

Datenbanksysteme I
WS 2007/08 – Übungsblatt 3

1. Aufgabe (Kardinalitätsangaben)

- a) Erstellen Sie als ER-Modell und UML-Modell im Vergleich die M:N-Beziehung „Student schreibt Klausur“, wobei zu einer Klausur Datum und Fach relevant sind. Für jeden Studenten soll pro Klausur eine Note gespeichert werden können.
- b) In UML soll durch Angabe der Multiplizität ausgedrückt werden: Eine Klausur werde von mindestens 3 Studenten geschrieben. Ein Student schreibe bis zu 50 Klausuren.
- c) Erstellen Sie ein Beispiel einer mehrstelligen Beziehung in UML mit Multiplizitätsangaben.

2. Aufgabe (UML-Entwurf)

Erstellen Sie einen UML-Entwurf (Festlegung von Klassen mit Attributen, sowie Assoziationen mit Multiplizität) zur nachfolgenden Banken-Miniwelt. Verwenden Sie dabei die UML-Konstrukte der Generalisierung (mit Angabe der Spezialisierungsart, d.h. Vollständigkeit und Disjunktheit) und Aggregation (Referenzsemantik- bzw. Wertesemantik).

Änderungen zur Beschreibung aus Blatt 2 sind kursiv hervorgehoben.

Es gibt Banken und Kunden.

Eine Bank, welche durch ihre Bankleitzahl und ihren Namen beschrieben wird, besitzt mindestens eine Zweigstelle. Jede Zweigstelle hat eine Adresse und eine innerhalb der Bank eindeutige Zweigstellenummer.

Eine Zweigstelle verwaltet Konten. Wir unterscheiden Guthaben- und Kreditkonten. Zu beiden Arten gibt es wiederum Unterarten (z. B. bei den Guthabekonten Girokonten und Sparbücher, bei Kreditkonten z.B. Konten für allgemeine Kredite und Konten für Baukredite).

Jedes Konto hat eine innerhalb der Miniwelt eindeutige Kontonummer, einen Kontostand und einen Zinssatz. Bei Kreditkonten gibt es eine vereinbarte monatliche Tilgungssumme.

Von den Kunden werden Vorname, Nachname, Geburtsdatum, Wohnanschrift und eine bankintern eindeutige Personenkennzahl gespeichert. Ein Kunde kann mehrere Guthaben- und Kreditkonten haben. *Er muss aber mindestens ein Guthabekonto besitzen.*

3. Aufgabe (ER nach UML)

Überführen Sie das ER-Modell einer Bibliothek (Vorlesungsskript Kapitel 2, Folie 25) in ein UML-Diagramm. Entwerfen Sie dazu geeignete, zusätzliche Oberklassen zur Generalisierung/Spezialisierung soweit sinnvoll und benutzen sie Aggregation wenn möglich.

4. Aufgabe (Referentielle Integrität)

Gegeben sei folgendes relationales Schema:

```
CREATE TABLE Angestellter (  
  ANr int PRIMARY KEY,  
  AName varchar(30)  
)  
  
CREATE TABLE Projekt (  
  PNr int PRIMARY KEY,  
  PName varchar(30),  
  Projektleiter int,  
  FOREIGN KEY (Projektleiter) REFERENCES Angestellter, NOT NULL  
)  
  
CREATE TABLE Mitarbeit (  
  ANr int,  
  PNr int,  
  Arbeitsstd int,  
  PRIMARY KEY (ANr, PNr),  
  FOREIGN KEY (ANr) REFERENCES Angestellter, NOT NULL,  
  
  FOREIGN KEY (PNr) REFERENCES Projekt, NOT NULL  
)
```

- a) Ergänzen Sie die Fremdschlüsseldefinitionen im relationalen Schema um geeignete Löschr- und Update-Regeln. Die Löschrregeln sollen folgendes sichern:
1. Scheidet ein Angestellter aus (d.h. wird er gelöscht), so wird auch die Information gelöscht, in welchen Projekten er mitgearbeitet hat.
 2. Wird ein Projekt gelöscht, so auch alle Informationen, welche Mitarbeiter mit wie vielen Stunden dort gearbeitet haben.
- b) Welche Löschrregel ist für die Projektleiterbeziehung zu wählen, damit die Konsistenz nicht gefährdet wird? Begründen Sie Ihre Entscheidung.