

Aufgabenblatt 4

Verteilte Anfragebearbeitung

- Letzter Abgabetermin: **Montag, 02. Juli 2007**
- Abgabe: Per E-Mail an Alexander Albrecht, Fachgebiet Informationssysteme.

Implementiere in Java einen Join über zwei Relationen in zwei Datenbanken. Das Ziel ist es, bei der Joinberechnung möglichst wenig Daten über das Netzwerk zu übertragen. Es stehen die zwei Datenbanken MOVIE1 und MOVIE2 in DB2 zur Verfügung.

```
CREATE TABLE Movie1.Filme (Titel CHARACTER(100) NOT NULL PRIMARY KEY,  
Jahr Integer)  
CREATE TABLE Movie2.Filme (Titel CHARACTER(100) NOT NULL PRIMARY KEY,  
Regie CHARACTER(50))
```

Für die Datenbanken gelten die folgenden Zugriffsberechtigungen:

- Lesend auf die Tabellen
- CREATE TABLE auf die Datenbanken
- INSERT auf neue Tabellen

Verwende die bereits für Aufgabenblatt 3 installierte DB2 Version. Lege die zwei Datenbanken MOVIE1 und MOVIE2 an und importiere die Tabellen in jeweils eine Datenbank. Die Tabellen findest Du als Flatfiles im Übungsverzeichnis R:\InfoInt07_Naumann\Uebung\0626\.

Beantworte die folgende Anfrage, die Titel und Regisseure aller Filme liefert, die jünger als 1999 sind:

```
SELECT F1.Titel, F2.Regie FROM Movie1.Filme F1, Movie2.Filme F2  
WHERE F1.Titel = F2.Titel AND F1.Jahr > 1999
```

Da die beiden Tabellen in unterschiedlichen Datenbanken liegen, kann die Anfrage so nicht ausgeführt werden. Verwende das Prinzip der *passing bindings* bzw. Semi-Join Techniken. Zu diesem Zweck können auch temporäre Tabellen verwendet werden. Jede neu erzeugte Tabelle muss nach der Beantwortung der Anfrage wieder gelöscht werden.

Abgabe

Dein Java-Programm soll neben dem Anfrageergebnis, der Anzahl der Tupel im Ergebnis und sämtlichen an beide Datenbanken geschickten SQL-Befehle auch die Anzahl der übertragenen Bytes ausgeben. Diese berechnen sich wie folgt:

- CREATE Befehle kosten soviel wie die String-Länge des Befehls selbst.
- Für INSERT Befehle zählen die eingefügten Tupel multipliziert mit der Tupellänge.
- Für SELECT Anfragen zählt die Tupelanzahl im Anfrageergebnis multipliziert mit der Tupellänge
- Die Tupellänge ist die Summe der Typlängen der einzelnen Attribute: CHARACTER entsprechend der Länge, INTEGER haben 4 Byte.

Im Übungsverzeichnis liegt bereits die Java Klasse DISTRIBUTEDJOIN, die einen naiven Join Ansatz implementiert. Erweitere dieses Java-Programm und dokumentiere den Programmcode.

Kontrolle

```
SELECT count(*) FROM Movie1.Filme F1, Movie2.Filme F2  
WHERE F1.Titel = F2.Titel (Ergebnis: 5586).
```

```
SELECT count(*) FROM Movie1.Filme F1, Movie2.Filme F2  
WHERE F1.Titel = F2.Titel AND F1.Jahr > 1999 (Ergebnis: 1991)
```

Auswertung & Punkte

Nach Ablauf der Abgabe lassen wir das Java Programm laufen. Dabei wird die Anzahl der übertragenen Bytes summiert. Das Ziel ist es natürlich, möglichst wenig Bytes zu übertragen.

Das Team mit der minimalen Anzahl übertragener Bytes ist Sieger dieses Wettbewerbs. Am Ende des Semesters wird der Gesamtsieger aller Wettbewerbe ermittelt. Die Preisverleihung findet am 19.07. im Rahmen der Vorlesung statt.