl	Rufnummer
01.01.2001	20095	Petra	Bayram	040	123456
02.02.2002	10115	Claudia	Schmidt	030	654321
03.03.2003	67433	Hans Dieter	Hüsch	09161	1234

2. Normalform -> Beispiel

Definition „echte Teilmenge“

Da in der Mathematik jede Menge auch eine Teilmenge ihrer selbst ist, wurde der Begriff „echte Teilmenge“ eingeführt, um Teilmengen anzusprechen, die tatsächlich kleiner als ihre Obermenge sind. In unserem Fall ist die **Teilmenge „PLZ“** *echt kleiner* als der zusammengesetzte Primärschlüssel aus „Geburtsdatum+PLZ“.

=> Das Attribut „PLZ“ ist also eine **echte Teilmenge des Primärschlüssels**:

Geburtsdatum	PLZ	Vorname	Nachname	Vorwahl	Rufnummer
01.01.2001	20095	Petra	Bayram	040	123456
02.02.2002	10115	Claudia	Schmidt	030	654321
03.03.2003	67433	Hans Dieter	Hüsch	09161	1234

2. Normalform -> Beispiel

Definition „funktional abhängig“

Ein Attribut „A“ ist **funktional abhängig** von einem Attribut „B“, wenn (bei vollständiger Kenntnis aller relevanten Informationen) **A aus B zwingend gefolgert werden kann**.

In unserem Beispiel können wir aber die **Vorwahl** zwingend aus der **PLZ** folgern:

Geburtsdatum	PLZ	Vorname	Nachname	Vorwahl	Rufnummer
01.01.2001	20095	Petra	Bayram	040	123456
02.02.2002	10115	Claudia	Schmidt	030	654321
03.03.2003	67433	Hans Dieter	Hüsch	09161	1234

2. Normalform -> Beispiel

Definition „funktional abhängig“

Ein Attribut „A“ ist **funktional abhängig** von einem Attribut „B“, wenn (bei vollständiger Kenntnis aller relevanten Informationen) **A aus B zwingend gefolgert werden kann**.

In unserem Beispiel können wir aber die **Vorwahl** zwingend aus der **PLZ** folgern:

Geburtsdatum	PLZ	Vorname	Nachname	Vorwahl	Rufnummer
01.01.2001	20095	Petra	Bayram	040	123456
02.02.2002	10115	Claudia	Schmidt	030	654321
03.03.2003	67433	Hans Dieter	Hüsch	09161	1234

2. Normalform -> Lösung

Die fast schon amüsante Lösung besteht nun schlicht darin, echte Teilmengen eines Primärschlüssels auszuschließen, indem stets ein (einspaltiges) ID-Attribut eingeführt wird. (Dieses „künstliche“ Attribut wird auch als **„Surrogatschlüssel“** bezeichnet). Auf diese Weise entkommen wir der doch recht komplizierten Formulierung der 2. Normalform, indem wir allen Tabellen quasi „die Chance“ nehmen, Abhängigkeiten zu dann nicht mehr existierenden echten Teilmengen des Primärschlüssels zu besitzen.

Geburtsdatum	PLZ	Vorname	Nachname	Vorwahl	Rufnummer
01.01.2001	20095	Petra	Bayram	040	123456
02.02.2002	10115	Claudia	Schmidt	030	654321
03.03.2003	67433	Hans Dieter	Hüsch	09161	1234

2. Normalform -> Lösung

Die fast schon amüsante Lösung besteht nun schlicht darin, echte Teilmengen eines Primärschlüssels auszuschließen, indem stets ein (einspaltiges) ID-Attribut eingeführt wird. (Dieses „künstliche“ Attribut wird auch als **„Surrogatschlüssel“** bezeichnet). Auf diese Weise entkommen wir der doch recht komplizierten Formulierung der 2. Normalform, indem wir allen Tabellen quasi „die Chance“ nehmen, Abhängigkeiten zu dann nicht mehr existierenden echten Teilmengen des Primärschlüssels zu besitzen.

Person_ID	Geburtsdatum	PLZ	Vorname	Nachname	Vorwahl	Rufnummer
1	01.01.2001	20095	Petra	Bayram	040	123456
2	02.02.2002	10115	Claudia	Schmidt	030	654321
3	03.03.2003	67433	Hans Dieter	Hüsch	09161	1234

2. Normalform -> Lösung

Für jede Datenbank in der 1. Normalform, in der alle Tabellen ausschließlich **1-spaltige Primärschlüssel** besitzen, gilt also:
Diese Datenbanken sind dann stets – und also quasi **„automatisch“** – zugleich auch in der **2. Normalform**!

Person_ID	Geburtsdatum	PLZ	Vorname	Nachname	Vorwahl	Rufnummer
1	01.01.2001	20095	Petra	Bayram	040	123456
2	02.02.2002	10115	Claudia	Schmidt	030	654321
3	03.03.2003	67433	Hans Dieter	Hüsch	09161	1234

2. Normalform -> Lösung ... allerdings:

Für jede Datenbank in der 1. Normalform, in der alle Tabellen ausschließlich **1-spaltige Primärschlüssel** besitzen, gilt also: Diese Datenbanken sind dann stets – und also quasi **„automatisch“** – zugleich auch in der **2. Normalform**!

Freilich ändert diese Vorgehensweise dann aber noch nichts an der **funktionalen Abhängigkeit** zwischen Vorwahl und PLZ. Aus diesem Grund werden wir uns nun abschließend auch noch mit der **3. Normalform** beschäftigen.

Person_ID	Geburtsdatum	PLZ	Vorname	Nachname	Vorwahl	Rufnummer
1	01.01.2001	20095	Petra	Bayram	040	123456
2	02.02.2002	10115	Claudia	Schmidt	030	654321
3	03.03.2003	67433	Hans Dieter	Hüsch	09161	1234

3. Normalform

3. Normalform -> Definition + Motivation

- Auch die **formale Definition** der 3. Normalform klingt durchaus herausfordernd:
- Eine Relationale Datenbank erfüllt die 3. Normalform, wenn sie zum einen die 2. NF erfüllt und darüber hinaus keine Tabelle besitzt, in der ein Nicht-Schlüssel-Attribut **in transitiver Abhängigkeit** zu einer echten (oder unechten) Teilmenge des Primärschlüssels steht.
- Auch dies soll mittels eines **anschaulichen Beispiels** erläutert werden.
- Motivation ist erneut Redundanz-Vermeidung mittels **monothematischer Tabellen**.

Transitivität -> Definition

- Eine Beziehung heißt **„transitiv“,** wenn aus der Relation von **„A nach B“** und **„B nach C“** stets folgt, dass dann auch zwingend eine Beziehung von **„A nach C“** gilt.
- Beispiele:
  - Die Beziehung **„ist größer als“** ist transitiv, denn wenn **„A größer als B“** und **„B größer als C“** dann folgt stets: **„A ist größer als C“**.
  - Die Beziehung **„ist teurer als“** ist transitiv, denn wenn **„A teurer als B“** und **„B teurer als C“** dann folgt stets auch: **„A ist teurer als C“**.
  - Die Beziehung **„ist Vater von“** ist **nicht transitiv**, denn wenn **„A Vater von B“** und **„B Vater von C“** dann folgt nicht: **„A ist Vater von C“**.
- Insbesondere gilt dann aber auch: Die Beziehung **„funktional abhängig von“** ist transitiv!

3. Normalform -> Beispiel

Das Attribut **Vorwahl** ist transitiv abhängig von einer (echten) Teilmenge des Primärschlüssels:

Person_ID	Geburtsdatum	PLZ	Vorname	Nachname	Vorwahl	Rufnummer
1	01.01.2001	20095	Petra	Bayram	040	123456
2	02.02.2002	10115	Claudia	Schmidt	030	654321
3	03.03.2003	67433	Hans Dieter	Hüsch	09161	1234

3. Normalform -> Lösung

Eine solche Transitivität demonstriert die fehlende Monothematik der Tabelle.
Lösung wird die **Auslagerung** in eine eigenständige **Tabelle** sein:

Person_ID	Geburtsdatum	PLZ	Vorname	Nachname	Vorwahl	Rufnummer
1	01.01.2001	20095	Petra	Bayram	040	123456
2	02.02.2002	10115	Claudia	Schmidt	030	654321
3	03.03.2003	67433	Hans Dieter	Hüsch	09161	1234

3. Normalform -> Lösung

Eine solche Transitivität demonstriert die fehlende Monothematik der Tabelle.
Lösung wird die **Auslagerung** in eine eigenständige **Tabelle** sein:

n

Person_ID	Geburtsdatum	PLZ (FK)	Vorname	Nachname	Vorwahl	Rufnummer
1	01.01.2001	20095	Petra	Bayram	040	123456
2	02.02.2002	10115	Claudia	Schmidt	030	654321
3	03.03.2003	67433	Hans Dieter	Hüsch	09161	1234

1

PLZ (PK)	Vorwahl
20095	040
10115	030
67433	09161

3. Normalform: Redundanz-Vermeidung ? -> Erläuterung

Würden wir auf eine Auslagerung in eine eigenständige Tabelle verzichten,
so müssten wir viele **Informations-Dubletten** akzeptieren:

Person_ID	Geburtsdatum	PLZ	Vorname	Nachname	Vorwahl	Rufnummer
1	01.01.2001	20095	Petra	Bayram	040	123456
2	02.02.2002	10115	Claudia	Schmidt	030	654321
3	03.03.2003	67433	Hans Dieter	Hüsch	09161	1234

3. Normalform: Redundanz-Vermeidung ? -> Erläuterung

Würden wir auf eine Auslagerung in eine eigenständige Tabelle verzichten,
so müssten wir viele **Informations-Dubletten** akzeptieren:

Person_ID	Geburtsdatum	PLZ	Vorname	Nachname	Vorwahl	Rufnummer
1	01.01.2001	20095	Petra	Bayram	040	123456
2	02.02.2002	**10115**	Claudia	Schmidt	**030**	654321
3	03.03.2003	67433	Hans Dieter	Hüsch	09161	1234
…	…	…	…	…	…	…
70	…	**10115**	…	…	**030**	…
…	…	…	…	…	…	…
93	…	**10115**	…	…	**030**	…