

Informatik am Beispiel von Datenbanken

Stefan Brass

Institut für Informatik
Martin-Luther-Universität Halle-Wittenberg

Inhalt

- 1 Informatik
- 2 Datenbank-Entwurf
- 3 Abfragen
- 4 Implementierung
- 5 Logik
- 6 Studium

Informatik: Einige Richtigstellungen

- Informatik besteht nicht darin, mit Windows, Word und Powerpoint gut umgehen zu können.
- Die meisten Informatiker sind nicht den ganzen Tag mit Programmieren beschäftigt.
- Die meisten Informatiker sind nicht Einzelkämpfer oder absonderliche “Nerds”, sondern arbeiten in einem Team.
- Man kann erfolgreicher Informatiker sein, ohne einen PC zusammenschrauben zu können (oder gar Löten zu können).
- Bessere Informatiker verstehen wirklich, was sie tun, und probieren nicht nur herum.
- Informatik ist für Frauen und Männer gleichermaßen geeignet!

Tipps zur Berufswahl

- Man sollte einen Beruf wählen, indem es ausreichend viele freie Stellen gibt: Informatiker sind sehr gesucht!
- Wer gut ist in Mathematik und Physik, kann auch Informatik.
 - Von der Mathematik braucht man das präzise Denken, Formalisieren und Beweisen, aber eher nicht spezielle Techniken zum Ableiten etc.
 - Von der Physik braucht man, dass man sich technische Zusammenhänge (Maschinen) auch abstrakt vorstellen kann. Erfahrungen mit Lego, Fischertechnik oder einem Elektronik-Baukasten wären nützlich.
- Natürlich soll einem der Beruf auch Spass machen, aber vieles macht Spaß, wenn man sich näher damit beschäftigt.
 - Informatik hat auch viele spannende Anwendungen. Beispiel: Bioinformatik.
- Ein Nutzen für die Menschheit wäre auch gut.
 - Informatik wird überall gebraucht, auch in der Medizin, zur Verbesserung der Sicherheit von Autos, und bei intelligenten Steuerungen zum Energiesparen.

Inhalt

- 1 Informatik
- 2 Datenbank-Entwurf**
- 3 Abfragen
- 4 Implementierung
- 5 Logik
- 6 Studium

Anwendung

- Es soll eine Webseite entwickelt werden,
 - die Informationen über Silvester-Feuerwerksartikel zur Verfügung stellt, und
 - Benutzern erlaubt, diese Artikel zu bewerten.

Ich habe einen Großfeuerwerker-Schein und veranstalte regelmäßig ein Feuerwerk zur Langen Nacht der Wissenschaften. Ich verwende dazu eine selbst entwickelte Planungssoftware, und arbeite mit einem Kollegen an einer computer-gesteuerten Zündanlage. Dies ist für mich persönlich eine spannende Anwendung der Informatik.

- Aus Zeitgründen müssen wir leider vieles weglassen, und uns auf einige wenige Daten von Feuerwerksartikeln beschränken.

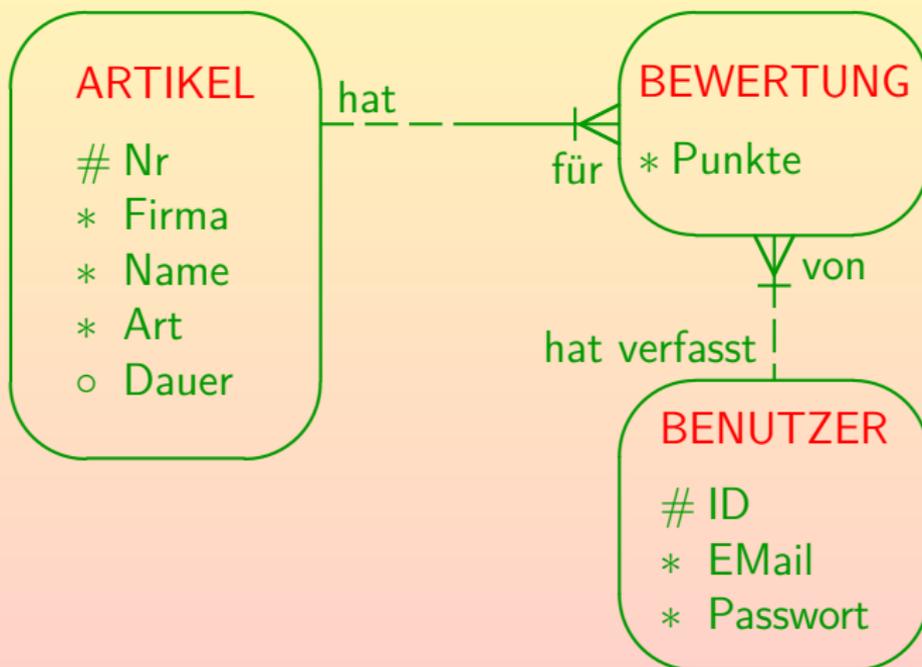
Etwas mehr Informationen und Sicherheitstipps unter:

[http://users.informatik.uni-halle.de/~brass/db12/xf_firew.pdf]

Datenbank-Entwurf

- Es gehört zu den Aufgaben eines Informatikers, mit Anwendungsexperten (“dem Kunden”) zu reden, und herauszufinden, welche Daten gebraucht werden.
- Im Beispiel gibt es drei **Arten von Objekten** der realen Welt, die in der Datenbank abgebildet werden müssen: Feuerwerksartikel, Benutzer, Bewertungen.
- Zwischen diesen Objekten gibt es **Beziehungen**: Eine Bewertung ist für genau einen Feuerwerksartikel, aber ein Feuerwerksartikel kann mehrere (≥ 0) Bewertungen haben.
- Die Objekte haben **Eigenschaften** (Attribute), die gespeichert werden müssen, z.B. haben Benutzer eine EMail-Adresse. Der Wert gehört zu einem Datentyp.

Entity-Relationship-Diagramm (Barker-Notation)



Relationale Datenbanken

- Hier werden die Daten in Form von Tabellen gespeichert:

ARTIKEL				
<u>Nr</u>	Firma	Name	Art	Dauer
1	Weco	Überraschungsvulkan	Vulkan	20
2	Jorge	Malachit	Batterie	25
3	Weco	Saphir	Batterie	70
4	Zink	Supervulkan No. 2	Vulkan	50

BENUTZER		
<u>ID</u>	E-Mail	Passwort
sb	brass@acm.org	...
xyz	xyz@gmx.de	...

BEWERTUNGEN		
<u>Nr</u>	<u>ID</u>	Punkte
1	sb	5
1	xyz	4
2	sb	5
3	xyz	2

Inhalt

- 1 Informatik
- 2 Datenbank-Entwurf
- 3 Abfragen**
- 4 Implementierung
- 5 Logik
- 6 Studium

Datenbank-Abfragen in SQL (1)

- Eine Datenbank ist ein Informationsspeicher:
 - Man kann mit Einfüge-/Lösch- und Änderungsoperationen Fakten darin ablegen, und
 - Diese Fakten müssen der zuvor entwickelten Struktur entsprechen.
 - mit Abfragen Informationen selektiv und ggf. aggregiert wieder herausbekommen.
- Die Kommunikation mit der Datenbank geschieht üblicherweise in der Datenbanksprache SQL.
- Das ist eine spezielle Programmiersprache für Datenbanken.
 - Sie ist recht einfach, und “deklarativ”: Man schreibt nur Bedingungen auf für die Daten, die man haben will, und nicht ein Berechnungsverfahren.

Datenbank-Abfragen in SQL (2)

- Beispiel: Alle Vulkane mit mehr als 30s Brenndauer:

```
SELECT Firma, Name
FROM Artikel
WHERE Art = 'Vulkan' AND Dauer > 30
```

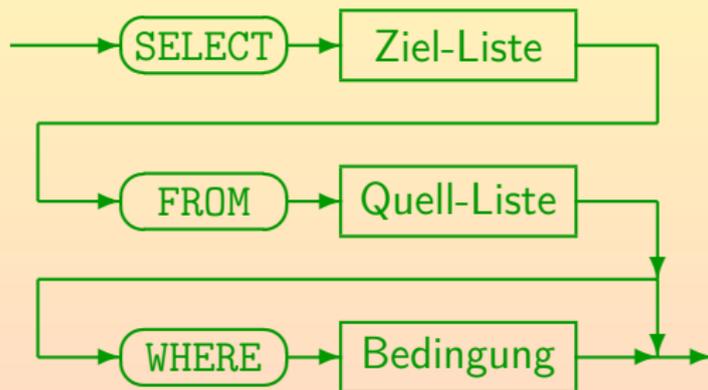
- Ergebnis:

Firma	Name
Zink	Supervulkan No. 2

- Wenn man etwas Englisch kann, kann man dieses "Mini-Programm" schon verstehen.
- Programmieren ist aber nicht wie einen Aufsatz zu schreiben, mehr wie eine mathematische Formel, oder wie das Zusammenstecken von Lego-Bausteinen aus Text.

Syntax-Graphen für SQL (Auszug)

SELECT-Ausdruck (vereinfacht):



Ziel-Liste (vereinfacht):



Inhalt

- 1 Informatik
- 2 Datenbank-Entwurf
- 3 Abfragen
- 4 Implementierung**
- 5 Logik
- 6 Studium

Implementierung eines Datenbank-Systems (1)

- Datenbank-Managementsysteme wie Oracle, MySQL, Microsoft SQL Server sind große Software-Systeme, die Daten verwalten und die Sprache SQL implementieren.
- Sie laufen auf großen Rechnern mit vielen parallelen Prozessoren, riesigen Hauptspeichern, und Schränken voller Platten (und natürlich auch normalen PCs).

Wem es Spass macht, aus großer Hardware die optimale Leistung herauszuholen, wäre in den Entwickler-Teams der großen DBMS-Hersteller richtig.

- Es gibt “TPC-Benchmarks” (Leistungs-Vergleiche), in denen jeder Hersteller gerne gewinnen will.

Im DBMS stecken die cleveren Algorithmen (Berechnungs-Verfahren) und trickreichen Datenstrukturen. Gerade weil die Datenbank-Sprache kein Berechnungs-Verfahren vorschreibt, hat man viele Freiheiten, muss diese für gute Leistung aber auch geschickt nutzen.

Implementierung eines Datenbank-Systems (2)

- Von einer einzelnen Platte kann man heute ungefähr 80 MByte/s sequentiell (am Stück) lesen.
Alternativ ca. 100 Blöcke zu z.B. 8 KByte wahlfrei (über die Platte verstreut).
- Die Daten der Beispiel-Anwendung sind leider eher klein. Wenn die Bewertungen noch einen Text enthalten, kommt man aber schon in einen Bereich, bei dem sequentielles Lesen aller Daten nicht mehr praktikabel ist.
Besonders wenn vor Silvester viele Benutzer über das Web Anfragen stellen.
- Z.B. wurden Index-Strukturen entwickelt, bei denen die Suchbegriffe wie in einem Buch-Index sortiert sind (mit einem Verweis auf die Vorkommen in der Tabelle).
Die Schwierigkeit liegt darin, dass nicht nur Suchen, sondern auch Einfügungen schnell gehen müssen. Außerdem muss das System selbständig über die beste Auswertungsmöglichkeit (z.B. mit/ohne Index) entscheiden.

Inhalt

- 1 Informatik
- 2 Datenbank-Entwurf
- 3 Abfragen
- 4 Implementierung
- 5 Logik**
- 6 Studium

SQL und Logik

- SQL-Anfragen können komplizierter sein als hier gezeigt, z.B. kann man mit Unteranfragen auch “es gibt” und “für alle” Bedingungen ausdrücken. Damit sind SQL-Anfragen praktisch Formeln der mathematischen Logik.
- Man kann (zumindest den SQL-Kern) mathematisch völlig präzise definieren, und auch Aussagen darüber beweisen.
- In der Informatik hat man die ganze Bandbreite:
 - von mathematischer Präzision, Reinheit und Abstraktion
 - über tolle Technik
 - und Design-Fragen (visuell und Programm-Architektur),
 - zu spannenden Anwendungen (mit Kundengesprächen).

Grenzen von Computern

- Es gibt SQL-Anfragen, die inkonsistent sind, d.h. ihr Ergebnis ist immer leer, egal, was der Tabelleninhalt ist, z.B.

```
... WHERE Dauer > 30 AND Dauer < 20
```
- Solche Anfragen sind natürlich sinnlos: Der Programmierer hat einen Fehler gemacht, die Datenbank sollte eine Warnung ausgeben.
- Man kann beweisen, dass kein Programm jemals geschrieben werden kann, das diesen Test für beliebige SQL-Anfragen ausführt (und immer nach endlicher Zeit das korrekte Ergebnis ausgibt).
- Es gibt also prinzipielle Grenzen der Berechenbarkeit.
Dies sichert auch die Arbeit: Man kann natürlich schon versuchen, immer größere Teilmengen von SQL zu prüfen.

Inhalt

- 1 Informatik
- 2 Datenbank-Entwurf
- 3 Abfragen
- 4 Implementierung
- 5 Logik
- 6 Studium**

Informatik-Studium (1)

- Studiengänge, die vom Institut angeboten werden:
 - Informatik: Bachelor (6 Semester) und Master (4 Semester)
 - Bioinformatik: Bachelor und Master
 - Lehramt Informatik
- Außerdem Beteiligung an Wirtschaftsinformatik u.a.
- Studiengänge von AQUIN akkreditiert (“Qualitätssiegel”).
 - Weitere gute Zeichen: Institut ist Mitglied im Fakultätentag Informatik, Studiengänge folgen Empfehlungen der Gesellschaft für Informatik (GI).
- Bachelor Informatik und Bioinformatik haben in Halle keinen “Numerus Clausus”, d.h. man kann mit Abitur ohne Einschränkungen studieren.

Beginn ist jeweils im Wintersemester (ab Oktober). Bewerbungstermin: 30.09.

Informatik-Studium (2)

- Vorlesungen im ersten Semester Informatik:
 - Objektorientierte Programmierung
 - Mathematische Grundlagen der Informatik und Konzepte der Modellierung
 - Einführung in die Rechnerarchitektur
 - Mathematik B (Teil 1: Diskrete Strukturen, lineare Algebra)
 - Allgemeine Schlüsselqualifikationen (wählbar z.B. Englisch, Medien, Rhetorik, wissenschaftliches Schreiben, Wirtschaft)
 - Außerdem ab 3. Semester ein Anwendungsfach, z.B. BWL, VWL, Biologie, Chemie, Physik, Mathematik, Geowissenschaften, Psychologie.
- Bei Bioinformatik statt Rechnerarchitektur und ASQ:
 - Zellbiologie
 - Allgemeine und Grundlagen der physikalischen Chemie

Informatik-Studium (3)

- Es ist nicht Voraussetzung für ein Informatik-Studium, dass Sie Informatik in der Schule hatten, oder schon programmieren können.
- Das Institut veranstaltet immer in den Sommerferien eine "SummerByte"-Woche, in der ein Spiel entwickelt wird, und es mehr Informationen zum Informatik-Studium gibt.

Für Schüler ab 10. Klasse. Bewerbungsschluss Anfang Mai.

Unterstützt von IT-Firmen, Einblick in Praxis. [<http://summerbyte.uni-halle.de/>]

- Mehr Informationen im Web:

[<http://www.informatik.uni-halle.de/studieninteressierte/>]

- Fachstudienberater: Dr. Christoph Bauer

Email: christoph.bauer@informatik.uni-halle.de