

Übungen zur Vorlesung „Informatisches Modellieren“ im Sommersemester 2010

Übungsblatt Nr. 4

Abgabetermin: entfällt!
Präsenzaufgabenblatt!

Aufgabe 12 (SQL-Anfragen)

Wir betrachten die Datenbank aus Aufgabe 11.

<i>Klassen</i>						<i>Schlachten</i>	
<i>Klasse</i>	<i>Typ</i>	<i>Land</i>	<i>Geschütze</i>	<i>Kaliber</i>	<i>Verdrängung</i>	<i>Name</i>	<i>Datum</i>
Iowa	Schlachtschiff	USA	9	16	46000	Nordkap	26.12.1943
...

<i>Ergebnisse</i>			<i>Schiffe</i>		
<i>Schlacht</i>	<i>Schiff</i>	<i>Status</i>	<i>Name</i>	<i>Klasse</i>	<i>Stapellauf</i>
Nordkap	Scharnhorst	versenkt	Iowa	Iowa	1943
Nordkap	Duke of York	ok	Missouri	Iowa	1944
...

Beispielinformation: „Das Schiff ‚Missouri‘ ist ein Schlachtschiff der Iowa-Klasse (hat also u.a. eine Verdrängung von 46000 Tonnen), das im Jahr 1943 vom Stapel gelaufen ist.“

In Aufgabe 11 waren relationale Ausdrücke zu den folgenden Fragen zu formulieren.

1. Welche Schiffsklassen besaß die USA?
2. Welche Schiffe sind zwischen 1901 und 1908 vom Stapel gelaufen?
3. Welche Schiffe waren 1943 älter als 10 Jahre?
4. Welche Schiffe sind nach 1940 vom Stapel gelaufen sind und hatten ein Kaliber von mindestens 12?
5. An welchen Tagen hat die „Duke of York“ an Schlachten teilgenommen?

Formulieren Sie für jede Teilaufgabe entsprechende SQL-Anfragen.

Aufgabe 13 (SQL – praktische Übungen)

Wir betrachten die Datenbank aus Aufgabe 11.

Die in Aufgabe 12 entwickelten SQL-Abfragen sollen auf einer realen Datenbank getestet werden.

Bearbeiten Sie dazu die folgenden Schritte:

1. Legen Sie zunächst die Tabelle *Schiffe* an: Verwenden Sie dazu den Befehl `CREATE TABLE Schiffe (Name VARCHAR(30), Klasse VARCHAR(30), Stapellauf INT);`, welchen Sie durch Drücken der *Return*-Taste ausführen können. Durch `INSERT INTO Schiffe VALUES('Iowa', 'Iowa', 1943);` kann anschließend die erste Zeile in die Tabelle eingefügt werden.

2. Zeigen Sie den aktuellen Inhalt der Tabelle *Schiffe* mittels `SELECT * FROM Schiffe;` an. Fügen Sie anschließend weitere Einträge in die Tabelle *Schiffe* ein. Zur Überprüfung können Sie die Tabelle zwischendurch anzeigen.
3. Legen Sie die Tabellen *Klassen*, *Schlachten* und *Ergebnisse* an. Füllen Sie diese jeweils mit einigen sinnvollen Einträgen und testen Sie anschließend die in Aufgabe 12 entwickelten SQL-Anfragen.

Aufgabe 14 (case-Anweisung)

Eine Erweiterung der „klassischen“ Version der Sprache SQL enthält eine Operation namens `case`, die die folgende Syntax hat:

```

CASE
  WHEN Prädikat1 THEN Ergebnis1
  :
  WHEN Prädikatk THEN Ergebnisk
  ELSE Ergebnis0
END

```

Dieser Ausdruck überprüft der Reihe nach, welches der Prädikate zu `true` ausgewertet und gibt dann das korrespondierende Ergebnis zurück. Wird kein solches Prädikat gefunden, so wird das Ergebnis der `ELSE`-Zeile zurück gegeben.

Nehmen Sie nun an, dass Sie in einer relationalen Datenbank eine Relation *EnglischArbeit* gespeichert haben, die dem Schema (SchülerNummer, PunkteInProzent) folgt und zu jeder Schülernummer die erreichten Punkte (in Prozent der Maximalpunktzahl) speichert.

1. Formulieren Sie eine SQL-Anfrage, die aus der Relation *EnglischArbeit* eine Relation erstellt, die zu jeder Schülernummer die Note im amerikanischen System enthält.

Das amerikanische Notensystem sei hierbei wie folgt definiert:

Punktebereich	Note
höchstens 40%	F
mehr als 40%, aber höchstens 60%	C
mehr als 60%, aber höchstens 80%	B
mehr als 80%	A

2. Bestimmen Sie nun die Anzahl aller Schüler, die eine bestimmte (vorgegebene) Note haben.