

Vorlesung

Logik und Datenbanken

Nicole Schweikardt

Johann Wolfgang Goethe-Universität Frankfurt am Main

“Was” statt “Wie” — am Beispiel von “Tiramisu”

Deklarativ statt Operationell

Tiramisu — Deklarativ

Aus Eigelb, Mascarpone und in Likör und Kaffee getränkten Biskuits hergestellte cremige Süßspeise

(aus: DUDEN, Fremdwörterbuch, 6. Auflage)

Tiramisu — Operationell

1/4 l Milch mit 2 EL Kakao und 2 EL Zucker aufkochen. 1/4 l starken Kaffee und 4 EL Amaretto dazugeben.

5 Eigelb mit 75 g Zucker weißschaumig rühren, dann 500 g Mascarpone dazumischen.

ca 200 g Löffelbiskuit.

Eine Lage Löffelbiskuit in eine Auflaufform legen, mit der Flüssigkeit tränken und mit der Creme überziehen. Dann wieder Löffelbiskuit darauflegen, mit der restlichen Flüssigkeit tränken und mit der restlichen Creme überziehen.

Über Nacht im Kühlschrank durchziehen lassen und vor dem Servieren mit Kakao bestäuben.

(aus: Gisela Schweikardt, handschriftliche Kochrezepte)

Einführung ins Thema

Der große Traum der Informatik

Operationelle Vorgehensweise:

Beschreibung, wie das gewünschte Ergebnis erzeugt wird “Wie”

Deklarative Vorgehensweise:

Beschreibung der Eigenschaften des gewünschten Ergebnisses “Was”

Traum der Informatik:

Möglichst wenig “wie”, möglichst viel “was”

D.h.: Automatische Generierung eines Ergebnisses aus seiner Spezifikation

Realität:

Software-Entwicklung: Generierungs-Tools

Programmiersprachen: Prolog

ABER: Operationeller Ansatz überwiegt immer noch

Datenbanken: Deklarative Anfragesprache ist Industriestandard! (SQL)

Datenbanksysteme

Datenbank (DB)

- ▶ zu speichernde Daten
- ▶ Beschreibung der gespeicherten Daten (Metadaten)

Datenbankmanagementsystem (DBMS)

- ▶ Softwarekomponente zum Zugriff auf die Datenbank
- ▶ Eigenschaften / Kennzeichen:
 - ▶ **Sichere** Verwaltung von Daten: langlebig, große Menge von Daten
... im Sekundärspeicher
 - ▶ **Effizienter** Zugriff auf (große) Datenmengen in der DB

Datenbanksystem (DBS)

- ▶ DB + DBMS

3-Schichten-Modell

Externe Schicht:

Verschiedene Ansichten der Daten

Ansicht 1:

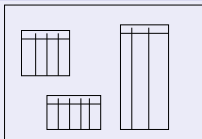
| | |
|---------|------------------|
| Name: | _____ |
| Straße: | _____ |
| PLZ: | _____ Ort: _____ |
| Tel: | _____ Fax: _____ |

Ansicht 2:

| | |
|-------------------|-------|
| Name: | _____ |
| BLZ: | _____ |
| KtoNr: | _____ |
| Kreditkartentyp: | _____ |
| Kreditkarten Nr.: | _____ |

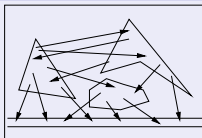
Logische Schicht:

Daten = Tabellen



Physische Schicht:

Datenstrukturen, Speicherorganisation



Wünschenswerte Eigenschaften eines DBS

- ▶ Unterstützung eines Datenmodells: "Darstellung" der Daten für den Zugriff
- ▶ **Bereitstellung einer DB-Sprache:**
 - ▶ zur Datendefinition: Data Definition Language (DDL)
 - ▶ zur Datenmanipulation und zum Datenzugriff: Data Manipulation Language (DML)
- ▶ **Zugangskontrolle:** Wer darf wann auf welche Daten zugreifen bzw. verändern?
- ▶ **Datenintegrität:** Wahrung der Datenkonsistenz und -korrektheit
- ▶ **Robustheit:** Wahrung eines konsistenten Zustands der DB trotz ...
 - ▶ Datenverlusts bei Systemfehlern (CPU Fehler, Plattencrash)
 - ▶ fehlerhafter Beendigung eines DB-Programms oder einer DB-Interaktion
 - ▶ Verletzung der Datenintegrität oder von Zugriffsrechten
- ▶ **Zugriffskoordination bei mehreren DB-Benutzern:** Synchronisation, korrekter Zugriff, korrektes Ergebnis / DB-Zustand
- ▶ **Effizienter Datenzugriff und Datenmanipulation:** schnelle Bearbeitung der Benutzeranfragen

Anfragesprachen

Wünschenswerte Eigenschaften:

Möglichst viel "Was"

Beschreiben der Eigenschaften des gewünschten Ergebnisses (deklarativ)

Möglichst wenig "Wie"

Beschreiben, wie das gewünschte Ergebnis erzeugt werden soll (operationell)

Möglichst unabhängig von den Details der Datenorganisation

Bezug auf logische Schicht oder externe Schicht, nicht auf physische Schicht

Der Preis der Bequemlichkeit und Unabhängigkeit:

- ▶ deklarative Anfragen verschieben die Arbeit vom Benutzer zum System
- ▶ System muss Anfrage in eine Folge von Operationen umwandeln
- ↔ **Gefahr der Ineffizienz**
- ↔ **Geht das überhaupt? Was ist die Auswertungskomplexität?**
- ▶ Andererseits: System hat große Freiheit in der Umsetzung, da kein Lösungsweg vorgeschrieben ist
- ↔ **Potenzial für Optimierung**

Hauptthema dieser Vorlesung: Anfragesprachen

Typische Fragestellungen für diese Vorlesung:

- ▶ Wie lassen sich deklarative Anfragen in ausführbare Operationen umsetzen?
↪ Äquivalenz von "Kalkül" und "Algebra"
- ▶ Welche Anfragen können in einer Anfragesprache gestellt werden, welche nicht?
↪ Ausdrucksstärke von Anfragesprachen
- ▶ Wie aufwendig ist die Auswertung von Anfragen prinzipiell?
↪ Auswertungskomplexität
- ▶ Wie lässt sich eine gegebene Anfrage möglichst effizient auswerten?
↪ Anfrageoptimierung, statische Analyse (Erfüllbarkeit, Äquivalenz, ...)

Inhaltsübersicht (2/2)

5. Konjunktive Anfragen (II)
 - ▶ Der Homomorphismus-Satz
 - ▶ Anfrageminimierung, Statische Analyse
 - ▶ Azyklische Anfragen
 - ▶ Mengen-Semantik vs. Multimengen-Semantik
6. Abhängigkeiten und Normalformen
 - ▶ Funktionale Abhängigkeiten und Join-Abhängigkeiten
 - ▶ The Chase
 - ▶ Normalformen, Informationstheoretischer Ansatz für Normalformen
7. Anfragesprachen mit Rekursion — Datalog
 - ▶ Syntax und Semantik
 - ▶ Auswertung von Datalog-Anfragen, Statische Analyse
 - ▶ Datalog mit Negation
8. Zusammenfassung und Ausblick auf weitere Themen
 - ▶ Datenaustausch und unvollständige Information
 - ▶ Probabilistische Datenbanken
 - ▶ Datenströme

Inhaltsübersicht (1/2)

1. Das Relationale Modell
 - ▶ Datenmodell
 - ▶ Anfragen
 - ▶ Datenkomplexität und kombinierte Komplexität
2. Konjunktive Anfragen (I)
 - ▶ Regelbasierte konjunktive Anfragen
 - ▶ Graphisch: Tableau-Anfragen
 - ▶ Logik-basiert: Konjunktiver Kalkül
 - ▶ Algebraisch: SPC-Algebra und SPJR-Algebra
3. Relationale Algebra
 - ▶ Definition und Beispiele
 - ▶ Anfrageauswertung und Heuristische Optimierung
 - ▶ Das Semijoin-Fragment der Relationalen Algebra
4. Relationenkalkül
 - ▶ Syntax und Semantik
 - ▶ Bereichsunabhängige Anfragen
 - ▶ Äquivalenz zur Relationalen Algebra
 - ▶ Auswertungskomplexität
 - ▶ Statische Analyse

Literatur

- [AHV] Abiteboul, Hull, Vianu: *Foundations of Databases*, Addison-Wesley, 1995
- [M] Maier: *The Theory of Relational Databases*, Computer Science Press, 1983
- [AD] Atzeni, de Antonellis: *Relational Database Theory*, Benjamin Cummings, 1992
- [SSS] Schweikardt, Schwentick, Segoufin: *Database Theory: Query Languages*. Kapitel 19 in *Algorithms and Theory of Computation Handbook*, 2nd Edition, Volume 2: Special Topics and Techniques, Mikhail J. Atallah and Marina Blanton (editors), CRC Press, 2009.

Organisatorisches

Organisatorisches

- ▶ **www-Seite der Vorlesung:**
www.tks.informatik.uni-frankfurt.de/teaching/logdb
- ▶ **Termine:** Di 14-17 (V), Do 14-16 (Ü), Magnus-Hörsaal
- ▶ **Übung:**
 - ▶ Ausgabe der Übungsblätter: Dienstags (in Vorlesung & auf www-Seite)
 - ▶ Abgabe der Lösungen: keine
 - ▶ Besprechung: in der Übungsstunde der darauffolgenden Woche
- ▶ **Voraussetzung für Scheinerwerb bzw Modulabschlussprüfung:**
siehe Webseite der Vorlesung!
- ▶ **Material zur Vorlesung:**
 - ▶ Folien auf www-Seite ... *vieles aber nur an der Tafel!*
 - ▶ Buch *“Foundations of Databases”* von S. Abiteboul, R. Hull, V. Vianu
(erhältlich auf www-Seite der Vorlesung)