# Materialisierte Sichten

*Seminar Business Intelligence* - Teil I:
*OLAP & Data Warehousing*

Sebastian Benz

25.07.2003

# Gliederung

1. Materialisierte Sichten
   - Prinzip und Anwendung
2. Auswahl zu materialisierender Sichten
   - Statische vs. Dynamische Auswahl
3. Aktualisierung
   - Inkrementelle Aktualisierung
4. Konsistenz
   - Aspekte und Konzepte
5. Zusammenfassung

# Motivation

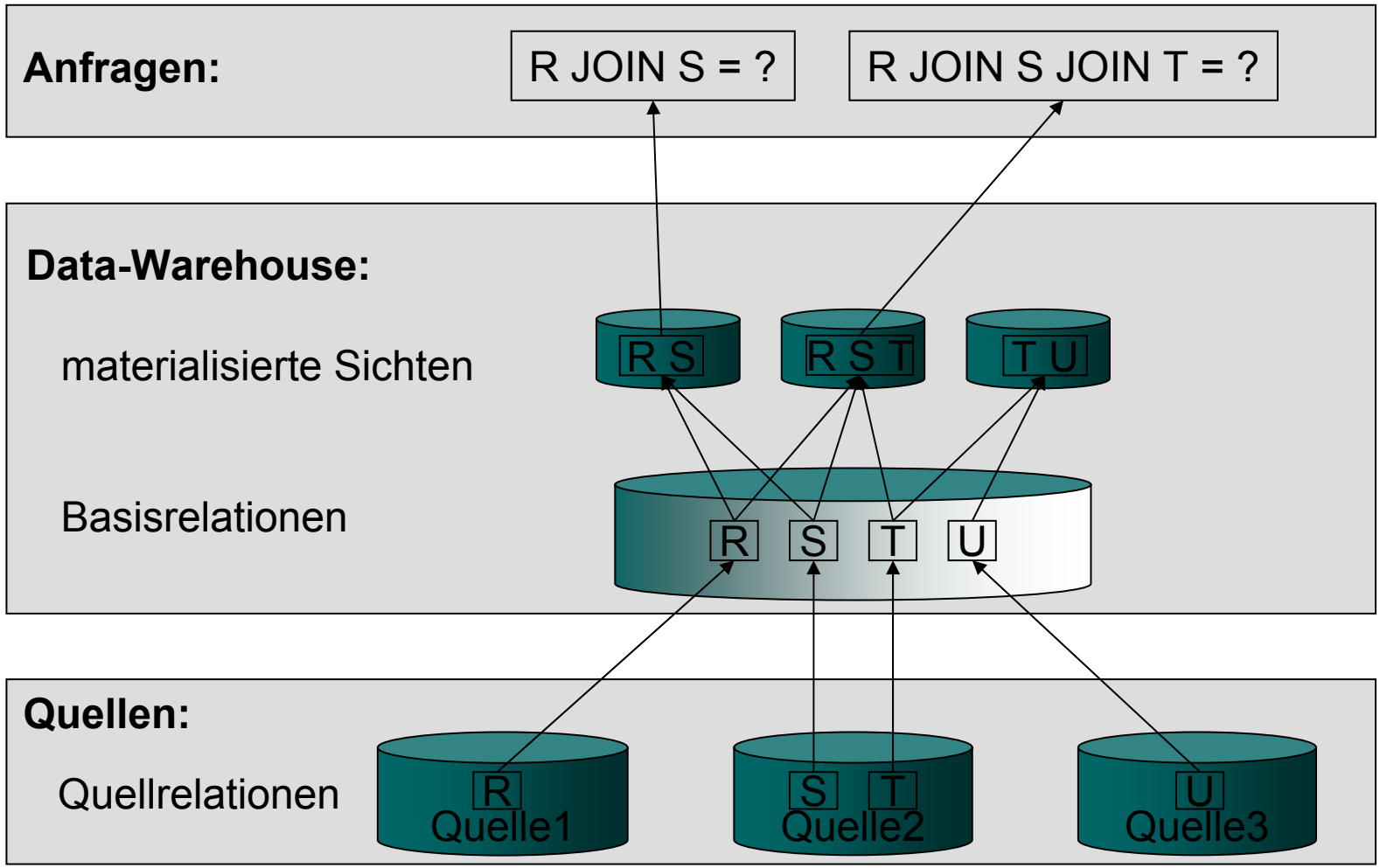
- Hauptzweck von OLAP Analyse von Daten
  - → hauptsächlich lesender Zugriff
- Sehr große Datenmengen
  - → aufwändige Anfrageauswertung
- Oft Aggregationsanfragen
  - → kleinere Anfrageergebnisse

→ möglichst gute Anfrage- und Antwortperformanz nötig

→ häufig benutze Anfragen im Data-Warehouse materialisieren

→ materialisierte Sichten (MS)

# Materialisierte Sichten



# Anfrageumformulierung

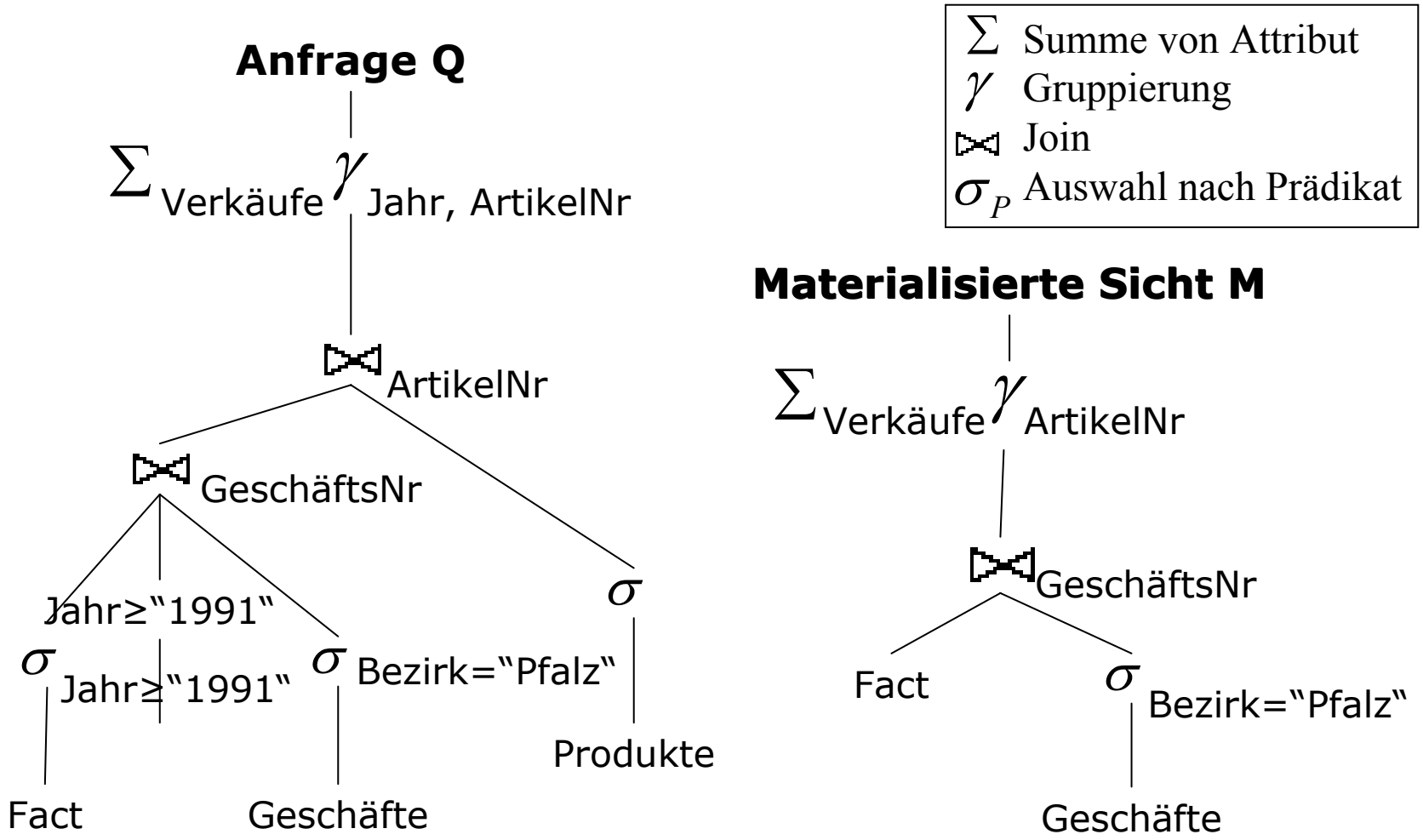
Vorausetzungen :

- Transparente Nutzung der MS
- Neue Anfrage ist gültige Ersetzung der alten Anfrage
- Additivität der Aggregationsfunktionen

Änderung der Anfrage mit Hilfe von Kompensationsoperationen, so dass sie eine materialisierte Sicht nutzen kann

→ Beispielverfahren: “verallgemeinerte Projektion“

# Beispiel für Nutzung der MS

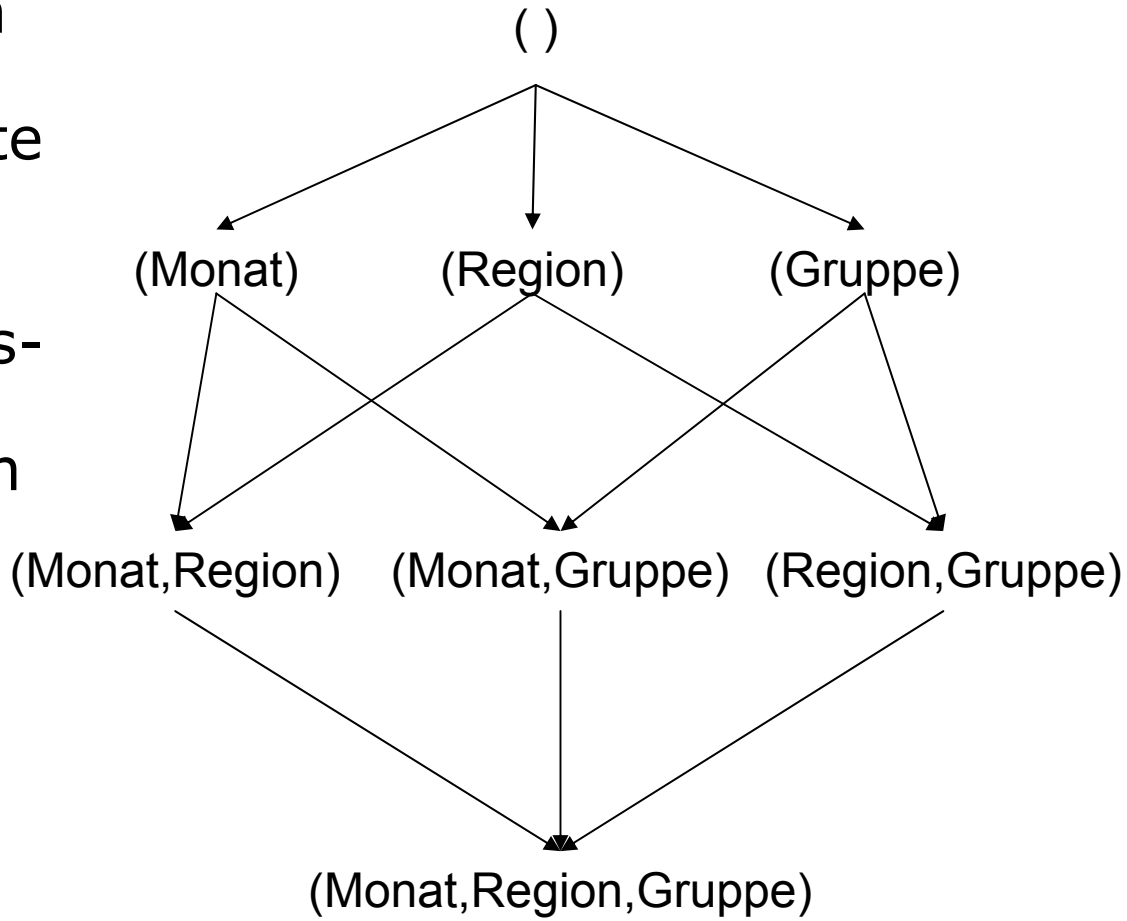


# Gliederung

1. Materialisierte Sichten
   - Prinzip und Anwendung
2. Auswahl zu materialisierender Sichten
   - Statische vs. Dynamische Auswahl
3. Aktualisierung
   - Inkrementelle Aktualisierung
4. Konsistenz
   - Aspekte und Konzepte
5. Zusammenfassung

# Aggregationsgitter

- Jeder Knoten mögliche materialisierte Sicht
- Für n Gruppierungs-attribute $2^n$ Möglichkeiten
- Welche auswählen?



# Statische Auswahl

- Anlegen der materialisierten Sichten an einem bestimmten Zeitpunkt von Algorithmus oder Administrator
- Algorithmus nach Greedy Prinzip
  - → für jeden Knoten Nutzen feststellen
  - → Knoten mit größtem Nutzen materialisieren
  - → solange bis max. Speicherplatz belegt
  - → liefert Menge an zu materialisierenden Sichten mit dem größten Nutzen

# Statische Auswahl

Beispielalgorithmus:

[Harinarayan, Rajaraman, Ullman]

- Eingabe:
  - Menge aller Gitterknoten N
  - maximaler Speichermehraufwand S
  - erwarteter Speicheraufwand |n| bei Materialisierung von Knoten n
- Ausgabe:
  - Menge aller zu materialisierenden
  - Knoten

# Statische Auswahl

Kandidatenmenge:
0.{(A1,A2,A3)}

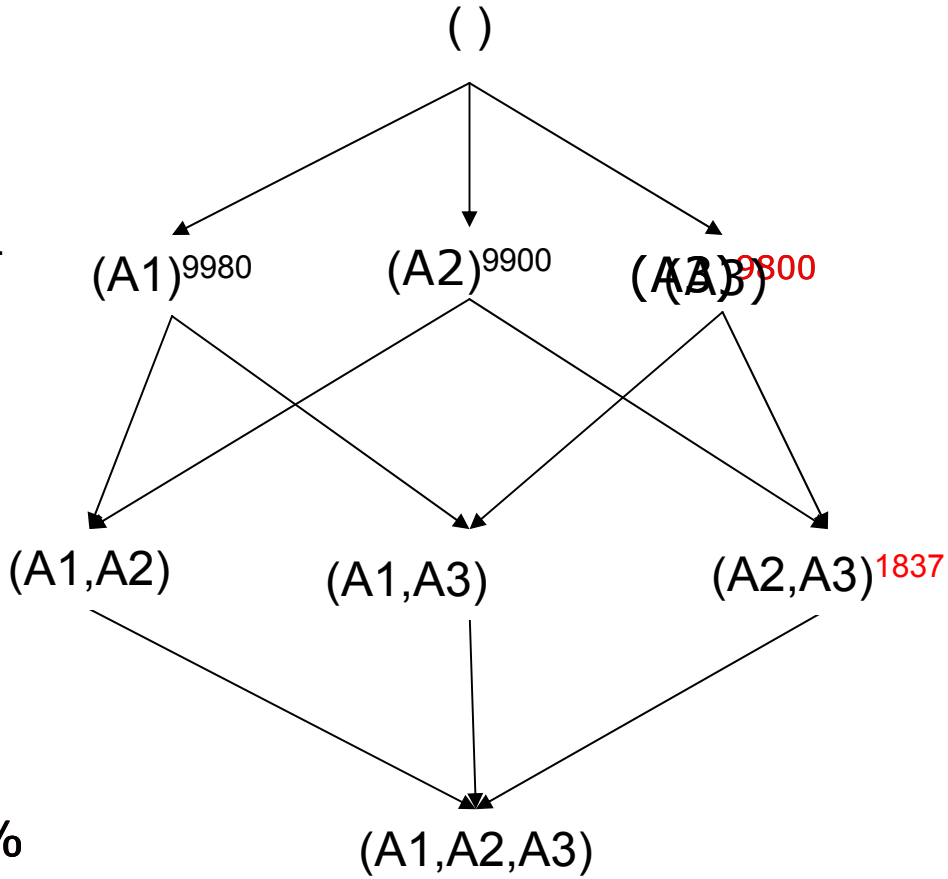
1.{(A1,A2,A3),(A1,A2)}

2.{(A1,A2,A3),(A1,A2),(A3)}

3.{(A1,A2,A3),(A1,A2),
(A3),(A1,A3)}

4.{(A1,A2,A3),(A1,A2),
(A3),(A1,A3),(A2,A3)}

Verwendeter zusätzlicher Speicher (max. 70%):



# Bewertung

- Aufwändig mit Komplexität von $O(n^3)$
  - → Optimierungen möglich
    - Nutzung von funktionalen Abhängigkeiten
    - Untere Schranke für Verdichtungsfaktor
    - Nur Auswahl von Knoten, deren Attribute in Anfragen aufgetreten sind
- Nachteile:
    - Keine Erkennung von Anfragemustern
    - Schnelle Alterung bei häufiger Aktualisierung
    - Keine Anpassung an wechselnde Anfragemuster
    - Keine Anfragebearbeitung möglich während Auswahl läuft

→ Dynamische Auswahl

# Dynamische Auswahl

Entscheidung welche Sichten materialisiert werden ist vom aktuellen Anfrageverhalten abhängig

Vorteile:

- Unterstützung von OLAP-Anwendungen (Bsp. Roll-up-Operation)
- Anfrageergebnisse oft klein
  - → Große Beschleunigung der Auswertung mit wenig Speicherplatzverbrauch

# Semantisches Caching

- Pufferung von Anfrageergebnissen anhand Semantik und Zusammenhang
- Speicherung in Hauptspeicher oder Festplatte
  - → Beschleunigung hauptsächlich durch schnelle Ermittlung der Ergebnismenge
- Speicherplatz beschränkt
  - → Verdrängungsverfahren für bestehende Sichten notwendig

# Verdrängung von MS

Verdrängungskriterien:

- Zeit des letzten Zugriffs
- Referenzierungshäufigkeit
- Größe der MS
- Kosten, die Neuberechnung/Aktualisierung der MS verursachen würde
- Anzahl Anfragen, die in Vergangenheit mit der MS hätten beantwortet werden können
- Anzahl Anfragen, die prinzipiell mit MS beantwortet werden könnten

→ Nutzwert berechnet sich aus gewichteten Faktoren

# Auswahl von Sichten

Kompromiss zwischen zusätzlichen Speicher-bedarf und Anfrageoptimierung

→ anwendungsabhängig

Frage: Wie bei Änderung der Basisdaten aktualisieren der materialisierten Sichten?

# Gliederung

1. Materialisierte Sichten
   - Prinzip und Anwendung
2. Auswahl zu materialisierender Sichten
   - Statische vs. Dynamische Auswahl
3. Aktualisierung
   - Inkrementelle Aktualisierung
4. Konsistenz
   - Aspekte und Konzepte
5. Zusammenfassung

# Rematerialisierung

Einfachste Möglichkeit:

Komplettes Löschen und Neuberechnen der Sicht nach Änderung

Nachteil:

Ineffizient, wenn sich nur ein kleiner Teil der Daten geändert hat

→ Sehr großer Datendurchsatz bei Aktualisierung

# Inkrementelle Aktualisierung

Idee: Nachvollziehen der Änderungen in der MS

Berechnung des neuen Zustands der MS aus dem alten Zustand der MS und der durchgeführten Änderung der Basisrelation

Ablauf:

1. Update Basisrelation
2. Nachricht an Sicht über Update
3. Anfragen der Sicht an andere Basisrelationen ob Aktualisierung notwendig
4. Aktualisierung der Sicht

# Beispiel

Basisrelationen B1, B2:

B1(Produkt, PGruppe)

| | |
|---|---|
| Becks | Bier |
| Cola | Softdrink |

B2(PGruppe, Herkunft)

| | |
|---|---|
| Bier | GER |
| Wein | FR |

Materialisierte Sicht V:

V(Produkt, Herkunft)

| | |
|---|---|
| Becks | GER |

Sichtdefinition:

```
CREATE VIEW V AS
(SELECT b1.Produkt, b2.Herkunft
      FROM b1, b2
      WHERE b1.PGruppe = b2.PGruppe)
```

# Beispiel

Basisrelationen:

B1(Produkt, PGruppe) :

| Becks | Bier |
|---|---|
| Cola | Softdrink |
| Perling | Wasser |
| Merlot | Wein |

Update $U_2$

Insert(B1,{Merlot, Wein})

Update $U_1$

Insert(B1,{Perling, Wasser})

Sicht V:

V(Produkt, Herkunft)

| Becks | GER |
|---|---|
| Merlot | FR |

B2(PGruppe, Herkunft)

| Bier | GER |
|---|---|
| Wein | FR |

Anfrage $Q_2$

$Q_2 = \{R\}$

SELECT Region

FROM B2

WHERE PGruppe = Wasser

# Anomalien

Problem:

Parallele Änderungen der Basisrelation(en)

→ mögliches Auftreten von Anomalien

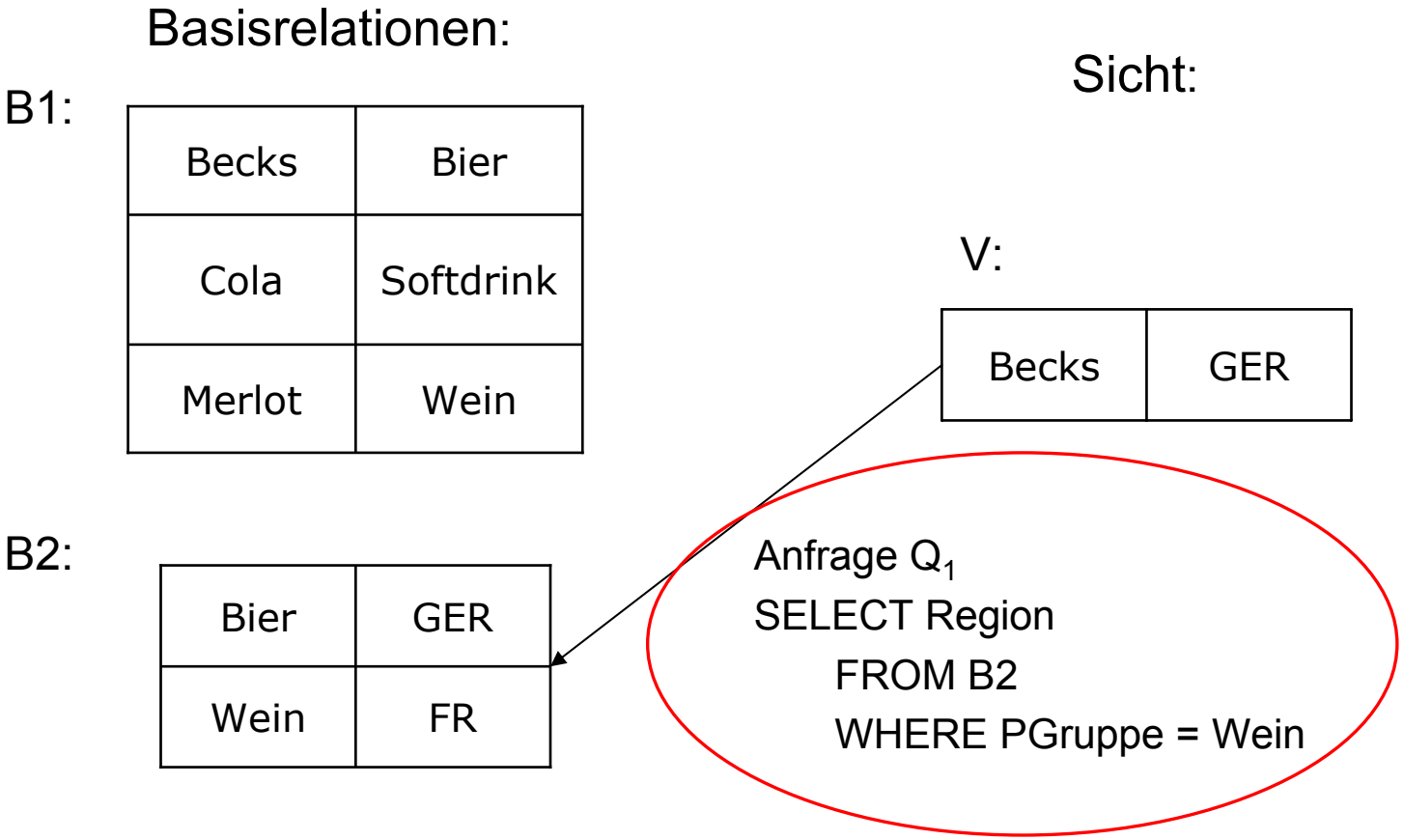
Lösung mit speziellen Algorithmen :

z.B. ECA-Verfahren:

→ Hinzufügen von Kompensationsanfragen bei Quellanfragen zum Ausgleich von parallelen Änderungen

# Optimierung der Aktualisierung

Ziel: Vermeidung von Zugriffen auf Basisrelationen



# Autonome Aktualisierung

Autonom aktualisierbar, wenn Aktualisierung ohne Zugriffe auf Basisrelationen stattfindet

- → In Praxis nur Forderung nach partieller Autonomie (nicht für alle Operationen)

Nutzung von Zusatzinformationen:

- Schemainformationen (Bsp. Primärschlüsseleigenschaften)
- Count einzelner Tupel (→ Counting Algorithmus)
- Verwendung von Hilfssichten

# Gliederung

1. Materialisierte Sichten
   - Prinzip und Anwendung
2. Auswahl zu materialisierende Sichten
   - Statische vs. Dynamische Auswahl
3. Aktualisierung
   - Inkrementelle Aktualisierung
4. Konsistenz
   - Aspekte und Konzepte
5. Zusammenfassung

# Konsistenz

Probleme:

- Sehr viele materialisierte Sichten ( >1000)
- Redundanzen zwischen verschiedenen Sichten

→ Konsistenzerhaltung wichtig

  - Basisrelationen ↔ materialisierten Sichten
  - materialisierten Sichten ↔ materialisierten Sichten

# Konsistenzaspekte

- Anwenderdefinierte Aktualitätsanforderungen
  - Zeitlicher Abstand
  - Wertemäßiger Abstand
  - Versionsbezogener Abstand
- Anfragekonsistenz
- Sitzungskonsistenz
- Aktualisierungsgranulat
  - Gesamtes Data-Warehouse
  - Einzelne Sichten
  - Gruppen von Sichten (→ Bsp. Viewgroup Konzept)

# Anforderungen

Nebenläufige Aktualisierung:

→ möglichst kleine Einschränkung der Anfragebearbeitung durch Aktualisierung

Unterstützung individueller Aktualisierungsstrategien:

→ Anpassbarkeit der einzelnen Aktualisierungsgranulate an unterschiedliche Anwendungsanforderungen

Ziel: Den besten Kompromiss zwischen Leistung und Konsistenz zu finden

# Mehrversionen Prinzip

- Erhöhung der Nebenläufigkeit durch mehrere Versionen eines Datenobjektes

  → Anfragen lesen ältere Version, während auf neuer Version aktualisiert wird

- Vorteil: Aktualisierung und Anfragen sind unabhängig

- Nachteil: Zugriffe auf veraltete Objekte möglich

  (↔ anwenderdefinierte Aktualitätsanforderung)

# Gliederung

1. Materialisierte Sichten
   - Prinzip und Anwendung
2. Auswahl zu materialisierender Sichten
   - Statische vs. Dynamische Auswahl
3. Aktualisierung
   - Inkrementelle Aktualisierung
4. Konsistenz
   - Aspekte und Konzepte
5. Zusammenfassung

# Zusammenfassung

- Anfrageauswertung wird stark beschleunigt
  - → stellt aber wieder neue Anforderungen und Probleme
  - → letztendlich Suche nach dem besten Kompromiss zwischen Leistung und Komplexität
- Entwicklung noch nicht abgeschlossen

# Ende

Materialisierte Sichten