

Arbeitsblätter zu Teil I des Praktikums

Allgemeine Hilfsmittel

- ✍ Bitte benutzen Sie bei Schwierigkeiten mit spezifischem Domänenwissen das Internet als Recherchemöglichkeit (beispielsweise Google oder Wikipedia).
- ✍ Einige Teilaufgaben erfordern die Konsultation der Benutzerdokumentation der verwendeten Software, da mitunter einige SQL Konstrukte datenbankspezifisch sind und somit nicht dem SQL-Standard entsprechen. Konsultieren Sie Ihren Tutor!

Zu den Werkzeugen

Nutzen sie - soweit möglich - nachfolgende Werkzeuge:

- ✍ *EclipseUML Studio* für das Objektmodell (erhältlich über <http://www.omondo.com/> bzw. eines der Netzlaufwerke des Fachbereiches Informatik). Die notwendige Lizenz ist gesondert erhältlich.
- ✍ Microsoft Access zur Datenverwaltung und zum Erstellen der Anwendungsmasken.
- ✍ Das Skriptum der Datenbankvorlesung.
- ✍ Papier + Bleistift + Ihr Ideenreichtum.

Aufgabe 1 :: Konzeptuelle Datenmodellierung (ERM)

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

Beispiel:



Abbildung 1: ER - Diagram Dozent, Veranstaltung, Semester

Attributliste:

Dozent


- ✍ Nummer
- ✍ Name
- ✍ Vorname
- ✍ Fachgebiet
- ✍ Strasse
- ✍ PLZ
- ✍ Ort

Veranstaltung


- ✍ Nummer
- ✍ Name
- ✍ Stundenblöcke
- ✍ Wochentage
- ✍ Prüfung (Ja/Nein)

Semester

- ✍ Nummer
- ✍ Fachbereich
- ✍ Anzahl Studenten

-  Ergänzen Sie die Attributliste um die zugehörigen Datentypen (s. Dokumentation der von Ihnen eingesetzten Datenbank. Studieren Sie vor allem das Kapitel der Datentypdefinitionen).


Überlegen Sie sich die weiteren Entitäten und Relationships: 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?

-  Vervollständigen Sie das ER - Diagramm um die fehlenden Entitäten, Relationships und Attribute.

Aufgabe 2 :: Relationaler Datenbankentwurf








Übersicht: Das Modell aus Aufgabe 1 ist in ein relationales umzusetzen und in Access zu implementieren

Aufgabe 2a :: Transformation ERM RM





-  Transformieren Sie Ihr ERM in ein relationales Modell und normalisieren Sie dieses bis zur 3. Normalform.

Beispiel:

Dozent

-  Nummer Integer ID-Schlüssel
-  Name String(20)
-  Vorname String(20)
-  Fachgebiet String(20)
-  Strasse String(20)
-  PLZ Integer(5)
-  Ort String(20)

Veranstaltung

-  Nummer Integer ID-Schlüssel
-  Name String(20)
-  Prüfung Boolesch
-  DozentenNr Integer Fremdschlüssel (Dozent)

Hinweise: Komplexe Objekte/Entitäten/Beziehungen sind zu normalisieren. n:m-Beziehungen sind durch eine Beziehungstabelle zu realisieren.

Aufgabe 2b :: Access Datenbank

-  Erstellen Sie aus dem relationalen Modell eine Access Datenbank.

Hinweise: Benützen Sie die Entwurfsansicht (Tabellen/Neu/Entwurfsansicht) zur Definition der Tabellen und ihrer Attribute. Konsistenzbedingungen können in das Feld Gültigkeitsregel eingetragen werden. Der Ausdruckeditor (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.

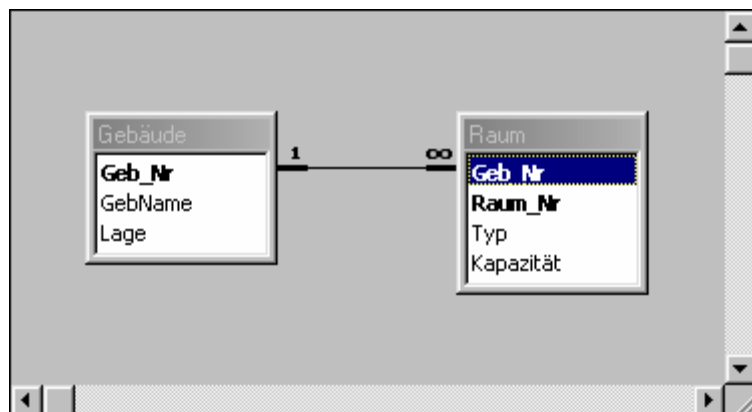


Abbildung 2: Erstellen der Beziehungen für einzelne Tabellen. (Ausschnitt aus dem DB-Schema)

Aufgabe 2c :: Dateneingabe

- ✍ Geben Sie so viele Stammdaten in der Tabellenansicht ein, dass Sie wenigstens einen Belegungsplan für Ihr Semester (WI4) erstellen können.

Hinweise: Wenn Sie auf eine Tabelle doppelklicken, erstellt Access eine Standardansicht der Tabelle in die Sie Daten eingeben können. ✍ siehe folgende Abbildung

	VorlesNr	Veranstaltung	Verantwortlicher	DozentenNr	Fachbe	Semester
▶ +	1	Einführung in die BWL	7	20	WI	1
+	2	Einführung in die EDV	21	21	WI	1
+	3	Mathematik	11	11	WI	1
+	4	Grundlagen der VWL	22	20	WI	1
+	5	Systemanalyse, Algorith.	6	6	WI	2
+	6	Englisch	10	23	WI	2
+	7	Mathematik	24	11	WI	2
+	8	Programmier-Praktikum	6	19	WI	2

Abbildung 3: Erstellen der Beziehungen für einzelne Tabellen. (Ausschnitt aus dem DB-Schema)

Aufgabe 2d :: Raumbellegung

- ✍ Erstellen Sie ein Formular für die Belegungsplanung.

Hinweise: Damit Sie leichter konsistente Belegungen erfassen können, sollte die Maske Auswahlfelder mit den zulässigen Daten erzeugen. Beachten Sie, dass die angezeigten Felder und das gebundene Feld (das in die Datenbank eingetragen wird) unterschiedlich sein können. Dadurch ist es möglich z.B. den Veranstaltungsnamen anzuzeigen obwohl nur der Identschlüssel der Veranstaltung als Fremdschlüssel als Datenbankeintrag übernommen wird.

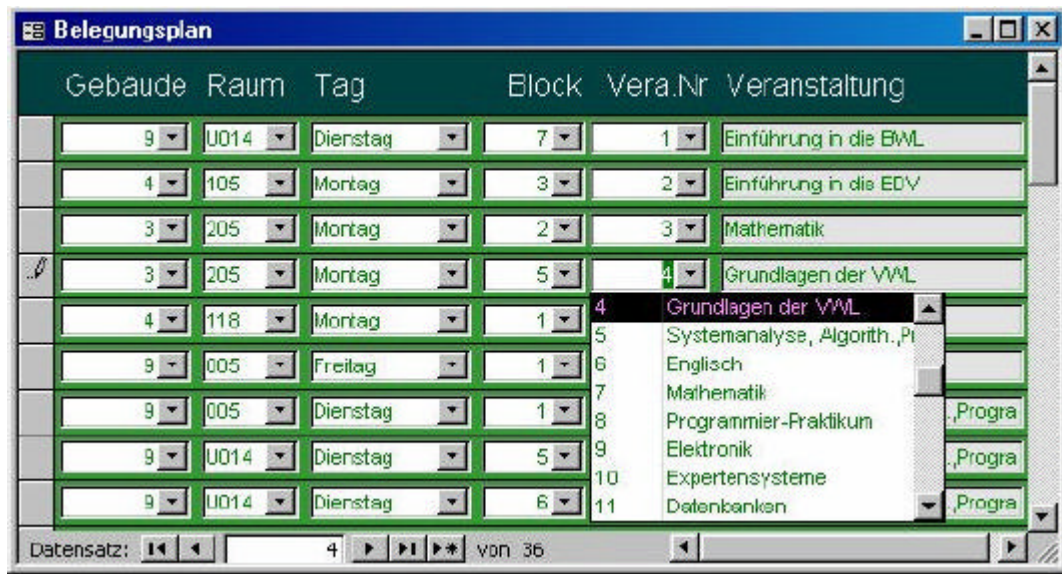


Abbildung 4: Beispiel eines Formulars für die Belegungsplanung.

Aufgabe 2e :: Stundenplan Studenten

✍ Erzeugen Sie einen matrixartigen Vorlesungsplan für die Studenten.

Hinweise: Die unten stehende Darstellung erhalten Sie durch die Erstellung einer geeigneten Sicht (In Access Abfrage genannt) und die Verwendung des Access „Kreuztabellenabfrage Assistenten“.

Block	Montag	Dienstag	Mittwoch	Donnerstag	Freitag
2	3-204, Expertensysteme, Frick	3-205, Datenbanken, Laux	3-204, Operation Research,	9-038, CIM 1 (Mawi), Glöckle	9-039, CIM 1 (Mawi), Glöckle
3	3-204, Expertensysteme, Frick	3-205, Datenbanken, Laux	3-204, Operation Research,	9-038, CIM 1 (Mawi), Glöckle	9-039, CIM 1 (Mawi), Glöckle
4	9-128, Unternehmensführung, Roth				
5	9-128, Unternehmensführung, Roth	9-031, Prakt. Datenbanken, Laux		3-213, Prakt. Expertensystem	
6		9-031, Prakt. Datenbanken, Laux		3-213, Prakt. Expertensystem	

Abbildung 5: Beispiel Vorlesungsplan in Matrixform für Studenten.

Aufgabe 3 :: Objektorientierter Datenbankentwurf

Übersicht: Es sind ein Klassenhierarchiediagramm und ein Sequenzdiagramme zur Darstellung der Transaktionen Raumbelugung zu erstellen. Für eine komplexe Beziehung (z.B. Dozent – Veranstaltung) ist die Verwaltung und Navigation dieser Beziehung zu implementieren.

Aufgabe 2a :: Entwurf UML Klassen und Sequenzdiagramm

Die Diagramme sollten mit einem UML Werkzeug Ihrer Wahl erstellt werden. Benutzen Sie vorzugsweise ein UML Werkzeug das in den Labors 9-029 und 9-030 installiert ist (z.B. EclipseUML von Omondo).

Achten Sie darauf, Beziehungen zwischen einzelnen Objekten immer mit einer Assoziation zu erstellen. Denken Sie bitte auch daran den Assoziationstyp gegebenenfalls genauer zu spezifizieren (z.B. Aggregation o. Komposition). Die endgültige Semantik einer Klasse oder Beziehung wird durch die Methoden bestimmt, welche Sie in Aufgabe 3b) programmieren werden.

Beispiele zur Praktikumsaufgabe

Klassenhierarchie

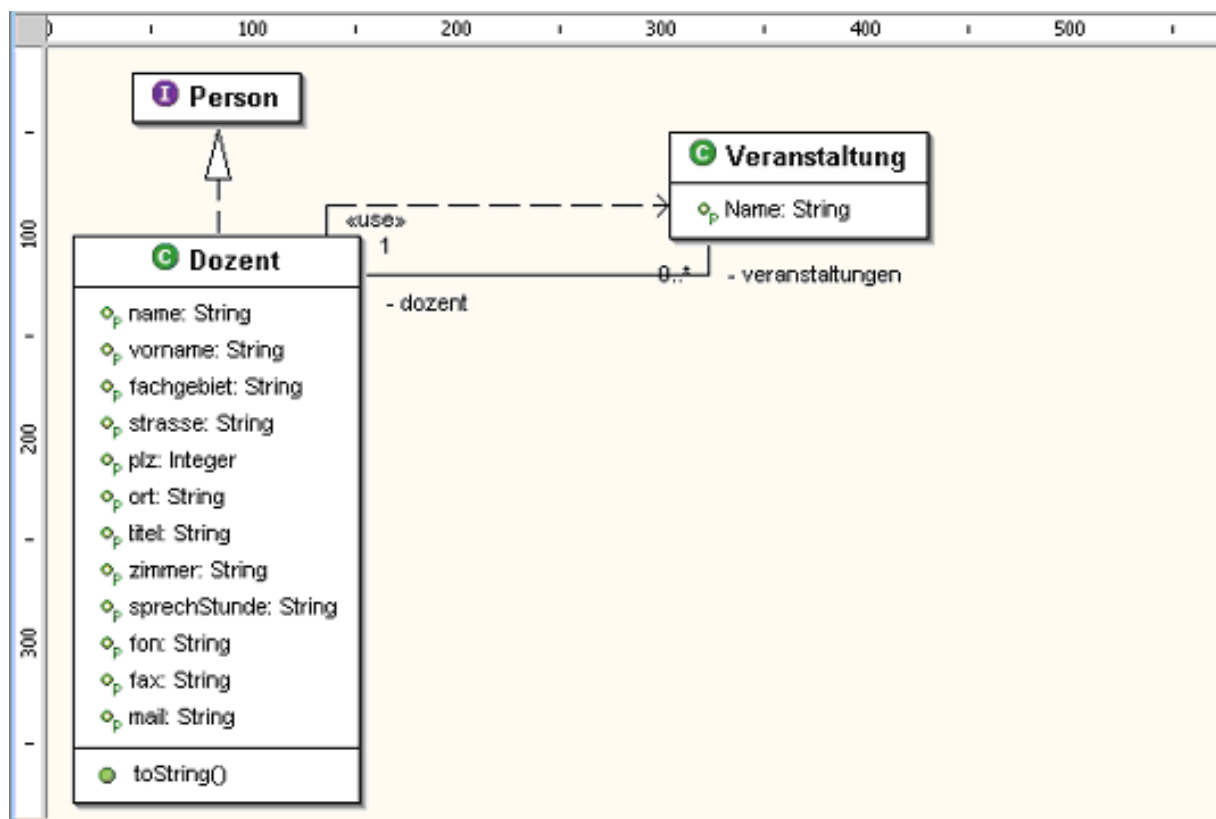


Abbildung 6: Ausschnitt eines Klassendiagramms – erstellt mittels EclipseUML.

Wie kann die Tatsache ausgedrückt werden, dass als Zeiträume nur die Wochentage Montag bis Freitag und die (bekannten) Zeitblöcke 1 ... 8 zulässig sind? Wie realisiert man eine bidirektionale Assoziation (Beziehung) in Java? Wie kann sichergestellt werden, dass jedes Beziehungstripel (Raum, Wochentag, Block) nur einmal belegt werden kann (Doppelbelegungsschutz)?

Klassenbeziehungen (Associations)

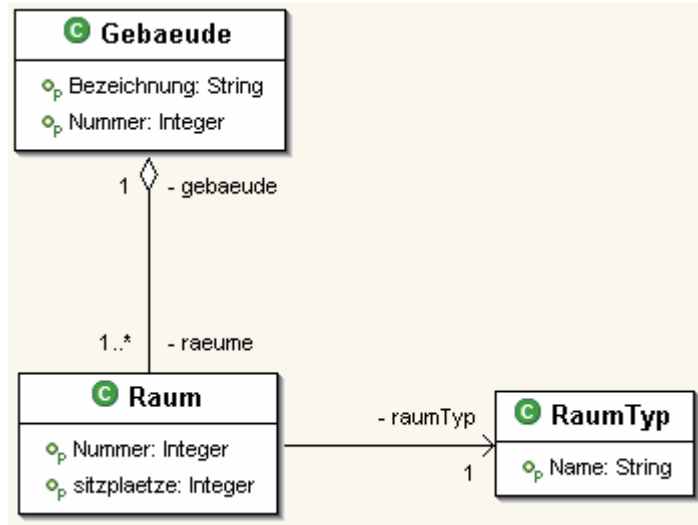


Abbildung 7: Beispiel der Klassenbeziehungen für die Vorlesungsplanung (unvollständig).

✍ Erstellen Sie ein Klassendiagramm, das die Hierarchie und die Beziehungen (Assoziationen) zwischen den Klassen beschreibt.

Hinweise: Entwickeln Sie das Klassendiagramm, indem Sie beide Beispiele in einem Diagramm zusammenfassen und die fehlenden Klassen und Beziehungen ergänzen und die o.g. Anforderungen möglichst weitgehend erfüllen.

Sequenzdiagramm (Transaktion Raum löschen)

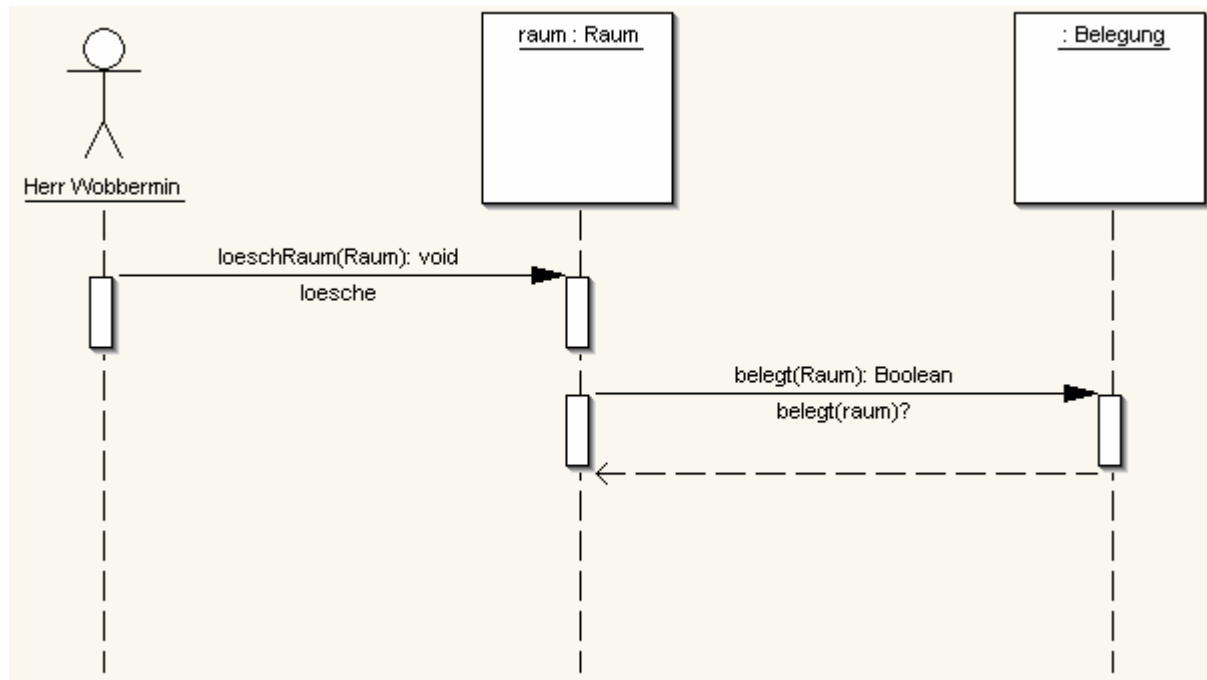


Abbildung 8: Beispiel eines Transaktionsdiagramms ‚Raum Löschen‘

✍ Erstellen Sie ein Sequenzdiagramm (Transaktion) für die Raumbellegung.

Aufgabe 2b :: Beziehungsmanagement Dozent – Veranstaltung

Übersicht: Erstellen Sie Klassendefinition von Ihrem Klassendiagramm soweit diese nicht automatisch von der DIE generiert wurden. Ergänzen Sie die Definitionen um technische Belange (z.B. Containerklassen für komplexe Beziehungen) und versehen Sie die Methoden mit der notwendigen Funktionalität zu ihrer Verwaltung.

Java Codebeispiel :: Raummanagement

```
package de.fhreitlingen.inf.wi.datenbanken.oo;

import java.util.Collection;
import java.util.Iterator;

public class Gebaeude {

    private Integer nummer;
    private String bezeichnung;

    private Collection raeume;

    public Iterator raeumeIterator() {
        return raeume.iterator();
    }
    public boolean addRaeume(de.fhreitlingen.inf.wi.datenbanken.oo.Raum element) {
        return raeume.add(element);
    }
    public boolean removeRaeume(de.fhreitlingen.inf.wi.datenbanken.oo.Raum element) {
        return raeume.remove(element);
    }
}


/*
 *
 * [ ... ]
 */


public class Raum {

    private Integer nummer;
    private Integer sitzplaetze;
    private RaumTyp raumTyp;

    private Gebaeude gebaeude;

    public Gebaeude getGebaeude() {
        return gebaeude;
    }
    public void setGebaeude(Gebaeude gebaeude) {
        this.gebaeude = gebaeude;
    }
}
```

 Schreiben Sie Methoden `addVeranstaltung(..)`, `removeVeranstaltung(..)`, `getVeranstaltungen()` für die Klasse „Dozent“.

 Schreiben Sie Methoden `setDozent(..)`, `getDozent()` für die Klasse „Veranstaltung“.

Zur Kontrolle, ob alle geforderten Funktionalitäten richtig implementiert sind, können Sie das Testprogramm „DozentVeranstaltungsTest.java“ verwenden. Die korrekten Ergebnisse können Sie exemplarisch der Datei „DozentVeranstaltungsTestResults.txt“ entnehmen.