

## Datenbanksysteme II

### SS 2009 – Übungsblatt 3

#### Aufgabe 1: Cursor-Konzept (Skript 3-5)

Bei der Anbindung einer Datenbank an eine Anwendungen kann das Cursor-Konzept genutzt werden.

Erläutern Sie das Cursor-Konzept. Welche Operationen gibt es für Cursor, und was ist ihre Funktionalität?

#### Aufgabe 2: SQLJ (Skript 3-21)

In der Datenbank eines Online-Shops sind Produkte mit ihrem Namen, dem aktuellen Preis und einer zugeordneten Kategorie erfasst. Die Kategorien sind dabei hierarchisch in einem Kategorienbaum angeordnet. Jede Kategorie besitzt dazu genau eine direkte Oberkategorie.

```
CREATE TABLE KATEGORIE (  
  KNR          INT          PRIMARY KEY,  
  KName       VARCHAR(60) NOT NULL,  
  OberKat     INT          REFERENCES KATEGORIE (KNR) )  
  
CREATE TABLE PRODUKT (  
  PNR          INT          PRIMARY KEY,  
  PName       VARCHAR(60) NOT NULL,  
  Preis       INT,  
  Kat         INT          REFERENCES KATEGORIE (KNR) )
```

- a) Skizzieren Sie eine Java-Funktion *getOberKategorien* unter Verwendung von SQLJ, welche zu einer gegebenen Kategoriennummer die Namen aller direkten und indirekten Oberkategorien ausgibt.
- b) Skizzieren Sie eine Java-Funktion *getProdukte* (ebenfalls mit Hilfe von SQLJ), die zu einer gegebenen Kategoriennummer die Namen und Preise der zugehörigen Produkte ausgibt. Wie ändert sich Ihre Funktion, wenn auch alle Produkte der (direkten und indirekten) Unterkategorien ausgegeben werden sollen?

### **Aufgabe 3: JDBC (Skript 3-16)**

Lösen Sie die Aufgabenstellungen von Aufgabe 2 mit JDBC. Verwenden Sie dabei *Prepared Statements*. Hierbei nehmen Sie der Einfachheit halber an, dass eine Datenbankverbindung `con` der Klasse *Connection* bereits besteht.

Zum Setzen und Lesen von Integer-Werten (wie z. B. der Kategoriennummer) verwenden Sie analog zu `setString/getString` die folgenden `java.sql`-Methoden:

```
Interface PreparedStatement: void setInt(int parameterIndex, int x)
```

```
Interface ResultSet:      int getInt(int columnIndex)
```

### **Aufgabe 4: Persistente SQL-Module: Funktionen/Stored Procedures (Skript 3-25)**

Erstellen Sie eine persistente SQL-Funktion 'preiskategorie', die zum Schema aus Aufgabe 2 den Durchschnitt aller Preise zu einer gegebenen Kategorie zurückgibt. Benutzen Sie diese Funktion, um die Namen der Kategorien auszugeben, deren durchschnittlicher Produktpreis zwischen 10 und 100 (Euro) liegt.