

Nachtrag Übungsserie 4

Folgendes sollten Sie durch die Bearbeitung der Übungsserie 4 gelernt haben:

- Lesen und Schreiben von SQL-Anfragen
- Unterschied zwischen der Mengen- und Multimengensemantik (speziell: UNION und UNION ALL)
- Übersetzung von Anfragen der relationalen Algebra in SQL-Anfragen
- Definition von temporären Sichten innerhalb von Anfragen (WITH-Anweisung)
- Semantik der TABLESAMPLE-Klausel (siehe <http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.admin.doc/doc/c0010970.htm>)
- Semantik der FETCH FIRST-Klausel (siehe dazu SQL Reference, Vol. 1: <http://www-1.ibm.com/support/docview.wss?rs=71&uid=swg27009552>)

Sichten

Eine Sicht ist eine Anfrage, die unter einem Namen gespeichert wird und in anderen Anfragen unter Angabe dieses Namens verwendet werden kann. Sichten bieten unter anderem folgende Vorteile:

- *logische Datenunabhängigkeit*: die der Sicht zugrunde liegenden Tabellen können erweitert werden (z. B. Anfügen von Spalten), ohne das Anwender / Anwendungen, die die Sicht verwenden, informiert werden müssen
- *Sicherheit*: Beschränkung des Zugriffs auf die Daten, indem Benutzer nur Anfragen auf Sichten, aber keine Anfragen auf die zugrunde liegenden Tabellen ausführen können
- *Wiederverwendbarkeit* von häufig auftretenden Anfragen

Mit WITH kann eine temporäre Sicht (innerhalb einer Anfrage) angelegt werden. Betrachten wir dazu die Anfrage aus Aufgabe 2a.

```
WITH product AS (  
  SELECT prod.name AS pname,  
         act.name AS aname,  
         COUNT(act.movie_id) AS countm  
  FROM (  
    SELECT name, movie_id  
    FROM actor  
    UNION ALL  
    SELECT name, movie_id  
    FROM actress  
  ) AS act  
  INNER JOIN movie AS mov ON act.movie_id = mov.mid  
  INNER JOIN producer as prod on prod.movie_id = mov.mid  
  GROUP BY prod.name, act.name  
  ORDER BY prod.name, act.name  
)  
  
SELECT product.pname,  
       product.aname,  
       product.countm
```

```
FROM product, (  
  SELECT pname, max(countm) AS maxvalue  
  FROM product  
  GROUP BY pname  
  ) AS maxcount  
WHERE product.pname = maxcount.pname  
      AND product.countm = maxcount.maxValue;
```

Die Sicht `product` liefert eine Tabelle mit Schema $(pname, aname, countm)$. Diese enthält ein Tupel (p, a, c) genau dann, wenn der Produzent p mit dem Schauspieler / der Schauspielerin a in insgesamt c Filmen zusammengearbeitet hat.

Ausgehend von dieser Tabelle wird die eigentliche Anfrage aufgebaut: Es wird der Produzent p^* bestimmt, der bezüglich `countm` den größten Wert aufweist (sei c^* dieser Wert). Hierbei ist zu bemerken, dass bezogen auf unseren Datenbankzustand mehrere Produzenten diese Eigenschaft erfüllen.

Für den Produzenten p^* werden nun die entsprechenden Tupel (p^*, a, c^*) zurückgeliefert. Das heißt, es wird für jeden Schauspieler / jede Schauspielerin a , der / die mit p^* am häufigsten zusammengearbeitet hat, p^* , a und die Anzahl der Filme ausgegeben, in denen p und a zusammengearbeitet haben (entspricht c^*).

Bezogen auf unseren Datenbankzustand werden 394 Zeilen zurückgegeben. Dabei werden insgesamt 29 Produzenten für p^* betrachtet.