Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

Informationsintegration

Materialisierte vs. Virtuelle Integration

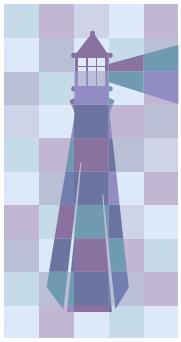
1.11.2005

Felix Naumann

Ankündigungen

Überblick

- Szenarien der Informationsintegration
  - Data Warehouse
  - Föderierte Datenbanken
- Einführung
- Materialisiert
  - Data Warehouse
- Virtuell
  - Mediator-Wrapper System
- Vergleich
  - Flexibilität
  - Antwortzeiten
  - Aktualität
  - etc.



Real-life Informationsintegration

Überblick: Zwei wesentliche Modelle

- Data Warehouses
  - Materialisierte Integration
  - Am Beispiel Buchhändler (Folien von Prof. Leser)
- Föderierte Datenbanken
  - Virtuelle Integration
  - Am Beispiel einer Life Sciences DB (DiscoveryLink)
  - Weitere Beispiele

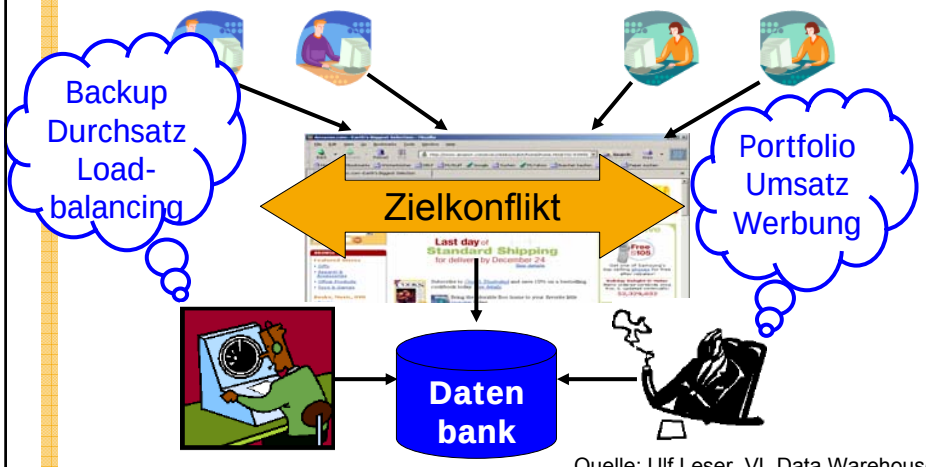
Data Warehouse

- Eine oder mehrere (ähnliche) Datenbanken mit Bücherverkaufsinformationen
- Daten werden oft aktualisiert
  - Jede Bestellung einzeln
  - Katalog Updates täglich
- Management benötigt Entscheidungshilfen (decision support)
- Komplexe Anfragen

Quelle: Ulf Leser, VL Data Warehouses

Bücher im Internet bestellen

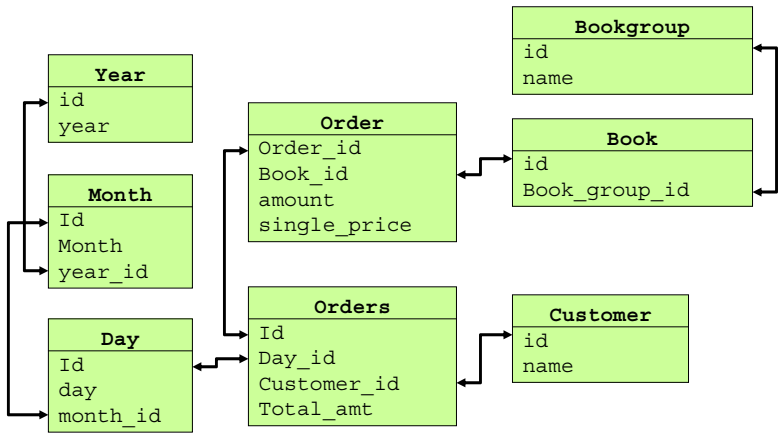
5



Quelle: Ulf Leser, VL Data Warehouses

Die Datenbank dazu

6

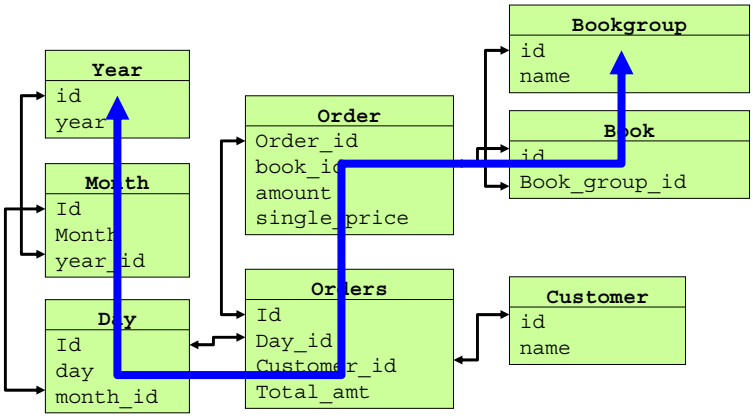


Quelle: Ulf Leser, VL Data Warehouses

Fragen eines Marketingleiters

7

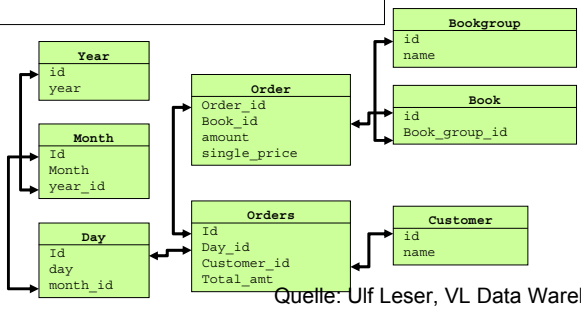
Wie viele Bestellungen haben wir jeweils im Monat vor Weihnachten, aufgeschlüsselt nach Produktgruppen?



Quelle: Ulf Leser, VL Data Warehouses

8

```
SELECT  Y.year, PG.name, count(B.id)
FROM    year Y, month M, day D, order O,
        orders OS, book B, bookgroup BG
WHERE   M.year = Y.id and
        M.id = D.month and
        O.day_id = D.id and
        OS.order_id = O.id and
        B.id = O.book_id and
        B.book_group_id = BG.id and
        day < 24 and month = 12
GROUP BY Y.year, PG.product_name
ORDER BY Y.year
```



Quelle: Ulf Leser, VL Data Warehouses

9

```
SELECT  Y.year, PG.name, count(B.id)
FROM    year Y, month M, day D, order O, orders OS,
        book B, bookgroup BG
WHERE   M.year = Y.id and
        M.id = D.month and
        O.day_id = D.id and
        OS.order_id = O.id and
        B.id = O.book_id and
        B.book_group_id = BG.id and
        day < 24 and month = 12
GROUP BY Y.year, PG.product_name
ORDER BY Y.year
```

6 Joins

- Year: 10 Records
- Month: 120 Records
- Day: 3650 Records
- Orders: 36.000.000
- Order: 72.000.000
- Books: 200.000
- Bookgroups: 100

Problem!

- Schwierig zu optimieren (Join-Reihenfolge)
- Je nach Ausführungsplan riesige Zwischenergebnisse
- Ähnliche Anfragen – ähnlich riesige Zwischenergebnisse

Quelle: Ulf Leser, VL Data Warehouses

10

In Wahrheit … noch schlimmer

Es gibt noch:

- Amazon.de
- Amazon.fr
- Amazon.it
- …

Verteilte Ausführung

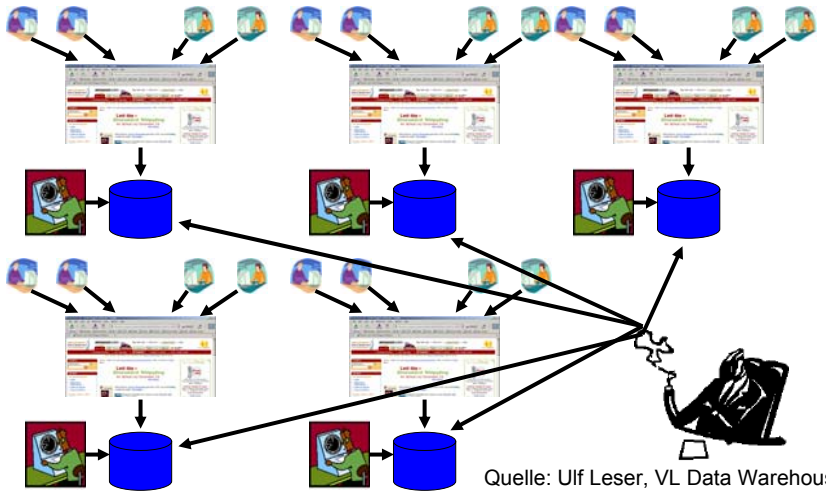
- Count über Union mehrerer gleicher Anfragen in unterschiedlichen Datenbanken

HILFE!

Quelle: Ulf Leser, VL Data Warehouses

In Wahrheit ...

11



Quelle: Ulf Leser, VL Data Warehouses

Technisch: Eine VIEW

12

```
CREATE VIEW christmas AS
        SELECT   Y.year, PG.name, count(B.id)
FROM             DE.year Y, DE.month M, DE.day D, DE.order O, ...
WHERE    M.year = Y.id and
...
GROUP BY Y.year, PG.product_name
ORDER BY         Y.year
UNION
        SELECT   Y.year, PG.name, count(B.id)
FROM             EN.year Y, EN.month M, EN.day D, DE.order O, ...
WHERE    M.year = Y.id and
...
```

```
SELECT      year, name, count(B.id)
FROM        christmas
GROUP BY    year, name
ORDER BY    year
```

Quelle: Ulf Leser, VL Data Warehouses

Probleme

13

Count über Union über verteilte Datenbanken?

- Integrationsproblem

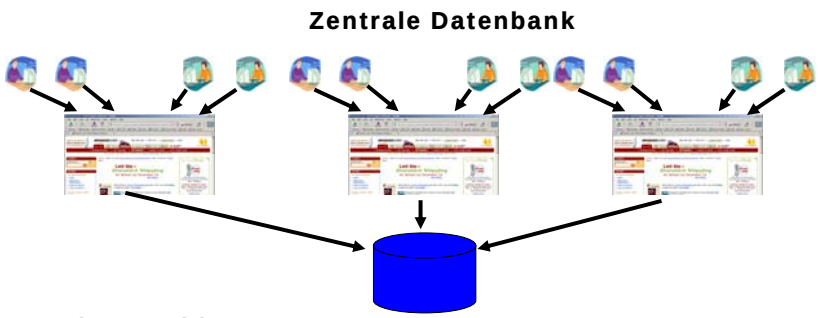
Berechnung riesiger Zwischenergebnisse bei jeder Anfrage?

- Datenmengenproblem

Quelle: Ulf Leser, VL Data Warehouses

Lösung des Integrationsproblems?

14



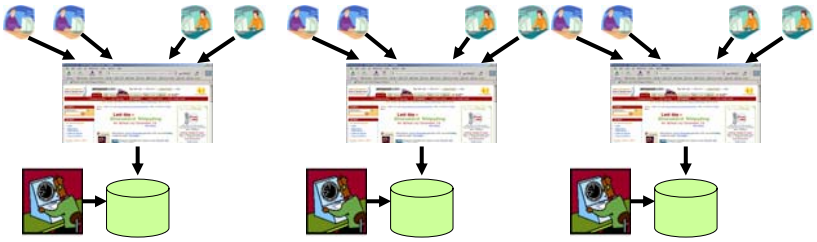
- Aber Probleme:
  - Zweigstellen schreiben übers Netz
  - Schlechter Durchsatz
  - Lange Antwortzeiten im operativen Betrieb

Quelle: Ulf Leser, VL Data Warehouses

Lösung Datenmengenproblem?

15

Denormalisierte Schema



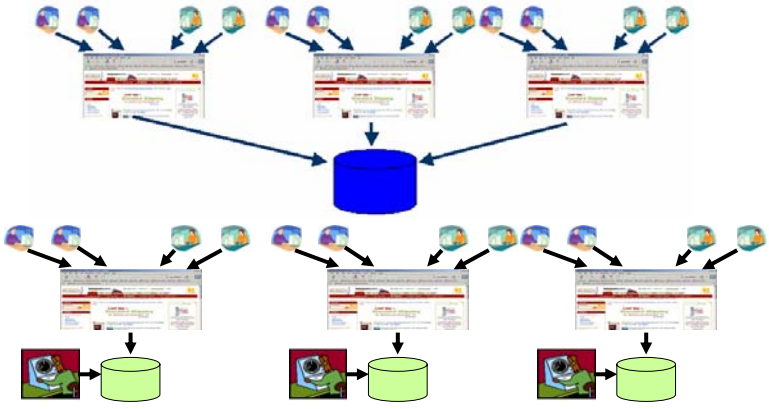
Aber Probleme:

- Jeder lesende / schreibende Zugriff erfolgt auf eine Tabelle mit 72 Mill. Records
- Lange Antwortzeiten im operativen Betrieb

Quelle: Ulf Leser, VL Data Warehouses

Zielkonflikt

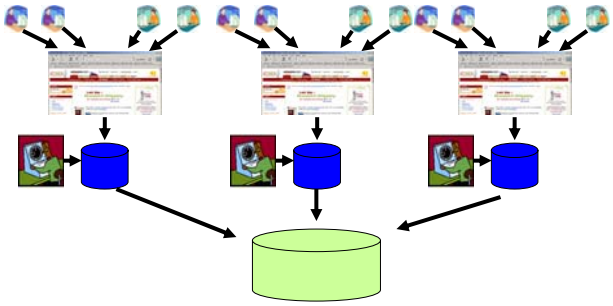
16



Tatsächliche Lösung

17

Aufbau eines Data Warehouse



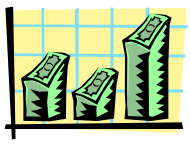
- Redundante, transformierte Datenhaltung
- Asynchrone Aktualisierung

Quelle: Ulf Leser, VL Data Warehouses

Weitere Anwendungsgebiete: Data Warehouses

18

- „Customer Relationship Management“ (CRM)
  - Identifikation von Premiumkunden
  - Personalisierung / Automatische Kundenberatung
  - Gezielte Massen-Mailings (Direktvertrieb)
- Controlling / Rechnungswesen
  - Kostenstellen
  - Organisationseinheiten
  - Personalmanagement
- Logistik
  - Flottenmanagement, Tracking
- Gesundheitswesen
  - Studienüberwachung, Patiententracking

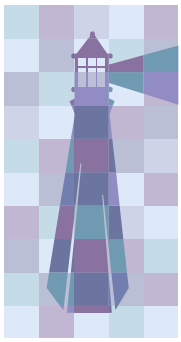


Quelle: Ulf Leser, VL Data Warehouses

Überblick

19

- Szenarien der Informationsintegration
  - Data Warehouse
  - Föderierte Datenbanken
- Einführung
- Materialisiert
  - Data Warehouse
- Virtuell
  - Mediator-Wrapper System
- Vergleich
  - Flexibilität
  - Antwortzeiten
  - Aktualität
  - etc.



Föderierte Datenbanken

20

- Mehrere autonome Informationsquellen
- Mit unterschiedlichsten Inhalten
  - Gene, Proteine, BLAST, etc.
- Und unterschiedlichsten Schnittstellen
  - HTML-Form, flat file, SQL, etc.
- Wissenschaftler (Biologe) benötigt z.B. möglichst viele Informationen über ein bestimmtes Protein
  - Funktion, Veröffentlichungen, verwandte Proteine usw.
- Sehr komplexe Anfragen
- Üblicher Ansatz: Browsing, Note-Taking, Copy & Paste
- Föderierte Datenbanken (wie DiscoveryLink) helfen.

Frage eines Biologen

21

Finde alle menschlichen EST Sequenzen, die nach BLAST zu mindestens 60% über mindestens 50 Aminosäuren identisch sind mit mouse-channel Genen im Gewebe des zentralen Nervensystems.

Quelle für das komplette Beispiel: *A Practitioner's Guide to Data Management and Data Integration in Bioinformatics,* Barbara A. Eckman in Bioinformatics by Zoe Lacroix and Terence Critchlow, 2003, Morgan Kaufmann.

Verschiedene Informationsquellen

22

Beteiligte Informationsquellen

- Mouse Genome Database (MGD) @ Jackson Labs
- SwissProt @ EBI
- BLAST tool @ NCBI
- GenBank nucleotide sequence database @ NCBI

Alle Quellen sind frei verfügbar

Herkömmlicher Ansatz: Browsing

23

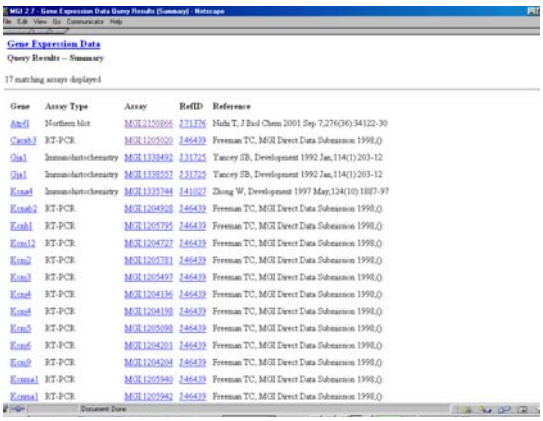
1. Suche „channel“ Sequenzen im Gewebe des ZNS durch MGD HTML Formular

Herkömmlicher Ansatz: Browsing

24

MGD Resultat

- 14 Gene aus 17 Experimenten



Herkömmlicher Ansatz: Browsing

25

- Details zu jedem der 14 Gene ansehen
- Durchschnittlich fünf SwissProt Links pro Gen

Herkömmlicher Ansatz: Browsing

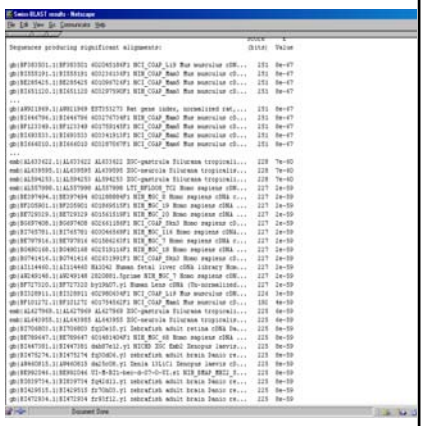
26

- Betrachten jedes SwissProt Eintrages
- Durch Klick BLAST Algorithmus anwerfen

Herkömmlicher Ansatz: Browsing

27

- Betrachten jedes BLAST Resultats um
  - nicht-menschliche Treffer zu eliminieren,
  - andere Bedingungen zu prüfen (>60% Identität, etc.).



Herkömmlicher Ansatz: Browsing

28

Für jeden verbleibenden Eintrag

- Komplette EST Sequenz bei GenBank holen

Alles sehr mühselig!

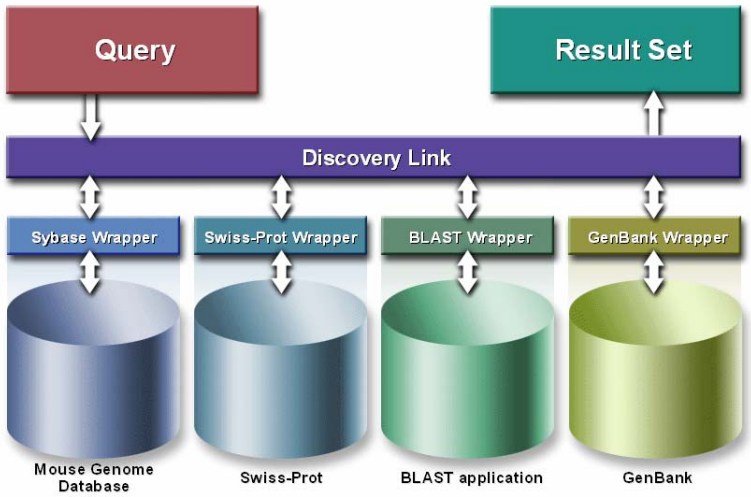
Idee der Integration

29

- Bildung eines globalen Schemas (Schemaintegration)
  - Gespeichert als Datenbankschema in DiscoveryLink
- Generierung von Wrappern für jede Datenquelle
  - Softwarekomponente
  - Mapping von lokalen Schemata auf globales Schema
  - Kennt Anfragefähigkeiten der Quellen

DiscoveryLink Architektur

30



Eigenschaften föderierter IS (und DiscoveryLink)

- Daten bleiben vor Ort.
- Informationsquellen sind autonom (und wissen oft nicht von ihrer Integration).
- Anfragen werden deklarativ an das globale Schema gestellt.
- Anfrage wird so verteilt wie möglich ausgeführt.
  - Je nach Mächtigkeit der Quellen
  - DiscoveryLink gleicht etwaige mangelnder Fähigkeiten aus.

Föderierter DBMS Ansatz

„Finde alle menschlichen EST Sequenzen, die nach BLAST zu mindestens 60% über mindestens 50 Aminosäuren identisch sind mit mouse-channel Genen im Gewebe des zentralen Nervensystems."

„Einfache" SQL-Anfrage um alle vorigen Schritte zu vereinen:

```
SELECT g.accnum,g.sequence
FROM   genbank g, blast b, swissprot s, mgd m
WHERE  m.exp = "CNS"
AND          m.defn LIKE "%channel%"
AND          m.spid = s.id AND s.seq = b.query
AND    b.hit = g.accnum
AND    b.percentid > 60 AND b.alignlen > 50
```

Föderierter DBMS Ansatz

- Effiziente Ausführung durch Optimierer
  - Herkömmliche Optimierung
  - Wrapper helfen mit
    - Kostenmodell
    - domänenspezifischen Funktionen
- Sichere Ausführung
  - Wiederholbar
  - Transaktional

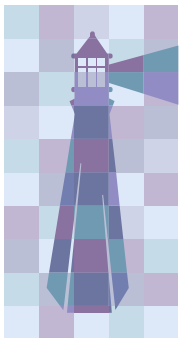
Weitere Anwendungsgebiete: Föderierte Datenbanken

- Meta-Suchmaschinen
- Unternehmensfusionen
  - Kundendatenbanken
  - Personaldatenbanken
- Grid
- Krankenhausinformationssysteme
  - Röntgenbilder
  - Krankheitsverlauf (Akte)
  - Verwaltung
  - Krankenkasse…
- Verteiltes Arbeiten („groupware“)
- Peer Data Management und P2P

Überblick

35

- Szenarien der Informationsintegration
  - Data Warehouse
  - Föderierte Datenbanken
- Einführung
- Materialisiert
  - Data Warehouse
- Virtuell
  - Mediator-Wrapper System
- Vergleich
  - Flexibilität
  - Antwortzeiten
  - Aktualität
  - etc.



Integration

36

Materialisiert

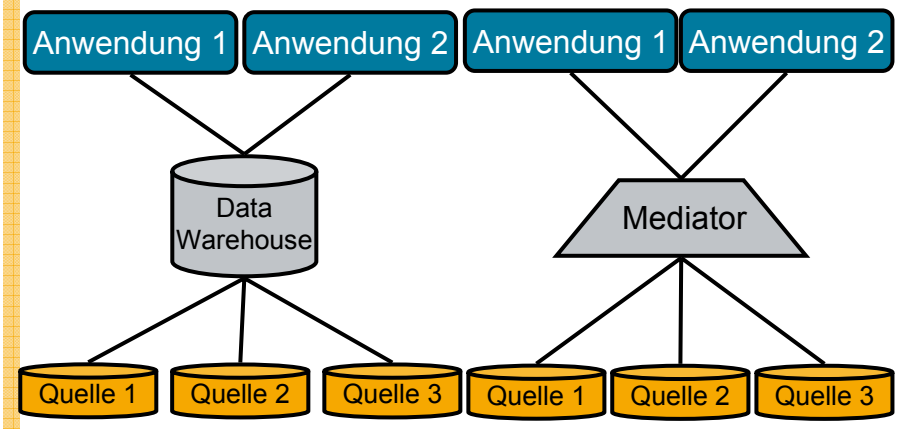
- A priori Integration
- Zentrale Datenbasis
- Zentrale Anfragebearbeitung
- Typisches Beispiel: Data Warehouse

Virtuell

- On demand Integration
- Dezentrale Daten
- Dezentrale Anfragebearbeitung
- Typisches Beispiel: Mediator-basiertes Informationssystem

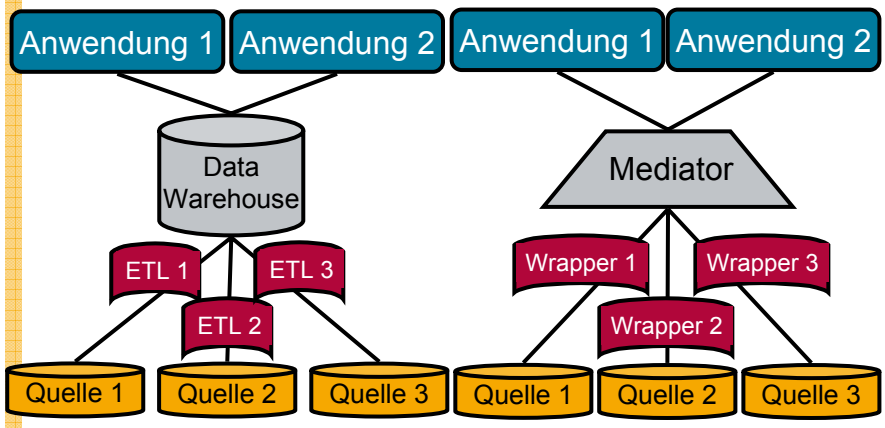
Data Warehouse vs. Mediator-basiertes Informationssystem

37



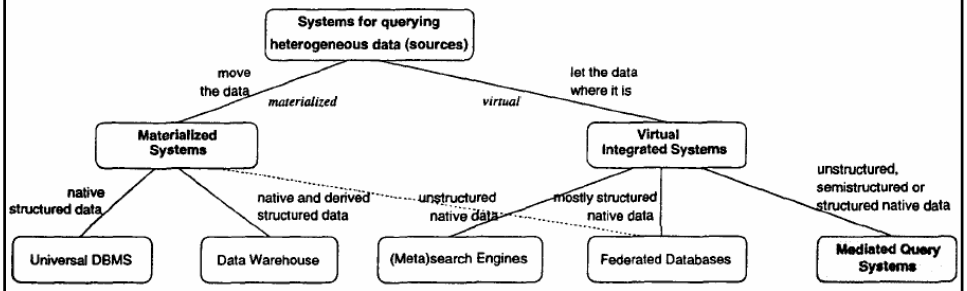
Data Warehouse vs. Mediator

38



Taxonomie nach [DD99]

39



Data Warehouse vs. Mediator

40

Jetzt jeweils kurzer Überblick

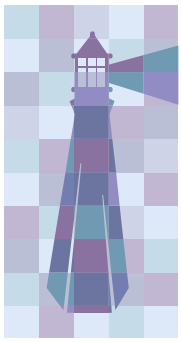
- Datenfluss
- Anfragebearbeitung
- Entwurf und Entwicklung (Schema)

Details in den folgenden Wochen

Überblick

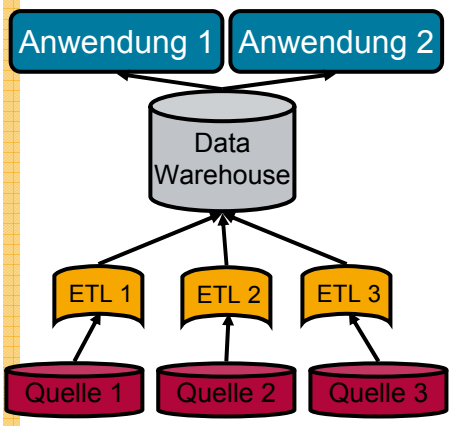
41

- Szenarien der Informationsintegration
  - Data Warehouse
  - Föderierte Datenbanken
- Einführung
- Materialisiert
  - Data Warehouse
- Virtuell
  - Mediator-Wrapper System
- Vergleich
  - Flexibilität
  - Antwortzeiten
  - Aktualität
  - etc.



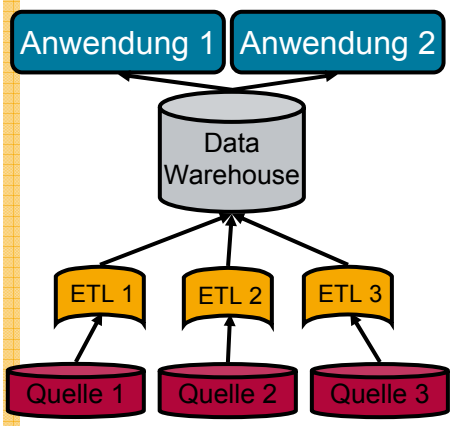
Materialisierte Integration - Datenfluss

42



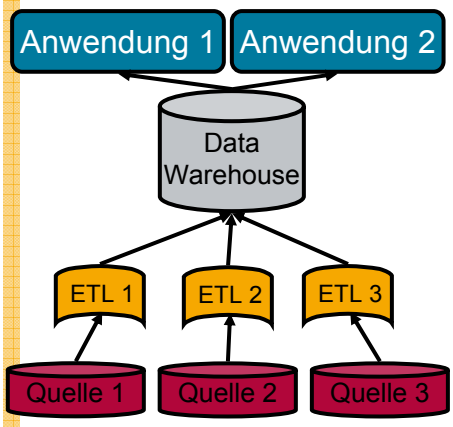
- Push
- Erstmalige „Bevölkerung“ (population) des DW
  - Data Cleansing
- Periodischer Datenimport
  - Stündlich / Täglich / Wöchentlich
  - Materialisierte Sichten / Sicht-Updates
- Redundante Datenhaltung
- Aggregation und Löschung alter Daten
  - Je älter, desto „aggregierter“

Materialisierte Integration - Anfragebearbeitung



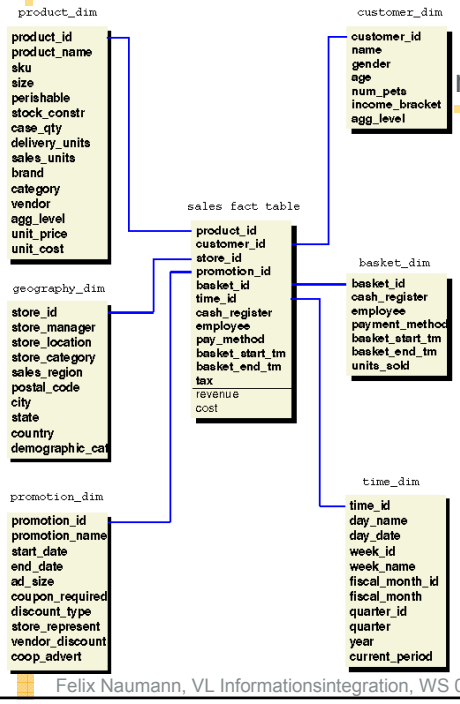
- Wie „normale“ DBMS
- Besonderheiten
  - Star Schema
  - Aggregation
  - Decision Support
- Siehe auch VL DWH

Materialisierte Integration - Schema



- Bottom-Up Entwurf
- Schemaintegration
- Star-Schema
  - *Fact-Table*
  - *Dimension Tables*

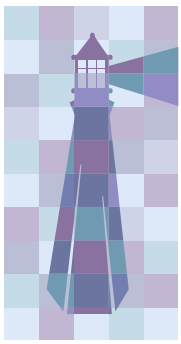
n - Schema



- Bottom-Up Entwurf
- Schemaintegration
- Star-Schema
  - *Fact-Table*
  - *Dimension Tables*

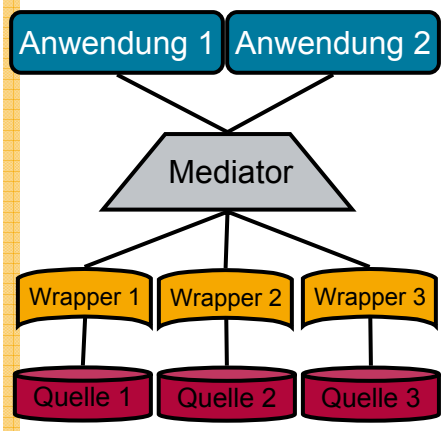
Überblick

- Szenarien der Informationsintegration
  - Data Warehouse
  - Föderierte Datenbanken
- Einführung
- Materialisiert
  - Data Warehouse
- Virtuell
  - Mediator-Wrapper System
- Vergleich
  - Flexibilität
  - Antwortzeiten
  - Aktualität
  - etc.



Virtuelle Integration - Datenfluss

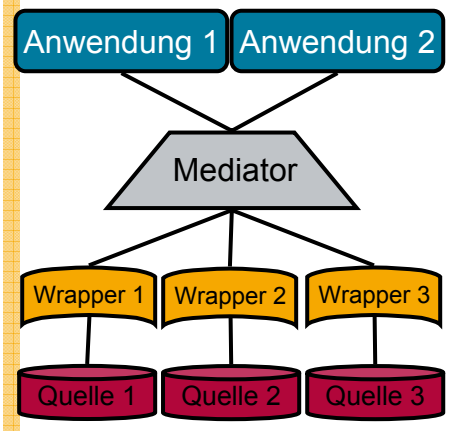
47



- Pull
- Daten sind in Quellen gespeichert.
- Nur die zur Anfragebeantwortung notwendigen Daten werden übertragen.
- Data Cleansing nur online möglich.

Virtuelle Integration - Anfragebearbeitung

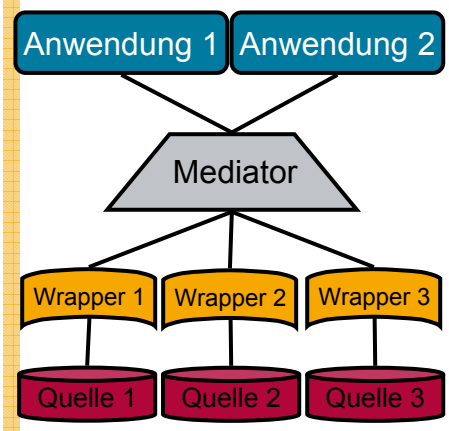
48



- Optimierung schwierig
  - Fähigkeiten der Quellen
  - Geschwindigkeit der Quellen
- Viele mögliche Pläne
  - Redundante Quellen
  - Redundante Pläne
- Dynamisch, um ausfallende Quellen auszugleichen

Virtuelle Integration - Schema

49

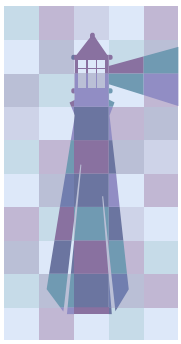


- Top-Down Entwurf
- Leicht erweiterbar
  - Global: Neue Quellen suchen
  - Lokal: Nur ein *mapping* verändern.
- Schema Mapping statt Schema-integration

Überblick

50

- Szenarien der Informationsintegration
  - Data Warehouse
  - Föderierte Datenbanken
- Einführung
- Materialisiert
  - Data Warehouse
- Virtuell
  - Mediator-Wrapper System
- Vergleich
  - Flexibilität
  - Antwortzeiten
  - Aktualität
  - etc.



Dimensionen des Vergleichs

51

- Aktualität
- Antwortzeit
- Flexibilität / Wartbarkeit
- Komplexität
- Autonomie
- Anfragebearbeitung / Mächtigkeit
- Read / Write
- Größe / Speicherbedarf
- Ressourcenbedarf
- Vollständigkeit
- Data Cleansing
- Informationsqualität

Aktualität (up-to-date-ness)

52

Materialisierte Integration

- Je nach Update-Frequenz
- In Unternehmen meist täglich (über Nacht)
- Beispiel SwissProt
  - Updates in SwissProt täglich
  - Aber: Release nur monatlich

Virtuelle Integration

- Sehr gut
- Abhängig von Aktualität der autonomern Quellen
- Manchmal: Caching

Antwortzeit (response time)

Materialisierte Integration

- Sehr gut
- Lokale Bearbeitung
- Wie DBMS
  - Optimierung
  - Materialisierte Sichten
  - Indices
  - ...
- Allerdings: Typische Anfragen sind komplex

Virtuelle Integration

- Nicht gut
- Daten sind entfernt
  - Übertragung durch das Netz
- Abhängig von Antwortzeit der Quellen
- Optimierung schwierig
- Komplexe Operatoren müssen naïv ausgeführt werden.
- Data Cleansing Operationen müssen nachgeholt werden.

Flexibilität / Wartbarkeit (flexibility / maintenance)

Materialisierte Integration

- Schwierig
- Entfernen / Ändern / Hinzufügen einer Quelle kann gesamte Integration verändern (bei GaV)
- Lokale Wartung eines großen und wachsenden Datenbestandes
  - Mit Indices etc.
- Tägliche Integration nötig

Virtuelle Integration

- Einfacher
- Entfernen / Ändern / Hinzufügen einer Quelle wirkt sich nur auf das mapping dieser Quelle aus (bei LaV)
- Quellen müssen Daten selbst warten.
  - Backups, DBMS Wartug etc.

Komplexität (complexity)

55

Materialisierte Integration

- Wie DBMS
- Komplexe Anfragen
- Anfrageplanung im GaV leicht
- Quellen sind oft untereinander ähnlich.
  - Oft sind es selbst DBMS

Virtuelle Integration

- Modellierung der Quellen wichtig
  - Fähigkeiten der Quellen
- Anfrageplanung in LaV schwierig
- Oft verschiedenste Quellen
  - Web Services
  - HTML Formulare
  - Flat Files
  - ...

Autonomie (autonomy)

56

Materialisierte Integration

- Quellen wenig autonom
  - Keine Kommunikations-autonomie
  - Geringe Ausführungs-autonomie
  - Geringe Design-autonomie
- Müssen bulk-read o.ä. zulassen
- Update notifications

Virtuelle Integration

- Quellen können autonom sein.
- Volle Design-Autonomie
- Fast volle Kommunikations-Autonomie
  - Gewisse Kommunikation ist nötig, sonst nicht Teilnehmer der Integration
- Fast volle Ausführungs-Autonomie
  - Nur: Anfragen müssen irgendwann beantwortet werden.

Anfragebearbeitung / Mächtigkeit (query planning / expressivenes)

Materialisierte Integration

- Anfragebearbeitung wie DBMS bzw. anderes globales System
- Anfragemächtigkeit wie globales System
  - z.B. volle SQL Mächtigkeit

Virtuelle Integration

- Anfragebearbeitung komplex
  - Verteilung
  - Autonomie
  - Heterogenität
- Mangelnde Fähigkeiten der Quellen können global eventuell ausgeglichen werden.
- Aber auch: Spezialfähigkeiten der Quellen können genutzt werden:
  - Image retrieval
  - Text Index

Lesen / Schreiben (Read / Write)

Materialisierte Integration

- Read immer möglich
- DW: Write oft nicht gewünscht, aber möglich
  - Kann zu Inkonsistenz mit Quellen führen

Virtuelle Integration

- Read meist möglich
- Verfügbarkeit!
- Write meist nicht möglich
  - Bei Redundanz: Wohin schreiben?
  - Transaktionen schwierig
  - Autonomie

Größe / Speicherbedarf (size / memory consumption)

Materialisierte Integration

- Hoch
  - Redundante Datenhaltung
  - DW: Historische Daten
- Wachstum
  - Stetig wachsend
  - Oder konstant durch zunehmende Aggregation im Laufe der Zeit
- *Footprint*: wie DBMS

Virtuelle Integration

- Gering
  - Metadaten
  - Cache
  - Zwischenergebnisse
- *Footprint*: wie DBMS

Ressourcenbedarf (*resource consumption*)

Materialisierte Integration

- Planbare Netzwerklast
- Daten werden eventl. unnötig übertragen
  - Abhängig von Anfrage
  - Aggregation
  - Pre-Aggregation

Virtuelle Integration

- Potentiell hohe Netzwerklast
- Daten werden mehrfach übertragen.
  - Cache kann helfen.
- Nur jeweils nötige Daten werden übertragen.

Je nach *Workload*.
Spannendes Optimierungsproblem!

Vollständigkeit (completeness)

61

Materialisierte Integration

- Gut
- Annahme: Materialisation ist vollständig

Virtuelle Integration

- Nur bei Verfügbarkeit aller nötigen Quellen
- Gegebenenfalls Anfrage unbeantwortbar oder nur unvollständig beantwortbar
  - Fuzzy Anfragesemantik:
    - Alle Tupel?
    - Alle Attribute?
- Definition der Vollständigkeit
  - Open World Assumption
  - Closed World Assumption

Datenreinigung (Data Cleansing)

62

Materialisierte Integration

- Viele Methoden
  - Aufwändig
- Offline (über Nacht)

Virtuelle Integration

- Online cleansing schwierig
  - Aufwand
  - Keine Interaktion mit Experten möglich

Informationsqualität (*information quality*)

Materialisierte Integration

- Hoch
- Kontrolliert
- Kann bei Bedarf verbessert werden.

Virtuelle Integration

- Abhängig von Quellen
- Oft zweifelhaft
  - Autonomie

Zusammenfassung Vor- und Nachteile

	Materialisiert	**Virtuell**
Aktualität	- (Cache)	+
Antwortzeit	+	-
Flexibilität	- (GaV)	+ (LaV)
Komplexität	-	--
Autonomie	-	+
Anfragemächtigkeit	+	-
Read/Write	+/+	+/-
Größe	-	+
Ressourcenbedarf	? (workload)	? (workload)
Vollständigkeit	+	? (OWA, CWA)
Datenreinigung	+	-
Informationsqualität	+	-

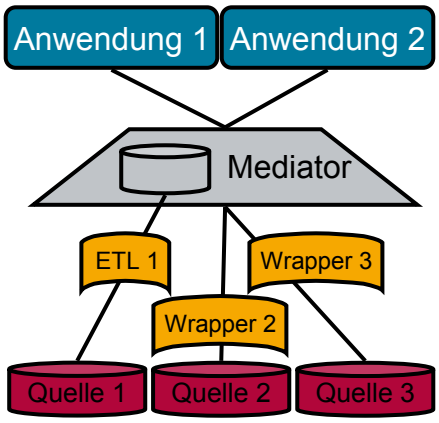
Hybrider Ansatz

65

Teile der Daten werden materialisiert
- Oft benötigte Daten (Cache)
- Als bulk verfügbare Daten
  - Dump Files
  - SQL Zugang
  - ...

Teile der Daten bleiben bei den Quellen
- Oft aktualisierte Daten
- Daten mit beschränktem Zugang
  - mind. eine gebundene Variable
  - Beschränkte Lizenzen

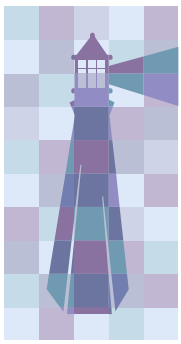
Optimierung bevorzugt lokale Daten
- Prüfung, ob Aktualisierung vorliegt



Rückblick

66

- Szenarien der Informationsintegration
  - Data Warehouse
  - Föderierte Datenbanken
- Einführung
- Materialisiert
  - Data Warehouse
- Virtuell
  - Mediator-Wrapper System
- Vergleich
  - Flexibilität
  - Antwortzeiten
  - Aktualität
  - etc.



Literatur

- [BKLW99] Busse, Kutsche, Leser, Weber, Federated Information Systems: Concepts, Terminology and Architectures. Forschungsbericht 99-9 des FB Informatik der TU Berlin, 1999. Online: http://www.informatik.hu-berlin.de/~leser/publications/tr_terminology.ps
- [DD99] Ruxandra Domenig, Klaus R. Dittrich: An Overview and Classification of Mediated Query Systems. SIGMOD Record 28(3): 63-72 (1999)