

Information

2 Einführung

2.1 Bedeutung des DB-entwurfes

Der Datenbankentwurf ist für eine datenbankgestützte Anwendung wie die Entwurfsplanung (Statik, Bauplan etc.) für ein zu bauendes Haus. In beiden Fällen kann bei falscher Planung das Gesamtsystem wie ein Kartentaus zusammenfallen. Wenn die Statik nicht stimmt, nutzt es gar nichts, wenn man die teuersten Dachziegel für das Haus verwendet! Es bleibt einsturzgefährdet. Genauso nützt die tollste grafische Oberfläche einer datenbankgestützten Anwendung nichts, wenn der zugehörige Datenbankentwurf nicht korrekt durchgeführt worden ist. So eine Anwendung wird den Benutzer früher oder später zur Verzweiflung bringen.

Dennoch wird dem Datenbankentwurf allzu oft nur geringe Bedeutung beigemessen, da die Schwächen einer Anwendung im Entwurfsbereich sicherlich nicht so schnell auffallen wie Unschönheiten bei der Gestaltung einer grafischen Oberfläche.

In diesem zweiten Kapitel werden deshalb anhand eines Beispiels einige Problemstellungen und die Notwendigkeit eines konzeptionellen Datenbankentwurfs aufgezeigt.

2.2 Eine erste Tabelle

Im folgenden Beispiel, sollen zunächst zu allen Schülern einer Schule Daten erhoben werden. Zu diesen Daten gehören der 'Nachname' und der 'Vorname' eines Schülers. Um einem Schüler Post zustellen oder ihn anrufen zu können, müssen noch 'Straße', 'PLZ', 'Ort' und die 'Telefonnummer' gespeichert werden. Auch das 'Alter' des Schülers ist von Interesse. (Sicherlich

gibt es noch vieles mehr, was an Daten zu einem Schüler erheben werden könnte, z.B. welche Noten er hat, in welcher Klasse er ist, etc. Für das einführende Beispiel soll die obige Datenmenge reichen.)

Erste handschriftliche Tabelle

Eine fast selbstverständliche Möglichkeit die Daten handschriftlich zu notieren besteht darin, eine Tabelle anzulegen, in die die Daten dann eingetragen werden. Diese erste Tabelle könnte folgendes Aussehen haben:

Schüler

Name	Adresse	Tel.	Alter
Ernst, Uwe	Weg 1; 51111 Köln	98765	14
Müller, Kai	89077 Ulm; Straße 2	87654	16
Specht, Eva	Pfad 15; 11111 Berlin	76543	17
Groß, Ute	Allee 9, 24103 Kiel	45678	15

Die Auswertung dieser scheinbar selbsterklärenden und sehr einfachen handschriftlichen Tabelle kann allerdings zu Problemen führen.

Probleme

- Bei den Namen kann nicht zwischen Vor- und Nachname unterschieden werden. Ist Ernst oder Uwe der Vorname des ersten Schülers?
- Beginnt die Adresse immer mit der Straße oder mit der PLZ und dem Ort? Die Tabelle lässt beliebige Möglichkeiten der Adressangabe zu und wird dadurch schwerer lesbar.
- Die Tabelle ist nur kurzzeitig aktuell, da sich das Alter eines Schülers bei jedem Geburtstag ändert.

2 Einführung

Deshalb sollte die Tabelle wie folgt verbessert werden.

Verbesserungen:

- Um die Eindeutigkeit von Vor- und Nachname zu gewährleisten, muss die Spalte 'Name' in zwei Spalten 'Vorname' und 'Nachname' aufgeteilt werden.
- Auch bei dem Merkmal 'Adresse' kann man durch das Aufteilen in die Merkmale 'Straße', 'PLZ', 'Ort' die Eindeutigkeit und Unverwechselbarkeit der Einträge erreichen.
- Das Merkmal 'Alter' muss durch das Merkmal 'Geburtsdatum' ersetzt werden. Das Alter des Schülers kann dann tagesaktuell bestimmt werden.

Zweite handschriftliche Tabelle

Schüler

Nach-name	Vor-name	Straße	PLZ	Ort	Tel.	Geburts-datum
Ernst	Uwe	Weg 1	51111	Köln	98765	05.06.2002
Müller	Kai	Straße 2	89077	Ulm	87654	08.12.2000
Specht	Eva	Pfad 15	11111	Berlin	76543	06.08.1999
Groß	Ute	Allee 9	24103	Kiel	45678	05.07.2001

Bei dieser Tabelle ist die Spaltenzuordnung eindeutig. Ohne Verwechslungsmöglichkeit kann beispielsweise der Vorname eines Schülers bestimmt werden. Dieses wurde dadurch erreicht, dass in jeder Tabellenspalte eindeutige Werte und nicht etwa Listen oder Kombinationen von Werten mit unterschiedlichen Bedeutungen eingetragen werden. Vorerst genügt es, sich Folgendes zu merken¹:

In Tabellenspalten dürfen nur eindeutige Werte einge-tragen werden.

Der zweite Fehler, in der ersten handschriftlichen Tabel-le bestand im Eintragen sich ständig verändernder Da-ten, wie dem Alter der Schüler. Die Daten der Tabelle sollten möglichst dauerhaft Gültigkeit haben. Daher ist auch folgender Merksatz wichtig:

In Tabellen sollten möglichst keine Daten abgelegt wer-den, die ständigen Änderungen unterworfen sind.

2.3 Grundlegende Datenbankbegriffe

Um eine gemeinsame Sprache als Grundlage für die weiteren Erklärungen zu haben, müssen zunächst einige Begriffe eingeführt und definiert werden.

Merkmal²

Spaltenüberschriften werden als Merkmal bezeichnet.

Beispiel:

Die Tabelle Schüler hat die folgenden Merkmale Name, Vorname, Straße, PLZ, Ort, Tel., Geburtsdatum

Datensatz³

Die Zeilen einer Tabelle werden als Datensätze be-zeichnet.

Beispiel:

Die Tabelle Schüler hat 4 Datensätze. Einer der Daten-sätze lautet:

'Müller, Kai, Straße 2, 89077, Ulm, 87654, 08.12.2000'

Datenfeld⁴

Eine einzelne Zelle einer Tabelle wird als Datenfeld be-zeichnet. Es wird durch den Datensatz (Zeile) und das Merkmal (Spalte) bestimmt.

Datenfeldwert

Ein Datenfeldwert ist der konkrete Dateninhalt eines Merkmals in einem bestimmten Datensatz.

Beispiel

In der Beispieldatatable Schüler enthält das Datenfeld des Merkmals Ort im zweiten Datensatz den Datenfeldwert 'Ulm'.

Schlüssel

Wie im Folgenden gezeigt wird, ist es sinn-voll, dass in einer Datenbank jeder Daten-satz einer Tabelle eindeutig identifiziert werden kann. Daher muss man ein Merkmal oder eine Kombination von Merkmalen bestimmen, mit denen eine solche **eindeu-tige Identifizierung** möglich ist. Dieses Merkmal (bzw. diese Kombination von Merkmalen) wird dann Schlüsselmerkmal (bzw. Schlüsselmerkmale) oder einfach Schlüssel genannt. Ein Schlüssel muss eine weitere Anforderung erfüllen: Es dürfen nicht mehr Merkmale als unbedingt notwendig zur Bildung des Schlüssels kombiniert werden. Man sagt, der Schlüssel einer Tabelle muss **minimal** sein. Das Merkmal bzw. die Merkmale, die die Funktion des Schlüssels über-nnehmen, werden unterstrichen. So können sie von den anderen Merkmalen, den **Nichtschlüsselmerkmalen**, unterschieden werden.



¹ Diese Eigenschaft entspricht dem, was durch die erste Normalform, siehe Kapitel 8.2 als Grundvoraussetzung für eine zulässige Tabellen gefordert wird.

² Häufig werden die Merkmale auch als Attribute bezeichnet.

³ Der Begriff Tupel hat sich ebenfalls als Bezeichnung für einen Datensatz etabliert.

⁴ Der Begriff Zelle ist ebenfalls anstelle von Datenfeld gebräuchlich.

2 Einführung

Schüler

Nach-name	Vor-name	Straße	PLZ	Ort	Tel.	Geburts-datum
Ernst	Uwe	Weg 1	51111	Köln	98765	05.06.2002
Müller	Kai	Straße 2	89077	Ulm	87654	08.12.2000
Specht	Eva	Pfad 15	11111	Berlin	76543	06.08.1999
Groß	Ute	Allee 9	24103	Kiel	45678	05.07.2001

Um die handschriftliche Tabelle in eine Datenbank umsetzen zu können, wurde beispielsweise das Merkmal 'Nachname' als Schlüssel der Tabelle ausgezeichnet. Dieser Schlüssel ist sicherlich minimal. Weniger als ein Merkmal als Schlüssel geht nicht. Eindeutig ist er zumindest in der Beispieltabelle auch! Dennoch ist die Wahl dieses Schlüssels nicht sinnvoll.

Da der Schlüssel eindeutig die Datensätze identifizieren muss, darf es in der Tabelle keine zwei Schüler mit demselben Nachnamen geben. Der Schüler 'Müller, Sebastian' hat somit schlechte Karten! Er kann nicht als Schüler in die Tabelle eingetragen werden.

Ein aus den Merkmalen 'Nachname' und 'Vorname' kombinierter Schlüssel ist sicherlich besser geeignet als das Merkmal 'Name' alleine. Der Schüler 'Müller, Sebastian' kann nun in die Tabelle aufgenommen werden. Die Wahrscheinlichkeit, dass zwei Schüler den selben Namen und Vornamen haben, ist deutlich geringer. Ausgeschlossen ist dies aber nicht!

Um diesem Problem aus dem Weg zu gehen, wird ein neutrales zusätzliches Merkmal (ID: Identitätsnummer) zur Tabelle hinzugefügt. Mit Hilfe dieses Merkmals können die Datensätze mit einer eindeutigen Nummer durchnummieriert werden. Dieses Merkmal eignet sich hervorragend als Schlüssel der Tabelle, da es die zwei Eigenschaften Minimalität und Eindeutigkeit erfüllt.

Schüler

ID_Schüler	Nachname	Vorname	Straße	...
1	Ernst	Uwe	Weg 1	...
2	Müller	Kai	Straße 2	...
3	Specht	Eva	Pfad 15	...

Schlüssel¹

Ein Schlüssel muss folgende beiden Bedingungen erfüllen:

- **Eindeutigkeit:** Ein Schlüssel identifiziert eindeutig die Datensätze der Tabelle.
- **Minimal:** Wird ein Schlüssel aus mehreren Merkmalen kombiniert, dürfen nicht mehr Merkmale als unbedingt notwendig an der Kombination beteiligt werden.

Auch bei konsequenter Verwendung von ID_Werten als Schlüsselmerkmal kann und wird es zusammengesetzte Schlüssel geben. Diese treten insbesondere bei Beziehungstabellen auf (Siehe Kapitel 2.5).

Tabellen einer Datenbank

Im Gegensatz zu einer handschriftlichen Tabelle muss eine Tabelle einer Datenbank folgende Eigenschaften erfüllen:

Tabelle²

Eine Tabelle ist eine Menge von Datensätzen, die tabellenförmig angeordnet werden und folgende eindeutigen Eigenschaften erfüllen:

- **Tabellenname:** Eine Tabelle besitzt einen, innerhalb der Datenbank, eindeutigen Namen.
- **Merkmalsnamen:** Der Name eines Merkmals ist innerhalb der Tabelle eindeutig.
- **Schlüssel:** Jede Tabelle besitzt einen Schlüssel.

Zudem gilt, dass die Anzahl und Reihenfolge der Merkmale und der Datensätze einer Tabelle beliebig sind.

2.4 Eine erste Beziehung

Als nächstes soll die Schülertabelle so erweitert werden, dass die Information, welcher Lehrer Klassenlehrer des jeweiligen Schülers ist, mit aufgenommen werden kann. Vom Klassenlehrer werden Vor- und Nachname als weitere Merkmale an die Tabelle angefügt.

Schüler

ID_Schüler	S_Nach-name	S_Vor-name	Straße	PLZ	...
1	Ernst	Uwe	Weg 1	51111	...
2	Müller	Kai	Straße 2	89077	...
3	Specht	Eva	Pfad 15	11111	...
4	Groß	Ute	Allee 9	24103	..

...	Ort	Tel.	Geburts-datum	L_Nach-name	L_Vor-name
...	Köln	98765	05.06.2002	Moll	Willi
...	Ulm	87654	08.12.2000	Moll	Willi
...	Berlin	76543	06.08.1999	Kurp	August
...	Kiel	45678	05.07.2001	Kurp	August

Anmerkung

Da die Merkmalsnamen einer Tabelle eindeutig sein müssen, wurde den Merkmalen 'Nach-' und 'Vorname' jeweils ein S_ bzw. ein L_ vorangestellt, um den Schüler- vom Lehrernamen unterscheiden zu können.

1 Synonym werden auch die Bezeichnungen Primärschlüssel oder Hauptschlüssel anstelle von Schlüssel verwendet.

2 Oft werden Tabellen auch als Relationen bezeichnet

Teil 2: Informationen

2 Einführung

Ein wesentliches Problem dieser Umsetzung besteht darin, dass unterschiedliche Lehrer mit gleichem Vor- und Nachnamen nicht unterschieden werden können. Um auch hier die Eindeutigkeit herzustellen, kann ebenfalls eine ID_Nummer, diesmal allerdings für Lehrer hinzugefügt werden.

Schüler

ID_Schüler	S_Nach-name	S_Vor-name	Straße	PLZ	Ort	...
1	Ernst	Uwe	Weg 1	51111	Köln	...
2	Müller	Kai	Straße 2	89077	Ulm	...
3	Specht	Eva	Pfad 15	11111	Berlin	...
4	Groß	Ute	Allee 9	24103	Kiel	...

...	Tel.	Geburts-datum	ID_Lehrer	L_Nach-name	L_Vor-name
...	98765	05.06.2002	1	Moll	Willi
...	87654	08.12.2000	1	Moll	Willi
...	76543	06.08.1999	2	Kurp	August
...	45678	05.07.2001	2	Kurp	August

Das Merkmal 'ID_Lehrer' dieser Umsetzung ist kein Schlüssel-Merkmal der Tabelle 'Schüler', da dadurch die Datensätze nicht eindeutig bestimmt werden. Lediglich gleichnamige Lehrer können durch diesen ID-Wert unterschieden werden.

Anomalien

Doch auch diese veränderte Tabelle weist noch erhebliche Probleme auf, die auch als Anomalien bezeichnet werden. Dabei werden die folgenden vier Anomalien unterschieden:

- Rechtschreibfehler führen zu inkonsistenten Daten. Alle Lehrerdatensätze mit dem selben 'ID_Lehrer' - Wert müssen auch den selben Lehrer - 'Nachnamen' und 'Vornamen' haben. Ist dies nicht der Fall liegt ein Fehler im Datenbestand vor. Die **Mutationsanomalie** entspricht dem versehentlichen Falschschreiben und somit dem unbeabsichtigten „Mutieren“ von Daten.
- Ein neuer (Klassen-)Lehrer kann erst hinzugefügt werden, wenn es auch einen Schüler gibt, der diesen Lehrer als Klassenlehrer hat. Andernfalls müsste ein „Dummy“-Schüler (Leerdatensatz) erfunden werden, um den neuen Lehrer in die Tabelle einzufügen zu können.

Unter **Einfügeanomalie** versteht man das ungewollte, aber zwingend notwendige Einfügen von Daten, z.B. von Dummydatensätzen, um die tatsächlich gewünschten Daten speichern zu können.

- Beim Löschen der Schüler können ungewollt auch Lehrerinformationen verschwinden. Immer wenn der letzte Schüler gelöscht wird, bei dem ein Lehrer Klassenlehrer ist, werden die zugehörigen Lehrerdaten auch gelöscht.

Die **Löschanomalie** ist das Gegenstück zur Einfügeanomalie. Unter einer Löschanomalie versteht man das ungewollte Löschen von Daten.

- Wenn ein Lehrer, beispielsweise nach einer Heirat, seinen Namen ändert, führt diese eine Namensänderung dazu, dass bei allen zugehörigen Schülerdatensätzen der Lehrername ebenfalls geändert werden muss.

Die **Änderungsanomalie** liegt vor, wenn eine einzelne Änderung weitere Folgeänderungen nach sich zieht.

Redundanz

Anomalien treten auf Grund von Redundanzen auf. Eine Redundanz ist die Wiederholungen von Daten OHNE tatsächlichen Informationsgewinn. Beispielsweise ist die Wiederholung des Namens Willi Moll eine Redundanz. Ist nämlich einmal bekannt, dass dem Lehrer mit der ID 1 der Name Willi Moll zugeordnet ist, enthält die erneute Nennung des Namens keine weitere Information². Wichtig ist die Erkenntnis:

Redundante Daten sind in einer Datenbank auf jeden Fall zu vermeiden!

Mit einer einzelnen Tabelle lassen sich die Redundanzen in dem Beispiel nicht vermeiden. Die Daten müssen auf zwei Tabellen, eine für die Schülertypen und eine für die Lehrertypen aufgeteilt werden.

ID_Lehrer	Nachname	Vorname
1	Moll	Willi
2	Kurp	August

Schüler

ID_Schüler	S_Nach-name	S_Vor-name	Straße	PLZ	Ort	Tel.	Geburts-datum
1	Ernst	Uwe	Weg 1	51111	Köln	98765	05.06.2002
2	Müller	Kai	Straße 2	89077	Ulm	87654	08.12.2000
3	Specht	Eva	Pfad 15	11111	Berlin	76543	06.08.1999
4	Groß	Ute	Allee 9	24103	Kiel	45678	05.07.2001

Die beiden Tabellen 'Schüler' und 'Lehrer' enthalten jetzt keine redundanten Daten mehr. Es besteht zwischen ihnen aber auch keine Beziehung, so dass die Information, welcher Lehrer der Klassenlehrer eines Schülers ist, nicht daraus abgelesen werden kann. Die beiden Tabellen sind völlig unabhängig voneinander.

1 Ein Dummy ist ein Platzhalter für einen noch nicht bekannten aber benötigten ersten Datensatz.

2 Im Zusammenhang mit dem Thema Normalformen (siehe Kapitel 8) spielen Anomalien und Redundanzen eine wichtige Rolle.

2 Einführung

Fremdschlüssel

Um zwei unabhängige Tabellen miteinander in Beziehung zu setzen, kann der Schlüssel der einen Tabelle als Identifikationsmerkmal an die andere Tabelle angefügt werden.

Lehrer

ID_Lehrer	Nachname	Vorname
1	Moll	Willi
2	Kurp	August



Schüler

ID_Schüler	Nach-name	Vor-name	Straße	PLZ	...
1	Ernst	Uwe	Weg 1	51111	...
2	Müller	Kai	Straße 2	89077	...
3	Specht	Eva	Pfad 15	11111	...
4	Groß	Ute	Allee 9	24103	...

...	Ort	Tel.	Geburtsdatum	ID_Lehrer
...	Köln	98765	05.06.2002	1
...	Ulm	87654	08.12.2000	1
...	Berlin	76543	06.08.1999	2
...	Kiel	45678	05.07.2001	2

Auf diese Weise kann die Information 'ist Klassenlehrer' in die Tabellen eingefügt werden. Die Tabellen 'Schüler' und 'Lehrer' sind jetzt nicht mehr unabhängig voneinander. Durch das zusätzliche Merkmal 'ID_Lehrer' in der Tabelle 'Schüler' werden die Tabellen zueinander in Beziehung gesetzt. Das Merkmal 'ID_Lehrer' bezeichnet man auch als Fremdschlüssel. Auf diese Weise wird die Beziehung 'ist Klassenlehrer' umgesetzt. Beispielsweise ist der Lehrer Kurp mit ID_Lehrer = 2 der Klassenlehrer der Schüler Specht und Groß.

Fremdschlüssel

Als Fremdschlüssel einer Tabelle wird ein Merkmal oder eine Kombination von Merkmalen bezeichnet, die in einer anderen Tabelle als Schlüssel vorkommen. Mit Hilfe von Fremdschlüsseln werden die Beziehungen zwischen Tabellen realisiert.



12.1.1

- 1 Häufig wird in der Literatur der Fremdschlüssel mit einer gestrichelten Linie unterstrichen, um das Fremdschlüsselmerkmal von anderen Merkmalen eindeutig unterscheiden zu können. In diesem Buch werden alle Schlüssel mit „ID_“ beginnen und die Fremdschlüssel bekommen immer denselben Namen wie der zugehörige Schlüssel. Daher ist bei dieser Namenskonvention die Zuordnung von Schlüsseln und Fremdschlüsseln sehr einfach. Auf die Unterstreichung der Fremdschlüssels kann deshalb verzichtet werden. Diese Namenskonvention hat sich als sehr praktikabel und hilfreich herausgestellt. Zwingend ist sie jedoch nicht! Beispielsweise kann man Fremdschlüssel auch Grundsätzlich mit FS_ beginnen lassen.

Datenbank

Um Daten redundanzfrei in einer Datenbank ablegen zu können, reicht eine Tabelle in der Regel nicht aus. Meistens müssen die Daten auf mehrere untereinander in Beziehung stehende Tabellen verteilt werden. Die Beziehungen zwischen den Tabellen realisieren sogenannte Fremdschlüssele. Die nachfolgende, sehr allgemeingehaltene Definition einer Datenbank greift diesen Grundgedanken der zerteilten und in Beziehung stehenden Daten wieder auf.

Datenbank

Eine Datenbank ist eine Sammlung von Daten, die miteinander in Beziehung stehen.

Tabellenschema

Tabellenschema

Die Gesamtheit aller Tabellen einer Datenbank wird als Tabellenschema bezeichnet.

Relationale Datenbank

Der Begriff der Relation wird in der Mathematik im Sinne von Beziehung verwendet. Übertragen auf den relationalen Datenbankbereich werden Relationen durch Tabellen umgesetzt. Sowohl das Verhältnis der Werte der unterschiedlichen Merkmale eines Datensatzes zueinander als auch die Fremdschlüssele zwischen Tabellen beschreiben Beziehungen. Dieses auf dem mathematischen Fundament begründete Datenbankmodell wurde erstmals 1970 von Edgar F. Codd vorgeschlagen und bildet noch heute die Grundlage der relationalen Datenbankmanagementsysteme.

Andere Datenbankmodelle

Es gibt neben dem relationalen noch weitere, jedoch weniger wichtige Datenbankmodelle.

- Hierarchisches Datenbankmodell
- Netzwerkdatenbankmodell
- Objektrelationales Datenbankmodell
- Objektorientiertes Datenbankmodell



12.1.2

RDBMS

Ein relationales Datenbankmanagementsystem (kurz RDBMS) umfasst neben den eigentlichen Daten und deren Beziehungen, die in Form von Tabellen gespeichert werden auch Verwaltungskomponenten. Diese Verwaltungskomponenten dienen dem Anlegen, Modifizieren, Sichern und Verwalten von Daten, Tabellen und Datenbanken ebenso wie dem Anlegen und Verwalten von Datenbankbenutzern und deren Datenbankzugriffsrechte. Die Datenabfrage und -manipulationssprache SQL als integralen Bestandteil eines jeden RDBMS ermöglicht viele dieser Aufgaben (Siehe Kapitel 10).

2 Einführung

2.5 Eine zweite Beziehung

Als zusätzliche Erweiterung des einführenden Beispiels sollen nun auch die Schulklassen mit in die Datenbank aufgenommen werden. Die drei Klassen

- BG (Berufsgrundschuljahr),
- IF (Technische/r Assistent/in Informationstechnik),
- FO13 (Fachoberschule Klasse 13)

sollen dabei als Beispieldatensätze genügen. Aus den Tabellen soll ablesbar sein, welcher Lehrer in welcher Klasse unterrichtet. Eine erste Idee besteht darin der Tabelle 'Lehrer' die zugehörigen Klassen in einer weiteren Spalte als Aufzählung anzufügen.

Lehrer

ID_Lehrer	Nachname	Vorname	Klasse
1	Moll	Willi	IF, TA-Informationstechnik FO13, Fachoberschule Kl.13
2	Kurp	August	BG, Berufsgrundschuljahr FO13, Fachoberschule Kl.13

Auch diese Tabellenumsetzung enthält wieder Listen und Redundanzen, die zu Anomalien und damit zu inkonsistenten Daten führen können. Um diese zu vermeiden, müssen die Daten wieder auf mehrere Tabellen verteilt und die Wertebereiche der Spalten eindeutig und unteilbar werden. Die eigenständigen Tabellen 'Klasse' und 'Lehrer' sehen wie folgt aus.

Klasse

ID_Klasse	Kürzel	Beschreibung
1	BG	Berufsgrundschuljahr
2	IF	TA-Informationstechnik
3	FO13	Fachoberschule Kl.13

Lehrer

ID_Lehrer	Nachname	Vorname
1	Moll	Willi
2	Kurp	August

In diesen Tabellen ist die Information 'welcher Lehrer in welcher Klasse unterrichtet' noch nicht enthalten! Anders formuliert: Die Beziehung 'unterrichtet' ist noch nicht umgesetzt. Auch im vorherigen Kapitel mussten die voneinander unabhängigen Tabellen 'Lehrer' und 'Schüler' in Beziehung gesetzt werden, um die Information 'ist Klassenlehrer' umsetzen zu können. Das Problem wurde gelöst, indem der Schlüssel der einen Tabelle als Fremdschlüssel der anderen Tabelle angefügt wurde. Diese Problemlösung kann auf die Tabellen 'Klasse' und 'Lehrer' nicht übertragen werden, denn:

ID_Klasse an die Tabelle 'Lehrer' anfügen

Fügt man das Merkmal 'ID_Klasse' als Fremdschlüssel in die Tabelle 'Lehrer' ein, so kann man für jeden Lehrer genau eine Klasse angeben, in der er unterrichtet.

ID_Lehrer an die Tabelle 'Klasse' anfügen

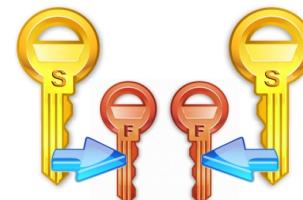
Wird hingegen das Merkmal 'ID_Lehrer' als Fremdschlüssel in die Tabelle 'Klasse' eingefügt, so kann für jede Klasse nur noch ein einziger Lehrer bestimmt werden, der in der Klasse unterrichtet.

Beide Varianten spiegeln nicht den tatsächlichen Sachverhalt wider. Das Hinzufügen von Fremdschlüsseln zu bestehenden Tabellen reicht also nicht aus, um die Beziehung 'unterrichtet' umzusetzen. Stattdessen muss für die Beziehung 'unterrichtet' eine weitere Tabelle, eine **Beziehungstabelle**, hinzugefügt werden.

Beziehungstabelle

Eine Tabelle, die der Umsetzung einer Beziehung dient, nennt man Beziehungstabelle.

In diese Tabelle können alle möglichen Kombinationen zwischen der Tabelle 'Lehrer' und der Tabelle 'Klasse' abgebildet werden.



unterrichtet

	ID_Lehrer	ID_Klasse	
Lehrer 1 (Moll)	1	2	(IF) Klasse 2
	1	3	(FO13) Klasse 3
Lehrer 2 (Kurp)	2	1	(BG) Klasse 1
	2	3	(FO13) Klasse 3

Die Beziehungstabelle 'unterrichtet' spiegelt folgende Informationen wider:

Der Lehrer 1 (Moll) unterrichtet in den Klassen

- 2 (IF),
- 3 (FO13),

Der Lehrer 2 (Kurp) unterrichtet in den Klassen

- 1 (BG)
- 3 (FO13).

Zusammengesetzter Schlüssel

Der Schlüssel der Beziehungstabelle wird aus der Kombination der Fremdschlüsselelemente gebildet. Die einzelnen Merkmale, aus denen sich Schlüssel zusammensetzen werden auch als Teilschlüssel bezeichnet.



2.6 Eine dritte Beziehung

In jeder Klasse wird zu Beginn eines Schuljahres ein Klassensprecher gewählt. Diese Beziehung 'ist Klassensprecher' zwischen den Tabellen 'Schüler' und 'Klasse' soll ebenfalls in die Datenbank aufgenommen werden.

Anmerkung

Um die Lesbarkeit des Beispiels zu erhöhen, werden alle Nichtschlüsselmerkmale der Tabelle 'Schüler' bis auf das Merkmal 'Nachname' weggelassen.

Schüler

ID_Schüler	Nachname	...
1	Ernst	...
2	Müller	...
3	Specht	...
4	Groß	...

Klasse

ID_Klasse	Kürzel	Beschreibung
1	BG	Berufsgrundschuljahr
2	IF	TA-Informationstechnik
3	FO13	Fachoberschule Kl.13

Betrachtet wird folgende Beispielsituation

- Die beiden Schüler 'Ernst' und 'Müller' gehen in die selbe Klasse IF TA-Informationstechnik.
- Der Schüler 'Ernst' ist der Klassensprecher dieser Klasse.
- Die Schülerin Specht ist Klassensprecherin der Klasse BG Berufsgrundschuljahr.
- Die Schülerin Groß ist Klassensprecherin der Klasse FO13 Fachoberschule.

Umsetzungsvarianten

Es gibt drei mögliche Varianten für die Beziehung 'ist Klassensprecher'. Entweder man fügt den Schlüssel einer Tabelle als Fremdschlüssel in die andere Tabelle ein oder man legt eine eigenständige Beziehungstabelle an.

Variante 1

'ID_Klasse' als Fremdschlüssel in der Tabelle 'Schüler'.

Schüler

ID_Schüler	Nachname	...	ID_Klasse
1	Ernst	...	2
2	Müller	...	
3	Specht	...	1
4	Groß	...	3

Diese erste Variante ist ungünstig. Da nur die wenigsten Schüler auch Klassensprecher sein werden, ist bei dieser Umsetzungsvariante mit sehr vielen leeren Datenfeldern zu rechnen.

Variante 2

'ID_Schüler' als Fremdschlüssel in der Tabelle 'Klasse'.

Klasse

ID_Klasse	Kürzel	Beschreibung	ID_Schüler
1	BG	Berufsgrundschuljahr	3
2	IF	TA-Informationstechnik	1
3	FO13	Fachoberschule Kl.13	4

Da in jeder Klasse ein Schüler als Klassensprecher gewählt wird, ist davon auszugehen, dass gar keine leeren Datenfelder entstehen. Somit ist diese zweite Variante eine sehr günstige Umsetzung.

Variante 3

Eine eigenständige Beziehungstabelle 'ist Klassensprecher' zwischen den Tabellen 'Klasse' und 'Schüler'.

Ist_Klassensprecher

ID_Klasse	ID_Schüler
1	3
2	1
3	4

Der Aufwand, eine extra Beziehungstabelle zu führen, ist erheblich größer, als lediglich den Schlüssel einer Tabelle als Fremdschlüssel einer anderen Tabelle zuzuordnen. Daher ist Variante 3 ungünstig.

Fazit

Da alle drei Varianten möglich sind (Redundanzen oder Listen treten nicht auf) wird man sich für die Variante entscheiden, bei der die wenigsten leeren Datenfelder und der geringste Verwaltungsaufwand zu erwarten ist. Für die Erweiterung der Datenbank um die Beziehung 'ist Klassensprecher' muss nur die Tabelle 'Klasse' um das Fremdschlüsselmerkmal 'ID_Schüler' ergänzt werden.

Bei mehreren möglichen Umsetzungsvarianten wird die Variante mit den wenigsten leeren Datenfeldern und dem geringsten Verwaltungsaufwand gewählt.

2.7 Eine fehlerhafte Beziehung

Abschließend soll die Beziehung 'ist Mitschüler' und somit die Information, welcher Schüler in welcher Klasse ist, in die Datenbank aufgenommen werden. Da jeder Schüler genau einer Klasse zuzuordnen ist, kann diese Beziehung 'ist Mitschüler' (analog zur Beziehung 'ist Klassenlehrer' aus Kapitel 2.4) umgesetzt werden, indem man das Schüsselmerkmal 'ID_Klasse' der Tabelle 'Klasse' als Fremdschlüssel an die Tabelle 'Schüler' anfügt.

Schüler

ID_Schüler	Nach-name	Vor-name	Straße	PLZ	...
1	Ernst	Uwe	Weg 1	51111	...
2	Müller	Kai	Straße 2	89077	...
3	Specht	Eva	Pfad 15	11111	...
4	Groß	Ute	Allee 9	24103	..

...	Ort	Tel.	Geburts-datum	ID_Klasse	ID_Lehrer
...	Köln	98765	05.06.2002	2	1
...	Ulm	87654	08.12.2000	2	1
...	Berlin	76543	06.08.1999	1	2
...	Kiel	45678	05.07.2001	3	2

'ID_Klasse' realisiert die Beziehung 'ist Mitschüler'. 'ID_Lehrer' realisiert die Beziehung 'ist Klassenlehrer'.

Klasse

ID_Klasse	Kürzel	Beschreibung	ID_Schüler
1	BG	Berufsgrundschuljahr	3
2	IF	TA-Informationstechnik	1
3	FO13	Fachoberschule Kl.13	4

'ID_Schüler' realisiert die Bez. 'ist Klassensprecher'.

Lehrer

ID_Lehrer	Nachname	Vorname
1	Moll	Willi
2	Kurp	August

unterrichtet

ID_Lehrer	ID_Klasse
1	2
1	3
2	1
2	3

Obwohl jeder Einzelschritt bei der Herleitung dieser Tabellen für sich betrachtet richtig und nachvollziehbar war, ist das entstandene Tabellenschema in seiner Gesamtheit trotzdem fehlerhaft!

Dadurch, dass mit Hilfe der Beziehung 'ist Mitschüler' jetzt bestimmt werden kann, in welche Klasse die Schüler gehen, muss nicht mehr für jeden Schüler einzeln festgelegt werden, welchen Klassenlehrer er hat. Es ist völlig ausreichend einen Klassenlehrer je Klasse anzugeben. Dazu müssen an den Tabellen folgende Änderungen durchgeführt werden:

1. Das Fremdschlüsselmerkmal 'ID_Lehrer' aus der Tabelle 'Schüler' entfernen
2. In der Tabelle Klasse das Merkmal 'ID_Lehrer' als Fremdschlüssel hinzufügen.

Beziehungsmerkmal

Besonders deutlich wird der Fehler in diesem Tabellenschema, wenn man beispielsweise die Information „seit wann ist ein Lehrer Klassenlehrer einer Klasse“ ebenfalls in der Datenbank speichern möchte. Dieses zusätzliche Merkmal 'seit wann' hängt logisch mit der Beziehung 'ist Klassenlehrer' zusammen und ist somit als Beziehungsmerkmal der Beziehung 'ist Klassenlehrer' zuzuordnen. Würde man aber dieses Merkmal, wie die Beziehung 'ist Klassenlehrer' ebenfalls fälschlicherweise durch Anfügen an die Tabelle 'Schüler' umsetzen, käme es wieder aufgrund von Redundanzen zu Anomalien. Dieses Problem kann nur sinnvoll umgangen werden, wenn beide Merkmale, das Fremdschlüsselmerkmal 'ID_Lehrer' und das zusätzliche Merkmal 'seit wann' in die Tabelle 'Klasse' überführt werden. (Um diese Problematik nachvollziehen zu können vergleichen Sie bitte unbedingt das hier vorgestellte fehlerhafte Tabellenschema mit dem korrigierten Tabellenschema im Kapitel 5.1)

Fazit

Dieses Beispiel veranschaulicht sehr gut, wie schwierig es ist, ohne Datenanalyse und strukturierten Datenbankentwurf, zu einem brauchbaren Tabellenschema zu gelangen. Welches Vorgehen beim Datenbankentwurf am sinnvollsten ist und welche Hilfsmittel dabei zur Verfügung stehen, wird in den folgenden Kapiteln gezeigt.

Der Datenbankentwurf kann nicht ohne Datenanalyse durchgeführt werden.



12.1.4