

Praktikum: Datenbanken
Woche 10: Projektaufgabe:
Programmierung von Datenbankanwendungen

Raum: LF230

Bearbeitung: **27.6.-22.7. 2005**

Als Abschlußaufgabe soll selbstständig ein vorgegebenes Programmgerüst zu einer Java-Anwendung erweitert werden, die lesend und schreibend auf einer Datenbank arbeitet. Der Zugriff erfolgt dabei über JDBC.

Die Projektaufgabe ist einzeln oder in Gruppenarbeit (in den gleichen Kleingruppen wie im Laufe des Praktikums) zu bearbeiten. Bei Einzelarbeit sollten beide Gruppenmitglieder eine eigene Datenbank innerhalb ihrer Instanz erstellen. Der Bearbeitungszeitraum endet am **22. 7. 2005**. Während dieser Zeit finden keine Praktikumssitzungen mehr statt. Den Programmcode zur Abgabe schickt Ihr bitte als E-Mail an den für Eure Gruppe zuständigen Tutor.

Zur Abnahme der Aufgaben könnt Ihr zwischen dem 18. und dem 29. Juli mit Eurem Tutor einen Termin vereinbaren.

Die vorgegebenen Klassen für das Spiel und das Benutzerinterface können von der Webseite geladen oder aus dem Verzeichnis `~dbprak/java` kopiert werden. Die Klassen geben ein Gerüst vor, das Ihr so übernehmen oder auch anpassen könnt.

Aufgaben

Aufgabe P10.1 Vorbereitung der Datenbank

Damit alle Datenbanken ein einheitliches Schema und einen einheitlichen Datenbestand haben, benutzt die im Skriptverzeichnis zur Verfügung stehenden SQL-Skripte um eine frische Datenbank zu erstellen. Ihr müsst dazu nicht Eure Datenbank löschen, sondern könnt stattdessen auch eine neue Datenbank in Eurer Instanz anlegen.

Erstellt Indexe auf allen Attributen, über die im Folgenden gesucht werden soll, damit die Anwendung flott arbeiten kann.

Aufgabe P10.2 Fragen-Tabelle

Erstellt eine neue Tabelle in der Datenbank, die Quiz-Fragen aus dem Bereich der Miniwelt enthält. Jeder Eintrag soll die folgenden Informationen enthalten: eine eindeutige, fortlaufende Nummer als Primärschlüssel, einen Zeitstempel des Einfügedatums, den Namen des Einfügenden, einen Fragetext und eine SQL-Anfrage auf dem Schema der Datenbank. Die SQL-Anfrage sollte als Ergebnis eine einzelne Spalte und mindestens eine Zeile liefern, die eine akzeptable Antwort auf den Fragetext wäre.

Beispiel:

```
(123,'2005-02-22-24.00.00','Sascha',  
'Nenne einen Bundesstaat der USA?',  
'SELECT name FROM provinz WHERE land=''USA''')
```

```
(124,'2005-03-12-22.00.00','David',  
'Was ist das flächenmäßig größte Land der Erde?',  
'SELECT name FROM land WHERE gebiet >= ALL (SELECT gebiet FROM  
land)')
```

```
(125,'2005-01-30-12.15.00','Christoph',  
'Wieviele Bundesländer hat die Bundesrepublik Deutschland?',  
'SELECT count(*) FROM provinz WHERE land=''D''')
```

Man beachte, dass der Apostroph, der als Zeichenkettenbegrenzer für CHAR- und VARCHAR-Werte dient, durch einen zusätzlichen Apostroph maskiert werden muß, wenn er explizit in der Zeichenkette erscheint (siehe erstes Beispiel).

Füllt diese Tabelle mit mindestens 20 angemessen anspruchsvollen Fragen. Jede Frage sollte zumindest eine WHERE-Klausel oder eine Aggregation besitzen. Variationen ein und der selben Frage zählen nicht als einzelne Fragen (also nicht zwanzig Varianten der Frage „*Was ist die Hauptstadt von X?*“).

Aufgabe P10.3 Highscore-Tabelle

Erstellt eine neue Tabelle in der Datenbank, die Highscore-Einträge verwalten soll. Jeder Eintrag soll einen eindeutigen Benutzernamen enthalten, eine bislang erreichte Höchstpunktzahl und einen Zeitstempel für den Highscore. Die Höchstpunktzahl soll nach oben durch ein von Euch zu wählendes Maximum begrenzt sein. Stellt das durch einen entsprechenden Constraint sicher.

Aufgabe P10.4 Erdkunde-Quiz

Erweitert das vorgegebene Benutzerinterface um Funktionalität. Ziel soll es sein, ein Erdkunde-Spiel zu schreiben, das die Datenbank benutzt, um

- geographische Daten bereitzustellen,
- Fragen zu verwalten, und
- eine Highscore-Tabelle zu führen.

Das Programm soll in zwei Modi laufen. Außerdem soll es jederzeit die Möglichkeit geben, die Liste der aktuellen Highscores abzufragen.

Spielmodus

Im **Spielmodus** soll das Programm nach Eingabe des Spielernamen zunächst die Anzahl vorhandener Fragen in der Datenbank ermitteln und dann eine vorgegebene Anzahl zufällig ausgewählter Fragen stellen. Die Punkte werden zu Beginn jedes neuen Spiels zurückgesetzt. Dabei soll die aktuelle Frage jeweils im vorgegebenen Fragefeld ausgegeben werden.

Dem Benutzer stehen dann zwei Möglichkeiten offen:

- (a) Eine Lösung in das Antwortfeld eintragen und den Knopf **Beantworten** auswählen. Das Programm soll dann mithilfe der abgespeicherten SQL-Anfrage die Richtigkeit der Anfrage überprüfen und gegebenenfalls den Punktestand des Spielers heraufzählen.
- (b) Eine SQL-Anfrage in das SQL-Feld eintragen und den Knopf **Nachschlagen** auswählen. Das Programm soll dann diese SQL-Anfrage abschicken und (wenn ein Ergebnis zurückgeliefert wird) dies im Ausgabefeld formatiert anzeigen.

Liefert die SQL-Anfrage die gesuchte Lösung kann der Spieler diese in das Antwortfeld eintragen und die Antwort abschicken. Pro Spielzug (Quizfrage) sind maximal zwei Anfragen an die Datenbank erlaubt, sonst gilt die Frage als falsch beantwortet.

Nach Ende des Spiels, wenn der Benutzer die vorgegebene Anzahl Fragen beantwortet hat, soll das Ergebnis in die Highscoretabelle eingetragen werden, wenn diese (a) noch keinen Eintrag für diesen Spieler enthält, oder (b) die bisherige Punktzahl niedriger war, als die aktuelle.

Eingabemodus

Im **Eingabemodus** soll das Programm nach Eingabe des Benutzernamen diesen auffordern, eine neue Frage und ein neues SQL-Statement in die bereitgestellten Felder einzutragen. Wenn der Benutzer den Knopf **Eintragen** anwählt, soll das Programm überprüfen, ob die Anfrage fehlerfrei ausgeführt wird, mindestens eine Zeile und genau eine Spalte als Ergebnis zurückliefert. Falls ja, soll das Ergebnis angezeigt und die Anfrage in die Tabelle mit den Quizfragen übernommen werden, sonst soll das Programm dem Benutzer eine Fehlermeldung zurückliefern.