

# Einführung in die Datenbanken

Felix Leitl

26. Juni 2024

## Inhaltsverzeichnis

|  |          |
|--|----------|
| <b>Grundlagen</b>  | <b>3</b> |
| Modellierung . . . . .   | 3        |
| Warum Datenbanken . . . . .                                      | 3        |
| Vorteile einer Datenbank . . . . .                               | 3        |
| Nachteile . . . . .  | 3        |
| Begriffe . . . . .   | 4        |
| Datenbank . . . . .  | 4        |
| Datenbank-Management-System . . . . .                            | 4        |
| Datenbanksystem . . . . .  | 4        |
| Datenbankanwendung . . . . .                                     | 4        |
| Datenmodell . . . . .  | 4        |
| Datenbankschema . . . . .  | 4        |
| Nutzdaten . . . . .  | 4        |
| Metadaten . . . . .  | 4        |
| Konzeptionelles Schema . . . . .                                 | 4        |
| Externes Schema . . . . .  | 5        |
| Internes Schema . . . . .  | 5        |
| Phasen des Datenbankentwurfs . . . . .                           | 5        |
| <b>ERM</b>   | <b>5</b> |
| <b>Relationenmodell</b>  | <b>5</b> |
| Bestandteile eines Datenmodells . . . . .                        | 5        |
| Begriffe . . . . .   | 5        |
| Erweiterte Attributdefinition . . . . .                          | 6        |
| Sicherstellung der Referenziellen Integrität . . . . .           | 6        |
| Löschen eines referenzierten Primärschlüssels . . . . .          | 6        |
| Ändern eines referenzierten Primärschlüssels . . . . .           | 6        |
| Integritätsbedingungen . . . . .                                 | 6        |
| „System-enforced Integrity“ . . . . .                            | 6        |
| Benutzerdefinierte oder „globale“ Integritätsbedingung . . . . . | 7        |

|  |           |
|--|-----------|
| <b>Mapping</b>                                       | <b>7</b>  |
| Abbildungskonzepte . . . . .                         | 7         |
| Algorithmus . . . . .                                | 7         |
| Reguläre Entity-Typen . . . . .                      | 7         |
| Schwache Entity-Typen . . . . .                      | 7         |
| M:N-Beziehungen . . . . .                            | 7         |
| N:1-Beziehungen . . . . .                            | 8         |
| 1:1-Beziehungen . . . . .                            | 8         |
| Mehrwertige Attribute . . . . .                      | 8         |
| Mehrstellige Beziehungen . . . . .                   | 8         |
| Generalisierung/Spezialisierung . . . . .            | 8         |
| Kategorien . . . . .                                 | 8         |
| <b>Normalisierung</b>                                | <b>8</b>  |
| Anomalien . . . . .                                  | 8         |
| Funktionale Abhängigkeit $X \rightarrow Y$ . . . . . | 9         |
| Volle Funktionale Abhängigkeit . . . . .             | 9         |
| Normalformen . . . . .                               | 9         |
| Erst Normalform (1NF) . . . . .                      | 9         |
| Zweite Normalform (2NF) . . . . .                    | 9         |
| Dritte Normalform (3NF) . . . . .                    | 9         |
| Boyce-Codd-Normalform (BCNF) . . . . .               | 9         |
| Vierte Normalform (4NF) . . . . .                    | 9         |
| Denormalisierung . . . . .                           | 9         |
| Wann ist eine Denormalisierung angebracht? . . . . . | 9         |
| <b>Relationenalgebra</b>                             | <b>10</b> |
| <b>SQL</b>   | <b>10</b> |
| <b>Multidimensionale Datenmodellierung</b>           | <b>10</b> |
| <b>Schichtenmodell</b>                               | <b>10</b> |
| <b>Transaktionen</b>                                 | <b>10</b> |
| <b>Pufferverwaltung</b>                              | <b>10</b> |

# Grundlagen

## Modellierung

Ein Modell ist ein zweckgerichtetes Abbild der Wirklichkeit

Zweck:

- Spezifizieren
- Konstruieren
- Visualisieren
- Dokumentieren

## Warum Datenbanken

- Große Software-Systeme
- Viele Anwendungen/Benutzer arbeiten mit den gleichen Daten
- Daten sollen auch nach Ende eines Programms verfügbar bleiben
- Daten sollen vor Verlust geschützt werden
- Daten sollen konsistent bleiben

## Vorteile einer Datenbank

- Anwendungsneutralität
- Vermeidung redundanter Daten
- Zentrale Kontrolle der Datenintegrität
- Synchronisation im Mehrnutzerbetrieb
- Fehlertoleranz
- Performance
- Skalierbarkeit
- Verkürzte Entwicklungszeiten für Anwendungen
- Umsetzung von Standards

## Nachteile

- Hohe initiale Kosten
- General purpose software
- Signifikanter Overhead

## **Begriffe**

### **Datenbank**

Eine Datenbank ist eine Sammlung zusammenhängender Daten.

- repräsentiert einen Ausschnitt der realen Welt (Miniwelt)
- Logisch kohärente Sammlung von Daten
- Hat definierten Zweck

### **Datenbank-Management-System**

Sammlung von Programmen zur Verwaltung einer Datenbank

- Erzeugung von DB
- Wartung von DB
- Konsistenter Zugriff auf DB

### **Datenbanksystem**

- DB + DBMS

### **Datenbankanwendung**

- DBS + Anwendungsprogramme

### **Datenmodell**

- Strukturierungsvorschrift für Daten (z.B. Tabellenform)

### **Datenbankschema**

- Beschreibung einer konkreten Datenbank

### **Nutzdaten**

- Eigentliche Datenbank

### **Metadaten**

- Struktur der DB
- Information über Speicherungsstrukturen

### **Konzeptionelles Schema**

- Beschreibt sämtliche Daten auf logischer Ebene
- z.B. Patient (NR. Krankenkasse, Laborwerte)

## Externes Schema

- Beschreibt den für die Anwendung relevanten Teil einer DB auf logischer Ebene
- z.B. für den Arzt: `Patient (Nr., Laborwerte)` und für die Verwaltung: `Patient (Nr., Krankenkasse)`

## Internes Schema

- Beschreibt die interne Speicherungsstrukturen einer Datenbank
- Unsichtbar für Anwendung
- z.B. Index über Attribut `Nr.` von `Patient`

## Phasen des Datenbankentwurfs

- Konzeptioneller Entwurf
  - Abbildung auf Semantisches Datenmodell (z.B. E/R-Modell)
- Logischer Entwurf
  - Abbildung auf Datenmodell

## ERM

Siehe Vorlesungsfolien

## Relationenmodell

### Bestandteile eines Datenmodells

- einfache Datentypen und Konstruktoren für zusammengesetzte Datentypen
- Konsistenzregeln:
  - inhärente Konsistenzregeln:  
gelten für ein Datenmodell per Konvention
  - explizite Konsistenzregeln:  
werden für eine Anwendung im Zuge der Datendefinition festgelegt
- Benennungskonvention für die Bezeichnung von Datenbankelementen

### Begriffe

- Relation: Menge von gleichartig aufgebauten Tupeln
- Tupel: Zeile einer Tabelle
- Kardinalität: Anzahl der Tupel in einer Relation
- Attribut: Spalte einer Tabelle

- Grad: Anzahl der Attribute
- Relationenschema:
  - Beschreibung einer Relation
  - besteht aus Relationennamen (z.B. **Personen**)
  - und einer Menge von Attributen (z.B. {PNr, Vorname, Nachname})
  - Jedes Attribut wird definiert über einen Attributnamen und einen Wertebereich
  - z.B. **Personen** (PRn, Vorname, Nachname)
- Relationales Datenbankschema: Menge von Relationaldatenbankschemata
- Wertebereich: zulässige Attribute
- Superschlüssel: definiert ein Tupel eindeutig
- Schlüsselkandidat: Minimaler Superschlüssel
- Primärschlüssel: Ausgewählter Schlüsselkandidat
- Fremdschlüssel: Attribut, dass mit Primärschlüssel einer Tabelle auf ein bestimmtes Tupel verweist

### **Erweiterte Attributdefinition**

- NOT NULL
- UNIQUE
- PRIMARY KEY

### **Sicherstellung der Referenziellen Integrität**

#### **Löschen eines referenzierten Primärschlüssels**

- RESTRICTED: ablehnen der Operation
- CASCADES: Alle referenzierenden Tupel werden auch gelöscht
- NULLIFIE: Referenzen werden auf NULL gesetzt
- SET DEFAULT

#### **Ändern eines referenzierten Primärschlüssels**

- RESTRICTED
- CASCADES

### **Integritätsbedingungen**

#### **„System-enforced Integrity“**

- Primärschlüsseleigenschaft
- Referenzielle Integrität

### Benutzerdefinierte oder „globale“ Integritätsbedingung

- Bedingungen aus der Anwendungsdomäne, die explizit formuliert werden müssen
- Kontrolliert durch das DBMS
- Operationen, die die Integritätsbedingungen verletzen werden abgelehnt

## Mapping

### Abbildungskonzepte

| ER-Modell                   | Relationenmodell                    |
|-----------------------------|-------------------------------------|
| Entity-Typ                  | „Entity“-Relation                   |
| 1:1- oder 1:N-Beziehungstyp | Fremdschlüssel oder                 |
| M:N-Beziehungstyp           | Beziehungstabelle mit 2 FS          |
| N-ärer Beziehungstyp        | Beziehungstabelle mit N FS          |
| Einfaches Attribut          | Attribut                            |
| Zusammengesetztes Attribut  | Menge von Attributen                |
| Mehrwertiges Attribut       | „Attribut“-Relation mit FS          |
| Wertebereich                | Wertebereich                        |
| Schlüsselattribut           | Schlüsselkandidat → Primärschlüssel |

### Algorithmus

#### Reguläre Entity-Typen

- Erzeuge eine Relation R, die alle einfachen Attribute von E umfasst
  - Bei zusammengesetzten Attributen nur Komponenten als eigenständige Attribute
- Wähle aus Schlüsselkandidaten einen Primärschlüssel
  - zusammengesetzt → Komponenten bilden zusammen den Primärschlüssel
  - Jeder Schlüsselkandidat, außer PS wird UNIQUE & NOT NULL

#### Schwache Entity-Typen

- Erzeuge eine Relation, die alle einfachen Attribute von W umfasst
- Füge als Fremdschlüssel alle PS-Attribute der Owner-Typen ein
- PS wird Kombination aller FSA, zusammen mit partiellem Schlüssel (falls vorhanden)

#### M:N-Beziehungen

- Erzeuge Relation die alle einfachen Attribute von X umfasst
- FS → PSA der beidem Relationen
- PS ist Kombination der FSA

### **N:1-Beziehungen**

- identifiziere die Relation, die dem Entity-Typ E auf der N-Seite des Beziehungstyps entspricht
- Füge den PS des anderen ET als FS in R ein
- Füge alle einfachen Attribute des Beziehungstyps X als Attribute in R ein

### **1:1-Beziehungen**

- Identifiziere Relationen R & S
- Nehme den PS von S bzw. R als FS von R bzw. S auf UNIQUE
- Füge alle einfachen Attribute in R bzw. S ein

### **Mehrwertige Attribute**

- Erzeuge Relation R mit folgenden Attributen:
  - Ein Attribut A, dass dem abzubildenden Attribut A entspricht
  - Den PS K der Relation S, die zu E gehört, als FS auf S
- Der PS der Relation R ist die Kombination von A & K

### **Mehrstellige Beziehungen**

- Erzeuge Relation R, die alle einfachen Attribute von B umfasst
- $FS \rightarrow PS$  aller Relationen
- $PS \rightarrow$  Kombination aller FS

### **Generalisierung/Spezialisierung**

siehe VL

### **Kategorien**

siehe VL

## **Normalisierung**

### **Anomalien**

- Einfüge-Anomalie (ohne hinzufügen von Info B, geht Info A nicht)
- Lösch-Anomalie
- Änderungs-Anomaile



## **Funktionale Abhängigkeit $X \rightarrow Y$**

$Y$  ist funktional abhängig von  $X$ , wenn es keine Tupel geben darf, in denen für gleiche  $X$ -Werte verschiedene  $Y$ -Werte auftreten

Linke Seite der FA wird „Determinante“ genannt

## **Volle Funktionale Abhängigkeit**

$Y$  ist voll funktional abhängig von  $X$ , wenn es keine echte Teilmenge  $Z \subset X$  gibt, für die gilt  $Z \rightarrow Y$

## **Normalformen**

### **Erst Normalform (1NF)**

Eine Relation, die nur atomare Attributwerte besitzt (keine Mengen als Attributwert)

### **Zweite Normalform (2NF)**

Eine Relation, in 1NF & deren Nicht-Schlüsselattribute voll funktional von jedem Schlüsselkandidaten abhängen

### **Dritte Normalform (3NF)**

Eine Relation, deren Nicht-Schlüsselkandidaten nicht transitiv abhängig von einem Schlüsselkandidaten sind

### **Boyce-Codd-Normalform (BCNF)**

Eine Relation, bei welcher jede Determinante einer FA ein Superschlüssel ist

### **Vierte Normalform (4NF)**

Eine Relation  $R$  ist in 4NF, wenn für jede nicht-triviale mehrwertige Abhängigkeit  $X \twoheadrightarrow A \in R$  gilt:  $X$  ist Superschlüssel von  $R$

Eine mehrwerte Abhängigkeit gilt, wenn die Attributwerte von  $C$  nur von  $A$  und nicht von  $B$  abhängig sind

$A \twoheadrightarrow C$  ist trivial, wenn  $C \in A$  oder  $B = \emptyset$

## **Denormalisierung**

Normalisierung kostet Zugriffszeit

### **Wann ist eine Denormalisierung angebracht?**

- Seltene Änderungen
- Viele Joins

Bei weiteren Fragen Anhang VL\_06 lesen

Relationenalgebra

SQL

Multidimensionale Datenmodellierung

Schichtenmodell

Transaktionen

Pufferverwaltung