



6. Übungsblatt

(Testwoche: 25. - 27. Mai 2010)

Einführung in Datenbanksysteme
Datenbanken für die Bioinformatik

Heinz Schweppe, Katharina Hahn

Hinweis: Bringen Sie das ausgedruckte Schema der Mondial Datenbank (<http://www.inf.fu-berlin.de/lehre/SS10/DBS-Intro/uebungen/mondial-abh.pdf>) zu den Tutorien mit!

Aufgabe 1 (Anwendungsarchitektur)

5 Punkte

Charakterisieren Sie den Unterschied zwischen den in der Vorlesung vorgestellten Client-Server Architektur und Server-seitiger Applikationslogik (*server-side computing*). Welche Aufgaben spielt der so genannter Applikationsserver? Geben Sie die zentralen Vor- und Nachteile von *server-side computing* im Vergleich zur Verwendung eines Applikationsservers an.

Aufgabe 2 (VIEWS)

7 Punkte

Erstellen Sie in ihrem Schema folgende Sichten (VIEWS) zur Mondial Datenbank. Geben Sie jeweils **gültige** SQL-Statements an.

- Eine Sicht, die alle Grenzinformationen (*borders*) der Datenbank in symmetrischer Form enthält. D.h., wenn ein Land $c1$ benachbart zu $c2$ ist, dann müssen die Tupel $(c1, c2, l)$ und $(c2, c1, l)$ in der Sicht sein, wobei l die Länge der Grenze ist.
- Eine Sicht, die die Information aller Organisationen (*organization*) enthält, die in einer Hauptstadt angesiedelt sind. Zu jeder Organisation soll zusätzlich die Anzahl der Einwohner der Stadt, in der sie ihren Sitz hat, enthalten sein.
- Eine Sicht, die alle Informationen aller Länder, die an Meeren liegen, mitsamt der Information über das/die Meer/e, an dem/denen sie liegen, enthält.

Aufgabe 3 (Funktionen)

9 Punkte

Schreiben Sie folgende Funktionen in PL/SQL (Oracle) bzw. PL/pgSQL (PostgreSQL) auf Basis des Schemas der MONDIAL Datenbank.

- Die Funktion *DENSITY* erhält ein Land als Eingabe und gibt für dieses die Bevölkerungsdichte (Einwohner pro Fläche) zurück.
- Die Funktion *BIG_CITIES* erhält ein Land und einen Schwellwert als Eingabe und gibt alle Städte in dem gegebenen Land aus, deren Einwohnerzahl über dem Schwellwert liegt. Für jede solche Stadt soll der Name, die Provinz und die Anzahl der Einwohner ausgegeben werden.

- c) Die Funktion *DISTANCE* gibt für 2 Städte ihre Distanz in Kilometern zurück. Beachten Sie, dass eine Stadt nur in Zusammenhang mit der Provinz und dem Land, in dem sie liegt, eindeutig ist. Berechnen Sie die Distanz mit Hilfe der Haversine Formel. Informationen zur Haversine Formel finden Sie z.B. unter <http://www.movable-type.co.uk/scripts/latlong.html>.

Informationen zur Programmierung von PL/SQL bzw. PL/pgSQL finden Sie in den jeweiligen Dokumentationen der beiden Datenbanksysteme.

Aufgabe 4 (Trigger)

8 Punkte

Schreiben Sie Trigger für die Mondial Datenbank, die die folgenden Aufgaben erfüllen. Beachten Sie, dass Sie auf Grund der Rechteverwaltung die Änderungen, die die Trigger auslösen auf *esel* nicht durchführen können. Sie können daher ihre Trigger nur auf bei Ihnen lokal installierten Datenbanken testen.

- a) Schreiben Sie einen Trigger, der bei Änderung der Länderkennung (*country.code*) ausgeführt wird und die Tabelle *province* entsprechend aktualisiert.
- b) Schreiben Sie einen Trigger, der vor dem Einfügen eines Tupels (*country1, country2, length*) in die Tabelle *borders* prüft, ob ein Tupel (*country2, country1, x*) bereits existiert. Ist dies der Fall, soll das Einfügen des Tupels verhindert werden. Ansonsten soll das Tupel als (*country1, country2, length*) eingefügt werden, wenn $country1 < country2$; ansonsten wird das Tupel (*country2, country1, length*) eingefügt.
- c) Schreiben Sie einen Trigger, der beim Einfügen/Ändern von Tupeln in die/der Relation *EthnicGroup* prüft, ob die Summe des Attributs *percentage* pro Land unter 100% bleibt. Wenn die Summe mit dem geänderten Tupel bzw. dem neu eingefügten Tupel größer als 100% ist, soll die Änderung bzw. das Einfügen abgelehnt werden und eine entsprechende Meldung ausgegeben werden.