

# Übung Blatt 5

## Materialized Views

Ulf Leser

Wissensmanagement in der  
Bioinformatik



# Achtung

---

- Die Tabellen, Views, Indexe etc. von Aufgabe 4 noch nicht löschen!

# Schema und Date

---

- Wir verwenden das TPC-H Schema
  - Skript zur Schemadefinition liegt im Web
  - In dieser Aufgabe brauchen wir die meisten Tabellen
  - Alle Tabellen mit Testdaten liegen im Schema **dwh2** und müssen von dort kopiert werden
    - Achtung: Neues Quellschema
    - Das ist eine 10GB Version der TPC-H Daten
  - Check:
    - **lineitem** hat ca. 30 Millionen Tupel
    - **orders** hat ca. 15 Millionen Tupel
    - **part** hat 2 Millionen Tupel
  - Auf den Tabellen dürfen keinerlei Indexe angelegt werden

# Anfragen

---

- Gegeben sind die folgenden Queries ohne Aggregation
  - Alle Bestellnummern von Kunden aus den UNITED STATES oder aus GERMANY mit Orderstatus 'F' und einem Gesamtpreis über 400.000
  - Alle Kunden der Region ASIA
  - Alle Supplier, die Teile der Firma Brand#15 liefern können, sortiert nach dem Teil, der Region und dem Preis des Teils
  - Alle Bestellnummern von Bestellungen, die Teile der Firma Supplier#000002990 beinhalten
  - Alle lineitem mit Kundennummer, Bestellnummer und Land aus der Region EUROPE, bei denen der Rabatt mindestens 0.1 beträgt und die per Post (L\_SHIPMODE = 'MAIL') versendet wurden

# Anfragen 2

---

- Und die folgenden Queries mit Aggregation
  - Die Bestellsumme aller abgeschlossenen Bestellungen in ASIA nach dem 31.12.1994, gruppiert nach Kunden, Nationen und Regionen
  - Die Anzahl sämtlicher Teile, die Bestandteil einer abgeschlossenen Bestellung aus dem Januar 1992 sind, gruppiert nach Name des Teils und Lieferant
  - Die Anzahl sämtlicher Teile, die Bestandteil einer abgeschlossenen Bestellung aus dem Januar 1992 sind, gruppiert nach Lieferant und Name des Teils
  - Die Anzahl sämtlicher Teile, die Bestandteil einer abgeschlossenen Bestellung aus dem Januar 1992 sind, gruppiert nach Lieferant und Nation
- Alle Queries sind im Web

# Aufgabe: Query Rewriting (20 Punkte)

---

- Erstellen Sie vier materialisierte Sichten, die die Ausführung aller neun Anfragen maximal beschleunigt
  - Die Anfragen müssen exakt so bleiben, wie sie sind
    - Der Optimierer muss also die Sichten selber einbauen und verwenden
    - „enable query rewrite“ nicht vergessen
  - Die Views müssen MATVIEW1 – MATVIEW4 heißen
  - Pro MatView dürfen maximal zwei Indexe angelegt werden
  - Kosten für die Aktualisierung der Views sind hier nicht relevant
  - Relevant ist aber die Größe der MatViews
- Zusammengesetztes Ranking
  - 2/3 zählt die Summe der Ausführungszeiten aller neun Queries
  - 1/3 zählt der Platzverbrauch aller MatViews zusammen
  - Berechnung: Normierung auf den jeweils größten Wert und gewichtete Summe beider Komponenten

# Punkte

---

- Pro angelegtem View, der in mindestens einer Query verwendet wird, gibt es 5 Punkte

# Größenmessungen

---

- Wenn Statistiken mit `dbms_stats.gather_table_stats()` ermittelt werden, wird auch die tatsächliche Tabellengröße ausgerechnet
- Wir verwenden als Abschätzung (Anzahl Tupel) \* (durchschnittliche Größe eines Tupels)
- Dazu ist im Web das Script `table_size.sql`
- Vor Ausführung müssen aktuelle Statistiken über die Objekte eingeholt werden!
- Gibt die Gesamtgröße über `dbms_output` aus
- Funktioniert für Tabellen und materialisierte Sichten

# Zeitmessung

---

- Im Web ist eine Prozedur, die exakt wie sie ist zur Zeitmessung verwendet werden wird
- Die neun Queries werden jeweils dreimal ausgeführt und die Gesamtzeit gemessen
- Einziger Parameter ist der Benutzername
- Das Skript kann gerne zur Entwicklung benutzt und modifiziert werden
- Größenordnung: Ein Durchlauf durch alle Queries ohne Indexe / MatView benötigt 1010 Sekunden
  - Nach gather\_schema\_stats

# Abgabe

---

- Per Mail bis Sonntag, 15.7.2007, 23:59 Uhr (2.5 Wochen Zeit)
  - Bitte als ASCII Text
  - Vorstellung ist am Donnerstag, den 19.7.2007
- Bitte Antworten zu den einz. Fragen mit <<<<<<<<<< beginnen
  1. Die Definition der vier Views (Auf den Namen achten!)
  2. Zu jedem View eine Liste der Queries (durchnummeriert), in denen er gemäß PLAN\_TABLE verwendet wird
  3. Die Größen der vier Views
  4. Die Ausführungszeit gemäß des Mess-Skripts
- Alle Tabellen, Views und PL/SQL Routinen müssen kompiliert in der Datenbank liegen
- Wettbewerbspunkte
  - Wir beachten die Gesamtzeit aller Queries und die Gesamtgröße aller Views
  - 1. Gruppe: 5 Punkte
  - 2. Gruppe: 3 Punkte
  - 3. Gruppe: 2 Punkte