

Informationsintegration  
Local-as-View: LaV

7.6.2007  
Felix Naumann

Überblick

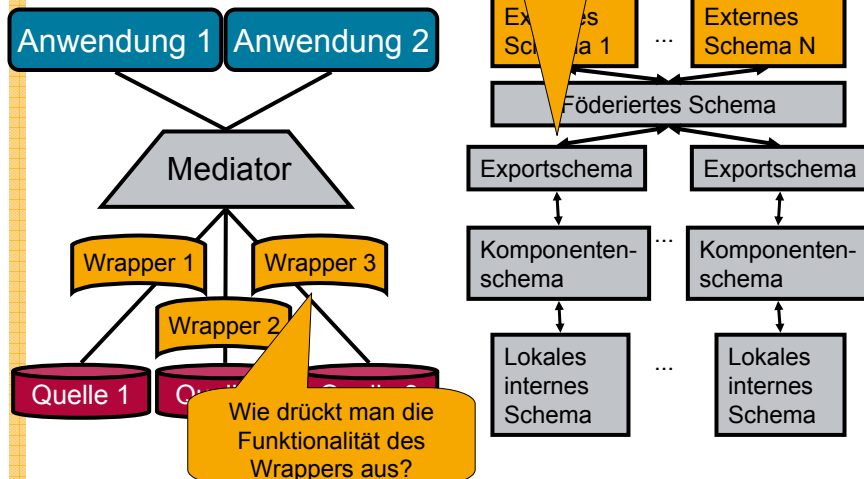
2

- Motivation
- Korrespondenzen
- Übersicht Anfrageplanung
- Global as View (GaV)
- ➔ ■ Local as View (LaV)
  - Modellierung
  - Anwendungen
  - Anfragebearbeitung
  - Containment
- Global Local as View (GLaV)
- Vergleich



## WdH.: Das Problem

3



Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

## Modellierung von Datenquellen

4

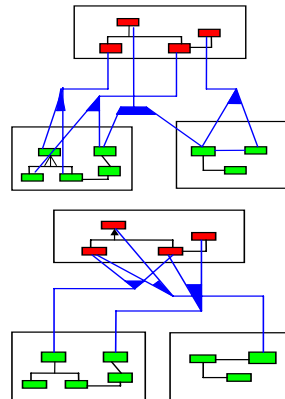
- Kernidee
  - Modellierung strukturell heterogener Quellen in Bezug auf ein globales Schema als Views (Sichten)
  - Relationales Modell
- Allgemein:
  - Eine Sicht verknüpft mehrere Relationen und produziert eine Relation.
- Sichten zur Verknüpfung von Schemata
  - Sicht definiert auf Relationen eines Schemas und produziert eine Relation des anderen Schemas
- Jetzt: Unterscheidung lokales und globales Schema

Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

## Global as View / Local as View

5

- Global as View
  - Relationen des globalen Schemas werden als Sichten auf die lokalen Schemas der Quellen ausgedrückt.
  
- Local as View
  - Relationen der Schemas der Quellen werden als Sichten auf das globale Schema ausgedrückt.



Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

## Überblick

6

- Motivation
- Korrespondenzen
- Übersicht Anfrageplanung
- Global as View (GaV)
- Local as View (LaV)
  - Modellierung
  - Anwendungen
  - Anfragebearbeitung
  - Containment
- Global Local as View (GLaV)
- Vergleich



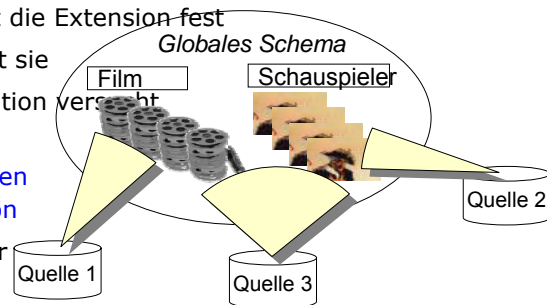
Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

## Warum LaV?

7

### Andere Sichtweise

- Es gibt in der Welt eine Menge von Filmen, Schauspielern, ...
- Das globale Schema modelliert diese Welt
- Theoretisch steht damit die Extension fest
  - Aber niemand kennt sie
  - Informationsintegration versucht sie herzustellen
- Quellen speichern Sichten auf die globale Extension
  - Also Ausschnitte der realen Welt
- Nur die können wir verwenden



Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

## Local as View (LaV) – Beispiel

8

Globales Schema  
 Film(Titel, Regie, Jahr, Genre)  
 Programm(Kino, Titel, Zeit)

S1: IMDB(Titel, Regie, Jahr, Genre)  
 S2: MyMovies(Titel, Regie, Jahr, Genre)  
 S3: RegieDB(Titel, Regie)  
 S4: GenreDB(Titel, Jahr, Genre)

```
CREATE VIEW S1 AS
SELECT * FROM Film

CREATE VIEW S2 AS
SELECT * FROM Film

CREATE VIEW S3 AS
SELECT Film.Titel, Film.Regie
FROM Film

CREATE VIEW S4 AS
SELECT Film.Titel, Film.Jahr,
       Film.Genre
FROM Film
```

Quelle: VL „Data Integration“, Alon Halevy, University of Washington, 2002

Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

## Local as View (LaV) – Beispiel

9

Globales Schema  
Film(Titel, Regie, Jahr, Genre)  
Programm(Kino, Titel, Zeit)

S9: ActorDB(Titel, Schauspieler, Jahr)

„Verpasste Chance“

```
CREATE VIEW S9 AS  
SELECT Titel, NULL, Jahr  
FROM Film
```

## Local as View (LaV) – Beispiel

10

Globales Schema  
Film(Titel, Regie, Jahr, Genre)  
Programm(Kino, Titel, Zeit)

S7: KinoDB(Kino, Genre)

```
CREATE VIEW S7 AS  
SELECT Programm.Kino, Film.Genre  
FROM Film, Programm  
WHERE Film.Titel = Programm.Titel
```

Assoziationen des globalen Schemas können in der Sicht hergestellt werden.

## Local as View (LaV) – Beispiel

11

Globales Schema  
 Film(Titel, Regie, Jahr, Genre)  
 Programm(Kino, Titel, Zeit)

S9: Filme(Titel, Jahr, Ort, RegieID)  
 Regie(ID, Regisseur)

```
CREATE VIEW S7Filme AS
SELECT Titel, Jahr, NULL, NULL
FROM Film
```

```
CREATE VIEW S7Regie AS
SELECT NULL, Regie
FROM Film
```

Assoziationen des lokalen Schemas können nicht abgebildet werden.

## Local as View (LaV) – Beispiel

12

Globales Schema  
 Film(Titel, Regie, Jahr, Genre)  
 Programm(Kino, Titel, Zeit)

S8: NeueFilme(Titel, Regie, Genre)  
 (Nebenbedingung: Jahr > 2000)

```
CREATE VIEW S8 AS
SELECT Titel, Regie, Genre
FROM Film
WHERE Jahr > 2000
```

- Nebenbedingung auf der Quelle kann modelliert werden
  - Wenn das Attribut im globalen Schema existiert
- IC müssen in der Quelle nicht explizit definiert werden
  - Auch implizite Einschränkungen können in den View

## Local as View (LaV) – Beispiel

13

Globales Schema  
 NeuerFilm(Titel, Regie, Jahr, Genre)  
 Programm(Kino, Titel, Zeit)  
 Nebenbedingung: Jahr > 2000

S1: IMDB(Titel, Regie, Jahr, Genre)  
 S2: MyMovies(Titel, Regie, Jahr, Genre)

```
CREATE VIEW S1 AS
SELECT * FROM NeuerFilm
(WHERE Jahr > 2000)
```

```
CREATE VIEW S2 AS
SELECT * FROM NeuerFilm
(WHERE Jahr > 2000)
```

- Nebenbedingungen auf dem globalen Schema können wir nicht modellieren
- Das ging aber bei GaV
- Also hat beides Stärken und Schwächen

Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

## Local as View (LaV) – lokale Nebenbedingungen

14

Globales Schema  
 Film(Titel, Regie, Jahr, Genre)

S1: AlleFilmeNett(Titel, Regie, Jahr, Genre)  
 S2: AlleFilmeBöse(Titel, Regie, Genre)  
 S3: NeueFilmeNett(Titel, Regie, Jahr, Genre)  
 (Nebenbedingung: Jahr > 2000)  
 S4: NeueFilmeBöse(Titel, Regie, Genre)  
 (Nebenbedingung: Jahr > 2000)  
 S5: AktuelleFilme(Titel, Regie, Genre)  
 (Nebenbedingung: Jahr = 2004)

```
CREATE VIEW S1 AS
SELECT * FROM Film
```

```
CREATE VIEW S2 AS
SELECT Titel, Regie, Genre
FROM Film
```

```
CREATE VIEW S3 AS
SELECT * FROM Film
(WHERE Jahr > 2000)
```

```
CREATE VIEW S4 AS
SELECT Titel, Regie, Genre
FROM Film
WHERE Jahr > 2000
```

```
CREATE VIEW S5 AS
SELECT Titel, Regie, Genre
FROM Film
(WHERE Jahr = 2004)
```

Modellierung zur Optimierung

Modellierung zur Beantwortbarkeit

Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

## Komplexere Beispiele

15

Datenquelle	Beschreibung
spielfilme(titel, regisseur, laenge)	Informationen über Spielfilme, die mindestens 80 Minuten Länge haben.
kurzfilme(titel, regisseur)	Informationen über Kurzfilme. Kurzfilme sind höchstens 10 Minuten lang.
filmkritiken(titel, regisseur, schauspieler, kritik)	Kritiken zu Hauptdarstellern von Filmen
us_spielfilme(titel, laenge, schauspieler_name)	Spielfilme mit US-amerikanischen Schauspielern
spielfilm_kritiken(titel, rolle, kritik)	Kritiken zu Rollen in Spielfilmen
kurzfilm_rollen(titel, rolle, schauspieler_name, nationalitaet)	Rollenbesetzungen in Kurzfilmen

```
film(titel, typ, regisseur, laenge);
schauspieler(schauspieler_name, nationalitaet);
spielt(titel, schauspieler_name, rolle, kritik);
```

```

    film(T, Y, R, L), L > 79, Y = 'Spielfilm' ⊇ spielfilme(T, R, L)
    film(T, Y, R, L), L < 11, Y = 'Kurzfilm' ⊇ kurzfilme(T, R)
    film(T, _, R, _) , spielt(T, S, O, K), O = 'Hauptrolle' ⊇ filmkritiken(T, R, S, K)
    film(T, Y, _, L), spielt(T, S, _, _) ,
    schauspieler(S, N), N = 'US', Y = 'Spielfilm' ⊇ us_spielfilm(T, L, S)
    film(T, Y, _, _) , spielt(T, _, O, K), Y = 'Spielfilm' ⊇ spielfilm_kritiken(T, O, K)
    film(T, Y, _, _) , spielt(T, S, O, _) ,
    schauspieler(S, N), Y = 'Kurzfilm' ⊇ kurzfilm_rollen(T, O, S, N)

```

Felix Nauma

## Local as View (LaV) – globale Nebenbedingungen



16

Globales Schema  
 NeuerFilm(Titel, Regie, Jahr, Genre)  
 Nebenbedingung: Jahr > 2000  
 oder  
 NeuerFilm(Titel, Regie, Genre)  
 Nebenbedingung: Jahr > 2000

S1: AlleFilmeNett(Titel, Regie, Jahr, Genre)  
 S2: AlleFilmeBöse(Titel, Regie, Genre)

```
CREATE VIEW AlleFilmeNett AS
SELECT * FROM NeuerFilm
(WHERE JAHR > 2000)
```

```
[CREATE VIEW AlleFilmeBöse AS
SELECT Titel, Regie, Genre
FROM NeuerFilm]
```

Problem:  
 Bekannte Nebenbedingung auf dem globalen Schema kann nicht modelliert werden.

Frage:  
 Ist das schlimm?

Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005



## Überblick

17

- Motivation
- Korrespondenzen
- Übersicht Anfrageplanung
- Global as View (GaV)
- Local as View (LaV)
  - Modellierung
  - Anwendungen
  - Anfragebearbeitung
  - Containment
- Global Local as View (GLaV)
- Vergleich



Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

## LaV – Anwendungen

18

- Anfrageoptimierung
  - Materialisierte Sichten auf Datenbankschema
- Datawarehouse Design
  - Materialisierte Sichten auf Warehouse-Schema
- Semantisches Caching
  - Materialisierte Daten beim Client
- Datenintegration
  - Datenquellen als Sichten auf globales (Mediator) Schema

Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

19

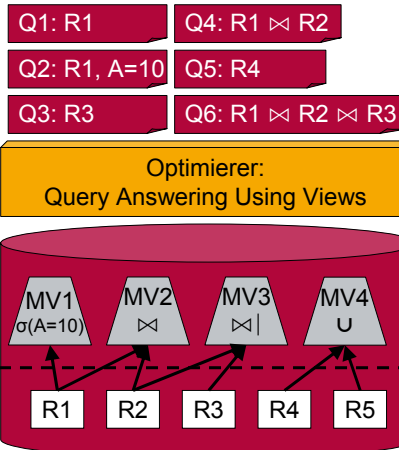
Materialisierte Sichten (materialized views, MV) auf Datenbankschema

- MQT: Materialized Query Table
- AST: Advanced Summary Table

Welche Sichten helfen bei der Beantwortung einer Datenbankanfrage durch Vorberechnung von Prädikaten?

Probleme:

- Es ist nicht immer besser eine MV zu verwenden (Indizes!).
- Aktualisierung von MVs
  - Write auf MV
  - Write auf Basisrelation



Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

20

- Materialisierte Sichten auf Warehouse-Schema
- Gegeben eine query workload
  - Query workload = Menge von Anfragen mit Häufigkeiten
- Welche Sichten sollte ich materialisieren?
  - Um alle Anfragen der Workload zu beantworten
- Allgemeiner:
  - Gegeben eine query workload, welche Sichten sollte ich materialisieren um die workload optimal zu beantworten.
    - Idee: Alle Kombinationen prüfen (AST Wizard)
  - Frage: Warum ist dieses Problem eigentlich ganz einfach?

Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

21

- In verteilten DBMS
- Materialisierte Daten beim Client
- Gegeben eine Anfrage
  - Welche Daten im Cache kann ich zur Beantwortung verwenden?
  - Welche Daten muss ich neu anfragen?
- Auch: Welche Sichten sollte ich vorberechnen?

Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

22

- Datenquellen als Sichten auf globales (Mediator) Schema
- Fragen:
  - Wie kann ich Antworten auf eine Anfrage an das globale Schema nur mittels der Sichten beantworten?
    - Unterschied zu Anfrageoptimierung: Keine Basistabellen verfügbar.
  - Kann ich die Anfrage vollständig (extensional) beantworten?

Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

## Überblick

23

- Motivation
- Korrespondenzen
- Übersicht Anfrageplanung
- Global as View (GaV)
- Local as View (LaV)
  - Modellierung
  - Anwendungen
  - Anfragebearbeitung
  - Containment
- Global Local as View (GLaV)
- Vergleich



Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

## Anfrageplanung

24

### Gegeben

- Eine Anfrage  $q$  an das globale Schema
- Lokale Schemata

### Gesucht

- Sequenz von Anfragen  $q_1 \diamond \dots \diamond q_n$
- Jedes  $q_i$  kann von einem Wrapper ausgeführt werden
- Die geeignete Verknüpfung von  $q_1, \dots, q_n$  beantwortet  $q$ 
  - Innerhalb eines Plans durch Joins:  $\diamond \rightarrow \bowtie$
  - Verschiedene Pläne werden durch UNION zusammengefasst :  $\diamond \rightarrow \cup$
- Von  $q_1 \bowtie \dots \bowtie q_n$  berechnete Tupel sind korrekte Antworten auf  $q$ 
  - Also fast  $q \equiv q_1 \bowtie \dots \bowtie q_n$

Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

## Anfrageplan

25

- Wir nennen  $q_1 \bowtie \dots \bowtie q_n$  einen Anfrageplan
- Definition  
Gegeben eine globale Anfrage  $q$ . Ein Anfrageplan  $p$  für  $q$  ist eine Anfrage der Form  $q_1 \bowtie \dots \bowtie q_n$  so dass
  - Jedes  $q_i$  kann von genau einem Wrapper ausgeführt werden
  - Jedes von  $p$  berechnete Tupel ist eine semantisch korrekte Antwort für  $q$ .
- Bemerkungen
  - „Semantisch korrekt“ haben wir noch nicht definiert
  - In der Regel gibt es viele Anfragepläne
  - Die  $q_i$  heißen Teilanfragen oder Teilpläne

Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

## Viele Anfragepläne

26

- Definition  
Gegeben eine globale Anfrage  $q$ . Seien  $p_1, \dots, p_n$  die Menge aller Anfragepläne für  $q$ . Dann ist das Ergebnis von  $q$  definiert als

$$result(q) = \bigcup_{i=1..n} result(p_i)$$

- Bemerkungen
  - Der UNION Operator entfernt Duplikate – dahinter verbirgt sich das Problem der Ergebnisintegration
  - Wie das Ergebnis berechnet wird, ist Sache der Anfrageoptimierung
    - Pläne können sich in Teilanfragen überlappen
  - Das Ergebnis von  $q$  hängt ab von den definierten Korrespondenzen

Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

## Anfragebearbeitung – LaV

27

### Gegeben:

- Anfrage an globales Schema
  - Insbesondere: Auf Relationen des globalen Schemas
- Für jede lokale Relation genau eine Sicht auf globales Schema

### Gesucht:

- Alle Tupel, die die Anfragebedingungen erfüllen
- Aber: Daten sind in lokalen Quellen gespeichert.

### Idee:

- Anfrageumschreibung durch Einbeziehung der Sichten
- Kombiniere geschickt die einzelnen Sichten zu einer Anfrage, so dass deren Ergebnis einen Teil der Anfrage (oder die ganze Anfrage) beantworten.
- Gesamtergebnis ist dann die UNION der Ergebnisse mehrerer Anfrageumschreibungen

**"Query Answering Using Views"**

Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

## LaV – Beispiel

28

### Ausschnitt Globales Schema

```
Lehrt(prof,kurs_id, sem, eval, univ)
Kurs(kurs_id, titel, univ)
```

### Quelle 1: Alle Datenbankveranstalt.

```
CREATE VIEW DB-kurs AS
SELECT K.titel, L.prof, K.kurs_id, K.univ
FROM   Lehrt L, Kurs K
WHERE  L.kurs_id = K.kurs_id
AND    L.univ = K.univ
AND    K.titel LIKE „%_Datenbanken“
```

### Quelle 2: Alle Humboldtvorlesungen

```
CREATE VIEW Hum-VL AS
SELECT K.titel, L.prof, K.kurs_id, K.univ
FROM   Lehrt L, Kurs K
WHERE  L.kurs_id = K.kurs_id
AND    K.univ = „Humboldt“
AND    L.univ = „Humboldt“
AND    K.titel LIKE „%VL_%“
```

### Globale Anfrage

```
SELECT prof
FROM   Lehrt L, Kurs K
WHERE  L.kurs_id = K.kurs_id
AND    K.titel LIKE „%_Datenbanken“
AND    L.univ = „Humboldt“
```



### Umgeschriebene Anfrage

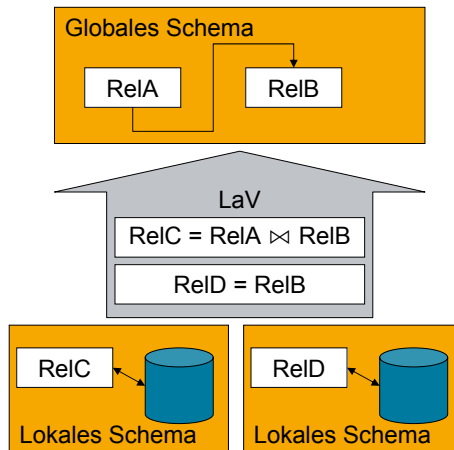
```
SELECT prof
FROM   DB-kurs D
WHERE  D.univ = „Humboldt“
```

**Frage:** Warum nicht Quelle 2 einbeziehen?

Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

## LaV Visualisierung (OWA)

29



Nutzer-Anfrage

```
SELECT ???
FROM RelB
WHERE ???
```

Anfrage-  
umschreibung

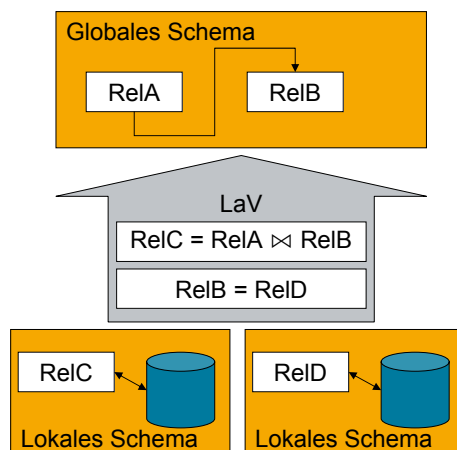
Umgeschriebene Anfrage

```
SELECT ???
FROM RelD
WHERE ???
UNION
SELECT Attr(B)
FROM RelC
WHERE ???
```

Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

## LaV Visualisierung

30



Nutzer-Anfrage

```
SELECT ???
FROM RelA, RelB
WHERE ???
```

Anfrage-  
umschreibung

Umgeschriebene Anfrage

```
SELECT ???
FROM RelC
WHERE ???
```

Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

## LaV – Beispiel

31

### Ausschnitt Globales Schema

Lehrt(prof,kurs\_id, sem, eval, univ)  
Kurs(kurs\_id, titel, univ)

### Quelle 1: Alle Datenbankveranstalt.

```
CREATE VIEW DB-kurs AS
SELECT K.titel, L.prof, K.kurs_id, K.univ
FROM Lehrt L, Kurs K
WHERE L.kurs_id = K.kurs_id
AND L.univ = K.univ
AND K.titel LIKE „%_Datenbanken“
```

### Globale Anfrage

```
SELECT titel, kurs_id
FROM Kurs K
WHERE L.univ = „Humboldt“
```



### Quelle 2: Alle Humboldtvorlesungen

```
CREATE VIEW Hum-VL AS
SELECT K.titel, L.prof, K.kurs_id, K.univ
FROM Lehrt L, Kurs K
WHERE L.kurs_id = K.kurs_id
AND K.univ = „Humboldt“
AND L.univ = „Humboldt“
AND K.titel LIKE „%VL_%“
```

### Umgeschriebene Anfrage

```
SELECT titel, kurs_id
FROM DB-kurs D
WHERE D.univ = „Humboldt“
UNION
SELECT titel, kurs_id
FROM Hum-VL
```

**Frage:**  
Warum hier doch  
Quelle 2 einbeziehen?

Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

## LaV – Beispiel Vergleich

32

### Globale Anfrage

```
SELECT prof
FROM Lehrt L, Kurs K
WHERE L.kurs_id = K.kurs_id
AND K.titel = „VL_Datenbanken“
AND L.univ = „Humboldt“
```



### Umgeschriebene Anfrage

```
SELECT prof
FROM DB-kurs D
WHERE D.univ = „Humboldt“
```

} Vollständige  
Antwort (CWA)

### Globale Anfrage

```
SELECT titel, kurs_id
FROM Kurs K
WHERE L.univ = „Humboldt“
```



### Umgeschriebene Anfrage

```
SELECT titel, kurs_id
FROM DB-kurs D
WHERE D.univ = „Humboldt“
UNION
SELECT titel, kurs_id
FROM Hum-VL
```

} Maximale  
Antwort

**Frage:**  
Was fehlt?

Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005



## CWA / OWA

33

- Closed World Assumption (CWA)
  - Vereinigung aller Daten der Basisrelationen entspricht der Menge aller relevanten Daten.
  - Beispiele: Data Warehouse (und traditionelle DBMS)
- Open World Assumption (OWA)
  - Vereinigung aller Daten der Datenquellen ist eine Teilmenge aller relevanten Daten.
  - Probleme
    - Inhalt der globalen Relation nicht fest
      - » Anfrageergebnisse können sich ändern
    - Definition der Vollständigkeit der Ergebnisse
      - » Welchen Anteil an der world hat das Ergebnis?
    - Negation in Anfragen

Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

## CWA / OWA – Beispiel

34

- Relation  $R(A,B)$
- View 1
  - CREATE VIEW V1 AS  
SELECT A FROM R
  - Extension: a
- View 2
  - CREATE VIEW V2 AS  
SELECT B FROM R
  - Extension: b
- Anfrage: SELECT \* FROM R
  - CWA: (a,b) muss in der Extension von R sein.
  - OWA: (a,b) muss nicht in der Extension von R sein.

R = (a,b)

R = (a,b)  
oder  
R = (a,x)  
(y,b)  
o.ä.

Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

## LaV – Anfragebearbeitung

35

- Gegeben: Anfrage  $Q$  und Sichten  $V_1, \dots, V_n$
- Gesucht: Umgeschriebene Anfrage  $Q'$ , die
  - bei Optimierung: äquivalent ist ( $Q = Q'$ ).
  - bei Integration: maximal enthalten ist.
    - D.h.  $Q \supseteq Q'$  und
    - es existiert kein  $Q''$  mit  $Q \supseteq Q'' \supseteq Q'$  wobei  $Q'' \neq Q'$ .
- Problem:
  - Wie definiert und testet man Äquivalenz und *maximal containment*?

Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

## Überblick

36

- Motivation
- Korrespondenzen
- Übersicht Anfrageplanung
- Global as View (GaV)
- Local as View (LaV)
  - Modellierung
  - Anwendungen
  - Anfragebearbeitung
  - Containment
- Global Local as View (GLaV)
- Vergleich



Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

37

- Gegeben
  - Anfrage Q (query)
  - Sicht V (view)
- Fragen
  - Ist Ergebnis von V identisch dem Ergebnis von Q?
  - Kurz: Ist V äquivalent zu Q,  $V = Q$  ?
- Rückführung auf „Enthalten sein“ (containment)
  - Ist das Ergebnis von V in Q enthalten?
  - Kurz: Ist V in Q enthalten,  $V \subseteq Q$  ?
- Denn
  - $V \subseteq Q, Q \subseteq V \Rightarrow V = Q$

Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

38

### Query containment (Anfrage-„Enthaltensein“)

- Sei S ein Schema. Seien Q und Q' Anfragen gegen S.
- Eine Instanz von S ist eine beliebige Datenbank D mit Schema S.
- Das Ergebnis einer Anfrage Q gegen S auf einer Datenbank D, geschrieben  $Q(D)$ , ist die Menge aller Tupel, die die Ausführung von Q in D ergibt.
- Q' ist contained (enthalten) in Q, geschrieben  $Q' \subseteq Q$ , gdw.  $Q'(D) \subseteq Q(D)$  für jedes mögliche D.

Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

## LaV – Anfrageumschreibungen

39

- *Query equivalence* (Anfrageäquivalenz)
  - Q ist äquivalent mit Q', geschrieben  $Q = Q'$ ,  
gdw.  $Q(D) \subseteq Q'(D)$  und  $Q(D) \supseteq Q'(D)$   
für jede mögliche Datenbank D.
  
- Wichtig
  - Es zählt das Ergebnis einer Anfrage, nicht die Syntax.

Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

## LaV – Beispiele

40

```
SELECT K.titel, L.prof, K.kurs_id, K.univ
FROM   Leht L, Kurs K
WHERE  L.kurs_id = K.kurs_id
AND    K.univ = „Humboldt“
AND    L.univ = „Humboldt“
AND    K.titel LIKE „%VL_%“
```

⊆

```
SELECT K.titel, L.prof, K.kurs_id, K.univ
FROM   Leht L, Kurs K
WHERE  L.kurs_id = K.kurs_id
AND    K.univ = „Humboldt“
AND    L.univ = „Humboldt“
```

Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

## LaV – Beispiele

41

```
SELECT K.titel, K.kurs_id
FROM Kurs K
AND K.univ = „Humboldt“
AND K.titel LIKE „%VL_%“
```

```
SELECT K.titel, K.kurs_id
FROM Kurs K
AND K.univ = „Humboldt“
```

≠

≠

```
SELECT K.titel, K.univ
FROM Kurs K
AND K.univ = „Humboldt“
AND K.titel LIKE „%VL_%“
```

```
SELECT K.titel
FROM Kurs K
AND K.univ = „Humboldt“
```

Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

## LaV – Beispiele

42

**Quelle 3:**  
 CREATE VIEW Hum-Kurse AS  
 SELECT K.titel, K.univ  
 FROM Leht L, Kurs K  
 WHERE L.kurs\_id = K.kurs\_id  
 AND K.univ = „Humboldt“  
 AND L.univ = „Humboldt“

⊆

**Quelle 5:**  
 CREATE VIEW Kurse2 AS  
 SELECT K.titel, K.univ  
 FROM Kurs K  
 AND K.univ = „Humboldt“

Prüfung von containment durch Prüfung aller möglichen Datenbanken?

- Zu komplex!

Prüfung von containment durch Existenz eines containment mapping.

- NP-vollständig in  $|Q| + |Q'|$  nach [CM77]
- Mehrere Algorithmen

Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

## Datalog Notation

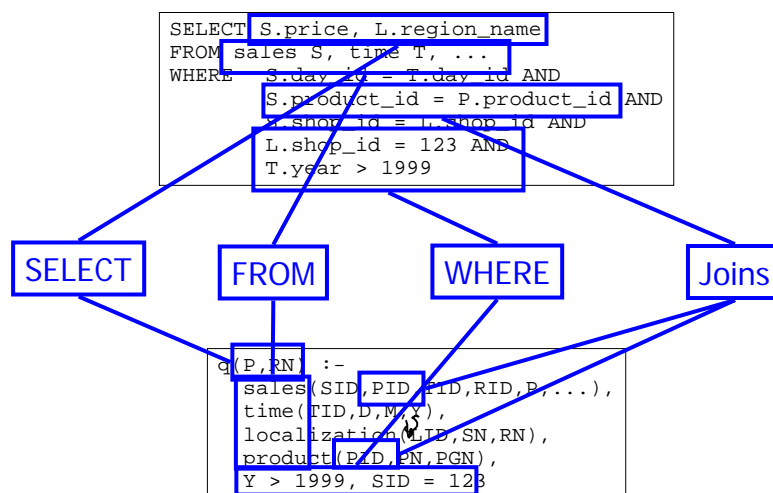
43

- Im Folgenden: Nur Konjunktive Anfragen
  - Nur Equijoins und Bedingungen mit =, <, > zw. Attribut und Konstanten
  - Kein NOT, EXISTS, GROUP BY, ≠, X>Y, ...
- Schreibweise: Datalog / Prolog
  - SELECT Klausel
    - Regelkopf, Exportierte Attribute
  - FROM Klausel
    - Relationen werden zu Prädikaten
  - WHERE Klausel
    - Joins werden durch gleiche Attributnamen angezeigt
    - Bedingungen werden explizit angegeben

Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

## SQL - Datalog

44



Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

## Query Containment

45

- A query  $p$  is contained in a query  $u$ ,  $p \subseteq u$ , iff all tuples computed by  $p$  are also computed by  $u$  for every DB.
- Beispiele (Regelkopf weggelassen)
  - $\text{map}(Mn, Ms) \subseteq \text{map}(Mn, Ms)$ ;
  - $\text{map}(Mn, Ms), Mn = \text{'HGM'} \subseteq \text{map}(Mn, Ms)$ ;
  - $\text{map}(Mn, Ms), Ms < 500 \subseteq \text{map}(Mn, Ms)$ ;
  - $\text{map}(Mn, Ms), \text{clone}(Mn, Cn, -) \subseteq \text{map}(Mn, Ms)$ ;
  - $\text{clone}(Mn, Cn, Cs), \text{clone}(Mn, Cn, Cs) \subseteq \text{clone}(Mn, Cn, Cs)$ ;

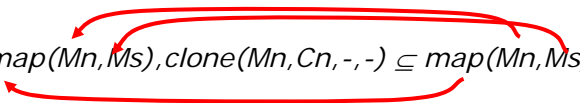
Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

## Beweis für Query Containment

46

- $p \subseteq u$  gdw ein containment mapping von  $u$  nach  $p$  existiert.
- Containment mapping:
  - $h: \text{sym}(u) \rightarrow \text{sym}(p)$  (Abbildung der Symbole)
  - CM1: Jede Konstante in  $u$  wird auf die gleiche Konstante in  $p$  abgebildet.
  - CM2: Jede exportierte Variable in  $u$  wird auf eine exportierte Variable in  $p$  abgebildet.
  - CM3: Jedes Literal (Relation) in  $u$  wird auf mindestens ein Literal in  $p$  abgebildet
  - CM4: Die Bedingungen von  $p$  implizieren die Bedingungen von  $u$
- Beweis: [CM77]

*$\text{map}(Mn, Ms), \text{clone}(Mn, Cn, -, -) \subseteq \text{map}(Mn, Ms)$*



Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

## Finden von Containment Mappings

47

Problem ist NP vollständig

- Exponentiell in der Anzahl der Literale
- Beweis: Reduktion auf „Exakt Cover“

Also: Alles ausprobieren

- Aufbau eines Suchbaums
  - Jede Ebene entspricht einem Literal
  - Auffächerung nach möglichen CMs

Algorithmus

- Nicht hier
- Siehe Buch

Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

## CM Beispiele

48

<code>product (PID, PN, PGID, ,Wasser `)</code>	$\subseteq$	<code>product (PID, PN, PGID, PGN)</code>
<code>product (PID, PN, PGID, PGN)</code>	$\not\subseteq$	<code>localization (SID, SN, RID, RN)</code>
<code>product (PID, PN, PGID, PGN) , PGN&gt;,Wasser `</code>	$\subseteq$	<code>product (PID, PN, PGID, PGN)</code>
<code>sales (SID, PID, ..., P, ...) , P&gt;80, P&lt;150</code>	$\not\subseteq$	<code>sales (SID, PID, ..., P, ...) , P&gt;100, P&lt;150</code>
<code>sales (SID, PID, ..., P, ...) , P&gt;100, P&lt;=150, P&lt;170, P&gt;=150</code>	$\not\subseteq$	<code>sales (SID, PID, ..., 150, ...)</code>
<code>sales (SID, PID, ..., P, ...) , product (PID, PN, ...)</code>	$\subseteq$	<code>sales (SID, PID, ..., P, ...)</code> (Bei Projektion auf sales)

Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005



## CM Beispiele

49

$q_1(B, D) :- \text{path}(A, B), \text{path}(B, C), \text{path}(C, D), \text{path}(D, E)$   
 $q_2(A, C) :- \text{path}(A, B), \text{path}(B, C), \text{path}(C, D)$   
 $q_1 \subseteq q_2$

$q_1(C, B) :- \text{path}(A, B), \text{path}(C, A), \text{path}(B, C), \text{path}(A, D)$   
 $q_2(X, Z) :- \text{path}(X, Y), \text{path}(Y, Z)$   
 $q_1 \subseteq q_2$

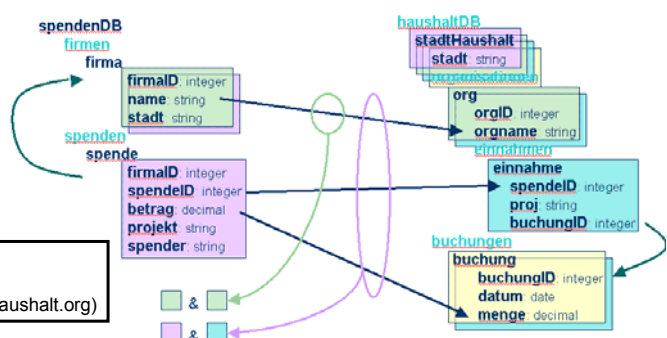
## Erzeugung der Anfragen

50

Beobachtung:  
Interpretation als  
*containment*

$\pi_{\text{name}}(\text{spendenDB.firmen})$   
 $\subseteq$   
 $\pi_{\text{orgname}}(\text{haushaltDB.stadtHaushalt.org})$

$\pi_{\text{name, spendeID, betrag}}(\text{spendenDB.firmen} \triangleright \langle \text{spendenDB.spenden})$   
 $\subseteq$   
 $\pi_{\text{orgname, spendeID, menge}}(\text{haushaltDB.stadtHaushalt.org.einnahmen} \triangleright \langle \text{haushaltDB.stadtHaushalt.buchungen})$



## Überblick

51

- Motivation
- Korrespondenzen
- Übersicht Anfrageplanung
- Global as View (GaV)
- Local as View (LaV)
  - Modellierung
  - Anwendungen
  - Anfragebearbeitung
  - Containment
- ➔ ■ Global Local as View (GLaV)
- Vergleich



Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

## Global-Local-as-View (GLaV)

52

- Kombination GaV und LaV
  - GaV:
    - Globale Relation = Sicht auf lokale Relationen
    - (Globale Relation  $\supseteq$  Sicht auf lokale Relationen)
  - LaV:
    - Sicht auf globale Relationen = lokale Relation
    - Sicht auf globale Relationen  $\supseteq$  lokale Relation
  - GLaV:
    - Sicht auf globale Relationen = Sicht auf lokale Relationen
    - Sicht auf globale Relationen  $\supseteq$  Sicht auf lokale Relationen
- Auch „BaV“: Both-as-View

Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

## Überblick

53

- Motivation
- Korrespondenzen
- Übersicht Anfrageplanung
- Global as View (GaV)
- Local as View (LaV)
  - Modellierung
  - Anwendungen
  - Anfragebearbeitung
  - Containment
- Global Local as View (GLaV)
- Vergleich



Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

## Vergleich GaV / LaV

54

### Modellierung

- GaV
  - Jede globale Relation definiert als Sicht auf eine oder mehr Relationen aus einer oder mehr Quellen.
  - Meist UNION über mehrere Quellen
  - Nebenbedingungen auf lokalen Quellen können nicht modelliert werden.
- LaV
  - Globale Relationen werden durch mehrere Sichten definiert.
  - Definition oft nur in Kombination mit anderen globalen Relationen
  - Nebenbedingungen auf globale Relationen können nicht definiert werden.

Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

## Vergleich GaV / LaV

55

- Anfragebearbeitung
  - GaV:
    - Anfrageumschreibung: Einfaches View unfolding
    - Eine große, umgeschriebene Anfrage
    - Interessante Optimierungsprobleme
  - LaV
    - Anfrageumschreibung: Answering queries using views
    - UNION über viele mögliche umgeschriebene Anfragen
    - Mehrere Algorithmen
    - Viele Anwendungen
    - Noch interessantere Optimierungsprobleme
- Flexibilität
  - GaV: Views setzen Relationen mehrerer Quellen in Beziehung
  - LaV: Jede View bezieht sich nur auf eine Quelle

Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005

## Literatur

56

Gute Zusammenfassung für LaV und weiterführende Literatur:

- [Levy01] Alon Y. Halevy: *Answering queries using views: A survey*, in VLDB Journal 10: 270-294, 2001.

Eher theoretisch

- [Ull00] Jeffrey D. Ullman: *Information Integration Using Logical Views*. TCS 2000: 189-210
- [Hull97] Managing Semantic Heterogeneity in Databases: A Theoretical Perspective. Richard Hull. PODS 1997 tutorial
- [Ull97] Jeffrey D. Ullman: Information Integration Using Logical Views. [ICDT 1997](#): 19-40
- [CM77] [Ashok K. Chandra](#) and [Philip M. Merlin](#). Optimal implementation of conjunctive queries in relational data bases. In Conference Record of the Ninth Annual ACM Symposium on Theory of Computing, pages 77-90, Boulder, Colorado, 2-4 May 1977.
- [LMSS95] Alon Y. Levy, [Alberto O. Mendelzon](#), [Yehoshua Sagiv](#), [Divesh Srivastava](#): Answering Queries Using Views. [PODS 1995](#): 95-104

Felix Naumann, VL Informationsintegration, WS 05/06 22.11.2005