

Datenbanksysteme II

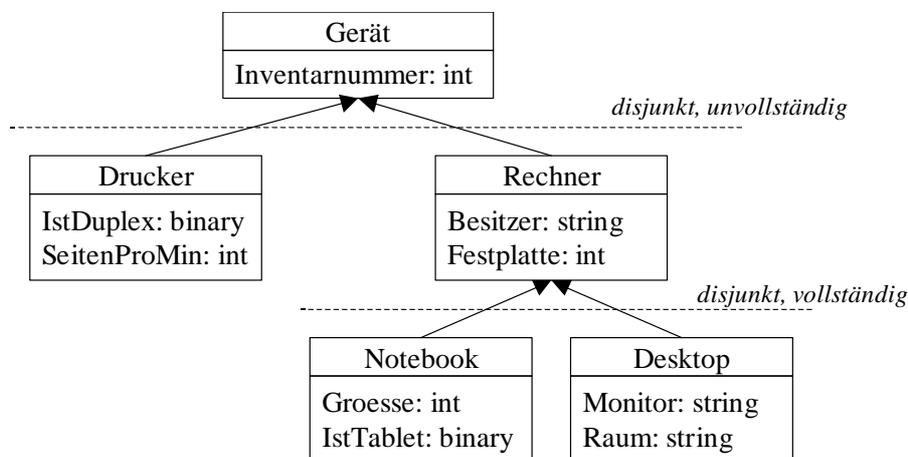
SS 2008 – Übungsblatt 4

Hinweis:

Für dieses Übungsblatt ist keine Bewertung, sondern lediglich Besprechung vorgesehen.

Aufgabe 1 (Objekt-Relationales-Mapping mit Hibernate)

Gegeben sei folgende Klassenhierarchie:



a) Geben Sie für jede der drei von Hibernate unterstützten Generalisierungsansätze

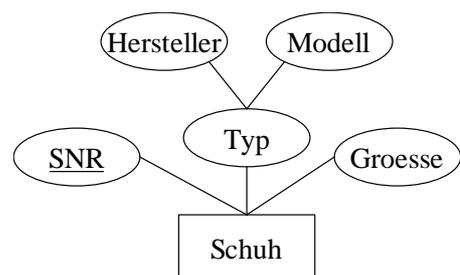
- Table-per-(concrete)-class,
- Table-per-subclass und
- Table-per-hierarchy

jeweils das relationale Modell für die Speicherung in einer relationalen Datenbank an.

b) Diskutieren Sie die drei Fälle im Hinblick auf ihre Effizienz bzgl. Anfragen an die Datenbank.

Aufgabe 2 (SQL:1999: DISTINCT-Typ, ROW-Typ)

Definieren Sie eine SQL:1999-Tabelle Schuhe zur Ab-
 speicherung der in nebenstehender Abbildung in ER-
 Notation spezifizierten Daten zur Verwaltung eines
 Schuhlagers. Verwenden Sie hierbei für das zusammen-
 gesetzte Attribut Typ einen ROW-Typ; für das Attribut
 Größe einen DISTINCT-Typ. Formulieren Sie dann auf
 Ihrer Tabelle eine SQL:1999-Anfrage, die Ihnen die
 Schuhnummern (SNR) für alle Schuhe des Herstellers
 Asics der Größe 43 liefert.



Aufgabe 3 (SQL:1999: Tabellendefinitionen, Anfragen)

Für eine Universitätsanwendung seien u.a. folgende SQL:1999-Typen definiert:

```
CREATE TYPE PersonT
(Name      VARCHAR (40),
 Fak      REF (FakultaetT));

CREATE TYPE StudentT UNDER PersonT
(MatNr     INT,
 Hauptfach VARCHAR (40),
 Nebenfach VARCHAR (40),
 Vordiplom BOOLEAN);

CREATE TYPE ProfT UNDER PersonT
(Buero     BueroT,
 Besoldung CHAR(2),
 DRTitel   VARCHAR(20),
 ForschGebiete VARCHAR(20) ARRAY[5]);

CREATE TYPE BueroT
(Gebaeude VARCHAR(40),
 Stockwerk INTEGER,
 Nummer   INTEGER,
 Telefon   INTEGER);

CREATE TYPE FakultaetT
(FName     VARCHAR (40),
 Dekan     REF (ProfT),
 Professoren REF(ProfT) ARRAY[50],
 Studenten REF(StudentT)
           ARRAY[3000]);
```

- a) Definieren Sie die zur Speicherung von Fakultäten, Studenten und Professoren benötigten Tabellen in SQL:1999 (ohne Tabelle zur Abspeicherung von *BueroT*-Instanzen).
- b) Definieren Sie zu dem Typ *FakultaetT* eine Methode *countStudents* (*vd* *BOOLEAN*) *RETURNS INTEGER* die zu einer gegebenen Fakultät (d.h. zu dem Fakultätsobjekt, auf dem sie aufgerufen wird), die Anzahl aller Studenten mit Vordiplom (*vd* = *TRUE*) bzw. ohne Vordiplom liefert.
- c) Schreiben Sie einen Konstruktor für *StudentT*. Dieser soll neben der Attributinitialisierung das neu generierte Objekt in das *ARRAY Studenten* des entsprechenden Fakultätsobjekts eintragen.
- d) Geben Sie SQL:1999-Anfragen (ohne Join) zur Beantwortung folgender Fragen an:
 1. In welchen Fakultäten sind Informatik-Studenten (Haupt-/Nebenfach) immatrikuliert?
 2. Wie heisst der Dekan der Fakultät, in der der Student mit Matrikelnummer 234567 immatrikuliert ist?
 3. In welchen Fakultäten sind Studenten mit Namen "Schneider" immatrikuliert?
 4. Finden Sie alle Professoren mit Name und Telefonnummer, die Dekan einer Fakultät sind und die Besoldungsstufe C4 haben.
- e) Definieren Sie in SQL eine Tabelle Dekane vom Typ *ProfT*. Schreiben Sie in SQL einen Ausdruck, der den Dekan der Fakultät 'Informatik' in die Tabelle Dekane einträgt.

Aufgabe 4 (Multimengen und Tupel)

Gegeben sei das folgende NF²-Schema einer Abteilungsdatenbank (vgl. Übungsblatt 3).

Abteilung							
AbtNr	Budget	Ausstattung		Mitarbeiter			
		Anzahl	Typ	MaNR	Funktion	Projekt	
						PNr	PName

- a) Schreiben Sie eine DDL-Anweisung, welche das Schema in einer SQL-Tabelle umsetzt. Nutzen Sie dafür die Tupel- und Multimengentypkonstruktoren von SQL:2003.
- b) Geben Sie SQL:2003-Anfragen zur Beantwortung folgender Anfragen an:
 1. Welche Abteilungen beschäftigen Mitarbeiter mit der Funktion eines Programmierers?
 2. Welche Projekte werden in mehr als einer Abteilung bearbeitet?