

# Datenbanken

## Inhalt

Eine Datenbank ist ein strukturiertes System zur Speicherung, Verwaltung und Abfrage von Daten. Typischerweise arbeitet man mit relationalen Datenbanken, bei denen die Daten in Tabellen organisiert sind.

Diese Tabellen stehen oft in Beziehungen zueinander - daher der Name "relationale Datenbank"

## Normalisierung

Normalisierung ist ein Prozess in der Datenbankmodellierung, bei dem man Tabellen in sinnvolle, saubere Strukturen bringt, um:

- Redundanzen zu vermeiden
- Dateinkonsistenzen zu verhindern
- Flexibilität und Wartbarkeit zu erhöhen

Dazu gibt es verschiedene Normalformen, die man Schritt für Schritt durchläuft.

## Normalformen

### Erste Normalform

Alle Attribute sind atomar - aber nicht teilbar.

Beispiel (nicht 1NF):

Kunde	Telefonnummern
Anna	0123,0456

Problem: Mehrere Telefonnummern in einem Feld -> nicht atomar

In die 1NF umgewandelt

Kunde	Telefonnummer
Anna	0123
Anna	0456

## Zweite Normalform

1. Muss in 1NF sein
2. Keine partiellen Abhängigkeiten vom Primärschlüssel (gilt nur bei zusammengesetzten Schlüsseln)

### Beispiel (2NF nicht erfüllt):

Student_ID	Student_Name	Kurs_ID	Kurs_Name	Dozent	Raum
1	Anna Müller	A1	Mathe	Dr. Koch	R101
2	Anna Müller	B2	Physik	Dr. Meyer	R102
3	Ben Schulze	A1	Mathe	Dr. Koch	R101

- Tabelle muss 1NF sein -> erfüllt
- Alle Nicht-Schlüsselattribute müssen voll funktional abhängig vom gesamten Primärschlüssel sein

### Problem:

- Primärschlüssel = (Student\_ID, Kurs\_ID)
- -> Aber Student\_Name hängt nur von Student\_ID ab, nicht vom Kurs!
- -> Verstoß gegen 2NF

### Lösung: Aufteilen in zwei Tabellen

#### Studenten

Student_ID	Student_Name
1	Anna Müller
2	Ben Schulze

#### Kursbelegung

Student_ID	Kurs_ID
1	A1
1	B2
2	A1

#### Kurse

Kurs_ID	Kurs_Name	Dozent	Raum
A1	Mathe	Dr. Koch	R101
B2	Physik	Dr. Meyer	R102

## Dritte Normalform

1. Muss in 2NF sein
2. Keine transitiven Abhängigkeiten (Nicht-Schlüsselattribute dürfen nicht voneinander Abhängen)

In der Kurs-Tabelle: Raum hängt eigentlich vom Dozent ab ab (z.B. Dr. Koch unterrichtet immer in Raum R101) -> transitives Abhängigkeit.

Lösung: Dozenten in eine eigene Tabelle auslagern

### Kurse

Kurs_ID	Kurs_Name	Dozent_ID
A1	Mathe	D1
B2	Physik	D2

### Dozenten

Dozent_ID	Dozent	Raum
D1	Dr. Koch	R101
D2	Dr. Meyer	R102

## Aufgaben

Die Datenbank, in der Angebote und Buchungen gespeichert werden sollen, muss die folgenden Anforderungen erfüllen:

- Jede Wohnung wird nur von einem Anbieter bereitgestellt
- Für eine Wohnung werden zu verschiedenen Buchungszeiten verschiedene Preise verlangt
- Ein Kunde kann mehrere Wohnungen buchen
- Zu jeder Buchung gehören An- und Abreisedatum
- PLZ und Ort der Kunden und Anbieter werden in einer gemeinsamen Tabelle angelegt

Geben Sie ein Datenbankschema in der 3. Normalform an, das die genannten Anforderungen erfüllt. Außer den angegebenen Attributen brauchen Sie nur die Schlüsselattribute sowie den Namen der Kunden und Anbieter anzugeben. Kennzeichnen Sie Primär- und Fremdschlüssel mit PK bzw. FK.

## Wohnung

ID (PK)	Anbieter (FK)

## Anbieter

ID (PK)	Name	Ort (FK)

## Kunde

ID (PK)	Name	Ort (FK)

## Preis

Wohnung (FK)	Wohnung (FK)	Beginn	Ende	Preis

## Buchung

Kunde (FK)	Wohnung (FK)	Abreisedatum	Anreisedatum

## Ort

ID (PK)	PLZ	Ort

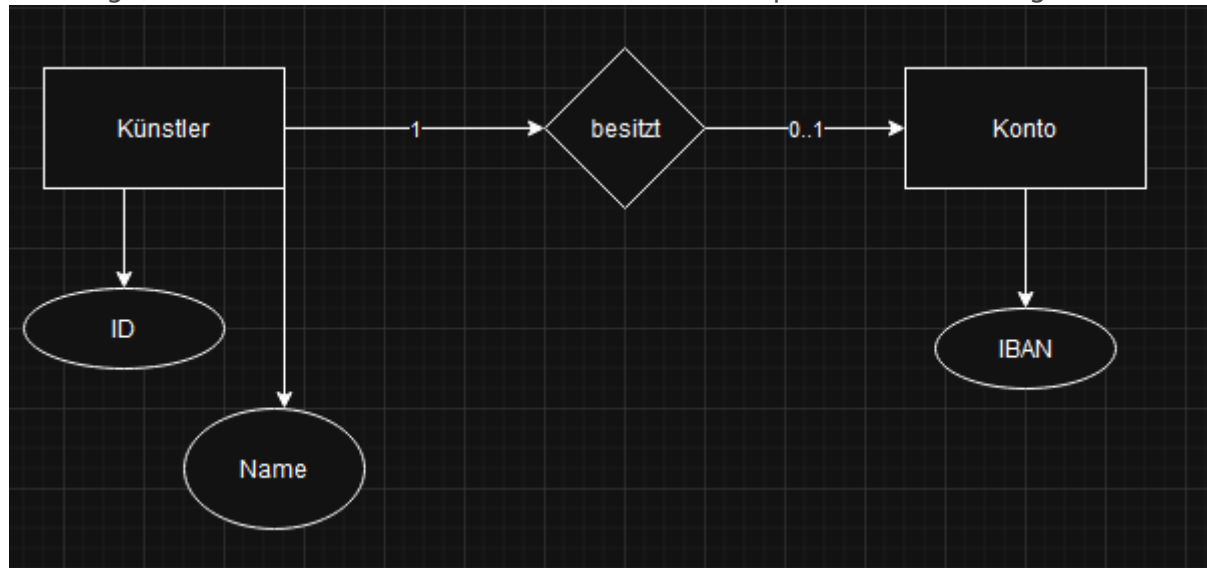
**Begründen Sie, warum die Adresse des Kunden gemäß der 1. Normalform nicht nur als ein einziges Attribut dargestellt werden sollte.**

Die gesamte Adresse in einem Feld wäre nicht Atomar. Dieses Problem löst die erste Normalform. Es sind Eingabefehler bei der Adresse nicht auszuschließen.

**Erklären Sie, warum bei der Implementierung der Datenbank für die An- und Abreise der Datentyp "DATE" anstelle eines Zeichenkettentyps verwendet werden sollte.**

Der Datentyp stellt sicher, dass sich nur Datumswerte im einheitlichen Format innerhalb dieses Feldes befinden. Über DATE ist eine chronologische Sortierfolge definiert, über Zeichenketten eine alphabetische. Dies führt bei Vergleichen und Sortierungen zu unterschiedlichen Ergebnissen.

Zu einigen Kunden enthält die Datenbank bereits entsprechende Zahlungsinformationem:



Eine solche Relation kann in einer relationalen Datenbank wahlweise in einer oder in zwei Tabellen dargestellt werden. Geben Sie beide Formen an.

## 2NF

ID (PK, FK)	Name	IBAN

## 3NF

Konto

ID (PK)	Name	KontoID (FK)

Konto

KundenID(FK)	IBAN

Welche Vor- oder Nachteile hat die Darstellung als getrennte Tabellen bei einer 0...1 Relation im Hinblick auf folgende Aspekte

- Speicherplatz
- Performant
- Datenschutz
- Datensicherheit

Speicherplatz:

- Nachteil: Zusätzliche Tabelle erforderlich.

Performanz:

- Nachteil: Zusätzliche Verknüpfung der Tabellen mit Abfragen erforderlich

Datenschutz:

- Vorteil: Separate Kontotabelle kann mit anderen Zugriffsrechten ausgestattet werden. Dazu dürfte es aber kein FK in der Kundentabelle geben, da dieser Aufschluss über die Kontodaten bieten würde.

Datensicherheit:

- Vor- und Nachteile bei der Datensicherheit: höherer Speicherbedarf auch bei Datensicherungen, aber besserer Zugriffsschutz möglich.

**Was ist bei einer SQL-Anweisung zu beachten, die aus zwei getrennten Tabellen die Namen aller Kunden und, falls vorhanden, deren Konto ermitteln soll?**

Eine solche SQL-Anweisung muss JOIN-Anweisungen (OUTER) enthalten, um die Tabellen miteinander zu Verknüpfen über den Fremdschlüssel

**Kunden wechseln gelegentlich ihre Kontoverbindungen. Geben Sie an, wie sich das ER-Modell ändert, wenn auch frühere Kontoverbindungen weiter gespeichert werden sollen.**

- Die 0...1 Relation ändert sich in eine 1:n Relation
- Konto wird um ein Attribut erweitert, der den Aktiven Datensatz markiert.

-----  
**Zunächst haben nicht alle User der Datenbank dieselben Rechte. Erläutern Sie, welche Zugriffsrechte bei einer Datenbank vergeben werden können**

Es können Lese, Löschen und Schreibberechtigungen, sowie den Zugriff auf bestimmte Tabellen eingeschränkt werden. Gleiches gilt für Prozeduren oder Views.

**Alle Daten werden unter der Berücksichtigung der referentiellen Integrität eingepflegt. Erklären Sie dieses Prinzip und seine Auswirkung auf die Dateneingabe.**

Das Prinzip der referentiellen Integrität verlangt, dass es zu jedem Fremdschlüsselwert, der in eine Datentabelle eingepflegt wird, bereits einen Primärschlüsselwert in der entsprechenden Mastertabelle gibt. Dies führt dazu, dass zunächst die Datensätze der Mastertabelle erstellt werden müssen und anschließend der Detailtabellen.

**Manche Kunden Löschen ihre Konten wieder. Erläutern Sie welche Maßnahmen ergriffen werden müssen, um Löschanomalien zu vermeiden.**

Es müssen alle Datensätze auch aus allen weiteren Tabellen mithilfe deren Primär- oder Fremdschlüsseln gelöscht werden, sodass keine Referenz mehr besteht.

**Auch die verwendeten COMMIT- und REVOKE-Anweisungen tragen zur Sicherheit bei. Erläutern Sie dies.**

COMMIT und REVOKE bewirken die sog. Transaktionssicherheit. Darunter versteht man, dass bestimmte Anweisungen nur zusammen oder gar nicht ausgeführt werden dürfen. Sollte also die Verarbeitung mitten in einer Transaktion abbrechen, müssen die bereits ausgeführten Transaktionen mittels REVOKE wieder rückgängig gemacht werden.

**Alle SQL-Anweisungen sind sorgfältig getestet worden. Beispielweise wurde für die folgende Anweisung, die alle Wohnungen in Italien ermitteln soll, zur Laufzeit als fehlerhaft erkannt:**

```
SELECT * FROM Wohnung WHERE Ort = (SELECT ID FROM Ort WHERE Land="Italien")
```

**Beschreiben Sie den Unterschied zwischen einem Laufzeitfehler und einem syntaktischen Fehler.**

Ein syntaktischer Fehler wird durch den Interpreter bzw. Compiler entdeckt und führt dazu, dass die betroffene Anweisung gar nicht erst ausgeführt werden kann. Bei einem Laufzeitfehler dagegen wird eine Anweisung zunächst ausgeführt, die Ausführung scheitert aber, z.B. weil die vorgefundenen Daten nicht zu den in einer Anweisungen getroffenen Annahmen passen.

**Der Laufzeitfehler tritt nur auf, wenn die Datenbank mehrere italienische Wohnungen enthält. Was bedeutet das für die Entwicklung entsprechender Testfälle?**

Bei der Entwicklung von Testfällen müssen insbesondere Grenzwerte Beachtet werden, z.B. dass es demselben Land kein, nur ein oder mehrere Datensätze vorhanden sind.

**Worin liegt der Fehler?**

Es muss IN statt = verwendet werden.

**Dieselbe Abfrage kann auch folgendermaßen formuliert werden. Warum entsteht in der umformulierten Fassung kein Laufzeitfehler mehr?**

```
SELECT * FROM Wohnung, Ort WHERE Wohnung.Ort = Ort.ID AND Land = "Italien";
```

Hier wird die Spalte Ort korrekt mit der Ortstabelle verknüpft, sodass für jede Wohnung genau ein Datensatz in der Ortstabelle zugeordnet ist.

**Erläutern Sie den Unterschied zwischen einer Stored Procedure und einem Trigger**

Eine Stored Procedure ist eine Prozedur die manuell ausgeführt werden muss, ein Trigger hingegen wird bei jedem Insert, Update oder Delete in eine bestimmte Tabelle ausgeführt.

## **Die ID der stornierten Wohnung wird der Stored Procedure als Parameter übergeben**

### **1. Erläutern Sie den Unterschied zwischen fakultativen und obligatorischen Parametern**

- Fakultative Parameter sind Optional und können bei der Ausführung weggelassen werden je nach Anwendungsfall, Obligatorische Parameter müssen bei jeder Ausführung der Prozedur übergeben werden

### **2. Erläutern Sie den unterschied zwischen Parametern und lokalen Variablen**

- Parameter werden einer Prozedur oder Funktion übergeben, Variablen sind im Quelltext deklariert und können nur im Quellcode selbst bearbeitet werden

---

Revision #3

Created 25 April 2025 04:37:00 by Admin

Updated 25 April 2025 08:52:25 by Admin