

Thema:	Datenbanken – Probeklausur		
Dozent:	Prof. Dr. Stephan Kleuker	Seitennummer:	Seite 1 von 12
Studiengang:	Wirtschaftsinformatik	Jahrgang:	I03
Datum:	15.11.2004	Bearbeitungszeit:	120 Minuten
Matrikelnummer:		Name:	(mögliche) Lösungsskizze



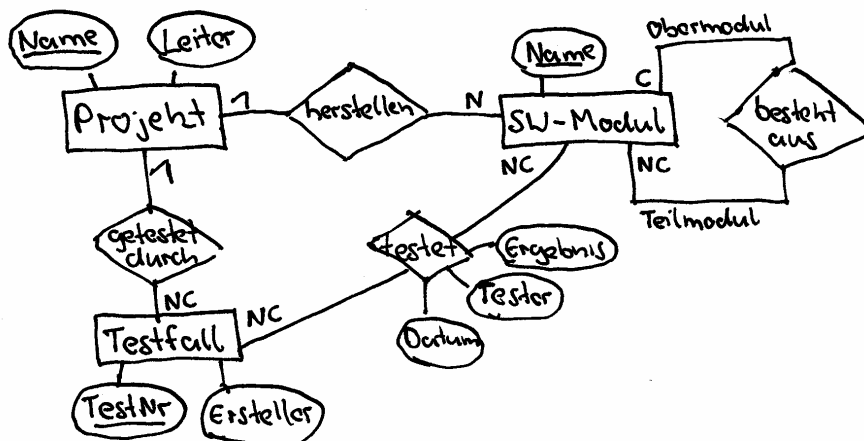
Hinweis: Die ersten vier Aufgaben sind ein Ausschnitt aus einer DB-Klausur für Wirtschaftsingenieure (d.h. sie sind vielleicht etwas leichter als das, was sie erwartet). Insgesamt handelt es sich um Beispielaufgaben, es ist also kein Schluss möglich, dass nicht genannte Themen nicht vorkommen oder dass genannte Themen nur in dieser Form abgefragt werden können. Insbesondere können sich Schwerpunkte verschieben. Man kann insgesamt 100 Punkte erreichen, 50 Punkte werden zum Bestehen benötigt.

Die hier gezeigten Lösungen sind teilweise mögliche Beispiellösungen, es kann Alternativen geben, die ebenfalls zur vollen Punktzahl geführt hätten. Alternative Lösungsvorschläge können bei mir zur Korrektur abgegeben werden. Falls Flüchtigkeitsfehler auffallen, bitte an mich melden.

Zugelassene Hilfsmittel: keine.

1) ER-Diagramm erstellen (5+3 = 8 Punkte)

- a) Formulieren sie folgende Sachverhalte als ER-Diagramm (Entitäten, Relationen, Attribute, Schlüssel, Kardinalitäten): Eine IT-Firma erstellt Software in verschiedenen Projekten. Die Entwicklungsinformationen zu den verschiedenen Projekten sollen festgehalten werden. Jedes Projekt, das durch seinen Namen eindeutig gekennzeichnet ist und einen Leiter hat, stellt einen oder mehrere SW-Module her. Jeder SW-Modul, der durch seinen Namen eindeutig erkennbar ist, gehört zu genau einem Projekt. Jeder SW-Modul kann aus mehreren SW-Modulen bestehen, jeder SW-Modul kann in maximal einem übergeordneten SW-Modul genutzt werden. Zu jedem Projekt gehört eine Liste von Testfällen, die durch eine Nummer eindeutig identifizierbar sind und einen Ersteller haben. Jeder Testfall bezieht sich auf einen oder mehrere SW-Module, jeder SW-Modul sollte in mindestens einem Testfall vorkommen, was aber nicht immer garantiert ist. Es gibt Testfälle, die frühzeitig erstellt werden und erst wesentlich später SW-Modulen zugeordnet werden. Für jede Ausführung eines Testfalls wird der letzte Zeitpunkt der Ausführung, der Name des Testers und das Testergebnis festgehalten (auf eine Historie der Testfälle wird verzichtet), dabei kann sich die Testausführung auf eine Teilmenge der zugeordneten Module beziehen.



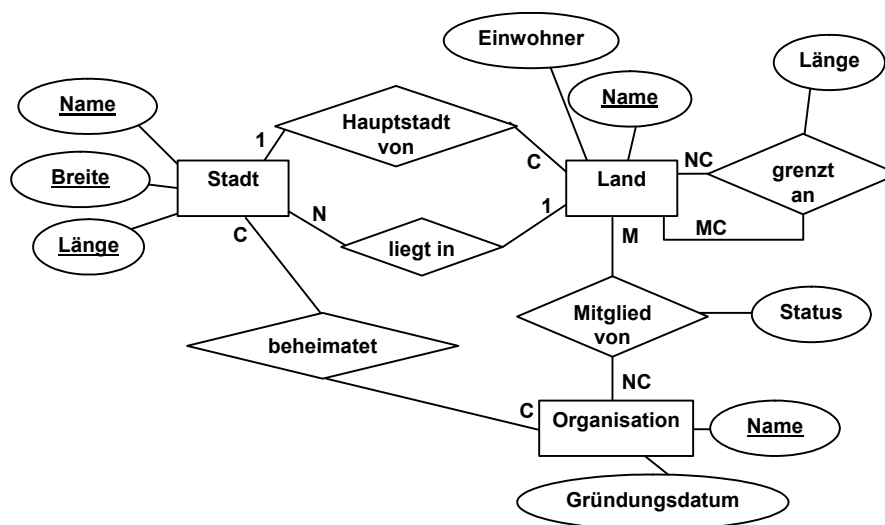
Thema:	Datenbanken – Probeklausur		
Dozent:	Prof. Dr. Stephan Kleuker	Seitennummer:	Seite 2 von 12
Studiengang:	Wirtschaftsinformatik	Jahrgang:	I03
Datum:	15.11.2004	Bearbeitungszeit:	120 Minuten
Matrikelnummer:		Name:	(mögliche) Lösungsskizze



- b) In Ihrem Diagramm sollte es einen Zyklus geben (zwei Entitätsmengen sind auf unterschiedlichen Wegen miteinander verknüpft). [Ist dies nicht der Fall, ergänzen sie einen Zyklus ihrer Wahl.] Beschreiben sie zunächst warum es sich generell dabei um ein Indiz für ein Modellierungsproblem handelt und warum der Zyklus in ihrer Modellierung beibehalten werden muss oder eine Relation gelöscht werden kann.

zu b): Bei einem Zyklus ist es möglich, dass eine Relation redundant ist, da sie aus den anderen Relationen berechnet werden kann. Die Relation „getestet durch“ kann nicht weggelassen werden, da es Testfälle geben kann, die noch keinen SW-Modulen zugeordnet wurden, diese wären über die anderen Relationen nicht erreichbar. Die Relation „herstellen“ kann nicht weggelassen werden, da nicht jeder SW-Modul garantiert in einem Testfall vorkommt. Die Relation test kann nicht weggelassen werden, da sonst nicht abgeleitet werden kann, welche Tests sich nun auf einen SW-Modul beziehen. (und da sie Zusatzinformationen enthält).

2) ER-Diagramm in Relationen (Tabellen) übersetzen (7+2 = 9 Punkte)
Gegeben sei das folgende ER-Diagramm:



- a) Leiten sie aus dem Diagramm Tabellen ab, die in den Spalten (insofern die Werte der im Diagramm genannten Attribute nicht NULL sind) keine NULL-Werte enthalten. Markieren sie einen Schlüsselkandidaten in jeder Tabelle. Vermeiden sie die Ableitung überflüssiger Einzeltabellen.
- b) Nennen sie zwei Korrekturen, die sie im ER-Diagramm durchführen würden, um das Modell realistischer zu machen, die sich nicht auf Attribute beziehen.

zu a)

Tabelle Stadt: Name | Breite | Länge | Land.Name

Tabelle Land: Einwohner | Name | Stadt.Name | Stadt.Breite | Stadt.Länge

Tabelle Organisation: Name | Gründungsdatum


Thema:	Datenbanken – Probeklausur			
Dozent:	Prof. Dr. Stephan Kleuker	Seitennummer:	Seite 3 von 12	
Studiengang:	Wirtschaftsinformatik	Jahrgang:	I03	
Datum:	15.11.2004	Bearbeitungszeit:	120 Minuten	
Matrikelnummer:		Name:	(mögliche) Lösungsskizze	

Tabelle Mitgliedschaft (Mitglied von): Land.Name | Organisation.Name | Status

Tabelle Residiert (beheimatet): Stadt.Name | Stadt.Breite | Stadt.Länge | Organisation.Name

Tabelle Grenze (grenzt an): Land1.Name | Land2.Name

zu b)

Die Kardinalität von „beheimatet“ sollte geändert werden:

Jede Stadt kann (NC) Organisationen beheimaten.

Jede Organisation ist in (N) Städten beheimatet. [Statt (N) könnte auch (1) stehen]

3) Relationen in Normalform (1+2+3+3 = 9 Punkte)

Grundsätzlich soll für folgende Umformungen gelten, dass sie dabei nur die unbedingt notwendigen Veränderungen vornehmen sollen. Markieren sie die Schlüsselkandidaten der Tabellen.

Nehmen sie an, dass zur Verwaltung von Studierenden folgende (Excel)-Tabelle genutzt wird:

MatNr	Name	TelefonNr	Studium	Firma	Betreuer	Fach	Note

Dabei gelten folgende funktionale Abhängigkeiten (weitere Abhängigkeiten lassen sich aus diesen berechnen)

- I. {MatNr} -> {Name, TelefonNr, Studium, Firma, Betreuer}
- II. {TelefonNr} -> {MatrNr}
- III. {MatNr, Fach} -> {Note}
- IV. {TelefonNr, Fach} -> {Note}
- V. {Firma} -> {Betreuer}
 - a) Begründen sie formal, warum {TelefonNr} -> {Firma} gilt.
 - b) Nennen sie alle Schlüsselkandidaten.
 - c) Bringen sie die Tabelle mit dem Standardverfahren in die zweite Normalform.
 - d) Bringen sie die Tabellen aus c) mit dem Standardverfahren in die dritte Normalform.

zu a)

kann aus II. mit I. transitiv geschlossen werden

zu b)

{MatNr, Fach} und {TelefonNr, Fach}

zu c)

Tabelle Note: MatNr | Fach | Note [statt MatNr könnte auch TelefonNr stehen]


Tabelle Studi: MatNr | Name | TelefonNr | Studium | Firma | Betreuer [zwei Schlüsselkandidaten {MatNr} und {TelefonNr}]

zu d)

nur Tabelle Studi nicht in 3NF, Umformung ergibt

Tabelle Studi: MatNr | Name | TelefonNr | Studium | Firma

Tabelle Betreuung: Firma | Betreuer

Thema:	Datenbanken – Probeklausur			
Dozent:	Prof. Dr. Stephan Kleuker	Seitennummer:	Seite 4 von 12	
Studiengang:	Wirtschaftsinformatik	Jahrgang:	I03	
Datum:	15.11.2004	Bearbeitungszeit:	120 Minuten	
Matrikelnummer:		Name:	(mögliche) Lösungsskizze	

Eine Tabelle MatNr | TelefonNr ist nicht notwendig (nicht Ergebnis des Verfahrens), da es sich um Schlüsselattribute handelt (es kann so auch keine redundante Information in den Tabellen entstehen)

4) SQL-Anfragen formulieren (3+3+3+3+3+3 = 18 Punkte)

Gegeben seien folgende Tabellen zur Projektverwaltung:

Projekt

PrNr	PrName	PrLeiter
1	Notendatenbank	Wichtig
2	Adressdatenbank	Wichtig
3	Fehlzeitendatenbank	Wichtig

Arbeitspaket

PrNr	PakNr	PakName	PakLeiter
1	1	Analyse	Wichtig
1	2	Modell	Wichtig
1	3	Implementierung	Mittel
2	4	Modell	Durch
2	5	Implementierung	Mittel
3	6	Modell	Schnitt
3	7	Implementierung	Hack

Arbeit


PakNr	MiName	Anteil
1	Wichtig	50
1	Klein	30
2	Winzig	100
3	Hack	70
4	Maler	40
4	Schreiber	30
6	Maler	30
6	Schreiber	40
7	Hack	50

Formulieren sie die folgenden Textzeilen jeweils als SQL-Anfragen.

- Geben sie die Namen der Personen an, die gleichzeitig Projektleiter und Arbeitspaketleiter sind (nicht unbedingt im gleichen Projekt !).
- Geben sie die Namen der Personen an, die mehrere Arbeitspakete leiten.
- Geben sie die Arbeitspaketnamen an, denen noch keine Arbeit (also kein Mitarbeiter) zugeordnet wurde.
- Geben sie eine Liste der Projektnamen zusammen mit der Summe der zugeordneten Arbeitsanteile (Anteil ist in Prozent angegeben, diese sind zu addieren) aus.
- Geben sie die Namen aller Mitarbeiter an, die zu mehr als 100% arbeiten.
- Geben sie alle Arbeitspaketnamen (PakName) aus, an denen zwei oder mehr unterschiedliche Personen beteiligt sind. („Implementierung“ gehört z.B. nicht dazu, da sie nur von „Hack“ durchgeführt wird.)

-- a)

```
SELECT Projekt.PrLeiter
FROM Projekt, Arbeitspaket
WHERE Projekt.PrLeiter=Arbeitspaket.PakLeiter;
```

Thema:	Datenbanken – Probeklausur			
Dozent:	Prof. Dr. Stephan Kleuker	Seitennummer:	Seite 5 von 12	
Studiengang:	Wirtschaftsinformatik	Jahrgang:	I03	
Datum:	15.11.2004	Bearbeitungszeit:	120 Minuten	
Matrikelnummer:		Name:	(mögliche) Lösungsskizze	

-- b)

```
SELECT A1.PakLeiter
FROM Arbeitspaket A1, Arbeitspaket A2
WHERE A1.PakNr<A2.PakNr
      AND A1.PakLeiter=A2.PakLeiter;
```

-- c)

```
SELECT Arbeitspaket.PakName
FROM Arbeitspaket
WHERE Arbeitspaket.PakNr IN ((SELECT PakNr FROM Arbeitspaket)
                             MINUS (SELECT PakNr FROM Arbeit));
```

-- d)

```
SELECT Projekt.PrName, SUM(Arbeit.Anteil)
FROM Projekt, Arbeitspaket, Arbeit
WHERE Projekt.PrNr=Arbeitspaket.PrNr
      AND Arbeitspaket.PakNr=Arbeit.PakNr
GROUP BY Projekt.PrName;
```

-- e)

```
SELECT Arbeit.MiName
FROM Arbeit
GROUP BY Arbeit.MiName
HAVING SUM(Arbeit.Anteil)>100;
```

-- f)


```
SELECT Arbeitspaket.PakName
FROM Arbeitspaket, Arbeit
WHERE Arbeitspaket.PakNr=Arbeit.PakNr
GROUP BY Arbeitspaket.PakName
HAVING COUNT(DISTINCT MiName)>1;
```

5) Tabelle in SQL definieren (5 Punkte)

Geben sie den SQL-Befehl zur Erzeugung der Tabelle Arbeitspaket aus der vorherigen Aufgabe in SQL an (nur **CREATE**, keine **INSERT**). Neben den dort „sichtbaren“ Randbedingungen, sollen folgende Bedingungen aufgenommen werden:

- Kein Eintrag darf leer sein.
- Die Person Winzig darf nie Arbeitspaketleiter sein.
- Alle Arbeitspakete mit dem Namen Analyse dürfen nie von der Person Hack geleitet werden.
- Alle Arbeitspakete mit dem Namen Implementierung müssen von der Person Mittel oder der Person Hack geleitet werden.

```
CREATE TABLE Arbeitspaket(
  PrNr NUMBER NOT NULL,
  PakNr NUMBER,
```

Thema:	Datenbanken – Probeklausur			
Dozent:	Prof. Dr. Stephan Kleuker	Seitennummer:	Seite 6 von 12	
Studiengang:	Wirtschaftsinformatik	Jahrgang:	I03	
Datum:	15.11.2004	Bearbeitungszeit:	120 Minuten	
Matrikelnummer:		Name:	(mögliche) Lösungsskizze	

```

PakName VARCHAR2(18) NOT NULL,
PakLeiter VARCHAR2(8) NOT NULL,
PRIMARY KEY(PakNr),
CONSTRAINT A1 FOREIGN KEY(PrNr) REFERENCES Projekt(PrNr),
CONSTRAINT A2 CHECK(PakLeiter<>'Winzig'),
CONSTRAINT A3 CHECK(NOT(PakName='Analyse')
                     OR NOT(PakLeiter='Hack')),
CONSTRAINT A4 CHECK(NOT(PakName='Implementierung')
                     OR (PakLeiter='Mittel' OR PakLeiter='Hack'))
);

```

6) Auswertung von DB-Anfragen (2+2+2 = 6 Punkte)

Gegeben seien folgende SQL-Anfragen an die Datenbank aus der Aufgabe 4.

- (a)

```

SELECT Projekt.PrName
FROM Projekt, Arbeitspaket
WHERE Projekt.PrNr=Arbeitspaket.PrNr
      AND Arbeitspaket.PakName='Implementierung'
      AND Arbeitspaket.PakLeiter='Mittel';

```
- (b)

```

SELECT A1.MiName
FROM Arbeit A1, Arbeit A2
WHERE A1.PakNr<A2.PakNr
      AND A1.MiName=A2.MiName;

```
- (c)

```

SELECT Arbeitspaket.PakName, COUNT(Arbeit.MiName)
FROM Arbeitspaket, Arbeit
WHERE Arbeitspaket.PakNr=Arbeit.PakNr
GROUP BY Arbeitspaket.PakName;

```

Geben sie jeweils die zugehörige Ausgabe an und beschreiben sie informell, was der Zweck der Anfrage ist.

-- a) In welchen Projekten wird die Implementierung von Mittel
-- geleitet?

PRNAME

Notendatenbank

Adressdatenbank


-- b) Welche Mitarbeiter arbeiten an mindestens zwei
-- unterschiedlichen Arbeitspaketen?

MINAME

Maler

Schreiber

Hack

Thema:	Datenbanken – Probeklausur			
Dozent:	Prof. Dr. Stephan Kleuker	Seitennummer:	Seite 7 von 12	
Studiengang:	Wirtschaftsinformatik	Jahrgang:	I03	
Datum:	15.11.2004	Bearbeitungszeit:	120 Minuten	
Matrikelnummer:		Name:	(mögliche) Lösungsskizze	

-- c) Gib zu jedem Arbeitspaket die Anzahl der Mitarbeiter an,
-- die zumindest anteilig am Projekt arbeiten.

PAKNAME **COUNT (ARBEIT.MINAME)**

```
-----
Analyse                      2
Implementierung              2
Modell                       5
```

3 Zeilen ausgewählt.

***/**

7) Änderung von Tabellen (3+1+1+1+1 = 7 Punkte)

Gehen sie davon aus, dass bei der Realisierung der Tabellen aus der Aufgabe 4 alle sichtbaren Beziehungen zwischen den Tabellen als FOREIGN KEYS spezifiziert sind.

- Mit welchen SQL-Befehl(en) ändern sie den PrName von „Notendatenbank“ in „Leistungsdatenbank“?
- Mit welchen SQL-Befehl(en) ändern sie die PrNr von 1 auf 8?

Gehen sie davon aus, dass alle FOREIGN KEY-Beziehungen mit ON DELETE CASCADE spezifiziert sind.

- Was passiert, wenn „**DELETE FROM Projekt WHERE Pr=1**“ ausgeführt wird?
- Was passiert, wenn „**DELETE FROM Arbeitspaket WHERE PakName='Implementierung'**“ ausgeführt wird?
- Was passiert, wenn „**DROP TABLE Arbeit**“ ausgeführt wird?

zu a)

```
UPDATE Projekt
SET PrName='Leistungsdatenbank'
WHERE PrName='Notendatenbank';
```

zu b)

Hinweis: Bei der Bewertung sollte eigentlich a) einen und b) drei Punkte erhalten.


```
ALTER TABLE Arbeitspaket DISABLE CONSTRAINT A1;
UPDATE Arbeitspaket
SET PrNr=8
WHERE PrNr=1;
UPDATE Projekt
SET PrNr=8
WHERE PrNr=1;
ALTER TABLE Arbeitspaket ENABLE CONSTRAINT A1;
```

zu c)

1. Zeile in Projekt, 1.-3. Zeile in Arbeitspaket, 1.-4. Zeile in Arbeit werden gelöscht

zu d)

3., 5., 7. Zeile in Arbeitspaket, 4., 9. Zeile in Arbeit werden gelöscht

Thema:	Datenbanken – Probeklausur			
Dozent:	Prof. Dr. Stephan Kleuker	Seitennummer:	Seite 8 von 12	
Studiengang:	Wirtschaftsinformatik	Jahrgang:	I03	
Datum:	15.11.2004	Bearbeitungszeit:	120 Minuten	
Matrikelnummer:		Name:	(mögliche) Lösungsskizze	

zu e)

Tabelle kann ohne Probleme gelöscht werden.

8) Trigger (5 Punkte)

Schreiben sie einen Trigger, der beim Einfügen eines Arbeitspakets überprüft, ob der neu einzutragende PakLeiter schon zweifach oder häufiger in der Tabelle vorkommt. Ist das der Fall, soll das Einfügen mit einer Fehlermeldung abgebrochen werden.

```

CREATE OR REPLACE TRIGGER NichtDreiPakete
BEFORE INSERT ON Arbeitspaket
FOR EACH ROW
DECLARE
    anzahl NUMBER;
BEGIN
    SELECT COUNT(*)
    INTO anzahl
    FROM Arbeitspaket
    WHERE Arbeitspaket.PakLeiter=:NEW.Pakleiter;

    IF anzahl>=2
    THEN
        RAISE_APPLICATION_ERROR(-20004,'Nicht mehr als zwei' ||
                                     ' Pakete');
    END IF;
END;
/

```

9) PL/SQL-Prozeduren (5+6 = 11 Punkte)


- Schreiben sie eine Prozedur **neuesProjekt**, die einen Projektnamen (PrName), einen Projektleiter (PrLeiter) und einen Arbeitspaketnamen (Pakname) übergeben bekommt. Es wird ein Projekt und ein Arbeitspaket mit dem Projektleiter als Paketleiter (PakLeiter) eingerichtet. Denken sie daran, dass neue Werte für die PRIMARY KEYS berechnet werden müssen (sie können davon ausgehen, dass alle Tabellen nicht leer sind).
- Schreiben sie eine Prozedur **arbeitVerteilen**, die für jeden Mitarbeiter (MiName sei eindeutig) die Summe der Arbeitsanteile addiert und auf 100% aufrechnet. Arbeitet z.B. ein Mitarbeiter in genau zwei Projekten mit 20% und mit 60%, soll nach der Aufrechnung 25% und 75% in der Tabelle stehen. $(25 = \frac{20}{20+60} * 100)$

zu a)

```

CREATE OR REPLACE PROCEDURE neuesProjekt(
    pname Projekt.PrName%TYPE,
    pleiter Projekt.PrLeiter%TYPE,
    pakname Arbeitspaket.Pakname%TYPE) IS

```


Thema:	Datenbanken – Probeklausur			
Dozent:	Prof. Dr. Stephan Kleuker	Seitennummer:	Seite 9 von 12	
Studiengang:	Wirtschaftsinformatik	Jahrgang:	I03	
Datum:	15.11.2004	Bearbeitungszeit:	120 Minuten	
Matrikelnummer:		Name:	(mögliche) Lösungsskizze	

```

neuprnummer Projekt.PrNr%TYPE;
neupaknr Arbeitspaket.PakNr%TYPE;
BEGIN
    SELECT MAX(PrNr)+1
    INTO neuprnummer
    FROM Projekt;

    SELECT MAX(PakNr)+1
    INTO neupaknr
    FROM Arbeitspaket;

    INSERT INTO Projekt VALUES(neuprnummer,pname,pleiter);
    INSERT INTO Arbeitspaket
VALUES(neuprnummer,neupaknr,pakname,pleiter);

END;
/


```

```

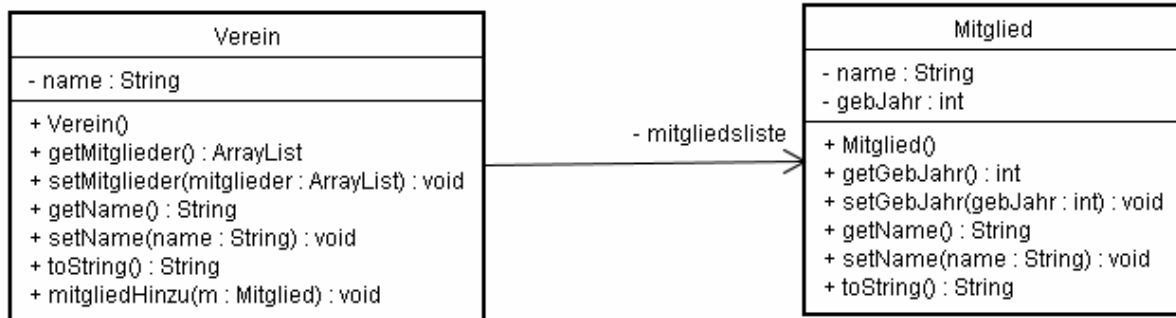
zu b)
CREATE OR REPLACE PROCEDURE arbeitVerteilen IS
    CURSOR Mitarbeiter IS
        SELECT DISTINCT Arbeit.MiName
        FROM Arbeit;
    summe NUMBER;
BEGIN
    FOR M IN Mitarbeiter
    LOOP
        SELECT SUM(Arbeit.Anteil)
        INTO summe
        FROM Arbeit
        WHERE Arbeit.MiName=M.Miname;

        UPDATE Arbeit
        SET Anteil=Anteil*100/summe
        WHERE Arbeit.MiName=M.Miname;
    END LOOP;
END;
/

```

Thema:	Datenbanken – Probeklausur			
Dozent:	Prof. Dr. Stephan Kleuker	Seitennummer:	Seite 10 von 12	
Studiengang:	Wirtschaftsinformatik	Jahrgang:	I03	
Datum:	15.11.2004	Bearbeitungszeit:	120 Minuten	
Matrikelnummer:		Name:	(mögliche) Lösungsskizze	

10) JDO (3+4 = 7 Punkte)




Gehen sie davon aus, dass das obige Klassendiagramm in Java umgesetzt und dass für die Klassen erfolgreich eine Datenbank aufgesetzt wurde, die JDO unterstützt. (Objekte der Klasse Verein haben eine Exemplarvariable mitgliedsliste vom Typ ArrayList). Ihre Aufgabe besteht darin, folgende Anfragen unter der Nutzung von JDO in Java zu formulieren. Dabei können sie davon ausgehen, dass eine erfolgreiche Verbindung zur Datenbank aufgebaut wurde, eine Transition gestartet wurde und ihnen ein PersistenceManager-Objekt zur Verfügung steht. Der Methodenkopf sieht also wie folgt aus:

```
public void aufgabeX(PersistenceManager pm) {
```

- Geben Sie die Namen aller Mitglieder aus, die vor 1990 geboren sind.
- Geben sie die Namen aller Vereine aus, in denen „Heinz Meier“, geboren 1984, Mitglied ist.

```
public static void aufgabe10a(PersistenceManager pm) {
    Extent alleMi = pm.getExtent(Mitglied.class, true);
    String filter="this.gebJahr<1990";
    Query q=pm.newQuery(alleMi,filter);
    Collection c= (Collection)q.execute();
    for(Iterator i=c.iterator();i.hasNext();)
        System.out.println((Mitglied)i.next().getName());
}
```

```
public static void aufgabe10b(PersistenceManager pm) {
    Extent alleVer = pm.getExtent(Verein.class, true);
    String filter="this.mitgliedsliste.contains(o)
        && (o.name=="Heinz Meier" && o.gebJahr==1984)";
    Query q=pm.newQuery(alleVer,filter);
    q.declareVariables("Mitglied o");
    Collection c= (Collection)q.execute();
    for(Iterator i=c.iterator();i.hasNext();)
        System.out.println((Verein)i.next().getName());
}
```

Thema:	Datenbanken – Probeklausur			
Dozent:	Prof. Dr. Stephan Kleuker	Seitennummer:	Seite 11 von 12	
Studiengang:	Wirtschaftsinformatik	Jahrgang:	I03	
Datum:	15.11.2004	Bearbeitungszeit:	120 Minuten	
Matrikelnummer:		Name:	(mögliche) Lösungsskizze	

11) Einige Fragen zu Datenbanken (2+3+3+2+2+3 = 15 Punkte)

- Wie kann man in einer Tabelle für eine bestimmte Spalte zählen, wie oft der Wert NULL in dieser Spalte steht?
- Beschreiben sie den Nutzen und die möglichen Probleme von Views in SQL.
- Beschreiben sie, was man unter „Impedance Mismatch“ versteht und nennen sie drei Lösungsstrategien für Projekte, die vor diesem Problem stehen, für das dahinter liegende Problem.
- Wenn ein DBMS keine Transaktionssteuerung nutzt, können „Unrepeatable Reads“ und „Lost Updates“ auftreten. Beschreiben sie diese Fehlerarten.
- Beschreiben sie, wozu die SQL-Befehle **COMMIT** und **ROLLBACK** dienen.
- Geben sie eine Tabelle an, die sich in dritter Normalform, aber nicht vierter Normalform befindet. Welche Umformungen machen sie, damit sie Tabellen in vierter Normalform mit gleicher Aussagekraft erhalten?

zu a)

```
SELECT COUNT(*)
FROM <Tabelle>
WHERE <Tabelle>.<bestimmteSpalte> IS NULL;
```

zu b) [mindestens zwei der folgenden Plus- und Minuspunkte sind zu nennen]


- Mit Views ist es möglich komplexe Anfragen unter einem einfachen Namen als virtuelle Tabelle abzuspeichern, auf die dann wie bei einer Tabelle zugegriffen werden kann.
- Man kann Rechte auf Views vergeben, so dass Personen nicht unmittelbar die Basistabellen einsehen können
- Man kann Views zur Bearbeitung zur Verfügung stellen, so dass Personen bestimmte Tabelleninhalte in den Basistabellen nicht sehen oder verändern können
- + Inhalte von Views können nur verändert werden, wenn sich die Änderungen unter Einhaltung aller Konsistenzregeln eindeutig in die Basistabellen übertragen lassen
- + Bestehen Veränderungsmöglichkeiten auf Daten, die Views sichtbar sind, können Daten aus den Views verschwinden oder neue Daten erscheinen
- + (Nicht materielle) Views müssen bei ihrer Nutzung immer wieder berechnet werden, so dass recht einfach aussehende Anfragen sehr viel Rechenzeit verbrauchen können

zu c)

Klassensysteme mit Vererbung und beliebigen dynamischen Beziehungen zwischen Objekten lassen sich nur schwer in einer relationalen Datenbank abbilden, so dass ein Speichern und Lesen von Objekten, die wieder andere Objekte referenzieren, sehr aufwändig wird.

Lösungsansätze sind:

- Auswahl einer objektorientierten Datenbank
- Auswahl eines Frameworks, wie JDO, dass die Verbindung zwischen Applikation und SW erleichtert
- Integration von Datenbank- und SW-Entwicklung z.B. durch EJB

Thema:	Datenbanken – Probeklausur			
Dozent:	Prof. Dr. Stephan Kleuker	Seitennummer:	Seite 12 von 12	
Studiengang:	Wirtschaftsinformatik	Jahrgang:	I03	
Datum:	15.11.2004	Bearbeitungszeit:	120 Minuten	
Matrikelnummer:		Name:	(mögliche) Lösungsskizze	

zu d)

T1: Lese A Lese A

T2: Schreibe A

T1 macht zweimal die gleiche Leseaktion, erhält aber unterschiedliche Ergebnisse
(Unrepeatable Read)

T1: Schreibe A Schreibe B

T2: Schreibe A

Nach Abschluss von T1 befindet sich nicht mehr die erwartete Information in A, da T2
zwischenzeitlich geändert hat (Lost Update)

zu e)

Mit COMMIT wird dem DBMS mitgeteilt, dass die zuletzt vom Nutzer in seiner Session
durchgeführten Änderungen endgültig übernommen werden sollen. Mit ROLLBACK sollen
die Änderungen der letzten Session zurück genommen werden. (Mit ROLLBACK A werden
Änderungen, die nach einem SAVEPOINT A erfolgten, zurück genommen.)

zu f)

Name	Kind	Auto
Ute	Uwe	BMW
Ute	Uwe	DKW
Ute	Udo	BMW
Ute	Udo	DKW

nach

Name	Kind	Name	Auto
Ute	Uwe	Ute	BMW
Ute	Udo	Ute	DKW

{Name} ->> {Kind}

{Name} ->> {Auto}