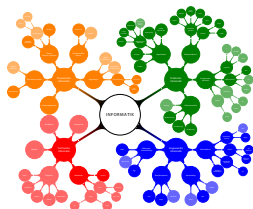


Kapitel 35

Einführung zu Datenbanken

Wie speichert man eine Milliarde Molekülstrukturen
– und wie findet man sie wieder?

Vorlesung Einführung in die Informatik 1 vom 13. Mai 2014 von Till Tantau



Lernziele von Kapitel 35

1. Das Konzept der Datenbank kennen und verstehen
2. Das E/R-Modell kennen
3. Daten mittels des E/R-Modells modellieren können

Gliederung von Kapitel 35

35.1 Datenbanksysteme

35.1.1 Was sind Datenbanken?

35.1.2 Aufbau von Datenbanken

35.1.3 Arten von Datenbanken

35.2 Datenmodelle

35.3 Das E/R-Modell

35.3.1 Einführung

35.3.2 Entitäten

35.3.3 Attribute

35.3.4 Relationships

35.1 Datenbanksysteme ◀

Was sind Datenbanken?
Aufbau von Datenbanken
Arten von Datenbanken

35.2 Datenmodelle

35.3 Das E/R-Modell

Einführung
Entitäten
Attribute
Relationships

- ▶ Die Firma *Molecular Sheep* hat an verschiedenen Standorten Labore, in denen das Genom von Schafen sequenziert wird.
- ▶ Sequenzierte Fragmente müssen *gesammelt* und *weiterverarbeitet* werden.
- ▶ Dabei stellt sich zunächst das Problem, die Daten an eine zentrale Stelle *zu übertragen*.
- ▶ Ein zweites Problem ist der *gleichzeitige Zugriff* verschiedener Personen auf die Daten.

35.1 Datenbanksysteme

- Was sind Datenbanken?
- Aufbau von Datenbanken
- Arten von Datenbanken

35.2 Datenmodelle

35.3 Das E/R-Modell

- Einführung
- Entitäten
- Attribute
- Relationships

Was sind Datenbanken?

Eine *Datenbank* ist eine strukturierte Sammlung von Daten.

- Man kann Daten in eine Datenbank *einfügen*.
- Man kann Daten aus einer Datenbank *löschen*.
- Man kann nach Daten in einer Datenbank *suchen*.

Was sind Datenbanksysteme?

Ein *Datenbanksystem* ist ein Programm, das eine oder mehrere Datenbanken verwaltet.

35.1 Datenbanksysteme

- Was sind Datenbanken?
- Aufbau von Datenbanken
- Arten von Datenbanken

35.2 Datenmodelle

35.3 Das E/R-Modell

- Einführung
- Entitäten
- Attribute
- Relationships

Ein Datenbanksystem kann:

1. Daten aus Datenbanken physikalisch effizient speichern.
2. Zugang zu Daten in Datenbanken herstellen.
 - Man kann Daten in Datenbanken *einfügen*.
 - Man kann Daten in Datenbanken *löschen*.
 - Man kann Daten in Datenbanken *suchen*.
3. Benutzer verwalten.
 - Mehrere Benutzer können *gleichzeitig* auf die Datenbanken zugreifen.
 - Benutzer können verschiedene *Rechte* haben (wie »darf nur Suchen«).

Sollte man eine Datenbank verwenden?

Was für Datenbanken spricht

- + Grundoperationen sind *viel schneller und besser* implementiert als man sie »selbst programmieren könnte«.
- + Daten sind immer *automatisch* auf der Festplatte gesichert.
- + *Mehrere Benutzer* können *gleichzeitig* zugreifen.
- + *Verschiedene Programme* können *gleichzeitig* zugreifen.
- + Datenbanksysteme können *sehr billig* sein.

Was gegen Datenbanken spricht

- Die Grundoperationen *Einfügen*, *Löschen* und *Suchen* kann man auch »selbst programmieren«.
- Man muss *neue Sprachen* lernen (zum Beispiel SQL).
- Datenbanksysteme können *sehr teuer* sein.

35.1 Datenbanksysteme

- Was sind Datenbanken?
- Aufbau von Datenbanken
- Arten von Datenbanken

35.2 Datenmodelle

35.3 Das E/R-Modell

- Einführung
- Entitäten
- Attribute
- Relationships

Schichten, durch die eine Anfrage an eine Datenbank durchläuft. I

35.1 Datenbanksysteme

Was sind Datenbanken?

- Aufbau von Datenbanken
- Arten von Datenbanken

35.2 Datenmodelle

35.3 Das E/R-Modell

Einführung

Entitäten

Attribute

Relationships

1. *Anwendungsprogramme*

Sie stellen Anfragen an die Datenbank mit Hilfe einer speziellen *Anfragesprache*.

2. *Externe Schemata*

Die Anfragen einer Anwendung beziehen sich auf eines von mehreren externen Schemata. Sie »gaukeln einen bestimmten Aufbau der Daten vor«. So kann ein Schema Teile der Datenbank ausblenden, auf die ein Programm keinen Zugriff haben soll.

3. *Konzeptionelles Schema*

Die Anfragen in Bezug auf ein externes Schema werden in Anfragen in Bezug auf das konzeptionelle Schema umgewandelt. Das konzeptionelle Schema beschreibt, wie die Daten logisch strukturiert sind. Dieses Schema ist das zentrale Schema, das man beim Aufbau einer Datenbank zu Anfang festlegen muss.

Schichten, durch die eine Anfrage an eine Datenbank durchläuft. II

35.1 Datenbanksysteme

Was sind Datenbanken?

- Aufbau von Datenbanken
- Arten von Datenbanken

35.2 Datenmodelle

35.3 Das E/R-Modell

Einführung

Entitäten

Attribute

Relationships

4. *Internes Schema*

Anfragen werden weiter verwandelt in Anfragen in Bezug auf ein internes Schema. Es wird vom Datenbanksystem automatisch erstellt und ist eine optimierte, hocheffiziente Verwaltungsstruktur.

5. *Externe Speicher*

Die Anfragen in Bezug auf das interne Schema werden in Zugriffe auf die *Festplatten* umgewandelt.

Welche Schichten gehören zu einer Datenbank?

- ▶ Nur die drei mittleren Schichten gehören zu einer Datenbank und werden vom Datenbanksystem verwaltet.
- ▶ Das *konzeptionelle Schema* wird bei der Erstellung der Datenbank einmal angegeben und dann in der Regel nicht mehr geändert.
- ▶ Die *externen Schemata* werden ebenfalls bei der Erstellung der Datenbank angegeben, können aber oft auch noch später geändert werden.
- ▶ Das *interne Schema* wird automatisch erzeugt und man hat darauf keinen Zugriff.

35.1	Datenbanksysteme
	Was sind Datenbanken?
▶	Aufbau von Datenbanken
	Arten von Datenbanken
35.2	Datenmodelle
35.3	Das E/R-Modell
	Einführung
	Entitäten
	Attribute
	Relationships

Merke

Die Anfragesprachen für Datenbanken erlauben zwei unterschiedliche Dinge:

1. Erstellung und Veränderung der Schemata.
2. Modifikation der Daten, wenn das Schema festgelegt ist.

35.1 Datenbanksysteme

Was sind Datenbanken?

- Aufbau von Datenbanken
- Arten von Datenbanken

35.2 Datenmodelle

35.3 Das E/R-Modell

Einführung

Entitäten

Attribute

Relationships

- Wenn mehrere Anwender gleichzeitig Daten in der Datenbank *suchen*, ist das im Allgemeinen kein Problem.
- Wenn sie aber gleichzeitig Daten *ändern* wollen (schlimmstenfalls sogar die gleichen Daten), so können *vielfältige Konflikte* entstehen.
- *Ein Datenbanksystem kümmert sich um all diese Probleme und löst sie automatisch auf.*

Datenbanksysteme unterscheiden sich unter anderem in folgenden Punkten.

- ▶ Menge der verwaltbaren Daten (von einigen Megabytes bis zu tausenden Terabytes).
- ▶ Anzahl der verwaltbaren Benutzer (von einem einzigen bis zu Millionen).
- ▶ Art der verwaltbaren Daten (Tabellen, Graphiken, Objekte, Filme).
- ▶ Art der verwaltbaren Schemata (relational, hierarchisch, objekt-orientiert).
- ▶ Geschwindigkeit und Sicherheit.
- ▶ Hersteller und Preis.

35.1 Datenbanksysteme

Was sind Datenbanken?

Aufbau von Datenbanken

▶ Arten von Datenbanken

35.2 Datenmodelle

35.3 Das E/R-Modell

Einführung

Entitäten

Attribute

Relationships

Ein fundamentaler Unterschied zwischen Datenbanken ist, wie ihre Schemata aufgebaut sein können:

relational In der Datenbank lassen sich (nur) Relationen (hocheffizient) speichern. Eine Relation setzt verschiedene Dinge in Beziehung, wie zum Beispiel Schafe mit den für sie sequenzierten Fragmenten.

hierarchisch In der Datenbank lassen sich (nur) hierarchische Strukturen speichern.

OO In der Datenbank lassen sich (nur) Objekte im Sinne der objektorientierten Programmierung speichern. Das Schema ist durch die Klassenstruktur gegeben.

Wir betrachten nur relationale Datenbanken.

Wir werden im Folgenden *nur relationale Datenbanken* betrachten.
Dies hat verschiedene Gründe:

- ▶ Diese Datenbanksysteme sind *ausgereift*.
- ▶ Es gibt *frei verfügbare, gute* Implementationen von relationalen Datenbanken.
- ▶ Sie lassen sich *extrem gut optimieren* und sind daher oft *sehr effizient*.
- ▶ Es gibt eine *einheitliche, einfache Anfragesprache* für sie, nämlich *SQL*.

Hier ein paar Nachteile von relationalen Datenbanken:

- ▶ Das relationale Modell passt schlecht zum objektorientierten Modell, das Anwendungsprogramme benutzen.
- ▶ In manchen Situationen sind hierarchische Datenbanken wesentlich schneller.

35.1	Datenbanksysteme
	Was sind Datenbanken?
	Aufbau von Datenbanken
	▶ Arten von Datenbanken
35.2	Datenmodelle
35.3	Das E/R-Modell
	Einführung
	Entitäten
	Attribute
	Relationships

Was sind Datenmodelle?

- ▶ Die verschiedenen Arten von Schemata spiegeln verschiedene Arten von *Datenmodellen* wider.
- ▶ Ein Datenmodell beschreibt *die konzeptionelle Struktur* der Daten.

Beispiel

In der Datenbank von Molecular Sheep sollen identifizierte Schafsgene gespeichert werden. Das Datenmodell legt nun fest:

- ▶ Welche Informationen über ein isoliertes Gen sollen gespeichert werden (Ort, Wissen über die Funktion, Länge)?
- ▶ Welche Informationen über Beziehungen zu anderen Genen sollen gespeichert werden (welche Gene bilden einen Pathway)?
- ▶ Welche Informationen über Beziehungen zu anderen Objekten in der Datenbank sollen gespeichert werden (welche Schafe haben dieses Gen, welche identifizierten Promoter hat das Gen)?

35.1 Datenbanksysteme

Was sind Datenbanken?
Aufbau von Datenbanken
Arten von Datenbanken

35.2 Datenmodelle ◀

35.3 Das E/R-Modell

Einführung
Entitäten
Attribute
Relationships

Es gibt zwei wichtige Arten von Datenmodellen:

- ▶ Entity-Relationship-Modelle.
 - ▶ Sie passen gut zu relationalen Datenbanken, weshalb wir diese betrachten werden.
 - ▶ Wie man vom E/R-Modell zu den Relationen in einer relationalen Datenbank kommt, wird im nächsten Kapitel erklärt.
- ▶ UML-Modelle (unified modelling language)
 - ▶ UML-Modelle passen gut zu objektorientierten Programmen, weshalb sie in der Softwaretechnik viel eingesetzt werden.

- Das *Entity-Relationship-Modell* ist ein *Datenmodell*.
- Es dient dazu, *Entitäten* und deren *Beziehungen* zu beschreiben.
- Ein E/R-Modell legt fest, welche *Arten* von Entitäten es geben kann und welche *Arten* von Beziehungen.
- Es legt noch nicht fest, welche Entitäten und Beziehungen zwischen konkreten Entitäten es gibt.

Ein E/R-Modell besteht aus drei Arten von Dingen:

1. *Entitätstypen*

Dies sind Arten von Objekten oder Dingen, über die Informationen in der Datenbank gespeichert werden sollen.

Beispiel: Schafe, Gene

2. *Attribute*

Dies sind Eigenschaften von Entitäten, die in der Datenbank gespeichert werden sollen.

Beispiel: Farbe des Schafs, Basenposition des Gens

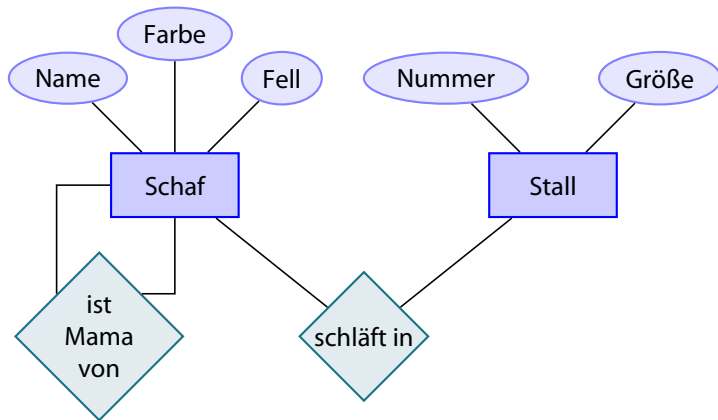
3. *Relationships*

Dies sind Beziehungen, die zwischen Entitäten bestehen.

Beispiel: Schafe haben Gene, Gene wirken zusammen mit anderen Genen

Beispiel: Ein Entity-Relationship-Diagramm.

- Einführung
- Entitäten
- Attribute
- Relationships



- Eine *Entität* ist ein Objekt, über das wir Informationen speichern.
Beispiel: Das Schaf Dolly
- Eine *Entitätsmenge* ist eine Menge von Entitäten. Man spricht aber (salopp und fälschlicherweise) auch oft von *entities*, obwohl man Entitätsmengen meint.
Beispiel: Schafe, schwarze Schafe

Zur Übung

Geben Sie Entitätsmengen an, die Molecular Sheep in seiner Datenbank halten sollte.

- Eine *Attribut* ist eine Eigenschaft von Entitäten.
Beispiel: Die Farbe von Schafen
- Die Menge aller Entitäten, die bestimmte Attribute haben, bilden den *Entitätstyp* dieser Attribute.
Beispiel: Schafe haben eine Farbe, eine Fellart, einen Namen.
- In einem E/R-Modell malt man Entitätstypen als *Rechtecke* auf.
- In einem E/R-Modell malt man Attribute als *Ovale* auf mit einer Kante zum Entitätstyp.

Zur Übung

Geben Sie Attribute für die Entitäten aus der vorherigen Aufgabe an.

Entitätstabellen enthalten Entitäten mit ihren Attributen.

- ▶ Für jeden Entitätstyp gibt es in der Datenbank später eine *Tabelle*.
- ▶ Diese enthält für jede Entität eine *Zeile*.
- ▶ Die *Spalten* sind die Attribute der Entität.
- ▶ Ein Spalte, anhand derer man die Entität eindeutig identifizieren kann, heißt *Schlüsselattribut* oder einfach nur *Schlüssel*.

- 35.1 Datenbanksysteme
 - Was sind Datenbanken?
 - Aufbau von Datenbanken
 - Arten von Datenbanken
- 35.2 Datenmodelle
- 35.3 Das E/R-Modell
 - Einführung
 - Entitäten
 - ▶ Attribute
 - Relationships

Beispiel

Name	Farbe	Fell
Dolly	weiß	lockig
Max	schwarz	wuschelig
Peter	schwarz	wuschelig
Flauschi	beige	glatt

- Eine *Relationship* ist eine Beziehung zwischen zwei oder mehr Entitäten.
Beispiel: Dolly *hat* das Asthma-Gen
- Der *Relationshiptyp* beschreibt, dass Beziehungen zwischen den Entitäten bestimmter Typen bestehen können.
Beispiel: Schafe *haben* Gene
- In einem E/R-Modell malt man Relationshiptypen als *Rauten* auf mit Kanten zu den Entitätstypen.

Zur Übung

Geben Sie Relationships für die Entitäten aus der vorherigen Aufgabe an.

Relationship Tabellen enthalten Tupel von Entitätsschlüsseln.

- Für jeden Relationshiptyp gibt es in der Datenbank später wieder eine *Tabelle*.
- Diese enthält für jede Relationship eine *Zeile*.
- Die *Spalten* sind die Schlüssel der beteiligten Entitätstypen.

Beispiel

Die Tabelle des Relationshiptyps »haben«.

Schafs-Name	Gen-Name
Dolly	Asthma-Gen
Dolly	Intelligenz-Gen
Max	Asthma-Gen
Peter	Intelligenz-Gen

Relationships können auch Attribute haben.

- Auch Relationships können *Attribute* haben.
- Diese werden wie bei Entitäten im Diagramm als Ovale dargestellt.

Beispiel

Die Tabelle des Relationshiptyps »leiden an«.

Schafs-Name	Krankheit	seit
Dolly	Scrapie	1. Juli 2006
Dolly	MKS	2. Juli 2006
Max	MKS	2. Juli 2006
Peter	Scrapie	5. Juli 2006

1. Datenbanken ermöglichen *effizientes* und *sicheres* Einfügen, Löschen und Suchen von Daten.
2. Datenbanken besitzen ein *konzeptionelles Schema*, das vorschreibt, wie die zu speichernden Daten organisiert sind.
3. E/R-Modelle bestehen aus *Entitäten*, *Attributen* und *Relationships*.

- 35.1 Datenbanksysteme
 - Was sind Datenbanken?
 - Aufbau von Datenbanken
 - Arten von Datenbanken
- 35.2 Datenmodelle
- 35.3 Das E/R-Modell
 - Einführung
 - Entitäten
 - Attribute
 - Relationships



H. P. Gumm, M. Sommer.

Einführung in die Informatik, Oldenbourg Verlag, 2004.

Kapitel 10.1, 10.2.1, 10.2.2, 10.2.3