Datenbanken

Lukas Hertel

23. Januar 2022

Inhaltsverzeichnis

Kapitel 1

Einleitung

1.1 Motivation für DBMS

- Redundanz und Inkonsistenz
  - Daten müssen oft in verschiedenen Kontexten gespeichert sein
- Zugriffsbeschränkung
  - Wenn Daten in unterschiedlichen Dateien liegen ist Verknüpfung schwierig
- Mehrbenutzerbetrieb
- Verlust von Daten
- Integritätsverletzung
  - Bedingungen die an die Daten gestellt werden
- Sicherheit
  - Rechte für unterschiedliche Nutzer
- Entwicklungskosten
  - Dateiverwaltung muss nicht neu implementiert werden

1.2 Datenbankabstraktion

1. Physische Ebene (Daten im Speicher)
2. Logische Ebene (Schema)
3. Sichten (zugeschnittene Teilmenge der Informationen)

Eine Änderung in der physischen Ebene darf die logische Ebene nicht verändern.

1.3 Datenmodelle

Um die Realität in einer Datenbank zu modellieren, muss man erst selbst ein konzeptuelles Schema entwerfen. Danach kann man dieses Schema halbautomatisch in verschiedene festgelegte logische Datenmodelle übersetzen. Das hier verwendete Modell nennt sich relationales Modell und beschreibt Relationen anhand von Tabellen.

Andere Darstellungsmöglichkeiten wären XML, Hierarchische Modelle oder objektorientierte Datenmodelle (...)

Kapitel 2

Datenbankentwurf

Wir werden uns nur mit der konzeptuellen Modellierung befassen.

2.1 Anforderungsanalyse

Rezept zur Erstellung eines Dokuments mit dem Ziel die Anforderungen an das System zu beschreiben. Die gegebenen Muster sollten, falls nötig, angepasst werden.

2.1.1 Objektbeschreibung

Aufbau:

```
<Objektname>
- Anzahl: <Anzahl>
- Attribute
  * <Attribut1>
    - Typ: <Datentyp>
    - Länge: <Länge der Zeichen oder der Ziffern einer Zahl>
    - Wertebereich: <Von...Bis>
    - Anzahl Wiederholungen: <Wie oft ein Objekt ein Attribut hat>
    - Definiertheit: <Wie viele Einträge bekannt sind in Prozent>
    - Identifizierend: <Ja/Nein je nach eindeutigkeit des Attributs>
  * <Attribut2>
  ...
```

2.1.2 Beziehungsbeschreibung

In einer Datenbank werden auch die Beziehungen zwischen Objekten gespeichert. Diese gilt es ebenfalls formal zu beschreiben.

Aufbau:

```
<Beziehung>
- Beteiligte Objekte
  * <Objekt1> als <Rolle>
```

```
    * <Objekt2> als <Rolle>
    ...
  - Attribute der Beziehung
    * <Attribut1>
    * <Attribut2>
    ...
  - Anzahl: <Anzahl> (pro <Zeiteinheit>)
```

2.1.3 Prozessbeschreibung

Die Prozessbeschreibung soll die Verarbeitung von Daten beschreiben, also den operativen Teil der Datenbank. Aufbau:

```
<Prozessname>
- Häufigkeit: <Intervall>
- benötigte Daten
  * <Objekt1>
  * <Objekt2>
  ...
- Priorität: <hoch, niedrig, ...>
- zu verareitende Datenmenge
  * <Anzahl> <Objekt1>
  ...
```

Ein beispielhafter Prozess in einer Universitätsdatenbank ist die halbjährliche Ausstellung von Zeugnissen.

2.2 Konzeptueller Entwurf

2.2.1 Entity-Relationship Modell

Mit dem Entity-Relationship Modell werden **Entites** und die **Beziehungen** zwischen ihnen modelliert. Neben den Entities und den Relationships gibt es auch **Attribute**, die sowohl Entites als auch Relationships haben können.

Beziehungen werden als Rauten dargestellt und Entities als Rechtecke. Attribute werden umkreist. Alle drei werden durch Kanten verbunden. Besitzt eine Entity ein Attribut (oder eine minimale Menge an Attributen), welches die Entity identifiziert, wird es unterstrichen und **Schlüssel** genannt.

2.2.2 Beziehungen

Eine Beziehung ist eine Teilmenge des kartesischen Produkts von Entititypen $R \subseteq E_1 \times E_2 \times ... \times E_n$, wobei n den Grad der Beziehung beschreibt. Ein Element in R nennt man **Instanz** des Beziehungstyps. Die **Funktionalität** einer Beziehung beschreibt wie viele Entities auf eine andere Entity abgebildet werden. Bei einer 1:1 Beziehung wird jedem Entity aus E_1 höchstens ein Entity aus E_2 zugewiesen und umgekehrt. Aus den Funktionalitäten lassen sich partielle Funktionen herleiten, wenn man aus einer Entity eine andere Entity herleiten kann.

Anmerkung: Im Buch wird die 1:N Beziehung partielle Funktion genannt, wie ist das möglich? Sobald ein Entity auf mehrere Entities abbildet kann es sich doch nicht mehr um eine Funktion handeln?

Die Funktionalitäten können auch auf nicht-binäre Beziehungen übertragen werden. Damit können Integritätsbedingungen definiert werden: Partielle Funktionen bilden immer auf ein eindeutiges Entity ab.

Die **(min, max)-Notation** kann Beziehungen ebenfalls beschreiben. Hierbei wird eine obere und untere Schranke festgelegt wie oft eine Entity in einer Beziehung vorkommen darf.