

# Crashkurs Relationale Datenbanksysteme

16.10.2003

Felix Naumann

## Voraussetzungen & Ziele

- Voraussetzungen
  - Abgeschlossenes Grundstudium
- Ziele
  - Vorbereitung für VL Informationsintegration I & II
  - Morgens + 1 Std. nachmittags:  
Relationales Modell, SQL, DBMS, Anfrageverarbeitung
  - Nachmittags:  
XML, XQuery
- Termin
  - Nur heute (16. Oktober) 9-13 und 14-18

## Forschungsgruppe Informationsintegration

- Juniorprofessor: Felix Naumann
- Mitarbeiter
  - Jens Bleiholder
  - Melanie Weis (ab 1.Nov.03)
- Themen

3



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Fragen, Wünsche und Vorstellungen

- Jetzt, oder...
- Raum: IV.105
- Sprechstunden: jederzeit
- Email: [naumann@informatik](mailto:naumann@informatik)
- Telefon: 2093 3905

4



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Überblick – Vormittag

- Motivation
  - Warum sind RDBMS und XML Systeme für die Informationsintegration wichtig?
- RDBMS – Relational Database Management System
  - Relationale Datenbanken
  - Relationales Modell (Datenmodell)
  - Relationale Algebra (Datenanfrage)
  - SQL (Anfragesprache)

5



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Überblick – Nachmittag

- RDBMS
  - Anfragebearbeitung
  - Optimierung
- XML
  - Grundkonzepte
  - XML Schema
  - XSLT & XQuery (Anfragesprachen)

6



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Quellen

- RDBMS Folien in Anlehnung an
  - VL “Data Warehousing”, Prof. Leser
  - VL “Einführung in Datenbanken”, Prof. Freytag

7



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Was ist Informationsintegration?

- Informationsintegration ist die Zusammenführung von Daten und Inhalt verschiedener Quellen zu einer einheitlichen Informationsmenge.
- Informationsintegration ist die **korrekte, vollständige** und **effiziente** Zusammenführung von Daten und Inhalt verschiedener, **heterogener** Quellen zu einer einheitlichen und **strukturierten** Informationsmenge zur **effektiven Interpretation** durch Nutzer und Anwendungen.

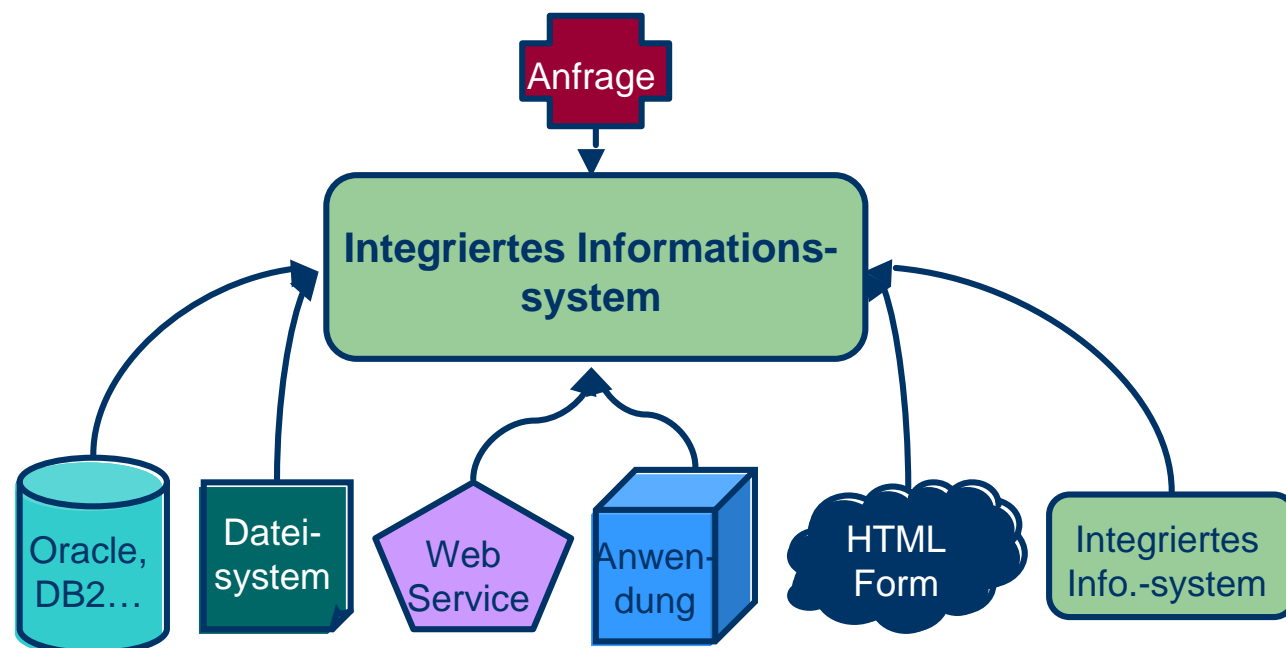
8



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

# Integrierte Informationssysteme



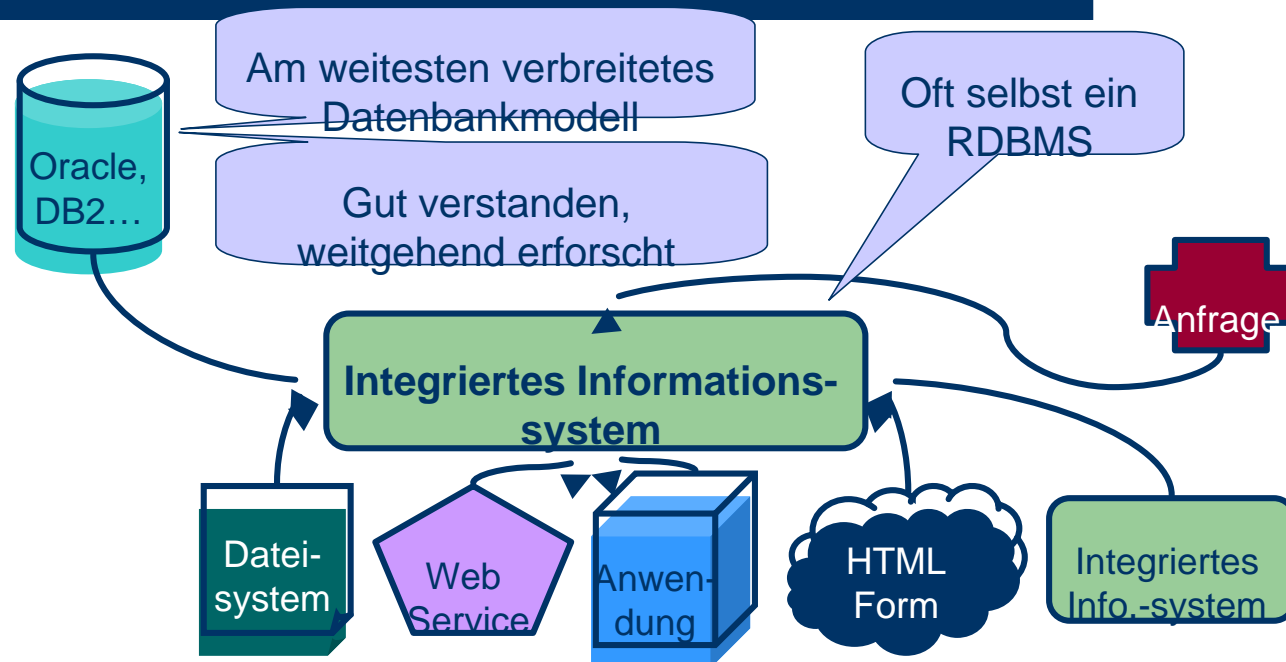
9



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

# Motivation – RDBMS



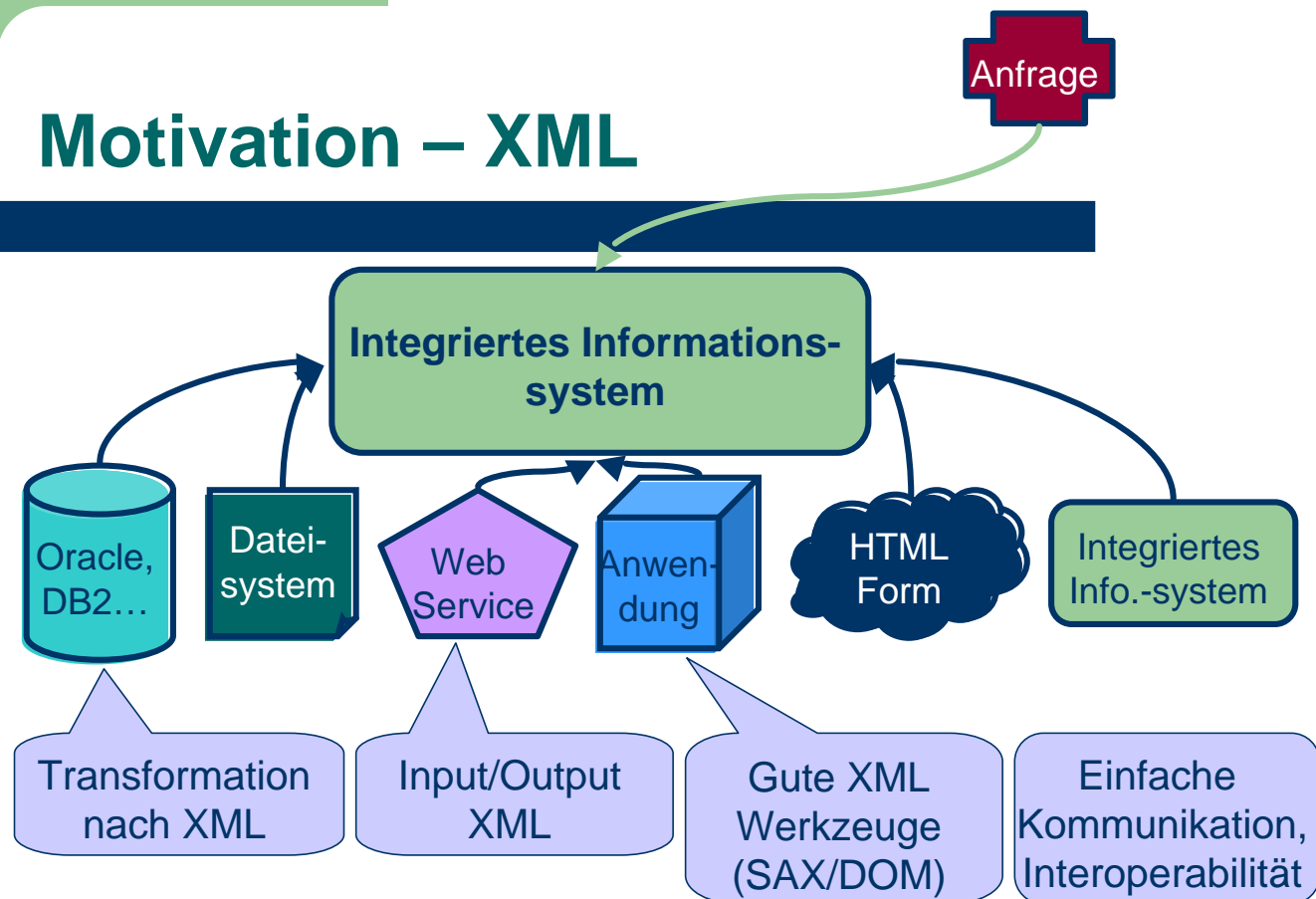
10



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Motivation – XML



11



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Was sind Daten?

- Digitale Repräsentation von
    - Dingen
    - Entitäten
    - Wissen
    - Information
  - in der wirklichen Welt.
  - Kernfragen:
    - Welche Daten speichere ich?
    - Wie speichere ich die Daten?
    - Wie frage ich Daten ab?
    - Wie erledige ich all dies effizient?
- } Unterstützung durch Datenbanksystem

12



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Datenbanksysteme

- Bestandteile

- Datenbank
  - Die Daten selbst
  - + Metadaten (Beschreibung der Daten)
- Datenbankmanagementsystem (DBMS)
  - Softwarekomponente zum Zugriff auf eine oder mehrere Datenbanken.
  - Server-basiert

13



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## DBMS Beispiele

- DBMS
  - Oracle, DB2
  - Informix, Sybase
  - NCR Terada, SQL-Server
  - PostgreSQL, Interbase
  - MySQL, ...
- Alle großen Softwaresysteme nutzen ein, mehrere oder sehr viele DBMS
  - SAP
  - Siebel
  - SRS
  - Sogar das neue Microsoft Windows File-System (Longhorn)

14



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## DBMS Aufgaben

- Unterstützung des Datenmodells
- Bereitstellung einer Anfragesprache (DDL & DML)
- Robustheit
  - Wahrung der Datenintegrität (Konsistenz etc.)
  - Abfangen von Systemfehlern
- Speicherverwaltung
- Transaktionsmanagement
  - Auch im Mehr-Benutzer-Betrieb
- Effiziente Anfragebearbeitung
- Userverwaltung & Zugangskontrolle

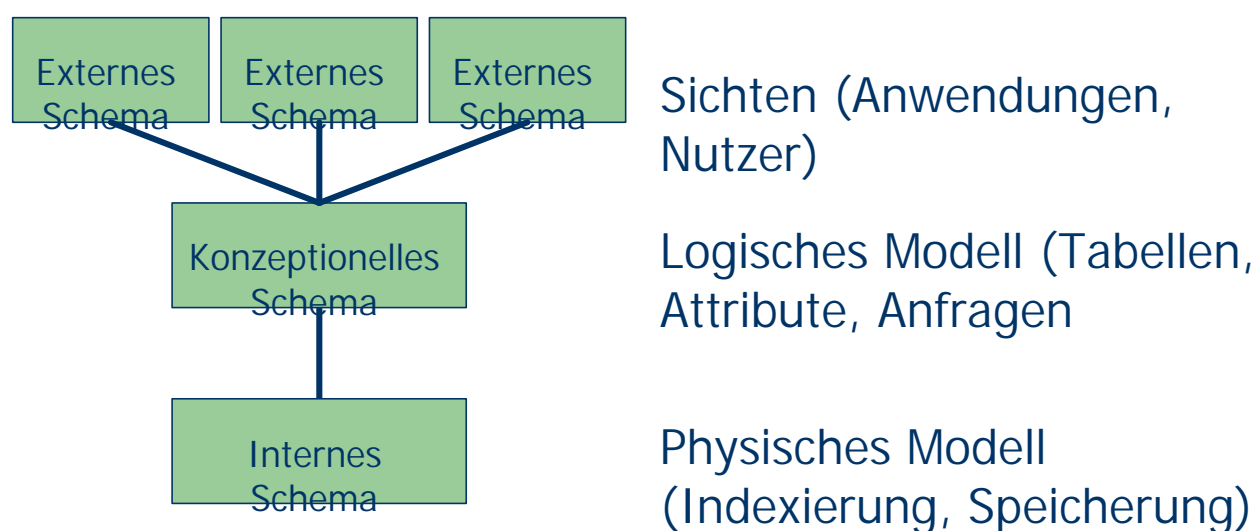
15



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Schichtenmodell (ANSI/X3/SPARC)



16

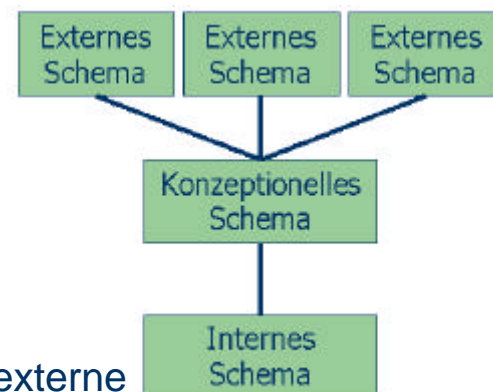


Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Schichtenmodell

- Interne (physische) Sicht
  - Speichermedium (Tape, Festplatte)
  - Speicherort (Zylinder, Block)
- Konzeptionelle (logische) Sicht
  - Unabhängig von physischer Sicht
  - Definiert durch Datenmodell
  - Stabiler Bezugspunkt für interne und externe Sichten
- Externe (logische) Sicht
  - Anwendungsprogramme
  - Nur auf die relevanten Daten
  - Enthält Aggregationen und Transformationen



17

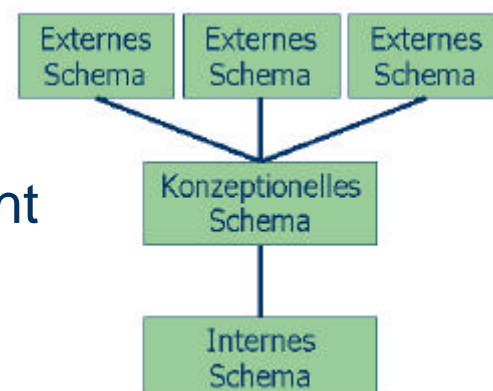


Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Schichtenmodell – Objekte

- Interne (physische) Sicht
  - Dateien
- Konzeptionelle (logische) Sicht
  - Schema
    - Relationen, Typen, Integritätsbedingungen
- Externe (logische) Sicht
  - „Views“



18

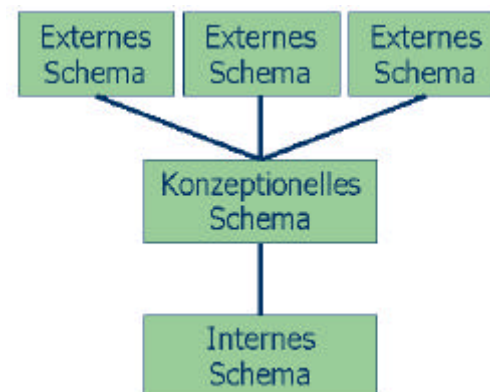


Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Schichtenmodell – Beispiel (Vorschau)

- Konzeptionelle (logische) Sicht
  - CREATE TABLE Book(  
ISBN VARCHAR(100)  
Title VARCHAR(100)  
Price DECIMAL(6,2))
- Externe (logische) Sicht
  - CREATE VIEW cheapBooks AS (  
SELECT \* FROM Book  
WHERE Price < 10,00 )
- Interne (physische) Sicht
  - CREATE INDEX BookIndex ON Book (ISBN)



## Datenbankentwurf

1. Konzeptioneller Entwurf
    - ER-Modell & ER-Diagramm
    - Relationenmodell
    - ...
  2. Physischer Entwurf
    - HW und SW abhängig
  3. Konzeptioneller Entwurf der externen Anwendungen
- Konzeptioneller Entwurf ist unabhängig vom physischen Entwurf!

## Überblick

- RDBMS – Relational Database Management System
  - Relationale Datenbanken
  - Relationales Modell (Datenmodell)
  - Relationale Algebra (Datenanfrage)
  - SQL (Anfragesprache)
  - Anfragebearbeitung
  - Optimierung

21



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Relationales Modell

- Auch
  - Relationenmodell
  - Relationales Datenmodell
- Ziele
  - Daten-Unabhängigkeit
  - Trennung vom hierarchischen und Netzwerk-Modellen

22



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Relationales Modell – Definitionen

- Domäne
  - Integer, Real, String
  - {rot, grün, blau}
- Relation (Tabelle)
  - Seien  $D_1, \dots, D_n$  Domänen
  - Relation  $R$  ist Teilmenge aus  $D_1 \times D_2 \times \dots \times D_n$
  - ...vom Grad  $n$
- Tupel (Zeile)
  - Ein Element aus  $R$
- Attribut (Spalte)
- Relationenschema
  - $R(A_1:D_1, \dots, A_n:D_n)$
- Kardinalität
  - $|R|$  ist die (aktuelle) Anzahl der Tupel in  $R$

23



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Relationales Datenmodell – Extension

- Extension einer Relation
  - Die Menge der momentanen Tupel
  - Kann sich durch Operationen ändern
  - Kann als Tabelle dargestellt werden
- Beispiel
  - Domänen:  
 $D_1 = \text{Integer}, D_2 = \text{String}, D_3 = \text{String}, D_4 = \text{Integer}$
  - Relationenschema:  
 $\text{Mitarbeiter}(P\_ID: D_1, \text{Vorname}: D_2, \text{Nachname}: D_3, \text{Alter}: D_4)$

24



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Relationales Datenmodell

Spalten/Attribute

Mitarbeiter			
P_ID	Vorname	Nachname	Alter
1	Peter	Müller	32
2	Stefanie	Meier	34
5	Petra	Weger	28
7	Andreas	Zwickel	44
...	...	...	...

Zeilen/  
Tupel

25



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Relationales Datenmodell vs. XML

- Relationen sind Mengen
  - Keine Duplikate
- Tupel sind ungeordnet
- Attributwerte sind Atomar
- XML ist Graphbasiert
  - Duplikate erlaubt (bzw. nicht definiert)
- In XML Dokumenten ist Reihenfolge wichtig
- Elemente sind geschachtelt

26



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Relationales Modell – Definitionen

- Datenbankschema
  - Menge von Relationenschemata
- Relationale Datenbank
  - Datenbankschema und alle zu dem Schema gehörenden Extensionen

27



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Relationales Modell – Schlüssel

- Sei  $R(A_1:D_1, \dots, A_n:D_n)$  ein Relationenschema und sei  $S \subseteq \{A_1, \dots, A_n\}$
- $S$  ist Schlüssel, falls
  - Eindeutigkeit:  
Falls zwei Tupel sich unterscheiden, unterscheiden sie auch in den Attributen des Schlüssels.
  - Minimalität:  
Es gibt kein  $S' \subsetneq S$ , welches Eindeutigkeit erfüllt.

28



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Relationales Modell – Schlüssel

P_ID	Vorname	Nachname	Alter
1	Peter	Müller	32
2	Stefanie	Meier	34
5	Petra	Weger	28
7	Andreas	Müller	44
11	Peter	Schmidt	32

Schlüssel: {PID}, {Vorname, Nachname}

Nicht Schlüssel (z.B.): {Vorname}: Nicht eindeutig  
{Vorname, Alter}: Nicht eindeutig  
{PID, Vorname}: Nicht minimal  
{Vorname, Nachname, Alter}: Nicht minimal

29



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Relationales Modell – Schlüssel

- Primärschlüssel
  - Ein „ausgezeichneter“ Schlüssel
- Schlüssel als Meta-Information
  - Kann nicht aus Extension abgelesen werden
  - Wichtige Entwurfsentscheidung

30



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Relationales Modell – Fremdschlüssel

- Seien  $R(A_1, \dots, A_n)$  und  $S(B_1, \dots, B_m)$  zwei Relationenschemata.
- Sei  $F$  ein Schlüssel für  $S$ 
  - Also  $F = \{B_{d1}, \dots, B_{dl}\} \subseteq \{B_1, \dots, B_m\}$
- Falls zusätzlich  $F \subseteq \{A_1, \dots, A_n\}$ , heißt  $F$  Fremdschlüssel von  $S$  in  $R$ .

31



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Relationales Modell – Fremdschlüssel

mitarbeiter			
p_id	vorname	name	...
1	Peter	Müller	32
2	Stefanie	Meier	34
5	Petra	Weger	28
7	Andreas	Zwiesel	44

projekte			
proj_id	p_id	name	kunde
1	1	DWH	
2	1	eCommerce	Metro
5	7	SAP	RAG

p\_id ist Schlüssel

proj\_id ist Schlüssel

p\_id ist Fremdschlüssel

32



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Relationales Modell – Integritätsbedingungen

- Jede Relation hat mindestens einen Schlüssel
- Attribute im Schlüssel dürfen nicht NULL sein
- Attributwerte in Domäne
- Zu jedem Fremdschlüsselwert muss es einen entsprechenden Schlüsselwert geben
- Assertions
  - Das Alter von Angestellten ist immer  $>15$ .

**Pause**

## Überblick

- RDBMS – Relational Database Management System
  - Relationale Datenbanken
  - Relationales Modell (Datenmodell)
  - Relationale Algebra (Datenanfrage)
  - SQL (Anfragesprache)
  - Anfragebearbeitung
  - Optimierung

35



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Relationale Algebra

- Anfragesprache
- 5 Grundoperationen
  - Vereinigung
  - Differenz
  - Kartesisches Produkt
  - Projektion
  - Selektion
- Abgeleitete Operationen
  - Join, Natural Join, Semi-Join
  - Division
  - Schnittmenge

Operationen bilden von Relation zu Relation ab (Abgeschlossenheit)

Operationen können zu Ausdrücken zusammengesetzt werden

36



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Relationale Algebra – Vereinigung

- $R \cup S$
- Attribute von R und S
  - Gleiche Anzahl
  - Gleiche Domänen
- Ergebnis
  - Mengentheoretische Vereinigung
  - gleiche Struktur wie R bzw. S
  - Duplikat-frei
  - $|R \cup S| = |R| + |S|$

37



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Relationale Algebra – Vereinigung

Mitarbeiter m

p_id	vorname	nachname	alter
1	Peter	Müller	32
2	Stefanie	Meier	34

Employees e

p_id	vorname	nachname	alter
5	Petra	Weger	28
7	Andreas	Zwickel	44

m ∪ e

p_id	vorname	nachname	alter
1	Peter	Müller	32
2	Stefanie	Meier	34
5	Petra	Weger	28
7	Andreas	Zwickel	44

38



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Relationale Algebra Vereinigung

- Jetzt: Kleine Vorschau, warum Informationsintegration so schwierig ist
  - Anderes Schema
  - Null Werte
  - Gleiche IDs
  - Gleiche real-world Objekte

39



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Relationale Algebra – Vereinigung

Mitarbeiter m

p_id	vorname	nachname	alter
1	Peter	Müller	32
2	Stefanie	Meier	34

Employees e

p_id	vorname	nachname	age
5	Petra	Weger	28
7	Andreas	Zwickel	44

Anderes Schema:  
 $m \cup e$  nicht  
definiert!

40



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Relationale Algebra – Vereinigung

Mitarbeiter m

p_id	vorname	nachname	alter
1	Peter	Müller	32
7	Andreas	Zwickel	34

Employees e

p_id	vorname	nachname	alter
5	Petra	Weger	28
7	Andreas	Zwickel	NULL

Null-Werte

$m \cup e$

p_id	vorname	nachname	alter
1	Peter	Müller	32
7	Andreas	Zwickel	34
5	Petra	Weger	28
7	Andreas	Zwickel	NULL



41



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Relationale Algebra – Vereinigung

Mitarbeiter m

p_id	vorname	nachname	alter
1	Peter	Müller	32
7	Stefanie	Meier	34

Employees e

p_id	vorname	nachname	age
5	Petra	Weger	28
7	Andreas	Zwickel	44

IDS: Schlüsseigenschaft  
gilt nicht mehr!

$m \cup e$

p_id	vorname	nachname	alter
1	Peter	Müller	32
7	Stefanie	Meier	34
5	Petra	Weger	28
7	Andreas	Zwickel	44

42



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Relationale Algebra – Vereinigung

Mitarbeiter m

p_id	vorname	nachname	alter
1	Peter	Müller	32
8	Andreas	Zwickel	44

Employees e

p_id	vorname	nachname	age
5	Petra	Weger	28
7	Andreas	Zwickel	44

$m \cup e$

p_id	vorname	nachname	alter
1	Peter	Müller	32
8	Andreas	Zwickel	34
5	Petra	Weger	28
7	Andreas	Zwickel	NULL

IDS: Schlüsseleigenschaft gilt zwar, aber Duplikate!

43



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Relationale Algebra – Differenz

- $R / S$
- Attribute von R und S
  - Gleiche Anzahl
  - Gleiche Domänen
- Ergebnis
  - Mengentheoretische Differenz
  - gleiche Struktur wie R bzw. S
  - $|R / S| = |R| - |S|$

44



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Relationale Algebra – Differenz

Mitarbeiter m

p_id	vorname	nachname	alter
1	Peter	Müller	32
7	Andreas	Zwickel	34

Employees e

p_id	vorname	nachname	alter
7	Andreas	Zwickel	34

m / e

p_id	vorname	nachname	alter
1	Peter	Müller	32

## Relationale Algebra – Kartesisches Produkt

- $R \times S$
- Attribute
  - Unterschiedliche Anzahl
  - Unterschiedliche Domäne
- Ergebnis
  - Keine doppelten Attributnamen
  - Alle Kombinationen von Tupeln in R und S
  - Grösse:  $|R| \cdot |S|$
  - Eher von theoretischem Interesse

## Relationale Algebra – Kartesisches Produkt

mitarbeiter

p_id	vorname	nachname	alter
1	Peter	Müller	32
2	Stefanie	Meier	34

projekte

proj_id	p_id	name	kunde
1	1	DWH	BMW
2	1	eCommerce	Metro

mitarbeiter x projekte

p_id	vorname	nachname	alter	proj_id	p_id	name	kunde
1	Peter	Müller	32	1	1	DWH	BMW
1	Peter	Müller	32	2	1	eCommerce	Metro
2	Stefanie	Meier	34	1	1	DWH	BMW
2	Stefanie	Meier	34	2	1	eCommerce	Metro

47



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Relationale Algebra – Projektion

- $\Pi_{\text{proj}}(R)$
- Sei  $R(A_1, \dots, A_n)$  ein Relationenschemata
- $\text{Proj} = \{A_{j_1}, \dots, A_{j_n}\} \subseteq \{A_1, \dots, A_n\}$
- Ergebnis
  - Hat Struktur  $A_{j_1}, \dots, A_{j_n}$
  - Es können Duplikate entstehen, die entfernt werden müssen.
  - $|\Pi_{\text{proj}}(R)| = |R|$

48



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Relationale Algebra – Projektion

mitarbeiter			
p_id	vorname	nachname	alter
1	Peter	Müller	32
2	Stefanie	Meier	34

mitarbeiter			
p_id	vorname	nachname	alter
1	Peter	Müller	32
2	Stefanie	<b>Müller</b>	34

$\Pi_{\text{vorname.nachname}}(\text{mitarbeiter})$	
vorname	nachname
Peter	Müller
Stefanie	Meier

$\Pi_{\text{nachname}}(\text{mitarbeiter})$
nachname
Müller

49



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Relationale Algebra – Selektion

- $\sigma_{\text{cond}}(R)$
- Sei  $R(A_1:D_1, \dots, A_n:D_n)$  ein Relationenschema
- cond ist „Selektionsbedingung“
  - Form: „ $A_i \text{ op } A_j$ “ oder „ $A_i \text{ op } \text{const}$ “
    - $A_i, A_j \in \{A_1, \dots, A_n\}$
    - $D_i = D_j$  und  $\text{const} \in D_i$
    - $\text{Op} \in \{=, ?, \neq, <, >\}$
  - Boolescher Ausdruck
    - Junktoren: UND, ODER, NICHT
- Ergebnis
  - Alle Tupel aus, die cond erfüllen
  - Gleiche Struktur wie R
  - $|\sigma_{\text{cond}}(R)| = \text{sf}_{\text{cond}} \cdot |R| = |R|$

50



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Relationale Algebra – Selektion

mitarbeiter

p_id	vorname	nachname	alter
1	Peter	Müller	32
2	Stefanie	Meier	34
5	Petra	Weger	28
7	Andreas	Zwickel	44

$\sigma_{\text{alter} > 33}(\text{mitarbeiter})$

p_id	vorname	nachname	alter
2	Stefanie	Meier	34
7	Andreas	Zwickel	44

$\sigma_{\text{alter} > 33, \text{nachname} = \text{„Meier“}}(\text{mitarbeiter})$

p_id	vorname	nachname	alter
2	Stefanie	Meier	34

$\sigma_{\text{alter} = \text{p\_id}}(\text{mitarbeiter})$

p_id	vorname	nachname	alter

51



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Relationale Algebra – Join

- $R \bowtie_{\text{cond}} S$
- Seien  $R(A_1:D_1, \dots, A_n:D_n)$  und  $S(B_1:E_1, \dots, B_m:E_m)$  Relationenschemata.
- cond ist „Joinbedingung“
  - Form: „ $A_i \text{ op } B_j$ “
    - $\text{Op} \in \{=, ?, \neq, <, >\}$
    - $D_i$  und  $E_j$  „kompatibel“
  - Boolescher Ausdruck
    - Junktoren: UND, ODER, NICHT
- $R \bowtie_{\text{cond}} S = \sigma_{\text{cond}}(R \times S)$

52



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Relationale Algebra – Join

mitarbeiter				projekte			
p_id	vorname	nachname	alter	proj_id	p_id	name	kunde
1	Peter	Müller	32	1	1	DWH	BMW
2	Stefanie	Meier	34	2	1	eCommerce	Metro
5	Petra	Weger	28	5	7	SAP	RAG
7	Andreas	Zwickel	44				

mitarbeiter $\triangleright \triangleleft_{p\_id=p\_id}$ projekte							
p_id	vorname	nachname	Alter	proj_id	p_id	name	kunde
1	Peter	Müller	32	1	1	DWH	BMW
1	Peter	Müller	32	2	1	eCommerce	Metro
7	Andreas	Zwickel	44	5	7	SAP	RAG

53



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Relationale Algebra – Schnittmenge

- $R \cap S$
- Attribute von R und S
  - Gleiche Anzahl
  - Gleiche Domänen
- Ergebnis
  - Mengentheoretische Schnittmenge
  - gleiche Struktur wie R bzw. S
  - Duplikat-frei
- $R \cap S = R / (R / S)$

54



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Relationale Algebra – Schnittmenge

Mitarbeiter m

p_id	vorname	nachname	alter
1	Peter	Müller	32
7	Andreas	Zwickel	34

Employees e

p_id	vorname	nachname	alter
5	Petra	Weger	28
7	Andreas	Zwickel	34

$m \cap e$

p_id	vorname	nachname	alter
7	Andreas	Zwickel	34

## Relationale Algebra – Weitere Operatoren

- Join Varianten
  - Semi-Join
  - Natural Join
  - $\theta$ -Join
  - Outer Join (Left-, Right- und Full-)
- Division

## Relationale Algebra - Anfragen

mitarbeiter

p_id	vorname	nachname	alter
1	Peter	Müller	32
2	Stefanie	Meier	34
5	Petra	Weger	28
7	Andreas	Zwickel	44
...	...	...	...

Vor- und Nachname aller Mitarbeiter älter als 40?

$\Pi_{\text{vorn.. nachn.}}(\sigma_{\text{alter} > 40}(\text{mitarbeiter}))$

57



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Relationale Algebra - Anfragen

mitarbeiter

p_id	vorname	nachname	alter
1	Peter	Müller	32
2	Stefanie	Meier	34
5	Petra	Weger	28
7	Andreas	Zwickel	44
...	...	...	...

Nachname und Projektname aller Mitarbeiter

projekte

proj_id	p_id	name
1	1	DWH
2	1	eCommerce
5	7	SAP
...	...	...

$\Pi_{\text{nachn., p.name}}(\sigma_{\text{m.p\_id} = \text{p.p\_id}}(\text{mit} \times \text{proj}))$

$\Pi_{\text{nachn., p.name}}(\text{mit} \bowtie_{\text{m.p\_id} > \text{p.p\_id}} \text{proj})$

$\Pi_{\text{nachn., p.name}}(\text{mit} \bowtie_{\text{m.p\_id} < \text{p.p\_id}} \text{proj})$

58



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Überblick

- RDBMS – Relational Database Management System
  - Relationale Datenbanken
  - Relationales Modell (Datenmodell)
  - Relationale Algebra (Datenanfrage)
  - SQL (Anfragesprache)
  - Anfragebearbeitung
  - Optimierung

59



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## SQL – Structured Query Language

- Erstmals: System R, IBM Almaden
- ANSI-SQL (89), SQL-2(92), SQL-3(98)
- Sehr weit verbreitet.
- Deklarativer Charakter:
  - Was soll das DBMS zeigen?
  - Nicht: Wie soll das DBMS das zu Zeigende finden (prozedural)?
- Fünf Grundbefehle:
  - Create, Insert, Update, Delete, Select
- DDL versus DML

60



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## SQL – Anfragen

- Grundform

- SELECT  $R_{i1}.A_1, R_{i2}.A_2, \dots, R_{im}.A_m$
  - FROM  $R_{i1}, R_{i2}, \dots, R_{im}$
  - WHERE  $\langle \text{cond} \rangle$

- Beispiel

- SELECT Vorname, Nachname
  - FROM mitarbeiter
  - WHERE Alter > 40

61



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## SQL – Beispiele

- SELECT Name  
FROM projekt
- SELECT \*  
FROM projekt  
WHERE Budget < 40.000
- SELECT mitarbeiter.Nachname, projekt.Name  
FROM mitarbeiter, projekt
- SELECT mitarbeiter.Nachname, projekt.Name  
FROM mitarbeiter, projekt  
WHERE mitarbeiter.p\_id = projekt.p\_id

Projektion

Selektion

Kartesisches  
Produkt

Join

62



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## SQL – Beispiele

- ```
SELECT mitarbeiter.Nachname, projekt.Name
FROM   mitarbeiter, projekt
WHERE  mitarbeiter.p_id = projekt.p_id
AND    projekt.Budget < 40.000
```
- ```
SELECT m.Nachname, p.Name
FROM   mitarbeiter m, projekt p
WHERE  m.p_id = p.p_id
```
- ```
SELECT m.Nachname
FROM   mitarbeiter m, projekt p
WHERE  m.p_id = p.p_id
AND    p.Budget < 40.000
```

Abkürzung

Es muss nicht jede  
Relation im SELECT  
auftauchen.

63



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## SQL – Weitere Funktionen

- DISTINCT
- ORDER BY
- UNION, DIFFERENCE, INTERSECTION
- Schachtelung
- Aggregation
- GROUP BY
  
- Alle sind fuer die Informationsintegration wichtig.

64



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## SQL – Beispiele

- ```
SELECT DISTINCT mitarbeiter.Vorname
FROM   mitarbeiter, projekt
WHERE  mitarbeiter.p_id = projekt.p_id
AND    projekt.Budget < 40.000
```
- ```
SELECT m.Nachname, p.Name
FROM   mitarbeiter m, projekt p
WHERE  m.p_id = p.p_id
ORDER BY p.budget
```

## SQL – Beispiele

- ```
SELECT mitarbeiter.Vorname, projekt.Name
FROM   mitarbeiter, projekt
WHERE  mitarbeiter.p_id = projekt.p_id
AND    projekt.Budget < 40.000
UNION
SELECT mitarbeiter.Vorname, projekt.Name
FROM   mitarbeiter, projekt
WHERE  mitarbeiter.p_id = projekt.p_id
AND    projekt.Budget > 100.000
```

## SQL – Beispiele

- ```
SELECT m.Vorname, p.Name
FROM   mitarbeiter m, projekt p
WHERE  m.p_id = p.p_id
AND    p.partner IN
      (SELECT company.Name
       FROM company
       WHERE company.country = "DE")
```
- Schachtelung

67



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## SQL – Beispiele

- Aggregation
  - AVG, COUNT, SUM, MIN, MAX
  - ```
SELECT AVG(Budget)
FROM   projekt
```
  - ```
SELECT SUM(p.Budget), MAX(p.Budget)
FROM   mitarbeiter m, projekt p
WHERE  m.p_id = p.p_id
AND    m.Nachname = "Schmidt"
```
  - ```
SELECT COUNT(*)
FROM   mitarbeiter
```

68



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## SQL – Beispiele

- ```
SELECT m.Nachname, SUM(p.Budget)
FROM   mitarbeiter m, projekt p
WHERE  m.p_id = p.p_id
GROUP BY m.Nachname
```
- ```
SELECT m.Nachname, AVG(p.Budget)
FROM   mitarbeiter m, projekt p
WHERE  m.p_id = p.p_id
GROUP BY m.Nachname
HAVING MIN(p.Budget) > 50,000
```

69



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## SQL – DDL

- DDL: Data Definition Language
- Definition von
  - Tabellen, Indexen, Views, ...
  - Administration: Tablespaces, Segmente, Parameter
  - Benutzerverwaltung: User, Gruppen, Rechte, ...

```
CREATE TABLE mitarbeiter (
  p_id          NUMBER,
  vorname       VARCHAR2(100),
  nachname      VARCHAR2(100),
  alter         NUMBER(2) CHECK (alter>0 AND alter<150),
  adresse       VARCHAR2(1000)
);
```

70



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## DQL – DDL

- Datentypen
  - INTEGER, SMALLINT
  - REAL, DOUBLE, NUMERIC(a,b), DECIMAL(a,b)
  - CHAR(n), VARCHAR(n)
  - BIT
  - DATE, TIME, TIMESTAMP, INTERVAL
- Attributnamen eindeutig

71



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## SQL – Insert

- Einfügen von Werten in Tabelle
- Prinzipiell ein Tupel pro Insert:

```
INSERT INTO mitarbeiter VALUES  
    (1, „Peter“, „Müller“, 38, „10101 Berlin“);
```

```
INSERT INTO projekte (proj_id,p_id,name) VALUES  
    (1, 3, „BMW“);
```

- Erweiterungen
  - Bulk-Insert  
INSERT INTO ... SELECT ...
  - Insert in mehrere Tabellen:  
INSERT INTO ... INTO ... INTO ...

72



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## SQL – Update

- Ändern von Werten in Tabellen
- Ändert mehrere Werte in einer Tabelle

```
UPDATE projekte
SET status = „abgebrochen“,
    kunde = kunde||“-insolvent“
WHERE kunde=„Grundig“
```

- Typisches Muster

```
UPDATE tabelle
SET ... = (SELECT ... FROM ... WHERE)
WHERE id in (SELECT ... FROM ... WHERE)
```

73



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Delete

- Löschen von Tupeln in einer Tabellen

```
DELETE FROM projekte
WHERE status=„abgeschlossen“
```

- Typisches Muster

```
DELETE FROM projekte
WHERE id in (SELECT ... FROM ... WHERE)
```

74



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

**PAUSE**

## Überblick

- RDBMS – Relational Database Management System
  - Relationale Datenbanken
  - Relationales Modell (Datenmodell)
  - Relationale Algebra (Datenanfrage)
  - SQL (Anfragesprache)
  - Anfragebearbeitung
  - Optimierung

## Anfragebearbeitung - Problem

- Anfragen sind deklarativ.
- Anfragen müssen in ausführbare (prozedurale) Form transformiert werden.
- Ziele
  - QEP – prozeduraler Query Execution Plan
  - Effizienz
    - Schnell
    - Wenig Ressourcenverbrauch (CPU, I/O, RAM, Bandbreite)

77



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Anfragebearbeitung

- Prinzipieller Ablauf
  1. Parsen der Anfrage (Syntax)
  2. Überprüfen der Elemente (Semantik)
  3. Berechnung von Ausführungsplänen
    - Exponentiell viele
  4. Wahl des optimalen Ausführungsplans
    - Regelbasierter Optimierer
    - Kostenbasierter Optimierer
  5. Anfrageausführung

78



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Abfragebearbeitung – Syntax

- SQL Anfrage wird in interne Darstellung übersetzt:
  - Z.B: Relationale Algebra
  - Syntaktische Korrektheit prüfen

79



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Anfragebearbeitung – Semantik

- Während der Übersetzung semantische Korrektheit prüfen
  - Existieren die Relationen/Attribute?
  - Korrekte Typen für Vergleiche?
  - Aggregation korrekt?
  - ...

80



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Anfragebearbeitung – Beispiel

- ```
SELECT m.Nachname
FROM   mitarbeiter m, projekt p
WHERE  m.p_id = p.p_id
AND    p.Budget < 40.000
```
- $\Pi_{m.Nachname}(\sigma_{m.p\_id = p.p\_id, p.Budget < 40.000}(m \times p))$

81

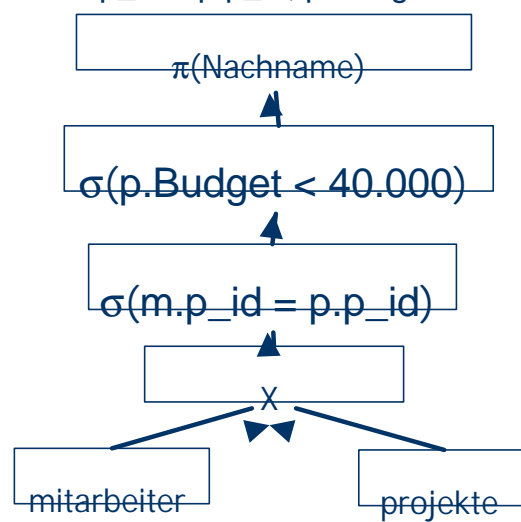


Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Anfragebearbeitung – Anfragebaum

- $\Pi_{m.Nachname}(\sigma_{m.p\_id = p.p\_id, p.Budget < 40.000}(m \times p))$



82

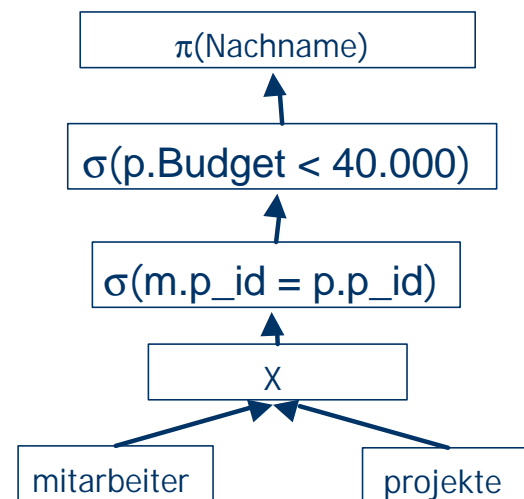


Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Anfragebearbeitung – Ausführung

- Interpretation des Baums als Ausführungsplan
  1. Kartesisches Produkt
  2. Zwei Selektionen
  3. Projektion
- Sehr aufwendig!



83



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Anfragebearbeitung – Transformationsregeln

- Transformation der internen Darstellung
  - Ohne Semantik zu verändern
  - Zur effizienteren Ausführung
    - Insbesondere: Kleine Zwischenergebnisse
- Äquivalente Ausdrücke
  - Zwei Ausdrücke der relationalen Algebra heißen äquivalent, falls
    - Gleiche Operanden (= Relationen)
    - Stets gleiche Antwortrelation

84



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Anfragebearbeitung – einige Transformationsregeln

- Kommutativität
  - $m \times p = p \times m$
- Assoziativität
  - $(m \times p) \times q = m \times (p \times q)$
- Kaskade von Projektionen
  - $\Pi_{m.Nachname}(\Pi_{m.Vorname, m.Nachname} m) = \Pi_{m.Nachname}(m)$
- Kaskade von Selektionen
  - $\sigma_{m.p\_id = p.p\_id}(\sigma_{p.Budget < 40.000}(m))$   
 $= \sigma_{m.p\_id = p.p\_id \text{ AND } p.Budget < 40.000}(m)$

85



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Anfragebearbeitung – einige Transformationsregeln

- Zusammenfassen von Selektion und Kartesischen Produkt zu Join
  - $\sigma_{m.p\_id = p.p\_id}(m \times p) = m \bowtie_{m.p\_id = p.p\_id} p$
- Vertauschen von Selektion und Projektion
  - $\Pi_{m.Nachname, m}(\sigma_{m.p\_id = p.p\_id}(m \times p))$   
 $= \Pi_{m.Nachname}(\sigma_{m.p\_id = p.p\_id}(\Pi_{m.Nachname, m.p\_id, p.p\_id}(m \times p)))$
- Vertauschen von Selektion und Join
  - $\sigma_{p.Budget < 40.000}(p \bowtie_{m.p\_id = p.p\_id} m)$   
 $= \sigma_{p.Budget < 40.000}(p) \bowtie_{m.p\_id = p.p\_id} m$
- Vertauschen von Projektion und Join

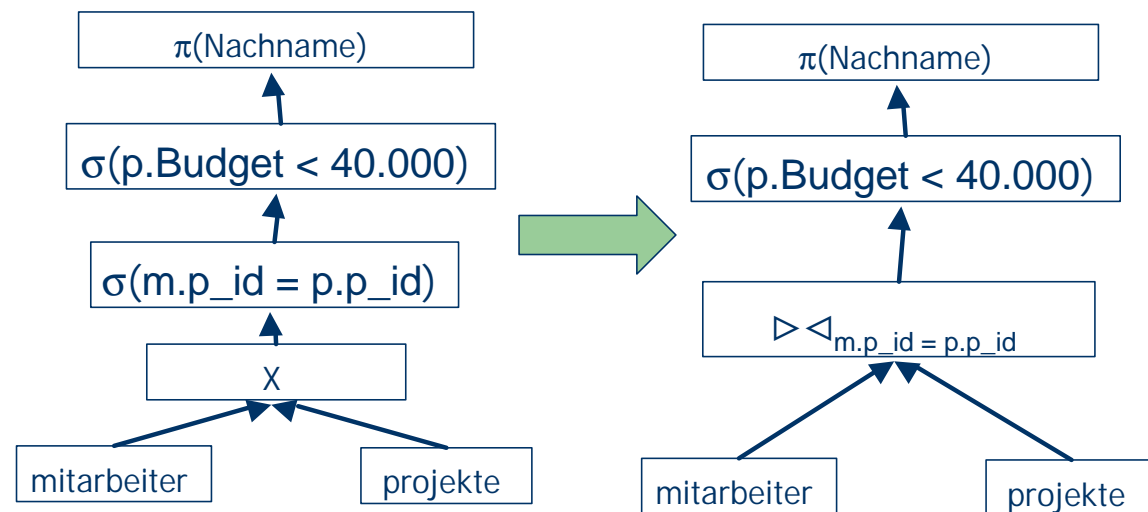
86



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Anfragebearbeitung – Beispiel



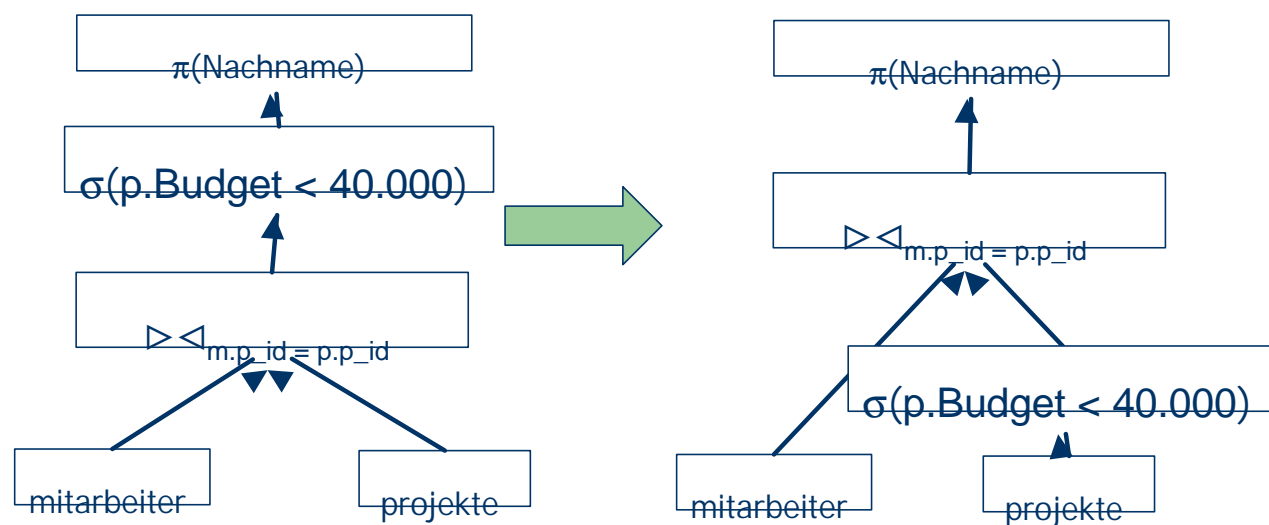
87



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Anfragebearbeitung – Beispiel



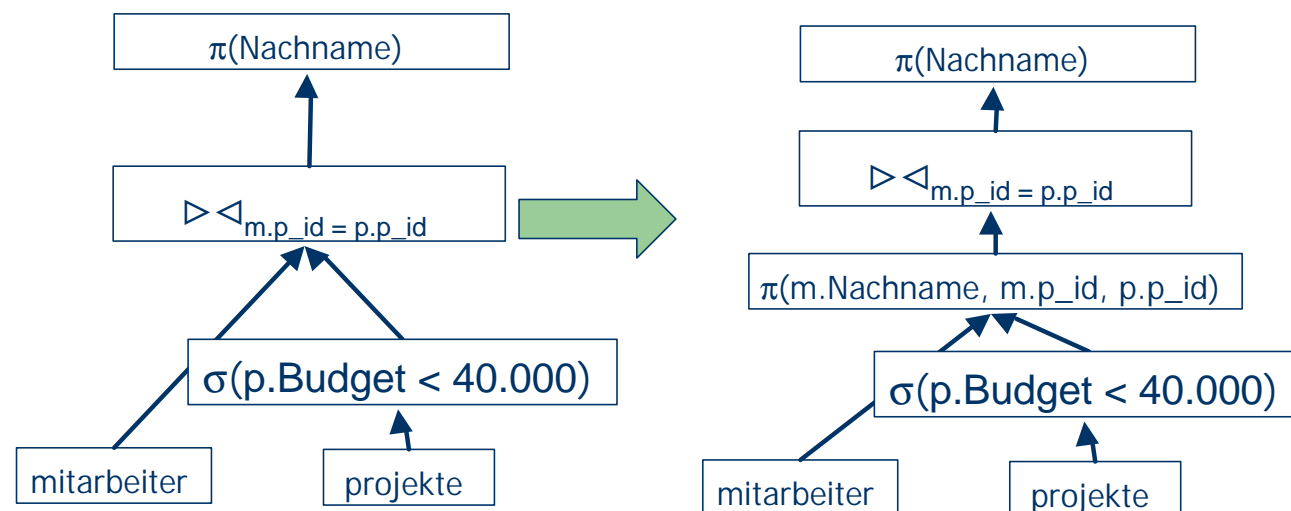
88



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Anfragebearbeitung – Beispiel



89



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Überblick

- RDBMS – Relational Database Management System
  - Relationale Datenbanken
  - Relationales Modell (Datenmodell)
  - Relationale Algebra (Datenanfrage)
  - SQL (Anfragesprache)
  - Anfragebearbeitung
  - Optimierung

90



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Anfragebearbeitung - Optimierung

- Regelbasierte Optimierung
  - Fester Regelsatz schreibt Transformationen gemäß der genannten Regeln vor.
  - Prioritäten unter den Regeln
  - Nicht hier.
- Kostenbasierte Optimierung
  - Kostenmodell
  - Transformationen um Kosten zu verkleinern

91



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Anfragebearbeitung - Optimierung

- Kostenmodell
  - Kardinalitäten der Relationen
  - Werteverteilungen (Histogramme)
  - Selektivität
    - Selektionen
    - Joins
- Hardware
  - Speichergröße
  - Bandwidth
  - I/O Zeiten

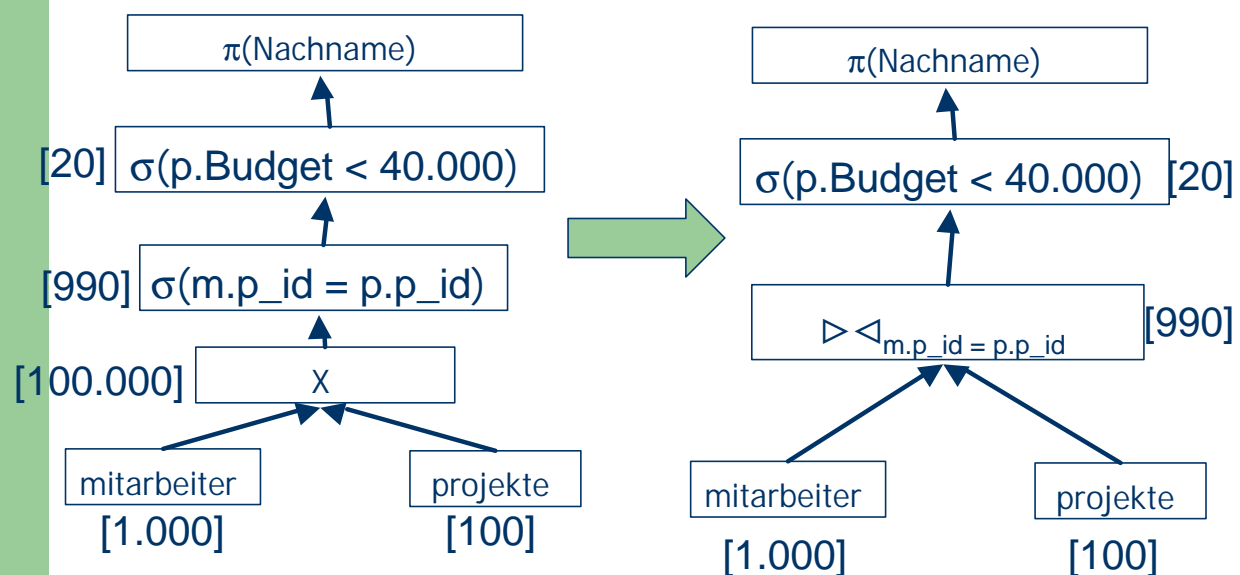
92



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Anfragebearbeitung – Optimierung



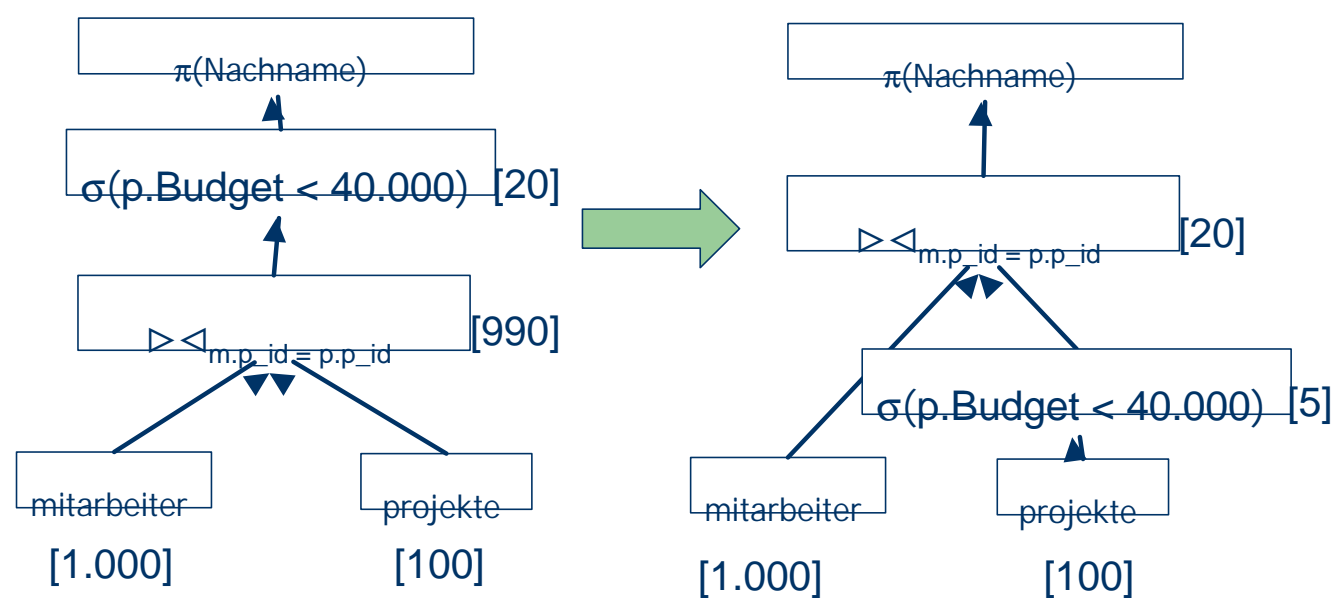
93



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Anfragebearbeitung – Optimierung



94



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Anfragebearbeitung – Optimierung

- Freiheitsgrade:
  - Algebraische Anfrageumformung
  - Joinreihenfolge
  - Joinmethode (Nested Loop, Sort-Merge, Hash ...)
  - Indexzugriff oder Full-Table-Scan
  - Operatorreihenfolge
  - ...

95



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Anfragebearbeitung – Optimierung

- Joinreihenfolge
  - Join ist i.d.R. teuerster Operator.
  - Optimierung konzentriert sich auf beste Reihenfolge.
  - Bei  $n$  Relationen  $n!$  Alternativen
    - Aber: Die meisten haben kartesisches Produkt.
  - Viele Algorithmen
    - Dynamische Programmierung
  - Bei mehr-Processor Systemen nicht nur Reihenfolge sondern auch Parallelisierung.

96



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Anfragebearbeitung – Optimierung

- Joinmethode
  - Nested Loops Join
    - Zwei geschachtelte Schleifen
    - Innere und äußere Relation
    - Aufwand:  $n \cdot m$
  - Sort Merge Join
    - Sortiere jede Relation
    - Ein mischender Cursor-Durchlauf durch jede Relation
    - Aufwand:  $n \log n + m \log m + n + m$
    - Günstig, falls Relationen sowieso sortiert, oder bei Index
  - Hash Join

97



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Anfragebearbeitung – Optimierung

- Nested Loops – Äußere vs. Innere Relation
- FOR EACH  $r$  in  $R$ 
  - FOR EACH  $s$  in  $S$ 
    - IF( $r.A = s.A$ ) THEN Output ( $r,s$ )
- Aufwandsunterschied  $R$  und  $S$ ?
- Ja, wegen Organisation der Tupel als Seiten auf Festplatte...

98



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Anfragebearbeitung – Optimierung

R: 

|                                                                                   |     |  |  |  |  |  |  |
|-----------------------------------------------------------------------------------|-----|--|--|--|--|--|--|
|  | ... |  |  |  |  |  |  |
|-----------------------------------------------------------------------------------|-----|--|--|--|--|--|--|

S: 

|                                                                                   |     |  |  |  |  |
|-----------------------------------------------------------------------------------|-----|--|--|--|--|
|  | ... |  |  |  |  |
|-----------------------------------------------------------------------------------|-----|--|--|--|--|

$P_R$  = Anzahl Seiten R

$P_S$  = Anzahl Seiten S

R außen:  $P_R + T_R \cdot P_S = 8 + 48 \cdot 6 = 296$

S außen:  $P_S + T_S \cdot P_R = 6 + 18 \cdot 8 = 150$

99



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003

## Literatur

- RDBMS
  - Principles of Database Systems, Jeffrey Ullman, Computer Science Press, 1988
- Informationsintegration
  - Principles of Distributed Database Systems  
[M. Tamer Özsu](#), [Patrick Valduriez](#)  
ISBN: 0136597076
  - ["Mediators in the Architecture of Future Information Systems"](#);  
Gio Wiederhold  
*IEEE Computer*, March 1992, pages 38-49.

10  
0



Felix Naumann, CK Informationssysteme, WS 03/04

16.10.2003