

AG Datenbanken und Informationssysteme

Wintersemester 2006 / 2007

Prof. Dr.-Ing. Dr. h. c. Theo Härder
 Fachbereich Informatik
 Technische Universität Kaiserslautern



<http://wwwdvs.informatik.uni-kl.de>

6. Übungsblatt

Für die Übung am Donnerstag, **07. Dezember 2006**,
 von 15:30 bis 17:00 Uhr in 13/222.

Machen Sie sich mit JDBC (Java Database Connectivity) vertraut, um die Programmieraufgaben mit Java und JDBC zu bearbeiten. Ergänzend zur Vorlesung finden Sie einen Verweis auf die Online-Dokumentationen von JDBC, wie etwa *JDBC-Technology Guide: Getting Started*, unter den Online-Ressourcen auf der Website zur Vorlesung <http://wwwdvs.informatik.uni-kl.de/courses/DBAW>.

Aufgabe 1: Eingebettetes SQL am Beispiel „Lieferung“

Gegeben seien folgende Relationen:

```
LIEFERANT (LNR, LNAME, STATUS, ORT)
PROJEKT (PNR, PNAME, ORT)
LIEFERUNG (LNR, TNR, PNR, MENGE)
TEIL (TNR, TNAME, FARBE, GEWICHT)
```

Skizzieren Sie ein Programm mit eingebetteten SQL-Anweisungen, das alle Lieferanten in der Reihenfolge ihrer Lieferantenummer ausgibt. Zu jedem Lieferanten soll direkt eine nach den Projektnummern sortierte Liste aller Projekte ausgegeben werden, die von diesem Lieferanten beliefert werden.

Aufgabe 2: Eingebettetes SQL mit Rekursion, Beispiel „Mitarbeiter-Hierarchie“

In der folgenden DB wird eine Mitarbeiterhierarchie in der üblichen Weise dargestellt:

```
MA (MANR, MANAME, MANAGER, GEHALT)
```

Skizzieren Sie eine Prozedur, die eine Managernummer als Eingabe-Parameter erhält und das Durchschnittsgehalt aller in der Hierarchie (evtl. über mehrere Stufen) untergeordneten Mitarbeiter berechnet.

Aufgabe 3: JDBC

Gegeben sei eine DB, die folgende Relationen enthält:

```
LIEFERANT (LNR, LNAME, STATUS, ORT)
PROJEKT (PNR, PNAME, ORT)
LIEFERUNG (LNR, TNR, PNR, MENGE)
TEIL (TNR, TNAME, FARBE, GEWICHT)
```

Schreiben Sie mit Hilfe von JDBC ein Java-Programm, das alle Lieferanten in der Reihenfolge ihrer Lieferantenummer ausgibt. Zu jedem Lieferanten sollen unmittelbar alle die von diesem Lieferanten belieferten Projekte in der Reihenfolge ihrer Projektnummer aufgelistet werden. Dabei soll weiterhin zu jedem Projekt eine nach den Teilenummern sortierte Liste aller Teile ausgegeben werden, die für dieses Projekt vom jeweiligen Lieferanten geliefert werden. Ihr Programm soll zwei Eingabeparameter erhalten: Benutzername und Passwort, die für die DB-Verbindung benötigt werden.

Aufgabe 4: JDBC und Metadaten

Gegeben sei eine nicht leere relationale Datenbank. Schreiben Sie mit Hilfe von JDBC ein Java-Programm, das für einen als Eingabeparameter eingegebenen Basisrelationennamen zunächst überprüft, ob die Relation in der Datenbank existiert. Ist die Relation vorhanden, so soll die zugehörige vollständige SQL-Anweisung für die Erzeugung der entsprechenden Relation generiert und ausgegeben werden. Die Ausgabe muss den Relationennamen und -attribute mit dem jeweiligen Datentyp und eventueller Längenspezifikation enthalten. Betrachten Sie dabei nur Attribute vom Typ VARCHAR, INTEGER und DECIMAL. Neben einem Relationennamen soll das Programm zwei weitere Eingabeparameter berücksichtigen: Benutzername und Passwort, die für die Verbindung zur Datenbank benötigt werden.

Hinweis:

Verwenden Sie die `getMetaData`-Methode des `Connection`-Objekts für den Zugriff auf das `DatabaseMetaData`-Objekt, mit dessen Methoden Sie dann die benötigten Metadaten abfragen.

Aufgabe 5: Abbildung von Sicht Operationen auf Tabellen (Basisrelationen)

In dieser Aufgabe betrachten wir die Umsetzung von SELECT-Anweisungen die Views verwenden. Dazu haben wir in der Vorlesung die schrittweise Ersetzung von View-Definitionen auf Basisrelationen betrachtet.

Geben seien die folgenden Relationen:

```
STUDENT(MatrNr, FbNr, Name, Vorname, PruefOrd)
MITARBEITER(PersNr, FbNr, Name, Vorname, Gehalt)
PROFESSOR(PersNr, Arbeitsgruppe)
FACHBEREICH(FbNr, Name, Dekan)
```

mit:

```
STUDENT.FbNr ist Fremdschlüssel auf FACHBEREICH.FbNr,
MITARBEITER.FbNr ist Fremdschlüssel auf FACHBEREICH.FbNr,
PROFESSOR.PersNr ist Fremdschlüssel auf MITARBEITER.PersNr,
FACHBEREICH.Dekan ist Fremdschlüssel auf PROFESSOR.PersNr,
MITARBEITER.PROFESSOR-Hierarchie ist abgebildet durch vertikale Partitionierung
```

und die folgenden Sichtdefinitionen:

```
CREATE VIEW StudFbPO96 (MatrNr, Name, Vorname, FbNr, FbName, PO) AS
SELECT s.MatrNr, s.Name, s.Vorname, f.FbNr, f.Name, s.PruefOrd
FROM STUDENT s, FACHBEREICH f
WHERE s.FbNr = f.FbNr
AND PruefOrd = 'PO96';
```

```
CREATE VIEW GehaltFB (FbNr, FBName, GehSum) AS
SELECT f.FbNr, f.Name, sum(m.Gehalt)
FROM MITARBEITER m, FACHBEREICH f
WHERE m.FbNr = f.FbNr
GROUP BY f.FbNr, f.Name
```

```
CREATE VIEW GehaltProfFB (FbNr, FBName, GehSum) AS
SELECT f.FbNr, f.Name, sum(m.Gehalt)
FROM MITARBEITER m, FACHBEREICH f, PROFESSOR p
WHERE m.FbNr = f.FbNr AND m.PersNr = p.PersNr
GROUP BY f.FbNr, f.Name
```

Formulieren Sie für die folgenden Aufgaben jeweils SQL-Anfragen (unter bestmöglichem Einsatz der vordefinierten Views) und geben sie für jede dieser Anfragen an, wie die schrittweise Abbildung auf Basisrelationen durchgeführt wird. Welche Anfragen lassen sich nicht so leicht abbilden? Welchen Zusatzmechanismus muss ein Datenbanksystem bereitstellen, damit alle Anfragen beantwortet werden können? Wie sehen ihre Antworten auf die zuvor gestellten Fragen aus, wenn alle Sichten materialisiert abgespeichert werden?

- a) Finde für alle Studenten die nach der Prüfungsordnung von 1996 studieren, die Professoren die sie unterrichten.
- b) Welcher Dekan verdient am meisten?
- c) In welchem Fachbereich verdienen die Professoren zusammen am meisten Geld?
- d) In welchem Fachbereich verdienen die Mitarbeiter ohne Berücksichtigung der Professorengehälter am meisten Geld?
- e) Wieviel verdienen die Professoren des Fachbereichs 'Informatik' im Mittel?