
3.2 Aufgaben

Wandeln Sie die ER-Modelle aus dem vorhergehenden Kapitel in Tabellenmodelle um.

4 Normalformen

Wenn Datenbanken über einen längeren Zeitraum verändert und angepasst werden, kann man erkennen, dass die Konsistenz der Daten, d.h. die "Stimmigkeit der Daten" abnimmt. Ebenso kommt es oft vor, dass Daten redundant gehalten und somit an vielen Stellen gepflegt werden müssen.

Dies versucht man durch das Konzept der **Normalisierung** zu verhindern. Sie soll

- redundante Daten
- Anomalien unter den Daten (Einfüge-, Änderungs-, Löschanomalie)

verhindern.

Empno(PK)	Ename	Hiredate	Manager	Deptno	Dname
1	Smith	10.12.02	34 100	1	Verkauf
2	Meyer	10.04.88	34 ?	2	Marketing
3	Jones	31.01.03	34 ?	2	Marketing
34	King	24.02.01	100	4	Management
<NULL>	<NULL>	<NULL>	<NUL>	5	Controlling

Einfüge-Anomalie

Eine Einfüge-Anomalie tritt auf, wenn ein Datensatz gespeichert werden soll und dieser keine oder keine eindeutigen Primärschlüsselwerte aufweist. Das Einfügen in eine Tabelle ist somit nicht möglich. Informationen können nicht gespeichert werden und gehen womöglich verloren. Das kann zum Beispiel der Fall sein, wenn für die Speicherung der Kundendaten zu Verifizierungszwecken die Personalausweisnummer als Primärschlüssel verwendet wird, diese aber leider vom Sachbearbeiter nicht erfasst werden konnte. Der Datensatz des Kunden kann nicht gespeichert werden.

Änderungs-Anomalie

Man spricht von einer Änderungs-Anomalie, wenn eine Entität redundant in einer oder sogar in mehreren Tabellen enthalten ist und bei einer Aktualisierung nicht alle berücksichtigt werden. Dadurch kommt es zur Inkonsistenz im Datenbestand. Es kann möglicherweise nicht mehr nachvollzogen werden welcher Wert der gültige ist. Dieser Sachverhalt lässt sich gut an einer Auftragstabelle darstellen. Diese speichert neben der Auftragsnummer auch den Namen eines Kunden und dessen Bestellung. Ein Kunde kann mehrere Bestellungen aufgegeben haben, wobei jede Bestellung in einem Datensatz erfasst wird. Wird nun aufgrund eines Schreibfehlers nachträglich der Name des Kunden „Reiher“ in „Reier“ bei einem Datensatz geändert, führt dies zu einem inkonsistenten Datenbestand. Nach der Änderung liegen demnach Aufträge für scheinbar zwei verschiedene Kunden vor und zwar für einen Kunden „Reiher“ und einen Kunden „Reier“.

Lösch-Anomalie

Enthalten die Datensätze einer Tabelle mehrere unabhängige Informationen, so kann es leicht zu Lösch-Anomalien kommen. Da sich die Daten in einem nicht-normalisierten Zustand befinden, kann durch Löschen eines Datensatzes ein Informationsverlust entstehen. Die Ursache liegt darin, dass in einer Tabelle unterschiedliche Sachverhalte gespeichert werden. Am Beispiel einer nicht-normalisierten Mitarbeitertabelle soll dies kurz skizziert werden. In der Mitarbeitertabelle werden die Personalnummer, der Name und die Abteilung gespeichert. Der Mitarbeiter „Krause“, der als einziger in der Abteilung „Lager“ war, ist aus dem Unternehmen ausgetreten und wird daher aus der Datenbank gelöscht. Da die Abteilung in der gleichen Tabelle gespeichert wird, verschwindet das „Lager“ aus der Datenbank, da „Herr Krause“ ja als einziger dieser Abteilung zugeordnet war.

Datenbank-Anomalien auflösen

Die beschriebenen Anomalien treten durch ein schlechtes Datenbank-Design auf. Daraus ergibt sich auch die redundante Datenhaltung. Um diese zu vermeiden, müssen die Tabellen einer Datenbank normalisiert werden. Die Normalisierung umfasst in der Praxis drei Stufen und sorgt für eine redundanzfreie und nach Entitäts-Typ getrennte Datenhaltung.

4.1 Normalform 1

- keine Wiederholungsgruppen
- Atomarität

Die 1. NF versucht, das Problem von Wiederholungsgruppen zu verhindern und fordert die Atomarität von Spaltenwerten. Folgendes Beispiel soll dies verdeutlichen.

In einer Tabelle soll der Warenkorb eines Shop-Besuchs abgelegt werden. Jeder Warenkorb wird durch seinen Primärschlüssel identifiziert. Wir gehen hier davon aus, dass der Warenkorb nur von angemeldeten Kunden gefüllt wird. Im Wesentlichen wird hier die Information gespeichert, welcher Artikel wie oft im Warenkorb abgelegt wurde. Das Ergebnis erster Überlegungen sieht wie folgt aus.

warenkorb_id	artikel	kunde_id
1	7856 30;7863 50;9015 1	12345
2	3006 1;3010 4	12346

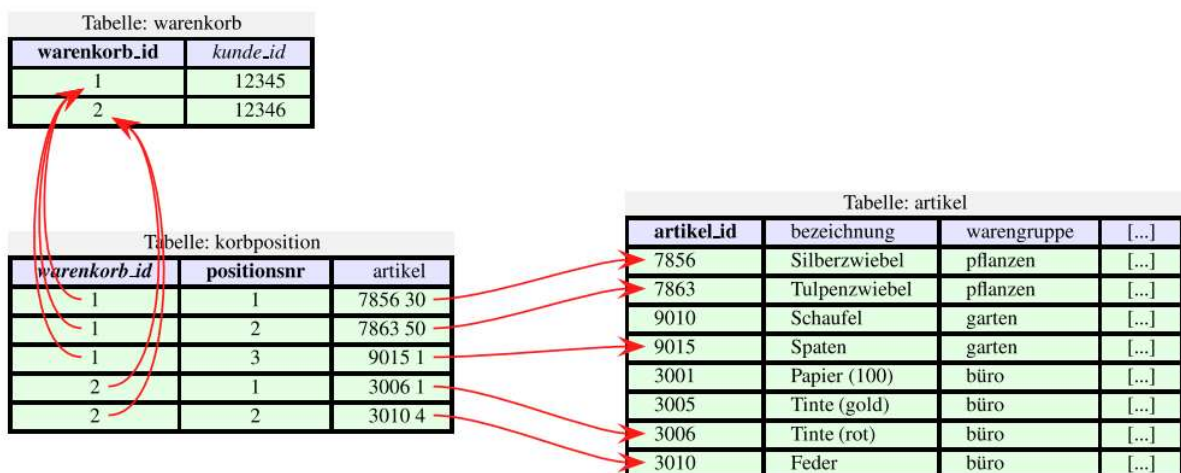
artikel_id	bezeichnung	warengruppe	[...]
7856	Silberzwiebel	pflanzen	[...]
7863	Tulpenzwiebel	pflanzen	[...]
9010	Schaufel	garten	[...]
9015	Spaten	garten	[...]
3001	Papier (100)	büro	[...]
3005	Tinte (gold)	büro	[...]
3006	Tinte (rot)	büro	[...]
3010	Feder	büro	[...]

Frage

Betrachten Sie die Inhalte der Tabelle **warenkorb** und diskutieren Sie die möglichen Nachteile. Gibt es auch Vorteile?

Die Werte der Spalte artikel in der Tabelle warenkorb sind im Grunde Listen, und die Listenelemente bestehen aus zwei Informationen: Artikelnummer und Anzahl. Dies sind zu vermeidende Wiederholungsgruppen: Die Listen sollen aufgelöst und die Teilinformationen in jeweils eigene Spalten überführt werden.

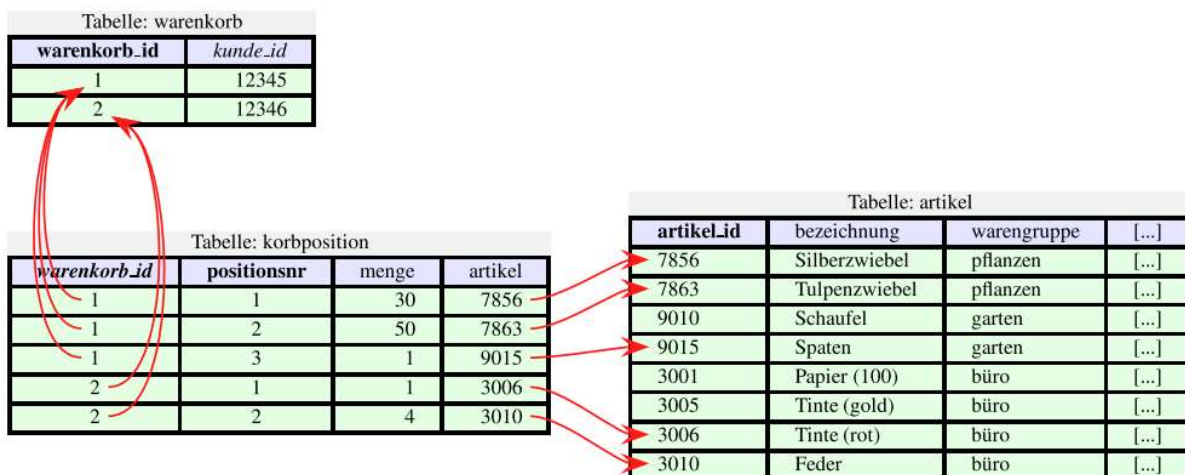
Um die Wiederholungsgruppenfreiheit herzustellen, brauchen wir eine neue Tabelle, die ähnlich der Tabelle position die Artikel aufnimmt.



Frage

Warum ist die Spalte artikel in der Tabelle **korbposition** keine Wiederholungsgruppe?

Was danach noch störend ins Auge fällt, ist die Spalte artikel in der Tabelle korbposition. Die Artikelnummer und die Anzahl sind in der gleichen Spalte abgelegt, was eine Auswertung und Veränderung der Daten erheblich erschwert. Die Spalte ist somit nicht atomar und sie kann noch weiter zerlegt werden. Zuletzt haben die Tabellen folgenden Aufbau.



Erste Normalform

Eine Tabelle ist dann in der 1. Normalform, wenn sie atomar und wiederholungsgruppenfrei ist. Eine Datenbank ist dann in der 1. Normalform, wenn alle Tabellen in der 1. Normalform sind.

4.2 Normalform 2

Volle funktionale Abhängigkeit der Nichtschlüsselattribute

Wir betrachten eine Tabelle tblPositionen, die die Bestellpositionen eines Auftrags zusammenfasst. Ein Auftrag kann mehrere Bestellpositionen besitzen; pro Bestellposition wird ein Artikel definiert. Der Primärschlüssel ist zusammengesetzt aus den Spalten AuftragNr und TeileNr.

TblPositionen				
<u>A_Nr</u>	<u>T_Nr</u>	T_Bezeichnung	A_Menge	P_Preis
99-8419	12088	P400 komplett	3	999,00
99-8420	11000	PC-Tisch	2	399,00
99-8420	11002	Monitor 19``	1	599,00
99-8420	12089	P500 komplett	10	1199,00
99-8421	12089	P500 komplett	5	1199,00
99-8421	23044	Tischfax	2	489,00
99-8422	33002	Toner HP	25	189,00
99-8422	11000	PC-Tisch	10	399,00
99-8422	11002	Monitor 19"	5	599,00

Die 2. NF fordert, dass die Werte alle Nichtschlüsselspalten voll funktional vom Gesamtschlüssel abhängig sind. Dies ist in der obigen Tabelle nicht immer der Fall, da der Artikelname lediglich abhängig von der TeileNr ist. Diese Spalten sind aus der Tabelle herauszulösen und in eigenen Tabellen zu speichern.

TblPositionen			tblArtikel		
<u>A_Nr</u>	<u>T_Nr</u>	A_Menge	<u>T_Nr</u>	T_Bezeichnung	P_Preis
99-8419	12088	3	12088	P400 komplett	999,00
99-8420	11000	2	11000	PC-Tisch	399,00
99-8420	11002	1	11002	Monitor 19"	599,00
99-8420	12089	10	12089	P500 komplett	1199,00
99-8421	12089	5	12089	P500 komplett	1199,00
99-8421	23044	2	23044	Tischfax	489,00
99-8422	33002	25	33002	Toner HP	189,00
99-8422	11000	10			
99-8422	11002	5			

4.3 Normalform 3

Keine transitive Abhängigkeit zwischen Nichtschlüsselspalten

Ermitteln Sie in untenstehender Grafik den Zusammenhang zwischen Bankleitzahl (blz) und dem Banknamen sowie ebenso zwischen der Bankleitzahl, der Kontonummer und der IBAN.

Tabelle: bankverbindung					
kunde_id	bankverbindung_nr	blz	kontonr	bankname	iban
12345	1	50041597	1234506789	Sparkasse Aulenland	DEXX500415971234506789
12345	2	50287667	5432109876	Volksbank Eriador	DEXX502876675432109876
12346	1	50287667	5432109880	Volksbank Eriador	DEXX502876675432109890

Wenn Nichtschlüsselspalten aus anderen Nichtschlüsselspalten herleitbar sind, bedeutet dies in der Regel, dass Informationen redundant in der Tabelle gehalten werden. Hier werden beispielsweise die Banknamen mehrfach genannt. Dies verbraucht nicht nur Speicherplatz, sondern macht eine Änderung der Banknamen teuer, da diese in vielen Zeilen durchgeführt werden müssen.

Tabelle: bankverbindung					Tabelle: bank		
kunde_id	bankverbindung_nr	kontonr	iban	blz	blz	bankname	lkz
12345	1	1234506789	[...]	50041597	50041597	Sparkasse Aulenland	DE
12345	2	5432109876	[...]	50287667	50287667	Volksbank Eriador	DE
12346	1	5432109880	[...]	50287667	50287667	Volksbank Eriador	DE

Transitive Informationen begegnen Ihnen relativ oft. Der Kontoinhaber ergibt sich aus der Kontonummer, der Ortsname aus der Postleitzahl, der Rabatt aus der Kundenart etc.

Aufgabe:

- Finden Sie Beispiele für Einfüge-, Änderungs- und Löschanomalien
- Erläutern Sie den Begriff Wiederholungsgruppe.
- Was bedeutet die Forderung nach Atomarität.
- Diskutieren Sie die Thematik Zusammengesetzter Schlüssel vs. Künstlicher Schlüssel

4.4 Referentielle Integrität

Im Bereich der relationalen Datenbanken wird die referentielle Integrität dazu verwendet die Konsistenz und die Integrität der Daten sicherzustellen. Dazu werden Regeln aufgestellt, wie und unter welchen Bedingungen ein Datensatz in die Datenbank eingetragen wird.

Bei der referentiellen Integrität können Datensätze die einen Fremdschlüssel aufweisen nur dann gespeichert werden, wenn der Wert des Fremdschlüssels einmalig in der referenzier-ten Tabelle existiert. Im Falle, dass ein referenzierter Wert nicht vorhanden ist, kann der Datensatz nicht gespeichert werden.

Warum wird die Referentielle Integrität benötigt?

Eine Datenbank kann schnell in einen inkonsistenten Zustand geraten. Im ungünstigsten Fall liegt eine nicht-normalisierte Datenbank vor, die starke Redundanzen aufweist. Dabei können Anomalien im Datenbestand auftreten, die verschiedene Formen annehmen. Man spricht hier von Einfüge-, Lösch- und Änderungsanomalien. Tritt eine oder mehrerer dieser Anomalien auf, kann das zur Verfälschung oder Löschung von Informationen führen.

Weitergabe

Während die RI grundsätzlich vor inkonsistenten Datenaktionen schützt, bieten viele Datenbanksysteme Zusatzfunktionen an, die bei Updates von Master-Datensätzen nützlich sein können:

Änderungswertweitergabe (ÄW)

Wenn der eindeutige Schlüssel eines Datensatzes geändert wird, kann das DBMS die Fremdschlüssel in allen abhängigen Datensätzen anpassen – anstatt die Änderung abzulehnen. Änderungswertweitergabe wird insbesondere dann benutzt, wenn natürliche Schlüssel (die sich ändern können; Familienname bei Heirat) verwendet werden; denn künstliche Schlüssel sind i. d. R. unveränderlich und eine Änderungswertweitergabe nicht erforderlich.

Löschwertweitergabe (LW)

In bestimmten Fällen ergibt es einen Sinn, abhängige Datensätze bei Löschung des Masterdatensatzes mitzulöschen.

```
create table ...  
... ,  
foreign key (id) references tabelle.pk  
on delete cascade  
on update restrict
```

Diese Funktionen können in der RI-Spezifikation optional gesetzt und (je nach DBMS) durch zusätzliche Bedingungen erweitert/präzisiert werden.