

Aufgabe: Datenbankmodellierung – Bücherei

Aufgabe: (Datenbank, Datenmodellierung und Normalisierung)

Eine städtische Bücherei verwaltet ihre Ausleihvorgänge mit einer Tabelle. Dabei werden sowohl Buch- als auch Kundendaten und die jeweilige Ausleihe in einer einzigen Tabelle gespeichert.

Nicht normalisierte Ausgangstabelle:

AusleiheID	Kundenname	Adresse	BuchID	Titel	Autor	Ausleihdatum
A001	Lisa Becker	Gartenweg 5, Köln	B10	Der Schwarm	Frank Schätzing	2024-02-12
A002	Tim Hoffmann	Brückenstr. 12, Bonn	B11	Homo Deus	Yuval Harari	2024-02-13
A003	Lisa Becker	Gartenweg 5, Köln	B11	Homo Deus	Yuval Harari	2024-02-14

Teilaufgaben

- Analysieren Sie die Tabelle auf Redundanzen und Anomalien. (10 BE)
- Normalisieren Sie die Tabelle bis zur 3. Normalform. Geben Sie die resultierenden Tabellen an (ohne Daten). (25 BE)
- Nennen und erläutern Sie drei konkrete Vorteile, die sich durch die Normalisierung für die Bücherei ergeben. (5 BE)

Lösung zu a) Analyse auf Redundanzen und Anomalien

Die gegebene Tabelle speichert Informationen zu Ausleihen, Kunden und Büchern in einer einzigen Relation. Dabei ergeben sich mehrere Probleme hinsichtlich Redundanzen und Anomalien:

1. Redundanzen:

- Kundendaten (z. B. **Lisa Becker**, **Gartenweg 5, Köln**) tauchen mehrfach auf, wenn ein Kunde mehrere Bücher ausleiht.
- Buchinformationen (**Titel**, **Autor**) wiederholen sich bei mehreren Ausleihen desselben Buchs (z. B. **Homo Deus**, **Yuval Harari**).

2. Anomalien:

- Änderungsanomalie:** Ändert sich z. B. die Adresse eines Kunden, muss diese an mehreren Stellen aktualisiert werden.
- Einfügeanomalie:** Ein neues Buch kann nicht erfasst werden, ohne dass es gleichzeitig ausgeliehen wird.
- Löschanomalie:** Wird ein Ausleihvorgang gelöscht, könnten wichtige Informationen über den Kunden oder das Buch verloren gehen – z. B. wenn es sich um den einzigen Ausleihdatensatz handelt.

3. Verstoß gegen die Normalformen:

- Die Tabelle ist **nicht in 1NF**, da die Adresse als zusammengesetztes Attribut gespeichert ist (Straße, Stadt).
- Sie ist **nicht in 2NF**, da Buchdaten und Kundendaten jeweils nur von einem Teil der zusammengesetzten Schlüsselattribute (z. B. **BuchID** bzw. **Kundenname**) abhängen.

- Die Tabelle ist **nicht in 2NF**, da sie einen zusammengesetzten Primärschlüssel besitzt (z. B. **AusleiheID** oder auch denkbar: Kombination aus **Kundenname** und **BuchID**) und es Attribute gibt, die nur von einem Teil dieses Schlüssels funktional abhängig sind:
 - Kundendaten wie **Kundenname** und **Adresse** hängen nur vom Kunden ab – nicht von der Buch-ID.
 - Buchdaten wie **Titel** und **Autor** hängen nur von der Buch-ID ab – nicht von der konkreten Ausleihe.
- Es liegt also eine **partielle Abhängigkeit** vor, die gegen die Anforderungen der 2NF verstößt.
- **Lösung:** Aufspaltung der Daten in separate Tabellen für Kunden, Bücher und Ausleihen.
- Die **3NF** ist ebenfalls verletzt, da es transitiv abhängige Daten gibt (z. B. **Autor** ist transitiv abhängig von **BuchID** über den **Titel**).

b) Normalisierung bis zur 3. Normalform

Schritt 1: 1. Normalform (1NF)

Alle Attributwerte müssen atomar sein, d. h. nicht weiter zerlegbar. Die Adresse (z. B. „Gartenweg 5, Köln“) ist ein zusammengesetztes Attribut.

⇒ Zerlegung der Adresse in Straße und Stadt.

Neue Tabelle (in 1NF):

- Ausleihe(AusleiheID, Kundenname, Straße, Stadt, BuchID, Titel, Autor, Ausleihdatum)

Schritt 2: 2. Normalform (2NF)

Voraussetzung: Tabelle ist in 1NF. Zudem dürfen alle Nichtschlüsselattribute nur vom gesamten Primärschlüssel abhängen – nicht nur von einem Teil davon.

⇒ Da die Tabelle einen zusammengesetzten Schlüssel enthält (AusleiheID oder ggf. (Kundenname, BuchID, Ausleihdatum)), sind z. B. Titel und Autor funktional nur von BuchID abhängig. Ebenso sind Straße und Stadt nur vom Kunden abhängig.

Zerlegung:

- Kunde(KundenID, Vorname, Name, Straße, Stadt)
- Buch(BuchID, Titel, AutorVorn, AutorNachn)
- Ausleihe(AusleiheID, KundenID, BuchID, Ausleihdatum)

Schritt 3: 3. Normalform (3NF)

Voraussetzung: Tabelle ist in 2NF. Zusätzlich dürfen keine transitiven Abhängigkeiten bestehen, d. h. alle Nichtschlüsselattribute müssen direkt vom Primärschlüssel abhängen.

⇒ In der aktuellen Zerlegung existieren keine transitiven Abhängigkeiten mehr.

Endgültige Tabellenstruktur in 3NF:

- Kunde(KundenID, Vorname, Name, Straße, Stadt)
- Buch(BuchID, Titel, AutorVorn, AutorNachn)
- Ausleihe(AusleiheID, KundenID (FK), BuchID (FK), Ausleihdatum)

Definition der drei Normalformen

1. Normalform (1NF)

Eine Tabelle befindet sich in der **1. Normalform**, wenn alle Attributwerte atomar sind – d. h.:

- Es gibt keine mehrwertigen oder zusammengesetzten Attribute.
- Jeder Tabellenwert enthält genau einen Eintrag (z. B. nicht „Müller, Schmidt“ als Nachname).

Beispiel: Die Adresse sollte nicht in einem einzigen Feld wie „Musterstraße 1, Berlin, 12345“ gespeichert sein, sondern in getrennten Feldern für Straße, Stadt und PLZ.

2. Normalform (2NF)

Eine Tabelle ist in der **2. Normalform**, wenn sie:

- bereits in 1NF ist und
- jedes Nicht-Schlüsselattribut **voll funktional abhängig** vom gesamten Primärschlüssel ist.

Dies bedeutet: Es dürfen keine sogenannten *partiellen Abhängigkeiten* existieren, bei denen ein Nicht-Schlüsselattribut nur von einem Teil eines zusammengesetzten Schlüssels abhängt.

Lösung: Aufspaltung in mehrere Tabellen, sodass jedes Attribut vollständig vom Schlüssel der jeweiligen Tabelle abhängt.

3. Normalform (3NF)

Eine Tabelle ist in der **3. Normalform**, wenn sie:

- bereits in 2NF ist und
- alle Nicht-Schlüsselattribute **direkt** vom Primärschlüssel abhängen, d. h. es gibt keine *transitiven Abhängigkeiten*.

Beispiel für transitive Abhängigkeit: Wenn die Telefonnummer eines Autors vom Namen abhängt und der Name wiederum vom AutorID, dann hängt die Telefonnummer transitiv vom AutorID ab. Diese Abhängigkeit sollte durch eine eigene Tabelle aufgelöst werden.

Ziel: Beseitigung aller indirekten (transitiven) Abhängigkeiten durch weitere Zerlegung.