Vorlesung
Logik und Datenbanken

Nicole Schweikardt

Johann Wolfgang Goethe-Universität Frankfurt am Main

Sommersemester 2008

Einführung ins Thema

“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)

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

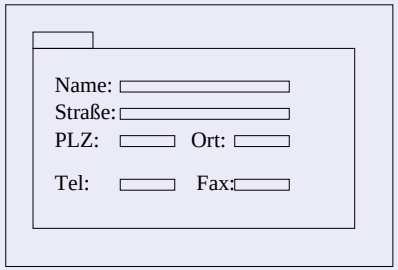
Wünschenswerte Eigenschaften eines DBS

- 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

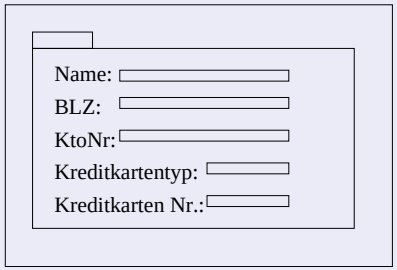
3-Schichten-Modell

Externe Schicht: Verschiedene Ansichten der Daten

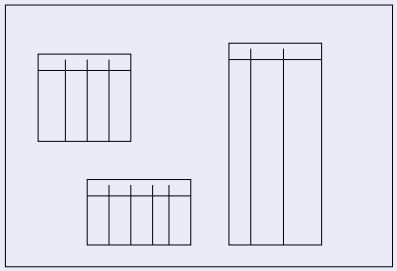
Ansicht 1:



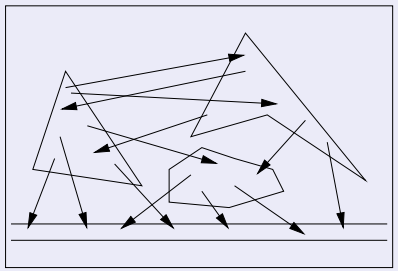
Ansicht 2:



Logische Schicht: Daten = Tabellen



Physische Schicht: Datenstrukturen, Speicherorganisation



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

$\rightsquigarrow$ Gefahr der Ineffizienz

$\rightsquigarrow$ Geht das überhaupt? Was ist die Auswertungskomplexität?

- Andererseits: System hat große Freiheit in der Umsetzung, da kein Lösungsweg vorgeschrieben ist

$\rightsquigarrow$ Potenzial für Optimierung

Hauptthema dieser Vorlesung: Anfragesprachen

Typische Fragestellungen für diese Vorlesung:

- Wie lassen sich deklarative Anfragen in ausführbare Operationen umsetzen?

$\rightsquigarrow$ Äquivalenz von “Kalkül” und “Algebra”

- Welche Anfragen können in einer Anfragesprache gestellt werden, welche nicht?

$\rightsquigarrow$ Ausdrucksstärke von Anfragesprachen

- Wie aufwendig ist die Auswertung von Anfragen prinzipiell?

$\rightsquigarrow$ Auswertungskomplexität

- Wie lässt sich eine gegebene Anfrage möglichst effizient auswerten?

$\rightsquigarrow$ Anfrageoptimierung, statische Analyse (Erfüllbarkeit, Äquivalenz, . . .)

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

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

Literatur

[AHV] Abiteboul, Hull, Vianu: *Foundations of Databases*, Addison-Wesley, 1995

[AD] Atzeni, de Antonellis: *Relational Database Theory*, Benjamin Cummings, 1992

[M] Maier: *The Theory of Relational Databases*, Computer Science Press, 1983

[ABS] Abiteboul, Buneman, Suciu: *Data on the Web*, Morgan Kaufmann Publishers, 2000

Organisatorisches

Organisatorisches

- **www-Seite der Vorlesung:**
  www.informatik.uni-frankfurt.de/~tkshp/lehre/SS08/LD/
- **Termine:** Di 12-14 (V), Di 14-16 (Ü), Do 12-14 (V)
  im Magnus-Hörsaal
- **Übung:**
  - Ausgabe der Übungsblätter: Donnerstags (in Vorlesung & auf www-Seite)
  - Abgabe der Lösungen: 1 Woche später (zu Beginn der Vorlesung)
  - Besprechung: in der darauffolgenden Übungsstunde (Dienstags)
  - keine Lösung ohne Begründung und Erklärung!
- **Voraussetzung für Zulassung zur Prüfung:** $\geqslant 40\%$ der Punkte
- **Material zur Vorlesung:**
  - Folien auf www-Seite *. . . vieles aber nur auf Tafel!*
  - Vorlesungsskript: auf www-Seite
  - Buch *"Foundations of Databases"* von S. Abiteboul, R. Hull, V. Vianu
    (erhältlich auf www-Seite der Vorlesung)

Das Relationale Modell

Das Relationale Modell

Datenmodell

Begriff “Datenmodell”:

- Rahmen zur Repräsentation / Speicherung von Daten
- Operationen zum Zugriff auf Daten
- Mechanismen zur Beschreibung von erwünschten Eigenschaften (Integritätsbedingugen)

Der Begriff “Datenmodell” ist nicht präzise definiert (im mathematischen Sinn).

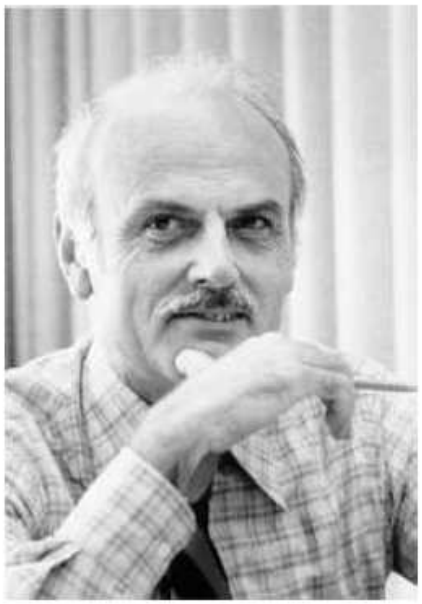
Im Folgenden wird eine präzise Definition des “Relationalen Modells” gegeben.

Das Relationale Modell

- Daten werden in Relationen (“Tabellen”) organisiert
- Mengen-orientierte Operationen
- Deklarative Anfragespezifikation
- Effiziente Anfragebearbeitung
- 1970 eingeführt von Edgar F. Codd
- seit Ende der 80er Jahre “Industriestandard”

Beispiel-Relation:

A	B	C	D
a	1	z	9
b	2	z	8
c	1	y	8
d	4	x	7



Edgar F. Codd (1923-2003)

Grundbegriff: Relationsschema

Beispiel-Relation:

A	B	C	D
a	1	z	9
b	2	z	8
c	1	y	8
d	4	x	7

ist vom Schema R mit $sort(R) = \{A, B, C, D\}$, $arity(R) = 4$

Wir legen ein für alle Mal fest:

- Eine abzählbar unendliche Mengen **att** von Attribut-Namen. Diese Menge sei geordnet via $\leqslant_{\mathbf{att}}$.
- Eine abzählbar unendliche Menge **dom** von "potentiellen Datenbankeinträgen" ("Konstanten") (**dom** steht für "Domain")
- Eine abzählbar unendliche Menge **relname** von Relations-Namen. Die Mengen **att**, **dom**, **relname** seien disjunkt.

- Eine Funktion $sort : \mathbf{relname} \to \mathcal{P}^{\text{fin}}(\mathbf{att})$, die jedem Relations-Namen eine endliche Menge von Attribut-Namen zuordnet . . . und zwar so, dass f.a. $U \in \mathcal{P}^{\text{fin}}(\mathbf{att})$ gilt: $sort^{-1}(U)$ ist unendlich.
- Die Stelligkeit eines Relations-Namens R ist $arity(R) := |sort(R)|$.
- Ein Relationsschema ist einfach ein Relations-Name R .
- Manchmal schreiben kurz $R[U]$ für $sort(R) = U$ und $R[k]$ für $arity(R) = k$.

Relationsschema vs. Relation

Beispiel-Relation:

A	B	C	D
a	1	z	9
b	2	z	8
c	1	y	8
d	4	x	7

ist vom Schema R mit $sort(R) = \{A, B, C, D\}$, $arity(R) = 4$

Relation $\hat{=}$ Tabelle

Tupel $\hat{=}$ Zeile in der Tabelle

Schreibweise: $t.A$ an Stelle von $t(A)$
für "Eintrag in Zeile t und Spalte A"

Beachte: Mengensemantik, d.h.:
Relation $\hat{=}$ die Menge aller Tabellenzeilen

Definition 1.1

Sei R ein Relationsschema.

- Ein R-Tupel ist eine Abbildung $t : sort(R) \to \mathbf{dom}$.
- Eine R-Relation ist eine endliche Menge von R-Tupeln.
- $inst(R)$ bezeichnet die Menge aller Relationen über R.
  (*inst* steht für "instances")

Grundbegriffe: Datenbankschema und Datenbank

Definition 1.2

- Ein Datenbankschema **R** ist eine endliche, nicht-leere Menge von Relationsschemata.

  Manchmal schreiben wir $\mathbf{R} = \{R_1[U_1], \ldots, R_n[U_n]\}$ um die Relationsschemata anzugeben, die zu **R** gehören.

- Eine Datenbank **I** vom Schema **R** ist eine Funktion, die jedem Relationsschema $R_i \in \mathbf{R}$ eine R_i-Relation zuordnet.

- $inst(\mathbf{R})$ bezeichnet die Menge aller Datenbanken vom Schema **R**.

- $schema(\mathbf{I}) := \mathbf{R}$ bezeichnet das Schema der Datenbank **I**.

Beispieldatenbank mit Kinodaten (1/5)

Zur Illustration von Anfragen verwenden wir eine kleine Datenbank mit Kinodaten, bestehend aus

- einer Relation *Orte*, die Informationen über Kinos (Kino, Adresse, Telefonnummer) enthält.
- einer Relation *Filme*, die Informationen über Filme enthält (Titel, Regie, Schauspieler)
- einer Relation *Programm*, die Informationen zum aktuellen Kinoprogramm enthält (Kino, Titel, Zeit)

Beispieldatenbank mit Kinodaten (2/5)

Orte-Relation:

Kino	Adresse	Telefon
Babylon	Dresdner Str. 2	61609693
Casablanca	Friedenstr. 12	6775752
Cinestar Cubix Alexanderplatz	Rathausstr. 1	2576110
Die Kurbel	Giesebrechtstr. 4	88915998
Filmpalast Berlin	Kurfürstendamm 225	8838551
International	Karl-Marx-Allee 33	24756011
Kino in der Kulturbrauerei	Schönhauser Allee 36	44354422
Moviemento	Kottbusser Damm 22	6924785
⋮	⋮	⋮

Beispieldatenbank mit Kinodaten (3/5)

Filme-Relation:

Titel	Regie	Schauspieler
Capote	Bennet Miller	Philip Seymour Hoffman
Capote	Bennet Miller	Catherine Keener
Das Leben der Anderen	F. Henkel von Donnersmarck	Martina Gedeck
Das Leben der Anderen	F. Henkel von Donnersmarck	Ulrich Tukur
Der ewige Gärtner	Fernando Meirelles	Ralph Fiennes
Der ewige Gärtner	Fernando Meirelles	Rachel Weisz
Good Night and Good Luck	George Clooney	David Strathairn
Good Night and Good Luck	George Clooney	Patricia Clarkson
Knallhart	Detlev Buck	Jenny Elvers
Knallhart	Detlev Buck	Jan Henrik Stahlberg
Raumpatrouille Orion – Rücksturz ins Kino	Michael Braun	Dietmar Schönherr
Raumpatrouille Orion – Rücksturz ins Kino	Theo Mezger	Dietmar Schönherr
Raumpatrouille Orion – Rücksturz ins Kino	Michael Braun	Eva Pflug
Raumpatrouille Orion – Rücksturz ins Kino	Theo Mezger	Eva Pflug
Raumpatrouille Orion – Rücksturz ins Kino	Michael Braun	Wolfgang Völz
Raumpatrouille Orion – Rücksturz ins Kino	Theo Mezger	Wolfgang Völz
Requiem	Hans-Christian Schmid	Sandra Hüller
Sommer vorm Balkon	Andreas Dresen	Nadja Uhl
Sommer vorm Balkon	Andreas Dresen	Inka Friedrich
Sommer vorm Balkon	Andreas Dresen	Andreas Schmidt
Syriana	Stephen Gaghan	George Clooney
Syriana	Stephen Gaghan	Matt Damon
V wie Vendetta	James McTeigue	Natalie Portman
Walk the Line	James Mangold	Joaquin Phoenix
Walk the Line	James Mangold	Reese Witherspoon

Beispieldatenbank mit Kinodaten (4/5)

Programm-Relation:

Kino	Titel	Zeit
Babylon	Capote	17:00
Babylon	Capote	19:30
Kino in der Kulturbrauerei	Capote	17:30
Kino in der Kulturbrauerei	Capote	20:15
International	Das Leben der Anderen	14:30
International	Das Leben der Anderen	17:30
International	Das Leben der Anderen	20:30
Filmpalast Berlin	Good Night and Good Luck	15:30
Filmpalast Berlin	Good Night and Good Luck	17:45
Filmpalast Berlin	Good Night and Good Luck	20:00
Kino in der Kulturbrauerei	Good Night and Good Luck	18:00
Kino in der Kulturbrauerei	Good Night and Good Luck	20:30
Kino in der Kulturbrauerei	Good Night and Good Luck	22:45
Babylon	Sommer vorm Balkon	21:45
Kino in der Kulturbrauerei	Sommer vorm Balkon	21:45
Filmmuseum Potsdam	Raumpatrouille Orion – Rücksturz ins Kino	22:00
⋮	⋮	⋮

Beispieldatenbank mit Kinodaten (5/5)

Datenbankschema der Kinodatenbank:

- Datenbankschema **KINO** $= \{Filme, Orte, Programm\}$
- $sort(Filme) = \{\text{Titel}, \text{Regie}, \text{Schauspieler}\}$
- $sort(Orte) = \{\text{Kino}, \text{Adresse}, \text{Telefon}\}$
- $sort(Programm) = \{\text{Kino}, \text{Titel}, \text{Zeit}\}$

Wir schreiben $\mathbf{I_{KINO}}$, um unsere konkrete Datenbank vom Schema **KINO** zu bezeichnen. Analog schreiben wir I_{Filme}, I_{Orte} und $I_{Programm}$ für die konkreten Relationen, die zur Datenbank $\mathbf{I_{KINO}}$ gehören.

Attribute: Benannte vs. Unbenannte Perspektive

Sind die Attribut-Namen Teil des expliziten Datenbankschemas?

In SQL: ja! Beispiel:

```
SELECT Titel FROM Filme WHERE Schauspieler="Natalie Portman"
```

Aber werden die Namen vom System nicht "weg-compiliert"?

- Benannte Perspektive:
  Ein Tupel über Relationsschema $R[U]$ ist eine Abbildung von U nach **dom**.
  Schreibweise: $t = \langle \mathrm{A} : \mathrm{a}, \mathrm{B} : 1, \mathrm{C} : \mathrm{z}, \mathrm{D} : 9 \rangle$
- Unbenannte Perspektive:
  Ein Tupel über Relationsschema $R[k]$ ist ein Element aus $\mathbf{dom}^k$ (Cartesisches Produkt aus k Kopien von **dom**).
  Schreibweise: $t = \langle \mathrm{a}, 1, \mathrm{z}, 9 \rangle$

Beispiel-Relation:

A	B	C	D
a	1	z	9
b	2	z	8
c	1	y	8
d	4	x	7

ist vom Schema R mit $sort(R) = \{A, B, C, D\}$, $arity(R) = 4$

Notation

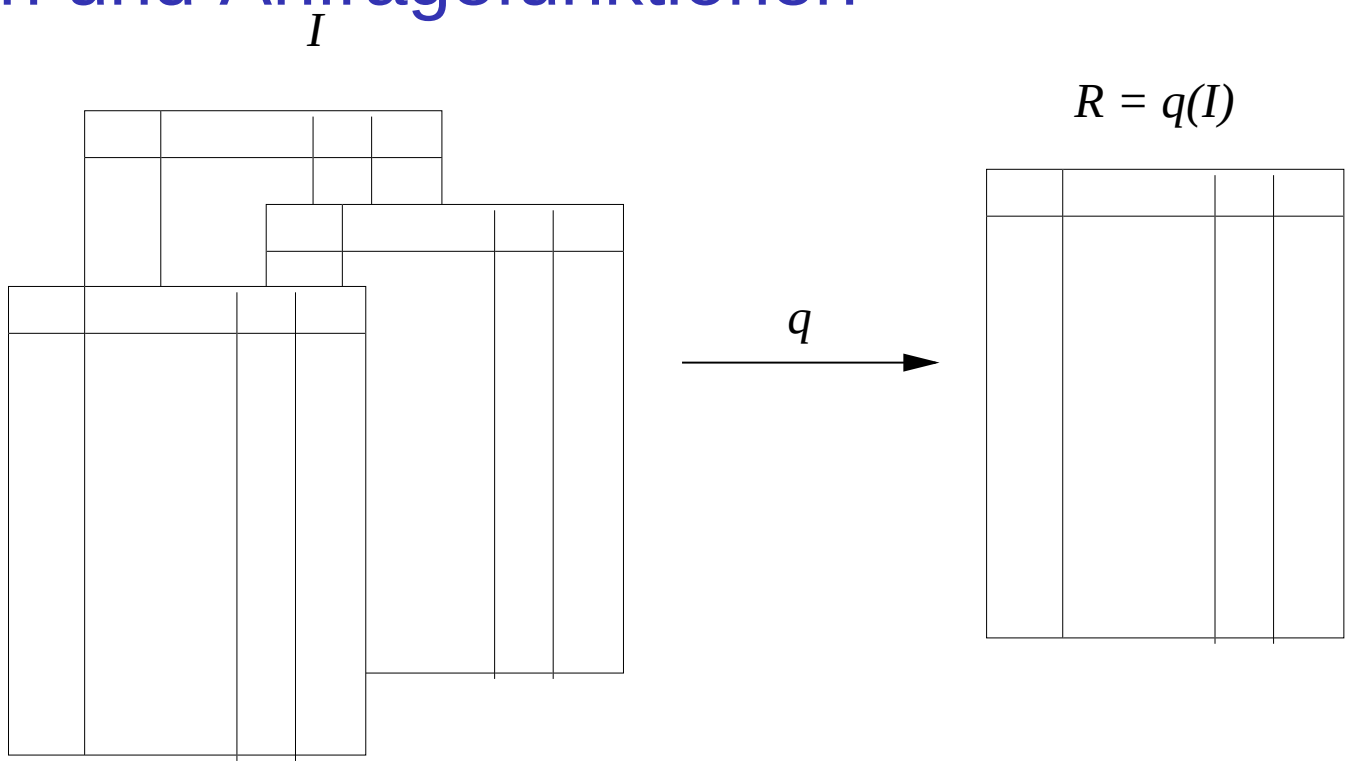
Konstanten (Elemente aus **dom**)	a, b, c, “George Clooney”, . . .
Attribute	A, B, C, . . .
Mengen von Attributen	U, V, . . .
Relations-Namen (-schemata)	R, S, $R[U]$, $S[V]$, . . .
Datenbankschemata	$\mathbf{R}$, $\mathbf{S}$
Tupel	t, s
Relationen (Relations-Instanzen)	I, J
Datenbanken (Datenbank-Instanzen) . . .	$\mathbf{I}$, $\mathbf{J}$

Das Relationale Modell

Beispiel-Anfragen

(1) Wer führt Regie in “Knallhart”?
(2) Wo läuft “Capote”?
(3) Welches sind Adresse und Telefonnummer des “Kino in der Kulturbrauerei”?
(4) Welche Kinos spielen einen Film mit “George Clooney”?
(5) Läuft zur Zeit ein “Detlev Buck” Film?
(6) Welche (je 2) Schauspieler haben schon in Filmen gespielt, in denen der jeweils andere Regie geführt hat?
(7) Welche Regisseure haben in einem ihrer eigenen Filme mitgespielt?
(8) Gib die 2-Tupel von Schauspielern an, die gemeinsam in einem Film gespielt haben!
(9) Egal für welche Datenbank, gib $\langle$“Knallhart”, “Detlev Buck”$\rangle$ als Antwort aus!
(10) Wo wird “Syriana” oder “Walk the Line” gespielt?
(11) Welche Filme laufen in mindestens 2 Kinos?
(12) In welchen Filmen spielt “George Clooney” mit oder führt Regie?
(13) Gib alle Filme aus, die im “International” laufen oder in denen “Andreas Dresen” Regie führt!
(14) Liste alle Schauspieler und den Regisseur von “Capote” auf!
(15) Welche Filme laufen nur zu 1 Uhrzeit?
(16) In welchen Filmen führt “George Clooney” mit, ohne Regie zu führen?
(17) Welche Filme haben nur Schauspieler, die schon mal in einem Film von “Stephen Spielberg” mitgespielt haben?

Anfragen und Anfragefunktionen



Definition 1.3

- Eine Anfragefunktion ist eine Abbildung q, die, für ein Datenbankschema **R** und ein Relationsschema R, jeder Datenbank **I** vom Schema **R**, eine Relation, $q(\mathbf{I})$ vom Schema R zuordnet. (q steht für “query”)
- Eine Anfrage ist eine Zeichenkette, die eine Anfragefunktion in einer bestimmten Syntax beschreibt.

Wünschenswerte Eigenschaften von Anfragefunktionen

Forderungen an eine Anfragefunktion q:

(a) Das Ergebnis sollte nicht von Details der Speicherung abhängen, sondern nur von der logischen Sicht auf die Daten

q ist eine Funktion $inst(\mathbf{R}) \rightarrow inst(R)$,
für ein DB-Schema $\mathbf{R}$ und ein Rel.schema R

(b) Sie sollte berechenbar sein.

q berechenbar

(c) Das Ergebnis sollte möglichst wenig von den einzelnen Datenwerten und möglichst viel von den Beziehungen der Daten untereinander abhängen

... siehe nächste Folie; Stichwort: “generisch”

Erläuterungen zu Eigenschaft (c)

Beispiel-Anfrage:

(3) Welches sind Adresse und Telefonnummer des “Kino in der Kulturbrauerei”?

Eigenschaft (c):

- Wenn sich die Telefonnummer vom “Kino in der Kulturbrauerei” in der DB ändert, soll sich das Ergebnis der Anfrage entsprechend ändern.
- Aber wenn der Name “Kino in der Kulturbrauerei” sich in der DB ändert, soll das Ergebnis leer sein.
- Allgemein:
  - Werden Elemente der Datenmenge **dom**, die in der Anfrage nicht explizit (als “Konstanten”) vorkommen, umbenannt, so sollen sie im Ergebnis auch umbenannt werden.
  - Dies gilt nicht für Elemente in **dom**, die in der Anfrage explizit vorkommen.
- mathematische Präzisierung: Begriff der generischen Anfragefunktion

Generische Anfragefunktionen

Definition 1.4

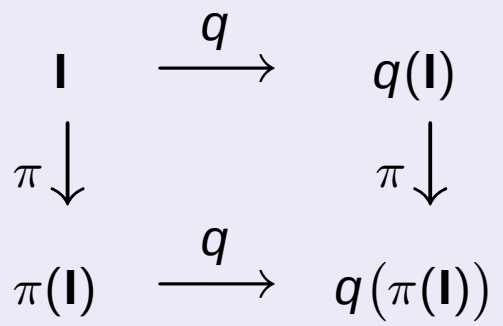
Sei C eine endliche Menge von Datenwerten (also $C \subseteq^{\text{fin}} \mathbf{dom}$).

Eine Anfragefunktion q heißt C-generisch, falls für jede Datenbank $\mathbf{I}$ (vom zu q passenden DB-Schema) und jede Permutation π von **dom** mit $\pi|_C = \text{id}$ (d.h. $\pi(x) = x$ für alle $x \in C$) gilt:

$$q(\pi(\mathbf{I})) \quad = \quad \pi(q(\mathbf{I})).$$

q heißt generisch, falls q $\emptyset$-generisch ist.

Illustration:



Beispiel:

(3) Welches sind Adresse und Telefonnummer des “Kino in der Kulturbrauerei”?
ist {“Kino in der Kulturbrauerei”}-generisch.

(7) Welche Regisseure haben in einem ihrer eigenen Filme mitgespielt?
ist generisch.

Boolesche Anfragen

Manche Anfrage lassen sich nur mit “ja” oder “nein” beantworten.

Beispiel: (5) Läuft zur Zeit ein “Detlev Buck” Film?

Konvention:

- Ergebnis ist eine 0-stellige Relation.
- Davon gibt es genau zwei Stück: $\emptyset$ und $\{\langle\rangle\}$.

  $\langle\rangle$ steht für das “Tupel der Stelligkeit 0”.
- Vereinbarung:
  - $\emptyset$ steht für “nein”
  - $\{\langle\rangle\}$ steht für “ja”

Anfragesprachen

Dieselbe Anfragefunktion kann in verschiedenen Anfragesprachen beschrieben werden.

Beispiel: (4) Welche Kinos spielen einen Film mit “George Clooney”?

- SQL:

```
SELECT   Orte.Kino, Orte.Adresse
FROM     Filme, Programm, Orte
WHERE    Filme.Schauspieler = “George Clooney” AND
         Filme.Titel = Programm.Titel AND
         Programm.Kino = Orte.Kino
```

- Regelbasierte Anfrage:

$$Ans(x_{Kino}, x_{Adr}) \leftarrow Filme(x_{Titel}, x_{Regie}, \text{“George Clooney”}), \; Programm(x_{Kino}, x_{Titel}, x_{Zeit}), \; Orte(x_{Kino}, x_{Adr}, x_{Tel})$$

- Relationenkalkül:

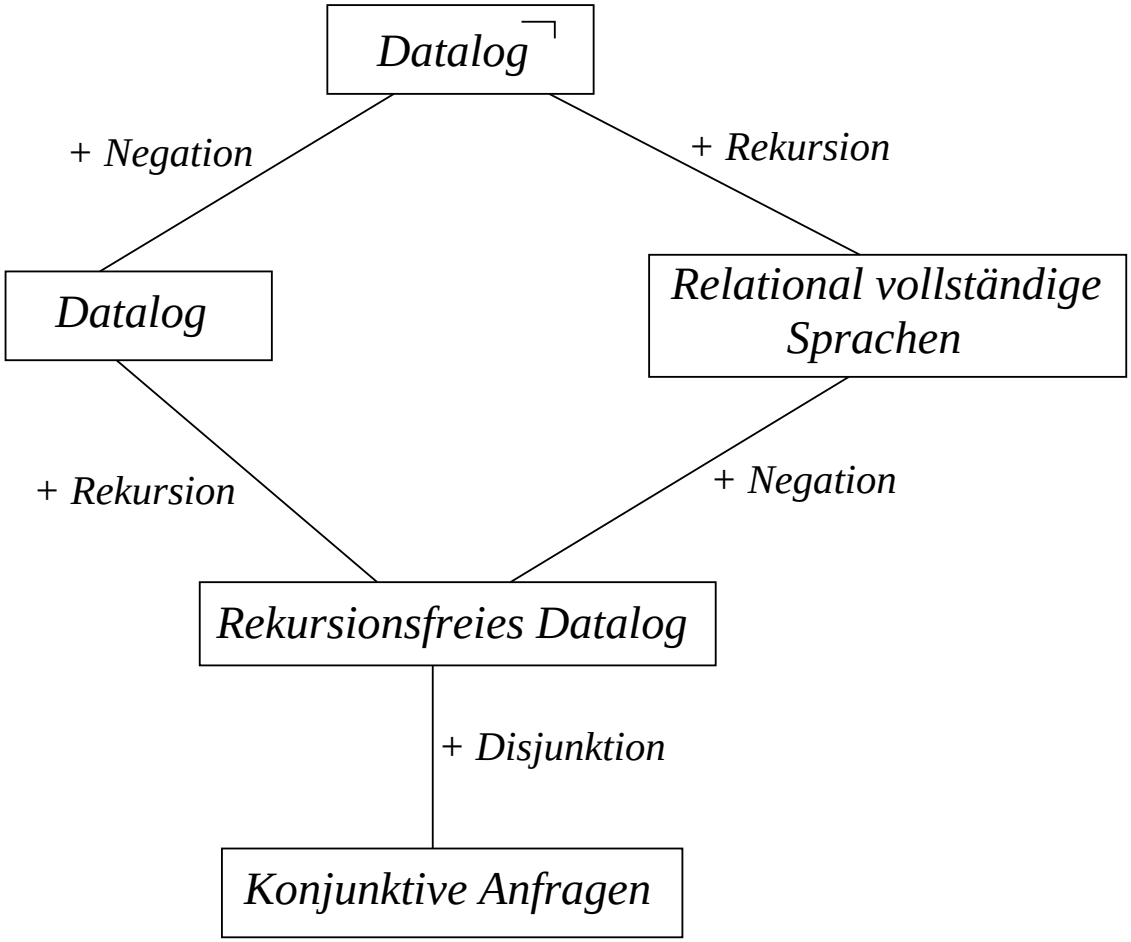
$$\left\{ \langle x_{Kino}, x_{Adr} \rangle \;:\; \exists x_{Titel}\, \exists x_{Regie}\, \exists x_{Zeit}\, \exists x_{Tel} \; Filme(x_{Titel}, x_{Regie}, \text{“George Clooney”}) \wedge Programm(x_{Kino}, x_{Titel}, x_{Zeit}) \wedge Orte(x_{Kino}, x_{Adr}, x_{Tel}) \right\}$$

- Relationale Algebra: $\pi_{Kino,Adresse}\left(\sigma_{\substack{Schauspieler = \\ \text{“George Clooney”}}}\left(Filme \bowtie Programm \bowtie Orte\right)\right)$

Hierarchie der Anfragesprachen

Bemerkung: Anfragesprachen unterscheiden sich in ihrer Ausdrucksstärke.

Übersicht:



Das Relationale Modell

Typische Problemstellungen bzgl. Anfrageauswertung

Sei $\mathcal{A}$ eine Anfragesprache.

AUSWERTUNGSPROBLEM FÜR $\mathcal{A}$:

Eingabe: Anfrage $Q \in \mathcal{A}$, Datenbank $\mathbf{I}$

Aufgabe: Berechne $Q(\mathbf{I})$

Variante:

LEERHEITSPROBLEM FÜR $\mathcal{A}$:

Eingabe: Anfrage $Q \in \mathcal{A}$, Datenbank $\mathbf{I}$

Aufgabe: Ist $Q(\mathbf{I}) = \emptyset$?

Wichtige Fragestellung:

Welche Ressourcen (etwa Zeit, Platz) sind nötig, um diese Probleme zu lösen?

Datenkomplexität und Kombinierte Komplexität

Die Komplexität der Anfrageauswertung kann unter zwei Blickwinkeln betrachtet werden:

1. Anfrage und Datenbank sind Eingabe $\rightsquigarrow$ Kombinierte Komplexität
   gemessen in n und k, wobei $n = ||\mathbf{I}||$ und $k = ||Q||$

2. Anfrage fest, Datenbank ist Eingabe: $\rightsquigarrow$ Datenkomplexität
   gemessen nur in $n = ||\mathbf{I}||$

Rechtfertigung für "Datenkomplexität":
i.d.R. ist die Anfrage kurz, die Datenbank aber sehr groß.

Typische Form von Ergebnissen, die im Laufe der Vorlesung bewiesen werden:

- Die Datenkomplexität der Relationalen Algebra ist in LOGSPACE.
- Die kombinierte Komplexität der Relationalen Algebra ist PSPACE-vollständig.