7. Versuch: SQL und Java

Setzen Sie für diesen Versuch bitte die folgenden Umgebungsvariablen:

```
export ORACLE_HOME=/usr/local/oracle/instantclient10_1
export CLASSPATH=.:$ORACLE_HOME/sqlj/lib/runtime12.jar:$ORACLE_HOME/classes12.jar
```

Bei JDBC und SQLJ wird auf externe Dateien zugegriffen, in der die Login-Daten für Oracle in Klartext gespeichert sind. Im Verzeichnis `$WORK` sind diese Dateien von allen Gruppenmitgliedern lesbar. Man sollte also in Erwägung ziehen, die Klassen im Home-Verzeichnis zu entwickeln.

Aufgabe 7.1 (Java Stored Procedures, Konsistenzerhaltung; 30 P.)
In Aufgabe 3.2 wurden bei der Löschung eines Flusses/Sees/Meeres die entsprechenden Tupel in *geo_River/geo_Lake/geo_Sea* ebenfalls gelöscht. Überlegen Sie sich, welche Aktionen bei Löschung eines Flusses/Sees/Meeres in der Datenbank sinnvoll sind. Schreiben Sie eine Java-Klasse mit drei Methoden `deleteRiver/Lake/Sea`, die ein(en) gegebenes/n Fluss/See/Meer als Parameter erhalten, löschen und alle relevanten Tabellen in der Datenbank entsprechend aktualisieren. Laden Sie die Klasse in die Datenbank und schreiben Sie Prozeduren als Wrapper.

Aufgabe 7.2 (JDBC; 40 P.)
Auf der Praktikumsseite finden Sie ein Java-File `frame.java`, das die Grundlage für ein Java-Programm welches per JDBC auf die Praktikumsdatenbank zugreift.

Das Programm setzt sich aus einer `TextArea` zur Ausgabe, einem `TextField` zur Eingabe von SQL-Anweisungen und einem `Button` zum Absenden der Anweisung an die Datenbank zusammen. Erweitern Sie diesen Rahmen so, dass beliebige DML-Anweisungen ausgeführt werden können (`SELECT`, `INSERT`, `DELETE`, `UPDATE`).

In `frame.java` ist außerdem die Methode `dispResult` zur implementieren. Passend zum ausgeführten SQL-Befehl soll die Ergebnismenge oder die Anzahl der geänderten Zeilen in der `TextArea` ausgegeben werden. Beachten Sie, dass die Anzahl der Spalten der Ergebnismenge je nach Anfrage variieren kann. Vor den Ergebnistupeln soll eine Zeile mit den Attributnamen der betreffenden Spalte erscheinen.

Aufgabe 7.3 (SQLJ, Bruttoinlandsprodukt; 30 P.)
Berechnen Sie mit Hilfe von SQLJ analog zu Aufgabe 5.3, welcher Anteil der Weltbevölkerung mindestens notwendig ist, um 50% des Welt-Bruttoinlandsproduktes zu erzeugen. Nehmen Sie dabei an, dass sich das Bruttoinlandsprodukt eines Landes gleichmäßig auf alle Einwohner verteilt. Geben Sie an, wieviele Einwohner aus welchen Ländern beteiligt sind. Verwenden Sie einen Iterator über ein geeignet gewähltes `SELECT`-Statement.

In SQLJ kann man den Verbindungsaufbau zur Datenbank mit einer Hilfsdatei bewerkstelligen:

```
#sqlj.props:
sqlj.url=jdbc:oracle:thin:@s4.ifi.informatik.uni-goettingen.de:1521:dbis
sqlj.user=<user>
sqlj.password=<passwd>
```

Diese Datei sollte im gleichen Verzeichnis wie die Java-Klasse liegen und wird so verwendet:

```
//Aufbau der Verbindung zu DB in SQLJ:
Oracle.connect(Countries.class, "sqlj.props");
```

Abgabe bis 08.04.2005