7. Versuch: JDBC

Bei diesen Aufgaben werden Oracle-Gruppenaccounts `JDBC_`n`/JDBC_`n verwendet (siehe vorgegebenen Rahmen auf der Praktikumsseite im WWW). Da diese Accounts naturgemäß noch leer sind, muss zuerst `create` aufgerufen werden, um die MONDIAL-Datenbank zu erstellen.

Aufgabe 7.1 (JDBC-Grundlage; 15 P.)
Auf der Praktikumsseite finden Sie ein Java-File `frame.java`, das die Grundlage für ein Java-Programm oder Applet bildet, welches per JDBC auf die Praktikumsdatenbank zugreift. Am Anfang der Datei stehen weitere Hinweise zum Setup, Umgebungsvariablen und Start.

Das Programm/Applet setzt sich im wesentlichen aus einer `TextArea` zur Ausgabe, einem `TextField` zur Eingabe von SQL-Anweisungen und einem `Button` zum Absenden der Anweisung an die Datenbank zusammen.

In `frame.java` ist die Methode `dispResult` zur Ausgabe der Ergebnismenge in der `TextArea` zu implementieren. Beachten Sie dabei, dass die Anzahl der Spalten der Ergebnismenge je nach Anfrage variieren kann. Zudem soll vor den Ergebnistupeln eine Zeile mit den Attributnamen der betreffenden Spalte erscheinen.
Die Implementierungen der folgenden Aufgaben sollten in das Programm/Applet integriert werden, es kann beispielsweise für jede Anfrage ein Button zur Verfügung gestellt werden. Dabei dient `frame.java` nur als Vorschlag, es kann auch ein eigenes Applet entworfen werden.

Aufgabe 7.2 (ResultSet; 20 P.)
Lösen Sie Aufgabe 5.3 in Java:
Berechnen Sie, welcher Anteil der Weltbevölkerung mindestens notwendig ist, um 50% des Bruttosozialproduktes zu erzeugen. Nehmen Sie dabei an, dass sich das Bruttosozialprodukt eines Landes gleichmäßig auf alle Einwohner verteilt. Geben Sie an, wieviele Einwohner aus welchen Ländern beteiligt sind.

Aufgabe 7.3 (Transitive Closure; 20 P.)
Erzeugen Sie mittels JDBC eine Relation, die für jeden Fluss die folgenden Daten enthält:

- Denjenigen Fluss, in dem das Wasser schlussendlich in ein Meer/See fließt, sowie
- dieses/n Meer/See.

Geben Sie die den Inhalt der erstellten Relation, sowie deren Anzahl von Tupeln an.

Aufgabe 7.4 (Konsistenzerhaltung; 20 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 referentielle Aktion bei Löschung eines Flusses/Sees/Meeres bezüglich der Relation *located* sinnvoll ist. Schreiben Sie drei Java-Prozeduren `delete_River/See/Meer`, die ein(en) gegebenes/n Fluss/See/Meer einlesen (z. B. im `TextField`), löschen und *located* entsprechend aktualisieren.

Aufgabe 7.5 (Meta-Anfragen; 25 P.)
JDBC erlaubt es, SQL-Anfragen als Strings zusammenzusetzen und dann an die Datenbank zu senden. Dies kann insbesondere verwendet werden, um auf Basis der Informationen des Data Dictionaries SQL-Befehle über alle Tabellen laufen zu lassen.

(Wartung) Schreiben Sie ein Skript, das alle Funktionen und Prozeduren löscht.

Abgabe bis 06.02.2004