

DEZSYS-01 „Entwurf verteilter Datenbanken“ von Stefan Erceg

Inhalt

1. Ziele	1
2. Voraussetzungen.....	2
3. Aufgaben	3
3.1 horizontale Fragmentierung	3
3.2 vertikale Fragmentierung	3
3.3 kombinierte Fragmentierung	4
4. Ergebnisse	5
5. Lessons learned	6
6. Zeitaufwand	6
7. Quellenangaben.....	6

1. Ziele

Unter Verwendung der Sample Database "Dell DVD Store" soll eine lokale Datenbank in eine verteilte Datenbank transferiert werden. Mit dieser Aufgabe soll die Fragmentierung einer Datenbank durchgeführt werden. Basierend auf einer Tabelle/View des DVD Stores sollen folgende Fragmentierungsarten umgesetzt werden:

- horizontale Fragmentierung nach mindestens 2 Kriterien
- vertikale Fragmentierung
- kombinierte Fragmentierung

Die Fragmente sollen jeweils in einem Schema mit der Bezeichnung

- Schema horizontal
- Schema vertical
- Schema combination

in Tabellen mit sinnvollen Namen gespeichert werden. Die Fragmentierung soll sinnvoll unter selbst definierten Annahmen spezifiziert werden.

Dokumentiere Deine Arbeitsschritte und die Definition Deiner Fragmente in einem Abgabeprotokoll und lade 2 Dateien (protocol.pdf, dump.zip) gemeinsam mit dem Dump der 3 Schemata hier hoch.

Punkte: 8 Punkte

- pro Fragmentierungsart (2 Punkte)
- Abgabeprotokoll (2 Punkte)

2. Voraussetzungen

- Der Aufbau der jeweiligen Fragmentierungsarten wurde uns in den Unterrichtsstunden vom Herrn Professor Micheler präsentiert.
- Die Sample Database „Dell DVD Store“ befindet sich auf folgender Seite: [1]. Ich habe mir die Dateien „ds21.tar.gz“ und „d21_postgresql.tar.gz“ heruntergeladen, da ich diese Aufgabe mit PostgreSQL erledigt habe.
- Danach wurden die beiden tar-Files entpackt. Dabei musste darauf geachtet werden, dass der Ordner „pgsqlds2“, welcher sich im PostgreSQL-tar-File befindet, in den Ordner „ds2“, welcher wiederum im tar-File „ds21“ zu finden ist, verlegt wird.
- In PostgreSQL wurde ein User „ds2“ mit dem Passwort „ds2“ und die Datenbank „ds2“ angelegt:

```
CREATE USER ds2 WITH SUPERUSER;  
ALTER USER ds2 WITH PASSWORD 'ds2';  
CREATE DATABASE ds2;
```

- Mit dem Befehl `chmod +x pgsqls2_create_all.sh` wurde das Shell-Script auf executable gesetzt, damit es ordnungsgemäß ausgeführt werden konnte.
- Das Shell-Script wurde mit dem Befehl `./ pgsqls2_create_all.sh` ausgeführt.
- Folgende Tabellen sind nun in der Datenbank zu finden:

List of relations			
Schema	Name	Type	Owner
public	categories	table	ds2
public	cust_hist	table	ds2
public	customers	table	ds2
public	inventory	table	ds2
public	orderlines	table	ds2
public	orders	table	ds2
public	products	table	ds2
public	reorder	table	ds2
(8 rows)			

3. Aufgaben

3.1 horizontale Fragmentierung

Bei der horizontalen Fragmentierung habe ich die Customers-Tabelle folgendermaßen aufgeteilt:

- alle Kunden, die zwischen 18 und 30 Jahre alt sind und aus der USA kommen
- alle Kunden, die zwischen 18 und 30 Jahre alt sind und nicht aus der USA kommen
- alle Kunden, die über 30 Jahre alt sind und aus der USA kommen
- alle Kunden, die über 30 Jahre alt sind und nicht aus der USA kommen

```
CREATE SCHEMA horizontal;
```

```
CREATE TABLE horizontal.customersUnder30FromUS AS SELECT * FROM  
customers WHERE age >= 18 AND age <= 30 AND country = 'US';
```

```
CREATE TABLE horizontal.customersUnder30FromRest AS SELECT * FROM  
customers WHERE age >= 18 AND age <= 30 AND NOT country = 'US';
```

```
CREATE TABLE horizontal.customersOver30FromUS AS SELECT * FROM  
customers WHERE age > 30 AND country = 'US';
```

```
CREATE TABLE horizontal.customersOver30FromRest AS SELECT * FROM  
customers WHERE age > 30 AND NOT country = 'US';
```

3.2 vertikale Fragmentierung

Bei der vertikalen Fragmentierung habe ich die Orders-Tabelle folgendermaßen aufgeteilt:

- alle generellen Informationen über die Bestellungen (Gesamtbetrag, Bestelldatum, usw.)
- alle zusätzlichen Informationen über die Bestellungen (Nettobetrag und hinzugerechnete Steuer)

```
CREATE SCHEMA vertical;
```

```
CREATE TABLE vertical.ordersGeneralInfo AS SELECT orderid, orderdate,  
customerid, totalamount FROM orders;
```

```
CREATE TABLE vertical.ordersAdditionalInfo AS SELECT orderid,  
netamount, tax FROM orders;
```

3.3 kombinierte Fragmentierung

Bei der kombinierten Fragmentierung habe ich die Customers-Tabelle folgendermaßen aufgeteilt:

- alle Kreditkartendetails der Kunden (Kreditkartentyp, Kreditkartennummer, usw.)
- alle restlichen Informationen über männliche Kunden (Wohnortdetails, Alter, usw.)
- alle restlichen Informationen über weibliche Kunden (Wohnortdetails, Alter, usw.)

```
CREATE SCHEMA combination;
```

```
CREATE TABLE combination.customersCreditCardDetails AS SELECT  
customerid, creditcard, creditcardtype, creditcardexpiration FROM  
customers;
```

```
CREATE TABLE combination.maleCustomers AS SELECT customerid,  
firstname, lastname, address1, address2, city, state, zip, country,  
region, email, phone, username, password, age, income, gender FROM  
customers WHERE gender = 'M';
```

```
CREATE TABLE combination.femaleCustomers AS SELECT customerid,  
firstname, lastname, address1, address2, city, state, zip, country,  
region, email, phone, username, password, age, income, gender FROM  
customers WHERE gender = 'F';
```

4. Ergebnisse

Die Datenbank wurde nun nach 3 Arten erfolgreich fragmentiert und die Schemen „horizontal“, „vertical“ und „combination“ existieren.

Folgende 4 Tabellen sind beim Schema „horizontal“ vorhanden:

List of relations				
Schema	Name	Type	Owner	
horizontal	customersover30fromrest	table	ds2	
horizontal	customersover30fromus	table	ds2	
horizontal	customersunder30fromrest	table	ds2	
horizontal	customersunder30fromus	table	ds2	

(4 rows)

Folgende 2 Tabellen sind beim Schema „vertical“ vorhanden:

List of relations				
Schema	Name	Type	Owner	
vertical	ordersadditionalinfo	table	ds2	
vertical	ordersgeneralinfo	table	ds2	

(2 rows)

Folgende 3 Tabellen sind beim Schema „combination“ vorhanden:

List of relations				
Schema	Name	Type	Owner	
combination	customerscreditcarddetails	table	ds2	
combination	femalecustomers	table	ds2	
combination	malecustomers	table	ds2	

(3 rows)

Mit den Befehlen

```
pg_dump -C -n horizontal -U ds2 ds2 > horizontal.sql
pg_dump -C -n vertical -U ds2 ds2 > vertical.sql
pg_dump -C -n combination -U ds2 ds2 > combination.sql
```

habe ich jeweils ein Dump-File pro Schema erstellt.

5. Lessons learned

- erstmalige Anwendung der horizontalen, vertikalen und kombinierten Fragmentierungen
- Datenbanken sinnvoll aufteilen

6. Zeitaufwand

In den beiden Unterrichtsstunden konnte ich die horizontale und vertikale Fragmentierung fertigstellen. Es wurden danach noch zusätzliche 40 Minuten in meiner Freizeit investiert, um die Arbeitsdurchführung im Protokoll detaillierter zu beschreiben und die kombinierte Fragmentierung zu implementieren.

7. Quellenangaben

- [1] Dell (2010, 2011). Dell DVD Store Database Test Suite [Online].
Available at: <http://linux.dell.com/dvdstore/> [zuletzt abgerufen am 02.10.2015]