

## TEIL I :: Modellierung und Realisierung einer Datenbank

### **Aufgabenbeschreibung**

Im Datenbankpraktikum wollen wir uns mit der Vorlesungsplanung des Fachbereiches Informatik beschäftigen. Ihnen ist der Sachverhalt aus studentischer Sicht (Vorlesungspläne) bestens bekannt, so dass auf eine detaillierte Beschreibung der Ist-Situation verzichtet werden kann. Dabei soll nicht nur die studentische Sicht, sondern auch das Deputat der Dozenten, die Raumplanung und -belegung berücksichtigt werden. Wir wollen zunächst verschiedene konzeptionelle Datenmodelle dazu entwickeln und diese anschließend in Form einer datenbankgestützten Anwendung umsetzen.

### **Zur Vorgehensweise**

Bitte verwenden Sie als Hilfe die zugehörigen Arbeitsblätter, diese enthalten hilfreiche Tipps und unterstützen Sie so bei der Suche nach einem Lösungsansatz.

Die nachfolgenden Aufgabenstellungen müssen in der aufgeführten Reihenfolge bearbeitet werden, da eine andere Vorgehensweise didaktisch nicht sinnvoll wäre.

Im Praktikum sollen Sie ein bestimmtes Problem gezielt analysieren, um dieses anschließend in einem Datenmodell abzubilden und dazu eine prototypische Anwendung zu entwickeln. Während der Analysephase werden Sie sich vor allem damit beschäftigen, ein so genanntes ER<sup>1</sup> – Diagramm zu erstellen. Ferner müssen Sie dieses Diagramm in ein objektorientiertes und ein relationales Modell transformieren. Verwenden Sie zur Dokumentation Ihrer Ergebnisse, die in der Datenbankvorlesung vorgestellten Notationen (ERD<sup>1</sup>, UML<sup>2</sup> und Bachman Diagramm).

Wir möchten Ihnen empfehlen, zumindest das ER – Diagramm zuerst von Hand auf Papier zu skizzieren.

### **Lernziele**

1. Modellieren eines ‚realen‘ Problems in Form eines ERM – bzw. objektorientierten Modells.
2. Üben verschiedener Notationen (UML, ERM).
3. Arbeiten mit einem CASE-Tool (EclipseUML).
4. Erkennen verschiedener Abbildungsprobleme zwischen der relationalen – und objektorientierten Welt.
5. Arbeiten mit Datenbanken (Microsoft Access, Oracle)

### **Aufgabe 1 :: Konzeptuelle Datenmodellierung (ERM)**

Erstellen Sie ein Entity-Relationship-Modell für die o. g. Vorlesungsplanung.

Gehen Sie systematisch vor, indem Sie zuerst die Entitäten, dann ihre Attribute und zuletzt die Beziehungen (Relationships) definieren. Beachten Sie, dass für jeden Entitätstyp ein Schlüssel (Attributkombination) festgelegt werden muss.

Eine Vorlesung ist genau einem Studiengang und einem Semester zugeordnet. Eine Vorlesung kann verschiedene Räume zu verschiedenen Zeiten belegen. Ein Dozent kann mehrere Vorlesungen und Übungen abhalten. Wo bietet sich die Verwendung einer schwachen Entität an? Wie kann strukturell eine Doppelbelegung der Räume verhindert werden?

Als Ergebnis sind ein ER-Diagramm und eine Attributliste mit Konsistenzbedingungen verlangt.

---

<sup>1</sup> Entity-Relationship Diagram (auch: ERD)

<sup>2</sup> (U)nified (M)odelling (L)anguage

## Aufgabe 2 :: Relationaler Datenbankentwurf

- a) Transformieren Sie das ER-Modell aus Aufgabe 1a in ein relationales Modell.  
Hinweis: Komplexe Objekte/Entitäten/Beziehungen sind zu normalisieren.
- b) Erstellen Sie eine Access-Datenbank.  
Hinweis: Benützen Sie die Entwurfsansicht (Tabellen/Neu/Entwurfsansicht) zur Definition der Tabellen und ihrer Attribute. Konsistenzbedingungen können in das Feld Gültigkeitsregel eingetragen werden. Der Ausdruckseditor (Schaltfläche [...] rechts neben dem Eingabefeld) hilft Ihnen bei der Erstellung syntaktisch korrekter Bedingungen. Die Beziehungen können am anschaulichsten in der Beziehungsansicht (Schaltfläche: Beziehungen), d.h. im Bachman-Diagramm erstellt werden.
- c) Geben Sie so viele Testdaten ein, dass Sie wenigstens einen Belegungsplan für Ihr Semester erstellen können.
- d) Erstellen Sie ein Formular für die Raumbelugung, bei welchem die zulässigen Werte aus einem Auswahlfeld ausgewählt werden können.  
Lernziel: Bei einer „guten Strukturierung“ der Datenbank kann Sie der Formularassistent bei der Erstellung einer Maske besser unterstützen, indem er die Daten für die Kombifelder aus Tabellen bezieht, die über Fremdschlüssel verbunden sind.
- e) Erzeugen Sie ein Ausgabeformular für die Vorlesungspläne der Studenten in Matrixform (Wochentage horizontal, Zeit vertikal, Matrixelemente enthalten Vorlesung, Dozent und Raum)  
Hinweis: Benützen Sie eine ‚lokale‘ Pivot-Tabelle.

## Aufgabe 3 :: Objektorientierter Datenbankentwurf

- a) Entwerfen Sie ein objektorientiertes Modell der Vorlesungsplanung. Erstellen Sie zuerst entweder von Hand oder mit Hilfe des „EclipseUML Studios“ (das UML-Java-Entwicklungswerkzeug von Omondo ein Hierarchiediagramm für die Fachklassen, Sequenzdiagramme zur Darstellung von wenigstens eine nichttriviale Methode (z.B. Transaktion: Belegung eines Raumes vornehmen). Achten Sie darauf, dass Ihr Modell die notwendige Präzision aufweist und die Navigationsmöglichkeiten bietet, welche Ihre Anwendung benötigt, z.B. wenn Ihr Programm die Vorlesungen eines Dozenten anzeigen soll, benötigen Sie eine Objektreferenz zu den Vorlesungen, wenn Sie auch den Dozenten einer Vorlesung ermitteln wollen, muss auch ein Rückverweis angelegt werden.
- b) Ergänzen Sie das OO – Diagramm aus Aufgabe um technische Belange (z.B. Containerklassen).  
Hinweis: Wenn Sie die Modellierung mit Omondo durchgeführt haben können Sie ausgehend vom Modell aus a) die bereits generierten Klassendefinitionen weiterverwenden. Die Klassendefinitionen sind mit einem Editor oder einem integrierten Entwicklungswerkzeug (IDEs) z.B. „EclipseUML Studio“ um Zugriffsmethoden (setXX, getXX) und um Objekt-Beziehungen zu ergänzen. Die Zugriffsmethoden, so genannten ‚Getter‘ – und ‚Setter‘ Methoden, werden vom EclipseUML Studio dann automatisch generiert, wenn Sie diese als Properties (Attribute) einer Klasse im „EclipseUML Studio“ anlegen. Daher brauchen Sie diese nicht selbst programmieren, sondern nur ggf. ergänzen, um die gewünschte Semantik zu erhalten. Für Mehrfachbeziehungen (1:n, n:m) verwenden Sie die Collection Klassen, welche im Java Package *java.util* enthalten sind.  
Als Ergebnis sollte wenigstens die Verwaltung und Navigation einer 1:n-Beziehung (z.B. Dozent – Veranstaltungen) möglich sein. Dazu müssen Sie die generierten Methoden ergänzen, um die notwendige Funktionalität (z.B. nur ein Dozent ist einer Vorlesung zugeordnet  $\Leftarrow$  Semantik der Beziehung) zu erhalten.

Aus Zeitgründen werden wir keinen Datenbankanschluss (z.B. über einen Objekt-Relational-Mapper wie Toplink, JPOX oder eine objektorientierte Datenbank wie FastObjects, Versant) durchführen.