Sammlung aller Staatsexamensaufgaben der Prüfungsnummer

66114

Datenbank- und Betriebssysteme (vertieft)

66114

Datenbank- und Betriebssysteme (vertieft)

Frühjahr 2006

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____	**Frühjahr**	
Kennwort: _____	**2006**	**66114**
Arbeitsplatz-Nr.: _____		

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

- Prüfungsaufgaben -

Fach: **Informatik (vertieft studiert)**

Einzelprüfung: **Datenbank- und Betriebssysteme**

Anzahl der gestellten Themen (Aufgaben): 4 Aufgaben, von denen 2 gemäß den untenstehenden Auswahlregeln zu bearbeiten sind!

Anzahl der Druckseiten dieser Vorlage: 14

Zu den zwei Themenschwerpunkten A (Datenbanksysteme) und B (Betriebssysteme) ist jeweils entweder die Aufgabe 1 oder 2 zu wählen. Auf der Vorderseite des Kopfbogens sind im Feld „Gewähltes Thema: Nr." die Nummern der beiden gewählten Aufgaben anzugeben (z. B. A1, B2)!

Bitte wenden!

Themenschwerpunkt A

(Datenbanksysteme)

Aufgabe 1:

1. Allgemeine Fragen

Bewerten Sie die folgenden Aussagen a) - i). Geben Sie für jede Aussage an, ob diese richtig oder falsch ist. Begründen Sie Ihre Aussage in jedem Fall.

a) Anwender müssen sich beim Mehrfachzugriff auf gleiche Daten eines Datenbanksystems absprechen um Fehler bzw. Chaos zu vermeiden.

b) Eine Relation kann maximal 255 Attribute besitzen.

c) Ein Tupel kann mehrfach in einer Relation enthalten sein.

d) Die physische Datenstruktur entspricht in einem relationalen Datenbanksystem immer der logischen Datenstruktur.

e) Primärschlüsselattribute verschiedener Relationen dürfen nicht den gleichen Namen haben.

f) In jeder Relation gibt es immer mindestens einen Schlüsselkandidaten.

g) Alle Relationen eines relationalen Schemas müssen miteinander in Beziehung stehen.

h) Alle Relationen in einem Datenbankschema müssen normalisiert sein.

i) Attribute der selben Relation dürfen nicht den gleichen Namen haben.

Definieren Sie folgende Begriffe bzw. Abkürzungen j) - n) im Kontext von Datenbanksystemen:

j) View

k) DDL

l) Domäne

m) Constraint

n) Referentielle Integrität

Fortsetzung nächste Seite!

2. Transaktionen und Transaktionsverarbeitung

a) Nennen und definieren Sie die vier wesentlichen Merkmale einer Datenbanktransaktion.

b) Beschreiben Sie zwei mögliche Probleme, die bei der unsynchronisierten Transaktionsausführung auftreten können.

c) Mit welchem Konzept versucht man die Probleme der unsynchronisierten Transaktionsausführung zu vermeiden? Beschreiben Sie, wie dieses Konzept in Datenbanksystemen umgesetzt wird.

3. Das Entity-Relationship-Modell

a) Bei einem 2-stelligen Relationship-Typ muss die Kardinalität (Funktionalität) festgelegt werden. Beschreiben, definieren und illustrieren Sie alle möglichen Formen.

b) Was versteht man unter der sog. (min,max)-Notation bei einem 2-stelligen Relationship-Typ? Erläutern Sie den Vorteil dieser Notation gegenüber der herkömmlichen Notation und geben Sie ein kurzes Beispiel.

c) Wie unterscheidet sich ein schwacher Entitätstyp von einem starken Entitätstyp? Skizzieren Sie an einem kleinen Beispiel, wie man einen schwachen Entitätstyp im ER-Modell darstellt.

d) Wie nennt man die Beziehung in einem ER-Modell, die Generalisierung bzw. Spezialisierung darstellt? Welche Eigenschaften haben Entities, die durch eine derartige Beziehung verbunden sind? Illustrieren Sie Ihre Erklärung ggf. an einem Beispiel.

Fortsetzung nächste Seite!

4. ER-Modellierung und Relationenmodell

a) ER-Modellierung

Erstellen Sie das Modell eines fiktiven Grundbuchamtes in ER-Notation. Wo möglich bzw. sinnvoll, sollen 3-fache Beziehungen und Generalisierung/Spezialisierung verwendet werden. Attribute von Entitäten und Beziehungen sind anzugeben; Schlüsselattribute werden durch Unterstreichen gekennzeichnet. Die Kardinalitäten von Beziehungen und - falls nötig - Rollennamen sollen ins Diagramm aufgenommen werden. Führen Sie Surrogatschlüssel nur ein, falls es nötig ist.

„Im Grundbuchamt"

Grundstücke werden im Grundbuchamt eindeutig durch eine Grundbuchnummer identifiziert und durch einen Lageplan genauer beschrieben. Grundstücke liegen nebeneinander und stehen dadurch in einer Nachbarschaftsbeziehung zueinander. Es gibt Grundstücke, die nicht neben anderen Grundstücken liegen und Grundstücke, die an mehrere andere Grundstücke angrenzen.

Ein **Haus** steht genau auf einem Grundstück. Grundstücke sind entweder nicht, mit einem oder mit mehreren Häusern bebaut. Jedes Haus kann eindeutig anhand seiner Hausnummer identifiziert werden.

Zu jedem Haus gehört mindestens ein **Stockwerk.** Aus städtebaulichen Gründen darf ein Haus aber maximal aus 30 Stockwerken bestehen. Ein Stockwerk gehört immer genau zu einem Haus. Jedes Stockwerk wird eindeutig durch seine Stockwerksnummer identifiziert. Ein Grundriss beschreibt die Aufteilung eines Stockwerks.

Ein Eigentümer kann beliebig viele Häuser besitzen. Ein Haus hat mindestens einen oder mehrere Eigentümer. Einem Eigentümer können beliebig viele Grundstücke gehören. Ein Grundstück kann immer nur einem Eigentümer gehören. Ein Eigentümer wird eindeutig durch seinen Namen und seine Adresse beschrieben.

b) Relationenmodell

Ausgehend von der ER-Darstellung ist ein Relationenschema in dritter Normalform (3. NF) zu entwerfen. Wie gewohnt, werden dabei Primärschlüssel durch Unterstreichen, Fremdschlüssel durch Überstreichen kenntlich gemacht.

Fortsetzung nächste Seite!

5. SQL

Szenario 1: Winzergenossenschaft

Für die Verwaltung ihrer Kunden in einer Datenbank verwendet eine kleine Genossenschaft der Amateurwinzer „AmMeiHü" die folgenden Relationen:

WEINFREUND (NR, NAME, VORNAME, ALTER, ADRESSE)
BESTELLUNG (NR, W_ID, DATUM, MENGE)
WEIN (W_ID, WEINBERG, JAHRGANG, PROZENT, REBSORTE)

Die Primärschlüssel der Relationen sind unterstrichen. **W_ID** in **BESTELLUNG** ist Fremdschlüssel zu **W_ID** in **WEIN**. **NR** in **BESTELLUNG** ist Fremdschlüssel zu **NR** in **WEINFREUND**. Formulieren Sie die folgenden Datenbankoperationen in SQL und geben Sie - falls möglich - die zugehörigen Ausdrücke in relationaler Algebra an.

a) Geben Sie den Namen und Vornamen aller gespeicherten Weinfreunde aus, die jünger als 30 Jahre sind.

b) Geben Sie eine Liste aller gespeicherten Weine des Weinbergs „Katzkopf" aufsteigend geordnet nach dem Jahrgang aus.

c) Geben Sie die durchschnittlich bestellte Menge Wein aller erfassten Bestellungen an.

d) Fügen Sie einen neuen Wein mit den folgenden Attributen in die Liste der Weine ein:

   Weinberg: „Katzkopf"
   Jahrgang: „2005"
   Prozent: „12"
   Rebsorte: „Silvaner"

   Die nächste freie W_ID ist 67.

e) Der Weinfreund mit dem Vornamen „Otto" und dem Namen „Normalverbraucher" hat noch nie einen Wein bestellt und ist daher aus der Liste der Weinfreunde zu löschen.

Fortsetzung nächste Seite!

Szenario 2: Fußball-Weltmeisterschaft

Zur Vorbereitung auf die Fußball-Weltmeisterschaft 2006 entwickeln die Stadtliga-Hobbyspieler des Dorffußballvereins **„EFC Lummerland"** einen Spielplan. Einige der Spieler sind selbst Hobbyprogrammierer und entscheiden sich für die folgenden Relationen:

```
NATION (Land, Kapitän, Trainer)
STADION (SID, Stadionname, Ort, Kapazität, Eintrittspreis)
MATCH (Stadion_ID, Datum, Land1, Land2, Ergebnis, Tore)
```

Es existieren folgende Fremdschlüsselbeziehungen: Das Attribut `Stadion_ID` der Relation `MATCH` ist ein Fremdschlüssel auf das Attribut `SID` der Relation `STADION`. Die Attribute `Land1` und `Land2` der Relation `NATION` sind Fremdschlüssel auf das Attribut `Land` der Relation `NATION`.

Geben Sie SQL-Anweisungen für folgende Problemstellungen an:

f) Erstellung des beschriebenen Relationenschemas mit Tabellen, Primary Key Constraints und Foreign Key Constraints.

g) Die Siegernation der letzen WM „Brasilien" wird mit ihrem Kapitän „Cafu" und Trainer „Scolari" als erste Nation eingetragen.

h) Das versehentlich vorzeitig eingetragene „Traumfinale" „Brasilien gegen Deutschland", am 09.07.2006 in Berlin, wird wieder gelöscht.

i) Kurz vor dem offiziellen Start der WM 2006 kehrt der ehemalige Trainer der Deutschen Nationalmannschaft, „Rudi Völler", wieder in sein Amt zurück und ersetzt den bisherigen Trainer „Jürgen Klinsmann".

j) Um den Ticketkauf planen zu können, möchten die Hobbyfußballer eine Liste aller Spiele („Match"), die im „Olympiastadion" in „Berlin" stattfinden, ausgeben.

k) Um die Kosten des Kartenkaufs zu kalkulieren, wollen die Hobbyfußballer die Summe der Eintrittspreise aller Spiele im „Olympiastadion" in „Berlin" ausgeben.

l) Um einen Vergleich mit ihren eigenen Fußballspielen zu haben, wollen die Hobbyfußballer nach der WM die durchschnittliche Anzahl an Toren pro Spiel berechnen.

Fortsetzung nächste Seite!

Aufgabe 2:

1. Datenbankanfragen in SQL

Wir betrachten eine FLUG-Datenbank mit den folgenden Tabellen:

- `Flug` (`Fnr`, `Start`, `Ziel`, `Distanz`, `Abflugszeit`, `Ankunftszeit`, `Preis`): Angaben zu einem Flug mit der Nummer `Fnr`.
- `Flugzeug` (`Anr`, `Fluggesellschaft`, `Reichweite`): zu einem Flugzeug mit der Nummer `Anr` wird die `Fluggesellschaft` und die `Reichweite` angegeben.
- `darf_fliegen` (`Pnr`, `Anr`, `Startdatum`): der Angestellte (Pilot) mit der Personalnummer `Pnr` darf das Flugzeug mit der Nummer `Anr` seit dem `Startdatum` fliegen.
- `Angestellter` (`Pnr`, `Pname`, `Lohn`): zu einem Angestellten (z.B. Pilot oder Stewardess) mit der Personalnummer `Pnr` wird der Name `Pname` und der `Lohn` angegeben.

Erstellen Sie in SQL folgende Anfragen:

a) Bestimmen Sie die Namen aller Angestellten, die seit dem '31.01.2002' ein Flugzeug der Fluggesellschaft `Lufthansa` fliegen dürfen.

b) Bestimmen Sie für jede Fluggesellschaft die maximale Reichweite ihrer Flugzeuge.

c) Bestimmen Sie gruppiert nach Start- und Zielflughäfen jeweils die Anzahl aller verbindenden Flüge.

d) Bestimmen Sie alle Fluggesellschaften, deren Angestellte zusammen mehr als 1.000.000 Euro verdienen, und geben Sie für jede solche Fluggesellschaft jeweils die Summe der Löhne an.

2. Datenmodellierung und Datenbankänderungen

Wir betrachten die Datenbank *FLUG* aus Teilaufgabe 1.

a) Geben Sie ein ER-Diagramm für *FLUG* an, in dem möglichst viele Relationen als Relationships modelliert sind.

b) Erklären Sie kurz die Begriffe Primär- und Fremdschlüssel im relationalen Datenmodell. Geben Sie geeignete `Create Table`-Statements mit Primär- und Fremdschlüsselbedingungen zur Erzeugung der Tabellen `Flugzeug`, `darf_fliegen` und `Angestellter` an.

c) Geben Sie `Insert`-Statements an, um in jede der Tabellen `Flugzeug`, `darf_fliegen` und `Angestellter` mindestens zwei Tupel einzufügen, so dass der entstandenen Datenbankinstanz alle Primär- und alle Fremdschlüsselbedingungen erfüllt sind.

d) Ändern Sie den Lohn des Angestellten mit der Personalnummer '17' auf 80.000 Euro.

Fortsetzung nächste Seite!

3. Funktionale Abhängigkeiten und Zerlegungen

Gegeben sei die Attributmenge $U = \{A, B, C, D, E\}$ und folgende Menge F von funktionalen Abhängigkeiten:

$$F = \{A \rightarrow B, AB \rightarrow C, BC \rightarrow DE\}$$

a) Bestimmen Sie die Attributhüllen $\{A\}_F^+$ und $\{B\}_F^+$.

b) Zerlegen Sie das Relationenschema $R = (U, F)$ mittels des Synthesealgorithmus in ein 3NF-Datenbankschema.

c) Zerlegen Sie das Relationenschema $R = (U, F)$ bezüglich der funktionalen Abhängigkeit $BC \rightarrow DE$ (Einzelschritt des Dekompositionsalgorithmus).

Themenschwerpunkt B

(Betriebssysteme)

Aufgabe 1:

Szenario:

A) Ein Programm zur Durchführung der Datensicherung auf einem Magnetbandgerät arbeite nach folgendem Schema: Eine Konfigurationsdatei "backup.conf" enthält eine Liste von Dateikatalogen, die zu sichern sind.

B) Das Datensicherungsprogramm liest jeden dieser Dateikataloge durch und schreibt nacheinander die Verwaltungsinformationen und den Inhalt der darin enthaltenen Dateien in einen Speicherblock der Größe 4 MB.

C) Das Magnetbandgerät arbeitet optimal, wenn es mit möglichst großen Schreibaufträgen versorgt wird. Deshalb wird ein zweiter Prozess erzeugt, der Zugriff auf den gleichen Speicherblock hat und der jeweils 1 MB große Datenbereiche daraus an das Magnetbandgerät überträgt.

1. Dateisystem

a) Ziel der Datensicherung ist natürlich eine möglichst originalgetreue Rekonstruktion im Fall von Datenverlust. Nennen Sie Dateiattribute (Verwaltungsinformationen des Dateisystems zu einer Datei), die Sie bei der Datensicherung mit abspeichern würden. Begründen Sie dies jeweils.

b) Dateisysteme sind meist hierarchisch aufgebaut. Was bedeutet dies und wie geht man damit in dem Datensicherungsprogramm um?

c) Skizzieren Sie in programmiersprachlicher Form den Ablauf von Schritt B in dem skizzierten Szenario in einem Betriebssystem Ihrer Wahl (z.B. UNIX, Linux oder Windows). Sie können hier vereinfachend davon ausgehen, dass der Speicherblock von 4 MB für die komplette Datensicherung groß genug ist.

2. Speicherverwaltung

a) Der in den Schritten B und C erwähnte Speicherblock soll von dem in Schritt B beschriebenen Prozess 1 gefüllt und von dem in Schritt C beschriebenen Prozess 2 ausgelesen werden. Wie erreicht man es, dass beide Prozesse gemeinsam auf diesen Speicherblock zugreifen können - welches Speicherverwaltungskonzept muss dafür von der Hardware und dem Betriebssystem unterstützt werden?

b) Beschreiben Sie für den Fall einer segmentierten Speicherverwaltung wie die Abbildung der logischen Adressen der beiden Prozesse auf diesen gemeinsam erreichbaren Speicherblock funktioniert! Skizzieren Sie hierzu die an dem Abbildungsvorgang beteiligten Datenstrukturen des Betriebssystems.

Fortsetzung nächste Seite!

3. Prozesse und Prozesskoordination

Grundsätzlich ist eine nebenläufige Arbeit von Prozess 1 und Prozess 2 möglich. Gehen Sie jetzt davon aus, dass der Speicherblock von 4 MB für die komplette Datensicherung nicht ausreicht.

a) Wie organisieren Sie den Speicherblock, um die nebenläufige Arbeit möglichst gut zu ermöglichen?

b) An welchen Stellen werden die beiden Prozesse typischerweise in den Zustand "blockiert" übergehen, so dass der jeweils andere Prozess die Zeit für seine Arbeit nutzen kann?

c) Um die in Schritt C geforderte Übertragung von jeweils 1 MB an das Bandgerät bei einem Schreibauftrag sicher zu gewährleisten, müssen die beiden Prozesse koordiniert werden. Welche Art von Prozesskoordinierung ist hier erforderlich?
Skizzieren Sie die Koordinierung für die beiden Prozesse in einer programmiersprachlichen Form.

Fortsetzung nächste Seite!

Aufgabe 2:

1. Interprozesskommunikation von Produzenten und Konsumenten

a) Erläutern Sie bitte knapp im Zusammenhang mit Interprozesskommunikation die Begriffe „sleep&wakeup" und „busy-waiting". Welches sind die Vor- und Nachteile der beiden Strategien?

b) Erläutern Sie knapp das Problem, das entsteht, wenn mehrere Prozesse parallel und unkoordiniert auf eine gemeinsame Ressource - beispielsweise eine globale Variable - lesend und schreibend zugreifen!

c) Korrigieren Sie die Fehler im unten stehenden Java-Programm, damit daraus ein Programm entsteht, welches parallele Prozesse koordiniert: **einen Produzenten** und eine beliebige Anzahl **n** von **Konsumenten.**
Der Produzent legt jeweils eine einzelne Zufallszahl in einer globalen Puffer-Variable ab. Die Konsumenten bedienen sich aus diesem globalen Puffer, wenn dort eine Zahl zur Verfügung steht.

Die Fehler befinden sich jeweils nur in den run()Methoden der beiden Klassen.
Schreiben Sie in Ihrer Lösung die beiden run()-Methoden in korrigierter Fassung.

```
class Konsument implements Runnable
{
  private Produzent myProduzent;

  private BinSemaphore myBSVoll;
  private BinSemaphore myBSLeer;
  private BinSemaphore myBSExklusiverZugriff;

  public Konsument (Produzent myProduzent, BinSemaphore myBSVoll
                   , BinSemaphore myBSLeer
                   , BinSemaphore myBSExklusiverZugriff)
  {
    this.myProduzent = myProduzent;
    this.myBSVoll = myBSVoll;
    this.myBSLeer = myBSLeer;
    this.myBSExklusiverZugriff = myBSExklusiverZugriff;
  }
  public void run()  // Bitte korrigieren
  {
    while(true)
    {
      myBSExklusiverZugriff.p();
      myBSVoll.p();
      System.out.println("konsumiert:"+myProduzent.globalerPuffer);
      myBSLeer.v();
      myBSExklusiverZugriff.v();
    }
  }
}
```

Fortsetzung nächste Seite!

```
class Produzent implements Runnable
{
   public int globalerPuffer;

   private BinSemaphore myBSVoll;
   private BinSemaphore myBSLeer;
   private BinSemaphore myBSExklusiverZugriff;
   java.util.Random zufallszahlenGenerator;

   public static void main (String[] args)
   {
     int n=10;

     BinSemaphore bsVoll = new BinSemaphore(false);
     BinSemaphore bsLeer = new BinSemaphore(true);
     BinSemaphore bsExklusiverZugriff = new BinSemaphore(true);

     Produzent derProduzent = new Produzent(bsVoll
                            , bsLeer, bsExklusiverZugriff);
     new Thread(derProduzent).start();
     for (int i=0; i<n; i++)
     {
       new Thread(new Konsument(derProduzent
                    , bsVoll, bsLeer, bsExklusiverZugriff)).start();

     }
   }

   public Produzent (BinSemaphore myBSVoll
                     , BinSemaphore myBSLeer
                     , BinSemaphore myBSExklusiverZugriff)
   {
      this.myBSVoll = myBSVoll;
      this.myBSLeer = myBSLeer;
      this.myBSExklusiverZugriff = myBSExklusiverZugriff;
      zufallszahlenGenerator = new java.util.Random();
   }

   public void run()    // Bitte korrigieren
   {
      while (true)
      {
         myBSLeer.v();
         myBSExklusiverZugriff.v();
         int neueZahl = zufallszahlenGenerator.nextInt(99);
         System.out.println ("produziert:"+neueZahl);
         globalerPuffer = neueZahl;
         myBSExklusiverZugriff.p();
         myBSVoll.p();
      }
   }
}
```

Fortsetzung nächste Seite!

2. Scheduling-Strategien

Zu einem bestimmten Zeitpunkt T kommen folgende Jobs in der gegebenen Reihenfolge in einer Rechenanlage mit einer CPU zur Bearbeitung an. Der zusätzliche Aufwand beim Umschalten zwischen den Jobs soll vernachlässigt werden.

Job	Bearbeitungszeit
1	10
2	1
3	2
4	1
5	5
6	9

a) Beschreiben Sie unter Verwendung von Gantt-Diagrammen (siehe Beispiel unten) die Bearbeitung der Jobs für die Schedulingverfahren **FCFS** (First-Come-First-Served), **SJF** (Shortest-Job-First) und **RR (q=2)** (Round-Robin mit Zeitquantum=2).

Beispiel für ein Gantt-Diagramm

Zeit	1	2	3	4	5	6	7	8	9
Prozessor#1	Job#1			Job#2	Job#3				leer

b) Geben Sie für alle drei Schedulingverfahren den „Turnaround" (Gesamtzeit im System) eines jeden Jobs an.

c) Geben Sie für alle drei Schedulingverfahren die Anfangswartezeiten, d.h. die Zeit eines jeden Jobs an, die er warten muss, bis er zum ersten Mal die CPU zugeteilt bekommt.

d) Welches Schedulingverfahren besitzt (angewandt auf das obige Beispiel und über alle Jobs gerechnet) die geringste durchschnittliche Gesamtwartezeit, d.h. bei welchem Verfahren ist die Zeit der Inaktivität der Prozesse durchschnittlich am geringsten?

Fortsetzung nächste Seite!

3. Speicherverwaltung

a) Beschreiben Sie knapp die drei Speicherverwaltungsstrategien **First-Fit, Best-Fit** und **Worst-Fit**. Nennen Sie (tabellarisch strukturiert) zu jeder Strategie mindestens einen Vorteil und einen Nachteil.

b) Gegeben sei die Liste mit den folgenden, freien Speicherblöcken (in genau dieser Reihenfolge, wobei 1MB = 1024KB). Diese Blöcke liegen nicht nebeneinander im RAM.
1000KB 1MB 200KB 110KB 4MB

Es werden nun an das Betriebssystem in folgender Reihenfolge Speicheranforderungen gestellt:
1000KB 900KB 200KB 100KB 110KB 2MB

Wie teilt jede der Strategien **First-Fit, Best-Fit** und **Worst-Fit** den Speicher zu? Zur vollständigen Lösung gehört die Liste aller freien Blöcke nach jeder erfüllten Anforderung.

c) Bezogen auf das obige Beispiel beantworten Sie bitte die folgenden Fragen mit knapper Begründung:

- Welches Verfahren schneidet bzgl. Erfüllung der Anfragen am schlechtesten ab?
- Welches Verfahren arbeitet bzgl. der Fragmentierung am effizientesten?

66114

Datenbank- und Betriebssysteme (vertieft)

Herbst 2006

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____ Kennwort: _____ Arbeitsplatz-Nr.: _____	**Herbst 2006**	**66114**

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

- Prüfungsaufgaben -

Fach: **Informatik (vertieft studiert)**

Einzelprüfung: **Datenbank- und Betriebssysteme**

Anzahl der gestellten Themen (Aufgaben): 4 Aufgaben, von denen zwei gemäß den untenstehenden Auswahlregeln zu bearbeiten sind!

Anzahl der Druckseiten dieser Vorlage: 10

Zu den zwei Themenschwerpunkten A (Datenbanksysteme) und B (Betriebssysteme) ist jeweils entweder die Aufgabe l oder 2 zu wählen. Auf der Vorderseite des Kopfbogens sind im Feld „Gewähltes Thema: Nr." die Nummern der beiden gewählten Aufgaben anzugeben (z.B. A1, B2)!

Bitte wenden!

Themenschwerpunkt A
Datenbanksysteme

AUFGABE 1

E/R und SQL

Teilaufgabe 1

Gegeben sei folgende Tabelle einer großen Kinokette:

KINO (Name, Stadt, Vorwahl, Telefon, MName, MGebDat, MGehaltsstufe, MGehalt, MHandy, dBZ, SaalNr, Sitzplätze, Soundsystem)

Der Buchstabe „M" steht hierbei für „Manager", unter „MGebDat" findet sich also das Geburtsdatum des Leiters des jeweiligen Kinos. „dBZ" bezeichnet die durchschnittliche Besucherzahl für ein Kino. Die SaalNr ist für jedes einzelne Kino eindeutig. Es existieren folgende funktionale Abhängigkeiten (auf die Angabe der trivialen Abhängigkeiten wurde verzichtet):

Stadt → *Vorwahl*
MGehaltsstufe → *MGehalt*
Name, Stadt → *Telefon, MName, MGebDat, MGehalt, MHandy, dBZ*
MName → *MGebDat, MGehaltsstufe, MGehalt, MHandy*
Name, Stadt, SaalNr → *Sitzplätze, Soundsystem*

a) Die Relation „Kino" sei in erster Normalform. Was muss für die Attribute der Tabelle gelten?

b) Bestimmen Sie den einzigen Schlüsselkandidaten von „Kino"!

c) Überführen Sie die Relation in die zweite (aber noch nicht dritte) Normalform und erläutern Sie Ihre Schritte kurz!

d) Finden Sie eine Anomalie, die Sie in diesem Beispiel durch Überführung in die zweite Normalform eliminieren konnten, sowie eine andere, die weiterhin vorhanden ist!

e) Überführen Sie nun die Relation, ebenfalls mit kurzer Erläuterung, in die dritte Normalform! Markieren Sie von allen Relationen die Primärschlüssel!

Teilaufgabe 2

Es seien folgende Tabellen eines Fahrradladens gegeben:

Bestand			
ArtNr	**Beschreibung**	**Größe**	**Preis**
F112	Prophete Herrenfahrrad	28	559,00 €
F123	Prophete Damenfahrrad	26	529,00 €
F023	Felix Kinderfahrrad	12	139,00 €
Z321	Prophete Pedale	Einheitsgröße	10,00 €
Z361	Magnum Bügelschloss	XL	99,00 €
Z721	Hella Speichenreflektoren	Einheitsgröße	2,99 €
MR231	Harley-Davidson - Street Rod	1130cm^3	14.500,00 €

Fortsetzung nächste Seite!

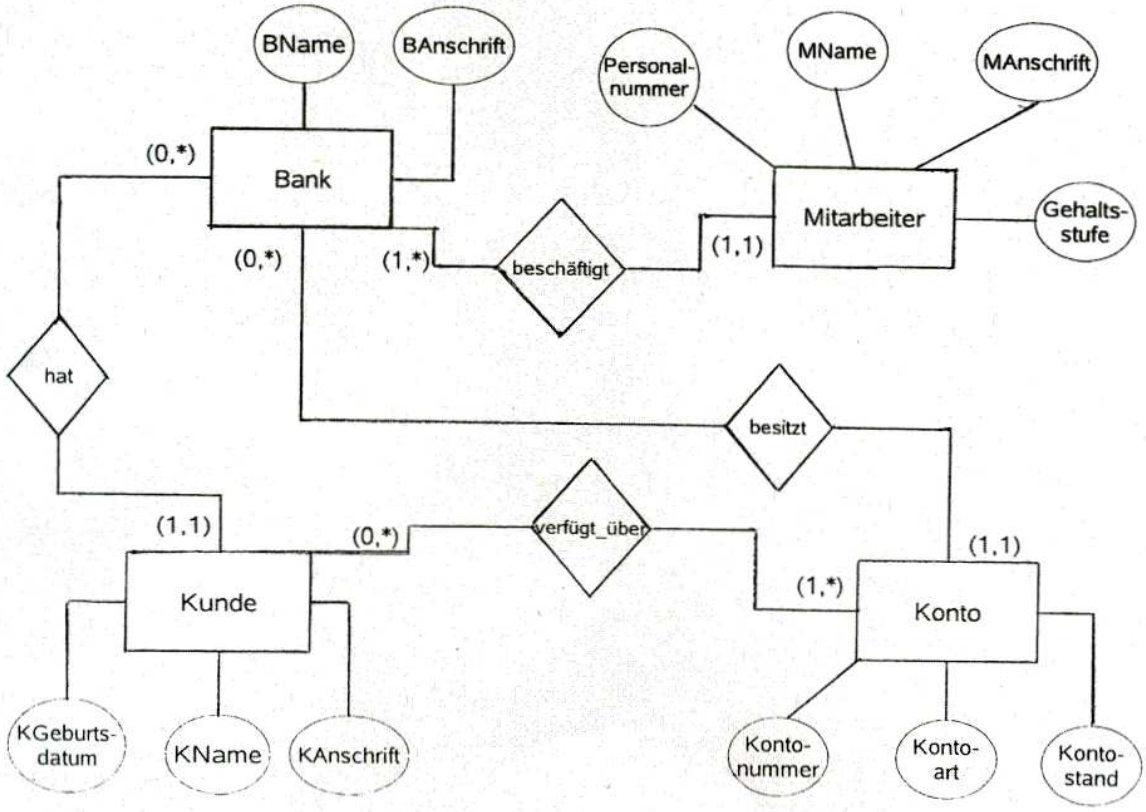
Zuordnung	
Beschreibung	**Typ**
Prophete Herrenfahrrad	Fahrrad
Prophete Damenfahrrad	Fahrrad
Felix Kinderfahrrad	Fahrrad
Prophete Pedale	Zubehör
Magnum Bügelschloss	Zubehör
Hella Speichenreflektoren	Zubehör
Harley-Davidson - Street Rod	Motorrad

a) Geben Sie die SQL-Anweisungen an, um die Tabelle *Bestand* (ohne Inhalt!) zu erstellen!

b) Formulieren Sie folgende Abfragen in SQL:
   I Welche Artikel, die dem Typ *Zubehör* angehören, sind verfügbar?
   II Bestimmen Sie alle Artikel, die Fahrräder sind und mehr als 500 € kosten!
   III Bestimmen Sie alle Artikel, die Zubehör sind und weniger als 10 € kosten!
   IV Sortieren Sie alle Artikel alphabetisch (Beschreibung)!
   V Bestimmen Sie den Durchschnittspreis aller Fahrräder!

Teilaufgabe 3

Gegeben sei folgendes Entity-Relationship-Diagramm einer Bank:



Fortsetzung nächste Seite!

a) Geben Sie die Schlüsselkandidaten an!

b) Überführen Sie das ER-Modell in ein verfeinertes Relationenschema!

c) Markieren Sie alle Fremdschlüssel innerhalb des Relationenschemas!

d) Wie würden Sie das Modell ergänzen, wenn die Bank mehrere Filialen unterhält?
Gehen Sie davon aus, dass Kontodaten etc. zentral bei der Bank gespeichert werden!

AUFGABE 2

1. Normalformenlehre

1.1 Welche Nachteile haben Redundanzen im Datenbestand?
Nennen und erklären Sie mindestens zwei Ihnen bekannte Anomalien!

1.2 Annahme: Relationenschema R sei in 2. Normalform. Welche Eigenschaft muss zusätzlich erfüllt werden, damit R in 3. Normalform ist?

1.3 Für die Zerlegung von Relationsschemata gibt es zwei grundlegende Korrektheitsbeziehungen. Benennen und erklären Sie diese kurz!

2. ER-Modellierung und Relationenmodell

2.1 ER-Modellierung

Erstellen Sie das Modell einer fiktiven Tanzschule in E/R-Notation! Wo möglich bzw. sinnvoll, sollen 3-fache Beziehungen und Generalisierung/Spezialisierung verwendet werden. Attribute von Entitäten und Beziehungen sind anzugeben; Schlüsselattribute werden durch Unterstreichen gekennzeichnet. Die Kardinalitäten von Beziehungen und - falls nötig - Rollennamen sollen ins Diagramm aufgenommen werden. Führen Sie Surrogatschlüssel nur ein, falls es nötig ist!

„Tanzschule“

Tanzschüler werden durch eine eindeutige Nummer, Namen, Geschlecht und Geburtsdatum beschrieben. Jeder Tanzschüler hat genau einen anderen Tanzschüler als „Tanzpartner“.

Tänze zeichnen sich durch eine eindeutige Kurzbezeichnung, einen Namen und einen gewissen Stil aus. Ein Tanzschüler hat bestimmte Tänze gelernt, andere nicht.

Ein **Tanzkurs** hat genau einen Tanzlehrer und mindestens einen Teilnehmer. Es werden in jedem Tanzkurs nur ausgewählte Tänze unterrichtet. Ein Tanzkurs hat einen Namen und ein Beginndatum. Er wird durch eine Nummer identifiziert. Ein Tanzschüler kann auch an mehreren Tanzkursen gleichzeitig teilnehmen oder pausieren. Nicht jeder Tanz wird in einem Tanzkurs unterrichtet.

Jeder **Tanzlehrer** kann nur ganz bestimmte Tänze unterrichten. Ein Tanzlehrer hat einen Vor- und Nachnamen und eine Zulassung. Er wird durch seinen Nachnamen identifiziert. Er kann keinen, einen oder mehrere Tanzkurse unterrichten. Für jeden Tanz im Repertoire muss es einen Tanzlehrer geben.

Dass an einem Kurs nur als Paar teilgenommen werden kann, ist nicht zu modellieren.

Fortsetzung nächste Seite!

2.2 Relationenmodell

Ausgehend von der ER-Darstellung ist ein Relationenschema in dritter Normalform (3. NF) zu entwerfen! **Primärschlüssel werden dabei durch Unterstreichen, Fremdschlüssel durch Nennung der referenzierten Relation in eckigen Klammern hinter dem Attributsnamen kenntlich gemacht, z.B.:**

```
Haus (Straße, OrtId[Ort])
Ort  (OrtId, PLZ, Name)
```

Das Attribut `OrtId` der Relation `Haus` verweist als Fremdschlüssel auf das Attribut `OrtId` der Relation `Ort`.

3. SQL

Bitte beachten Sie:
Primärschlüssel werden dabei durch Unterstreichen, Fremdschlüssel durch Nennung der referenzierten Relation in eckigen Klammern hinter dem Attributsnamen kenntlich gemacht, z.B.:

```
Haus (Straße, OrtId[Ort])
Ort  (OrtId, PLZ, Name)
```

Das Attribut `OrtId` der Relation `Haus` verweist als Fremdschlüssel auf das Attribut `OrtId` der Relation `Ort`.

Szenario „DJ forever!"
Eine Discjockey-Agentur verwendet folgendes einfaches Datenbankschema:

```
Club (CId, Name, AnzahlAreas, ChefDJ[DJ])
DJ (DJId, Name)
Booking (DJId[DJ], CId[Club], Tag, Gage)
Tag={Mo, Di, Mi, Do, Fr, Sa, So}
```

Jeder Club hat einen ChefDJ. DJs arbeiten an unterschiedlichen Tagen evtl. in verschiedenen Clubs. Aber alle arbeiten regelmäßig, d.h. in der Tabelle „Booking" wird nur der Wochentag, kein konkretes Datum hinterlegt.

Primärschlüssel sind unterstrichen. **`DJId`** in **`Booking`** ist Fremdschlüssel zu **`DJId`** in **`DJ`**, **`CId`** in **`Booking`** ist Fremdschlüssel zu **`CId`** in **`Club`** und **`ChefDJ`** in **`Club`** ist Fremdschlüssel zu **`DJId`** in **`DJ`**.

Formulieren Sie folgende Datenbankoperation in SQL:

I. Geben Sie alle Namen der DJs aus!
II. Was verdient „DJ Santos" am Mi?
III. Welcher Club zahlt die höchste Gage?
IV. Wie oft in einer Woche arbeitet „DJ Oetzi"?
V. Wie hoch ist die Durchschnittsgage aller DJs?
VI. Geben Sie Namen aller DJs aus, die „DJ Mike" als Chef haben! Es soll kein Name doppelt vorkommen.
VII. Fügen Sie „DJ Helmut" mit der DJId „0815" in die Datenbank ein!
VIII. „DJ Helmut" hat mittwochs ein Booking im Club „Almenhof" mit einer Gage von „80". Fügen Sie dieses Booking in die Datenbank ein!

Themenschwerpunkt B
Betriebssysteme

AUFGABE 1

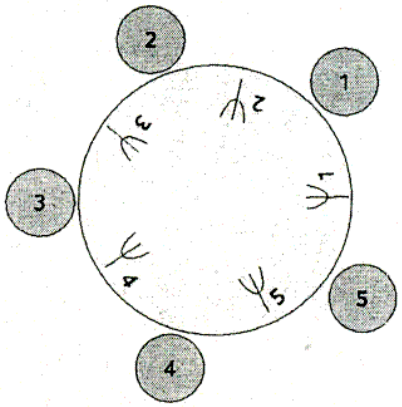
1. Synchronisation

1.1 Definieren Sie den Begriff Verklemmung!

1.2 Welche Bedingungen müssen erfüllt sein, damit eine Verklemmung entsteht?

1.3 Das Philosophenproblem

Fünf Philosophen sitzen an einem runden Tisch, in dessen Mitte ein immer voller Teller mit Nudeln steht. Rechts und links von jedem Philosophen liegt jeweils eine Gabel. Zum Essen benötigt ein Philosoph zwei Gabeln. Die Philosophen sind nur mit zwei Dingen beschäftigt: Denken und Essen. Ist einer hungrig, so greift er zuerst nach der linken und dann nach der rechten Gabel und fängt an zu essen. Ist eine Gabel belegt, so wartet er, bis sie wieder frei ist. Hat der Philosoph sich satt gegessen, so legt er die Gabeln wieder an ihren Platz und fährt fort mit Denken.



a) Betrachten Sie unten stehenden Pseudocode, der einen einzelnen Philosophen simuliert! Erläutern Sie in einem Satz, wie es in dieser Lösung zu einer Verklemmung kommen könnte!

```
N=5;
Procedure phil(i:INTEGER)
  WHILE (TRUE) DO
    think();                //Philosoph denkt nach
    take_fork(i);           //nimm die linke Gabel
    take_fork((i+1) %N)     //nimm die recht Gabel, % ist Modulo-Operator
    eat();                  //essen
    put_fork(i);            //lege linke Gabel zurück
    put_fork((i+1) %N);     //lege rechte Gabel zurück
  END;
END phil;
```

b) Neben einer Verklemmung könnte noch ein weiteres Problem auftauchen. Erläutern Sie kurz welches!

Fortsetzung nächste Seite!

1.4 Semaphore

An einem Flughafen gibt es eine Lagerhalle, an der Transportflugzeuge ihre Ware abliefern und aufnehmen können. Lieferanten bringen jeweils zwei Kisten, während Abholer jeweils nur eine Kiste abholen. Zum Be- und Entladen fliegen die Flugzeuge zuerst über eine Start- und Landebahn ein und fahren dann vor die Lagerhalle. Dort steht ein Kran, der die Flugzeuge be- und entlädt. Danach fliegen die Flugzeuge wieder von der gleichen Start- und Landebahn weg.

Beachten Sie folgende Bedingungen:

- Die Start- und Landebahn kann jeweils nur von einem Flugzeug befahren werden.
- Vor der Lagerhalle ist unbegrenzt Platz, so dass dort mehrere Lieferanten und Abholer stehen können.
- Die Lagerhalle hat eine Kapazität von höchstens 30 Kisten.
- Lieferanten dürfen nur zur Lagerhalle fahren, wenn die Lagerhalle noch für alle Kisten Kapazitäten hat. Ansonsten müssen sie nach dem Landen warten.
- Abholer dürfen nur zur Lagerhalle fahren, wenn noch mindestens eine Kiste in der Lagerhalle ist. Ansonsten müssen sie nach dem Landen warten.
- Es gibt nur einen Kran, der Flugzeuge nacheinander be- und entladen kann.
- Der Kran darf erst zu einem Flugzeug fahren, wenn es zur Lagerhalle gefahren ist.
- Ein Flugzeug darf erst zur Startbahn fahren, nachdem der Kran vom Flugzeug abgefahren ist.
- Die Lagerhalle ist zu Beginn leer.

Die Prozesse sehen folgendermaßen aus:

```
Lieferant
{
 while (TRUE)
 {
  <auf Start- und Landebahn landen>

  <zur Lagerhalle fahren>

  <2 Pakete abliefern>

  <zur Start- und Landebahn, starten>
 }
}
```

```
Abholer
{
 while (TRUE)
 {
 <auf Start- und Landebahn landen>

 <zur Lagerhalle fahren>

 <1 Paket abholen>

 <zur Start- und Landebahn, starten>
 }
}
```

```
Kran
{
 while (TRUE)
 {
 <Anfahrt zu Flugzeug>

 <be-/entlade Flugzeug>

 <Abfahrt von Flugzeug>
 }
}
```

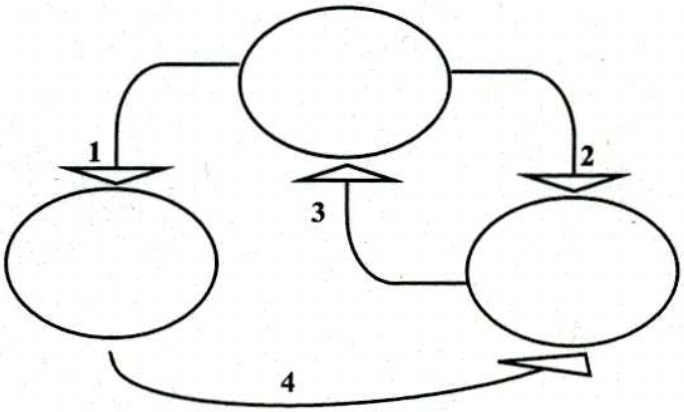
Vervollständigen Sie mit Hilfe von Semaphoren die obigen Prozesse so, dass es zu keiner Verklemmung kommen kann! Verwenden Sie ganzzahlige Semaphore und geben Sie zu den verwendeten Semaphoren die Startwerte an! Sperrphasen sind möglichst kurz zu halten.

2. Prozess- und Prozessorverwaltung

2.1 Erklären Sie kurz den Unterschied zwischen einem Programm und einem Prozess!

2.2 Nennen Sie die Zustände, die ein Prozess aus Betriebssystemsicht einnehmen kann! Übertragen Sie dafür unten abgebildetes Zustandsdiagramm auf Ihr Arbeitsblatt! Tragen Sie die Zustände in die Kreise ein und benennen Sie die Zustandsübergänge 1 bis 4!

Fortsetzung nächste Seite!



2.3 Erläutern Sie kurz die Vorgehensweisen der folgenden Scheduling-Algorithmen:

- Round Robin
- First-come, First-served
- priority scheduling (statisch)

Gehen Sie auch auf die Kriterien Fairness, Echtzeitfähigkeit und unterbrechend ein!

2.4 Fünf Stapelverarbeitungsaufgaben A bis E kommen in einem Rechenzentrum an. Die folgende Tabelle zeigt für jede Stapelverarbeitungsaufgabe die Ankunftszeit, die Laufzeit sowie die Priorität. Die Priorität wächst mit steigenden Zahlen.

	Ankunftszeit	Laufzeit	Priorität
A	0ms	10ms	3
B	1ms	6ms	5
C	2ms	2ms	2
D	3ms	4ms	1
E	4ms	8ms	4

a) Bestimmen Sie für jeden der in der Aufgabe 2.3 genannten Scheduling-Algorithmen die durchschnittliche Prozessdurchlaufzeit (Verweilzeit)! Vernachlässigen Sie dabei den Overhead des Prozesswechsels! Die Zeitscheibe für Round Robin betrage 2ms.

b) Welchen Algorithmus würden Sie auswählen? Begründen Sie kurz Ihre Entscheidung!

3. Speicherverwaltung

3.1 Eine Arbeitsspeicherverwaltung wende dynamische Segmentierung an und habe in der Freispeicherliste noch drei Segmente von 700, 450 und 300 Speichereinheiten zur Verfügung (Reihenfolge wie Aufzählung). Es treffen nun nacheinander Anforderungen von 100, 300, 400, 300 und 250 Speichereinheiten an. Wie würden mit dem first fit- und dem best fit-Verfahren die Speichereinheiten auf die Segmente verteilt werden? Stellen Sie Ihr Ergebnis in einer zu unten stehender äquivalenten Tabelle dar, vergleichen Sie die Ergebnisse und erläutern Sie die Unterschiede! Geben Sie hierbei jeweils die noch freien Speichergrößen an!

Anforderung	**First Fit**			**Best Fit**		
Segment	Segment 1	Segment 2	Segment 3	Segment 1	Segment 2	Segment 3
Initial	700	450	300	700	450	300
100						
300						
400						
300						
250						

Fortsetzung nächste Seite!

3.2 Als weitere Möglichkeit gibt es das Verfahren worst fit. Erläutern Sie kurz die Verfahrensweise, Vor- und Nachteile!

AUFGABE 2:

1. Speicherverwaltung

a) Skizzieren Sie (grafisch) die Abbildung einer logischen Adresse in eine physikalische Adresse in einem System mit Segmentierung!

b) Was muss das Betriebssystem tun, wenn aufgrund von Hauptspeichermangel ein Segment ausgelagert werden soll? Beschreiben Sie den Ablauf und welche Änderungen in welchen der in Teilaufgabe a beschriebenen Datenstrukturen vorgenommen werden müssen!

c) Was passiert, wenn der Prozess nach dem Auslagern erneut auf die Daten des Segments zugreift? Beantworten Sie in Ihrer Beschreibung vor allem die folgenden Fragen:
- Welche Einheit des Systems erkennt, dass das Segment nicht vorhanden ist?
- Woran wird das erkannt?
- Welche Aktivitäten finden daraufhin im Betriebssystem oder in der Anwendung statt?
- Welche Datenstrukturen werden in diesem Zusammenhang wie modifiziert?
- Welche Prozesszustände nimmt der betroffene Prozess in welcher Phase ein?

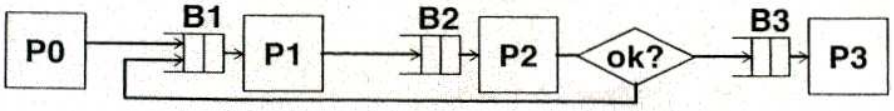
d) Ein Segment soll von zwei Prozessen gemeinsam genutzt werden. Was muss das Betriebssystem hierfür tun?

e) Nennen Sie drei wesentliche Unterschiede zwischen Segmentierung und Seitenadressierung!

2. Prozesse und Prozesskoordination, Teil 1

Gegeben sei folgendes Szenario: Ein Prozess PO erzeugt Daten, die er zur Vorverarbeitung einem Prozess P1 übergibt. Das Ergebnis der Vorverarbeitung übergibt P1 dann an Prozess P2, der eine Weiterverarbeitung vornimmt. Das Ergebnis der Weiterverarbeitung wird entweder an Prozess P3 zur Ausgabe übergeben oder muss - falls es bestimmten Kriterien noch nicht genügt - nochmals an Prozess P l zur erneuten Vorverarbeitung übergeben werden.

Die Prozesse P1 bis P3 verfügen jeweils über einen FIFO-Puffer (B1 - B3) mit n Speicherplätzen, über den sie ihre Eingaben entgegen nehmen. Pl hat nur einen solchen Puffer (B1), über den sowohl PO als auch P2 die Daten bei ihm anliefern. Die Prozesse geben ihre Daten grundsätzlich in Form von Datenpaketen weiter, ein Puffer-Speicherplatz kann ein solches Datenpaket aufnehmen.



a) Beschreiben Sie, an welchen Stellen Koordinierungsbedarf in diesem Szenario besteht und mit welchen Koordinierungsmechanismen man die Zugriffe jeweils koordinieren kann!

Fortsetzung nächste Seite!

b) Skizzieren Sie nun in einer programmiersprachlichen Form die Operationen `put` und `get`:
`void put(message m)`
nimmt eine Nachricht entgegen und kopiert sie in den Puffer. Falls gerade kein Platz in dem Puffer frei ist, blockiert die Operation so lange, bis wieder ein Speicherplatz für die Nachricht frei wird.
`message get()`
entnimmt die nächste Nachricht aus dem Puffer. Falls der Puffer leer ist, blockiert die Operation so lange, bis wieder eine Nachricht vorliegt.

c) Ist Ihre Lösung verklemmungsgefährdet? Begründen Sie Ihre Antwort, indem Sie beschreiben, wann eine Verklemmung auftreten kann bzw. warum Verklemmungen nicht auftreten werden!

3. Prozesse und Prozesskoordination, Teil 2

Gehen Sie von folgendem, gegenüber Aufgabe 2 veränderten Szenario aus:

Die Puffer können jeweils nur eine Nachricht aufnehmen. Die Prozesse verarbeiten die Nachricht direkt aus dem Eingangspuffer und schreiben das Ergebnis direkt in den Puffer des empfangenden Prozesses. Beide Puffer werden erst dann wieder freigegeben, wenn die Verarbeitung beendet ist. Der Prozess P2 stellt dabei zu Beginn der Verarbeitung fest, ob eine neue Vorverarbeitung durch P1 oder die Ausgabe durch P3 folgen wird (noch bevor die ersten Ergebnisdaten produziert werden).

Bevor ein Prozess seinen Eingabepuffer belegt, überprüft er, ob überhaupt Daten darin vorhanden sind. Liegen keine Daten vor, wartet er 10 Sekunden bevor er es erneut versucht.

a) Skizzieren Sie den Ablauf und die Koordinierung in den Verarbeitungfunktionen der Prozesse P1 (`Funktion work1 ( )`) und P2 (`Funktion work2 ( )`).

b) Welche Bedingungen für Verklemmungen treffen in dem geschilderten Szenario zu? Beschreiben Sie jeweils den Zusammenhang zwischen "abstrakter Bedingung" und konkreter Situation!

c) Welche Möglichkeiten zur Verklemmungsvermeidung gibt es in dem geschilderten Szenario ganz konkret? Wie müssten Sie Ihre Funktionen `work1` und `work2` dafür jeweils ändern?

66114

Datenbank- und Betriebssysteme (vertieft)

Frühjahr 2007

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer

Kennzahl: _____

Kennwort: _____

Arbeitsplatz-Nr.: _____

Frühjahr 2007

66114

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

— Prüfungsaufgaben —

Fach: **Informatik (vertieft studiert)**

Einzelprüfung: **Datenbank- und Betriebssysteme**

Anzahl der gestellten Themen (Aufgaben): **4** Aufgaben, von denen 2 gemäß untenstehender Auswahlregeln zu bearbeiten

Anzahl der Druckseiten dieser Vorlage: **11**

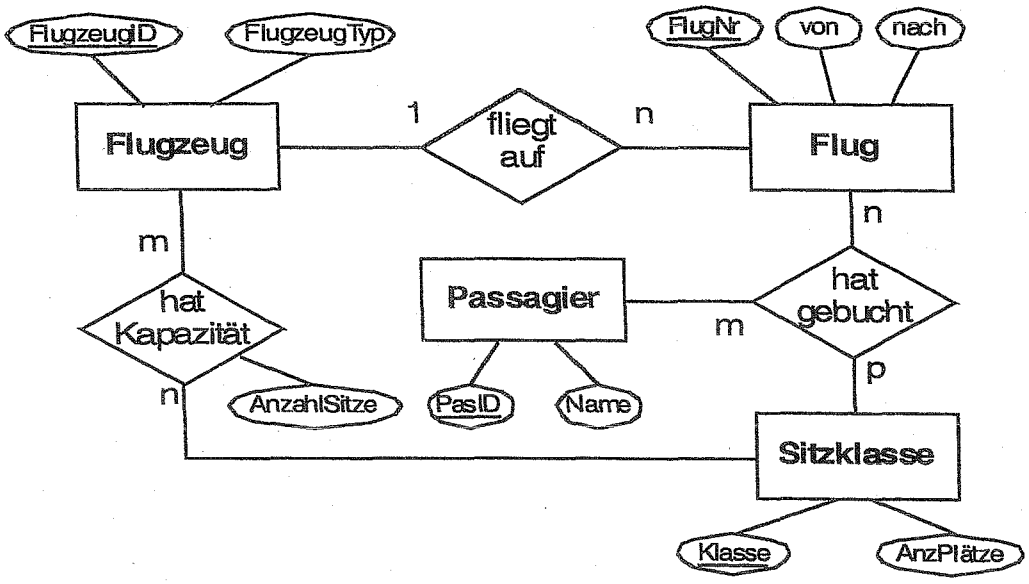
Zu den zwei Themenschwerpunkten A (Datenbanksysteme) und B (Betriebssysteme) ist jeweils entweder die Aufgabe 1 oder 2 zu wählen. Auf der Vorderseite des Kopfbogens sind im Feld „Gewähltes Thema: Nr." die Nummern der beiden gewählten Aufgaben anzugeben (z. B. A1, B2)!

Bitte wenden!

7. Datenbankentwurf

Datenbanksysteme dienen der Speicherung und effizienten Verarbeitung von Daten. Das Datenbankschema lässt sich mit Hilfe von Entity-Relationship-Diagrammen entwerfen.

Das folgende Entity-Relationship-Diagramm (ERD) beschreibt einen Ausschnitt aus der Datenbank eines Flugbuchungssystems. Bestimmen Sie das zu diesem ERD gehörige Relationenschema! Verwenden Sie als Namen für Fremschlüsselattribute die Namen der referenzierten Primärschlüsselattribute. Primärschlüsselattribute sind zu unterstreichen, Fremdschlüsselattribute zu überstreichen!



Aufgabe 2:

1. Datenbanksysteme und Modelle

Die Entwicklung der ersten Datenbanksysteme begann in den frühen 60er Jahren. Seitdem haben sie die Datenverwaltung auf der Grundlage einfacher Betriebssystemdateien auf vielen Gebieten abgelöst.

a) Definieren Sie den Begriff Datenbanksystem und grenzen Sie die Funktionalitäten eines Datenbanksystems von denen eines Dateisystems ab!

b) Nennen Sie die Phasen des Datenbankentwurfsprozesses und charakterisieren Sie diese kurz!

c) Entity-Relationship-Modell: Erläutern Sie die nachfolgenden Konzepte eines ER-Modells!

Entität,
Relationship/Beziehungstyp,
Attribut,
Kardinalität

Fortsetzung nächste Seite!

d) Relationales Modell: Erläutern Sie informell die folgenden Begriffe! Gehen Sie insbesondere auf die Bedeutung der verschiedenen Begriffe für die Anwendungsmodellierung ein!

Schlüsselkandidat,
Primärschlüssel,
Fremdschlüssel

e) Sie haben sowohl die Grundoperationen der relationalen Algebra als auch die daraus abgeleiteten Operationen kennen gelernt. Definieren Sie die Operation „Division"!

f) Sie haben die Normalformenlehre kennen gelernt. Welchen Sinn und Zweck hat die Normalformenlehre?

g) Transaktionen sind ein wichtiges Konzept in Datenbanksystemen. Was versteht man unter dem ACID-Prinzip? Welche Konsequenzen im Verhalten des Datenbanksystems hat das ACID-Prinzip für den Anwender und Entwickler?

2. ER-Modellierung und Relationenmodell

a) ER-Modellierung

Es soll ein Modell der beschriebenen universitären Welt in ER-Notation erstellt werden! Dabei können alle Ihnen bekannten Modellierungskonstrukte verwendet werden. Wo möglich bzw. sinnvoll, sollen 3-fache Beziehungen und Generalisierung/Spezialisierung verwendet werden. Attribute von Entitäten und Beziehungen sind anzugeben; Schlüsselattribute werden durch Unterstreichen gekennzeichnet. Die Kardinalitäten von Beziehungen und (falls nötig) Rollennamen sollen ins Diagramm aufgenommen werden!

Das folgende Beispiel beschreibt einen Ausschnitt der universitären Welt. Personen und Einrichtungen der Universität werden durch folgende Typen von Objekten beschrieben:

- Für **Personen** werden folgende Informationen aufgezeichnet: Ausweisnummer, Vorname, Nachname, Adresse (Land, PLZ, Ort, Straße, Hausnummer), Geburtstag und eine Liste von Telefonnummern.

- **Mitarbeiter** sind eine Gruppe von Personen, die durch zusätzliche Eigenschaften charakterisiert sind: Personalnummer, Fachrichtung, Zimmernummer, Gebäude, Anstellungsdatum und Gehalt.

- Einige Mitarbeiter sind **Professoren,** für die noch weitere Charakteristika hinzukommen: Lehrstuhl und Gehaltsstufe (C3, C4 etc.).

- Darüber hinaus gibt es eine weitere Personengruppe, die **Studenten.** Zusätzlich zu den Eigenschaften der Personen besitzen sie folgende Attribute:
  Matrikelnummer, Studienfach und Immatrikulationsdatum.

- Jeder **Lehrstuhl** wird durch seine Fachrichtung, das Institut, zu dem er gehört, und die Anzahl seiner Planstellen beschrieben.

Fortsetzung nächste Seite!

Darüber hinaus enthält die abzubildende universitäre Miniwelt noch Ausprägungen folgender Typen:

- Die Universitätsbibliothek enthält eine Reihe von **Büchern**, die durch ISBN-Nummer, Titel, Jahr der Veröffentlichung, Auflage, mehrere Suchbegriffe und eine Autorenliste beschrieben werden. Die Anzahl ihrer Seiten und ihr Preis sollen ebenfalls aufgenommen werden.

- **Buchverlage** sollen durch Namen und Ort repräsentiert werden.

- **Vorlesungen** haben eine Bezeichnung, eine vorgegebene Zahl von Semesterwochenstunden und werden im Sommer- oder Wintersemester abgehalten. Für den Besuch einer Vorlesung kann der vorherige Besuch einer Reihe anderer Vorlesungen, in denen Grundlagen vermittelt werden, gefordert sein.

Zwischen Instanzen der beschriebenen Objekttypen bestehen eine ganze Reihe von Beziehungen. In jedem der folgenden Punkte können eine oder mehrere Beziehungen beschrieben sein:

- Mitarbeiter und Studenten können Bücher bei der Bibliothek ausleihen. Die Ausleihfrist beträgt jeweils 14 Tage. Bei einer Überschreitung dieser Frist werden Gebühren (5 EUR) erhoben. Überschreiten die kumulierten Gebühren, die eine Person zu entrichten hat, eine vorgegebene Grenze (20 EUR), so kann die betreffende Person keine weiteren Bücher ausleihen.

- Professoren empfehlen den Studenten Bücher für ihre Vorlesungen.

- Einige Professoren sind Lehrstuhlinhaber; für jeden Lehrstuhl gibt es aber höchstens einen Lehrstuhlinhaber.

- Professoren halten ihre Vorlesungen entweder im Winter- oder im Sommersemester. Jeder Student kann mehrere Vorlesungen in einem Semester besuchen. Jede Vorlesung kann eine beliebige Zahl anderer Vorlesungen voraussetzen (Vorkenntnisse).

- Bücher werden von Verlagen herausgegeben.

- Studenten werden von Professoren in unterschiedlichen Fächern geprüft. Die Leistungen in jeder Prüfung werden mit einer Note bewertet.

b) Relationenmodell

Ausgehend von der ER-Darstellung ist ein Relationenschema in dritter Normalform (3. NF) zu entwerfen. Wie gewohnt, werden dabei Primärschlüssel durch Unterstreichen, Fremdschlüssel durch Überstreichen kenntlich gemacht.

Fortsetzung nächste Seite!

3. SQL

a) Szenario 1: Bücher und Verlag

Für die Verwaltung ihrer Bestände in einer Datenbank verwendet eine kleine Buchhandlung die folgenden Relationen:

BUCH (ISBN, Autor_Nachname, Titel, Verlags_Id, Lagerbestand)
VERLAG (Verlags_Id, Verlagsname, Verlagsadresse)

Die Primärschlüssel der Relationen sind unterstrichen, Verlags_Id in BUCH ist Fremdschlüssel zu Verlags_Id in VERLAG. Formulieren Sie die folgenden Datenbankoperationen in SQL und geben Sie - falls möglich - die zugehörigen Ausdrücke in relationaler Algebra an!

1. Welche ISBN hat das Buch "Kaufmännische Datenbanken" von Wedekind?

2. Welche Adresse hat der Verlag, in dem das Buch mit der ISBN 2-411-16541-3 erschienen ist?

3. Die Buchhandlung erwirbt fünf Exemplare des bei Adisson-Wesley erschienenen Buchs "Objektorientierte Datenbanken" von Andreas Heuer, ISBN 3-89319-315-4. Der Verlag ist bereits in der Relation VERLAG vorhanden und hat die ID "AW1". Formulieren Sie die Anweisungen für den Eintrag der Daten!

4. Löschen Sie den "Pleite-Verlag", "Erfolglosenstr. 13, 90439 Nürnberg"! Gehen Sie davon aus, dass von dem Verlag nie ein Buch gelagert wurde!

b) Szenario 2: Personaldatenbank

Für die Personalabteilung eines Unternehmens soll eine Personaldatenbank erstellt werden. Das Unternehmen ist in mehrere Abteilungen untergliedert. Jeder Angestellte gehört zu genau einer Abteilung; jede Abteilung hat einen Abteilungsleiter. Die folgenden Relationen stellen einen kleinen Ausschnitt aus der erarbeiteten Datenbankanwendung dar:

ABTEILUNG (ANr, Bezeichnung)
PERSONAL (Pnr, Name, Geburtsdatum, Gehalt, ANr)
LEITER (ANr, Pnr)

Die Fremdschlüsselbeziehungen sind sprechend und gemäß der Beschreibung gewählt.

Fortsetzung nächste Seite!

Geben Sie SQL-Anweisungen für folgende Problemstellungen an:

1. Erstellung des beschriebenen Relationenschemas mit Tabellen, Primary Key Constraints und Foreign Key Constraints.

2. Frau Schneider (geb. 12.5.61) wird zur Leiterin der neu gegründeten Abteilung "Kundensupport". Sie erhält die Personalnummer 4497.

3. Herr Müller (PNr 2371) scheidet aus dem Unternehmen aus.

4. Wie viele Personen arbeiten in der Abteilung, die von Frau Ziegler (geb. 30.12.58) geleitet wird?

5. Wie viele Mitarbeiter hat die Abteilung, in der Herr Schuster (geb. 18.1.83) arbeitet?

6. Wie hoch ist das Durchschnittsgehalt in den einzelnen Abteilungen?

7. Wessen Gehalt liegt über dem Durchschnittsgehalt der Abteilung, in der er beschäftigt ist?

Themenschwerpunkt B

(Betriebssysteme)

Aufgabe 1:

1. Speicherverwaltung

a) Beschreiben Sie die Seitenersetzungsstrategie LRU *(Least Recently Used)*. Gehen Sie dabei ein auf Funktionsweise und Motivation der Strategie sowie auf Probleme bei der Realisierung!

b) Welche Strategien zur näherungsweisen Realisierung von LRU gibt es? Erläutern Sie an einem Beispiel detailliert die Funktionsweise! Warum und wie werden durch die von Ihnen beschriebene Strategie die Probleme von LRU vermieden?

c) Was versteht man unter Seitenflattern *(Thrashing)*? Was sind die Ursachen? Welche Möglichkeiten gibt es, Seitenflattern zu vermeiden?

2. Prozesse

a) Skizzieren Sie die Zustände, die ein Prozess unter einem Mehrbenutzer-Betriebssystem wie z. B. UNIX typischerweise einnehmen kann!

b) Ein Prozess habe folgenden Ablauf (grob skizziert):

1. Start, Variablen initialisieren
2. Konfigurationsdatei einlesen
3. kurze Vorberechnungen durchführen (Dauer: 2 ms)
4. Berechnungsdaten aus Datei einlesen
5. Berechnung durchführen (Dauer: 800 ms)
6. Ergebnis in Datei ausgeben
7. Ende

- Nehmen Sie zunächst an, das Betriebssystem arbeitet mit einer "First Come First Served"-Scheduling-Strategie. Welche Zustände nimmt der Prozess während dieses Ablaufs ein, wodurch werden welche Zustandsübergänge ausgelöst?

- Nehmen Sie nun an, das Betriebssystem arbeitet mit der Strategie "Round Robin" und es gibt weitere, laufbereite Prozesse im System. Die Zeitscheibenlänge sei 300 ms. Wie sieht der Ablauf des Prozesses (Zustände, Zustandsübergänge, Ursachen) aus?

c) In vielen Betriebssystemen wird zwischen einem Benutzermodus und einem Systemmodus unterschieden. Beschreiben Sie die Eigenschaften und Unterschiede der beiden Modi. Wie erfolgt die Umschaltung zwischen den Modi? Was sind die Ziele dieser Unterscheidung? In welchen Rechensystemen ist die Unterscheidung sinnvoll oder sogar wichtig und in welchen Systemen ist sie verzichtbar?

Fortsetzung nächste Seite!

3. Verklemmungen

a) Welche Bedingungen müssen gegeben sein, damit eine Verklemmung auftreten kann?

b) Welche drei grundsätzlichen Verfahren gibt es, um mit der Verklemmungsproblematik umzugehen? Beschreiben Sie jedes Verfahren! Wie ist jeweils die grundsätzliche Vorgehensweise? Geben Sie ein Beispiel für einen Algorithmus an, der bei einem dieser Verfahren zum Einsatz kommt und beschreiben Sie den Algorithmus!

4. Dateisysteme

a) Was sind die grundlegenden Aufgaben eines Dateisystems?

b) Welche Informationen und Attribute werden zu einer Datei von einem Betriebssystem typischerweise verwaltet?

c) Skizzieren Sie in einer programmiersprachlichen Form den Ablauf eines Programms wie *ls* unter UNIX oder *dir* unter Windows!

Aufgabe 2:

1. Prozesse und Threads

a) Erläutern Sie die Begriffe *Programm*, *Prozess* und *Thread*!
- Wie entstehen Programme, Prozesse bzw. Threads?
- Wie werden sie vom Betriebssystem verwaltet?
- Welche Eigenschaften, Attribute etc. werden für die Verwaltung benutzt?

b) Was haben diese Konzepte miteinander zu tun, wo liegen die Unterschiede?

c) Beschreiben Sie die unterschiedlichen Möglichkeiten zur Realisierung von Threads! Was sind jeweils die Vor- und Nachteile?

2. Scheduling

a) Beschreiben Sie die grundlegenden Scheduling-Strategien und ihre Eigenschaften!

b) Wie könnte eine realistische Scheduling-Strategie für ein Mehrbenutzer- und Mehrprogramm-Betriebssystem aussehen?

- Welche Ziele sollte solch eine Scheduling-Strategie haben?

- Über welche Parameter muss man bei der Strategie entscheiden und was sind die Auswirkungen der Parameter?

- Welche Probleme können unter sehr starker Systemlast entstehen und was kann man dagegen tun?

Fortsetzung nächste Seite!

3. Speicherverwaltung

a) Skizzieren und beschreiben Sie den Ablauf einer Adressumrechung von einer virtuellen Adresse in eine reale Hauptspeicheradresse in einem seitenadressierten System!

b) Wie ist der Ablauf, wenn sich die adressierte Seite nicht im Hauptspeicher befindet?

c) Wie groß ist heute typischerweise eine Seite?

d) Welche Probleme würden sich ergeben, wenn man die Adressabbildung in ihrer einfachsten Form auf einem heute gebräuchlichen PC realisieren würde? Welche Konzepte gibt es, diese Probleme zu mildern?

4. Dateisystem

a) Was sind die grundlegenden Aufgaben eines Dateisystems?

b) Wie ist ein Dateisystem unter UNIX oder Windows prinzipiell strukturiert? Welche Verwaltungseinheiten gibt es?

c) Welche Informationen und Attribute werden zu einer Datei von einem Betriebssystem typischerweise verwaltet?

d) Skizzieren Sie in einer programmiersprachlichen Form den Ablauf eines Programms wie *ls* unter UNIX oder *dir* unter Windows!

5. Synchronisation

a) Welche Verfahren/Konzepte zur Absicherung eines kritischen Abschnitts gibt es?

b) Charakterisieren Sie diese Verfahren jeweils (Vor- und Nachteile, wo werden sie eingesetzt?)!

c) Wie funktionieren die P- und V-Operationen von Semaphoren? Skizzieren Sie den Ablauf der Operationen in einer programmiersprachlichen Form! Was ist bei der Implementierung der P- Operation zu beachten?

d) Skizzieren Sie in einer programmiersprachlichen Form die Abläufe in einem Erzeuger-Verbraucher-System, das die vollen und leeren Pufferplätze über zählende Semaphore koordiniert!

66114

Datenbank- und Betriebssysteme (vertieft)

Herbst 2007

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____	**Herbst 2007**	**66114**
Kennwort: _____		
Arbeitsplatz-Nr.: _____		

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

— Prüfungsaufgaben —

Fach: **Informatik (vertieft studiert)**

Einzelprüfung: **Datenbank- und Betriebssysteme**

Anzahl der gestellten Themen (Aufgaben): **4** Aufgaben, von denen zwei gemäß untenstehender Auswahlregeln zu bearbeiten sind!

Anzahl der Druckseiten dieser Vorlage: **10**

Zu den zwei Themenschwerpunkten A (Datenbanksysteme) und B (Betriebssysteme) ist jeweils entweder die Aufgabe 1 oder 2 zu wählen. Auf der Vorderseite des Kopfbogens sind im Feld „Gewähltes Thema: Nr." die Nummern der beiden gewählten Aufgaben anzugeben (z. B. A1, B2)!

Bitte wenden!

Themenschwerpunkt A
(Datenbanksysteme)

Aufgabe 1:

1. Allgemeine Fragen

Geben Sie für jede der folgenden Aussagen an, ob diese richtig oder falsch ist! Begründen Sie Ihre Aussage in jedem Fall!

a) Jede Relation muss mindestens ein Attribut besitzen.
b) Ein Primärschlüssel muss immer aus mindestens zwei Attributen zusammengesetzt sein.
c) Alle Relationen eines Datenbankschemas müssen in der ersten Normalform vorliegen.
d) Datenbanksysteme können gleichzeitig von maximal 32 Anwendern genutzt werden.
e) Ein Fremdschlüssel verweist auf genau ein Tupel einer anderen Relation.
f) Eine Datenbanktransaktion entspricht genau einem SQL-Statement.
g) Der Name eines Fremdschlüsselattributes darf nicht gleich dem Namen des referenzierten Primärschlüsselattributes sein.
h) Die logische Datenstruktur einer relationalen Datenbank beschreibt, wo welche Daten auf der Festplatte gespeichert sind.
i) Alle Attribute einer Relation müssen unterschiedliche Namen haben.
j) Kein Attribut einer Relation darf den gleichen Namen wie die Relation tragen.

Definieren Sie folgende Begriffe bzw. Abkürzungen im Kontext von Datenbanksystemen!

k) DML
l) Schlüsselkandidat
m) Index
n) Stored Procedure

2. Relationenmodell und relationale Anfragesprachen

a) Relationen werden oft mit Tabellen verglichen. Dieser Vergleich ist jedoch nicht ganz korrekt. Stellen Sie kurz dar, welche Unterschiede zwischen den Konzepten „Tabelle“ und „Relation“ existieren!
b) Stellen Sie kurz den Unterschied zwischen „Relationenschema einer Relation“ und der „Extension einer Relation“ dar! Illustrieren Sie Ihre Erklärung an einem kleinen Beispiel!
c) Nennen und beschreiben Sie in jeweils 1-2 Sätzen drei Grundoperationen der relationalen Algebra!
d) Benennen und beschreiben Sie kurz die prinzipiellen Möglichkeiten der Anfrage-Optimierung!

3. Normalformenlehre

a) Erläutern Sie kurz Sinn und Zweck der Normalformenlehre!
b) Charakterisieren und erläutern Sie an einem kurzen Beispiel die erste, zweite und dritte Normalform!
c) Stellen Sie kurz dar, welche Korrektheitskriterien man bei der Zerlegung eines Relationenschemas beachten muss!

Fortsetzung nächste Seite!

4. Transaktionen und Transaktionsverarbeitung

a) Nennen und definieren Sie die vier wesentlichen Merkmale einer Datenbanktransaktion!
b) Was versteht man unter dem sog. Lost-Update Problem? Skizzieren Sie den Sachverhalt an einem kurzen Beispiel!
c) Um die quasi parallele Ausführung von Transaktionen auf einem Datenbanksystem fehlerfrei zu ermöglichen, verwendet man sog. Sperren. Stellen Sie kurz dar, welche Arten von Sperren es gibt und wie sie zueinander in Beziehung stehen!
d) Welches Problem kann sich bei der Verwendung von Sperren ergeben? Wie kann man dieses Problem in der Praxis umgehen?

5. ER-Modellierung und Relationenmodell

a) ER-Modellierung
Erstellen Sie das Modell zur Organisation einer fiktiven Bundestagswahl in E/R-Notation! Wo möglich bzw. sinnvoll, sollen 3-fache Beziehungen und Generalisierung/Spezialisierung verwendet werden. Attribute von Entitäten und Beziehungen sind anzugeben; Schlüsselattribute werden durch Unterstreichen gekennzeichnet. Die Kardinalitäten von Beziehungen und – falls nötig – Rollennamen sollen ins Diagramm aufgenommen werden. Führen Sie Surrogatschlüssel nur ein, falls es nötig ist!

„Bundestagswahl"

Bei der Bundestagswahl werden **Wahlkreise** durch eine eindeutige Wahlkreisnummer identifiziert und durch einen Namen genauer beschrieben.

In einem Wahlkreis tritt mindestens ein **Kandidat** zur Wahl an, jedoch kann ein Kandidat maximal in einem Wahlkreis antreten. Jeder Kandidat hat einen Namen, der ihn eindeutig identifiziert.

Jeder Kandidat ist Mitglied maximal einer **Partei**, die jedoch mindestens 10 Kandidaten stellen muss. Eine Partei kann vor der Wahl eine Koalitionsaussage zu beliebig vielen anderen Parteien machen. Jede Partei wird durch ihren Namen eindeutig identifiziert und durch ein Regierungsprogramm charakterisiert.

Jede Partei stellt für die Wahl genau eine **Liste** mit Kandidaten auf. Jede Liste muss zu genau einer Partei gehören. Auf einer Liste sind mindestens fünf Kandidaten positioniert. Jeder Kandidat darf maximal auf einer Liste positioniert sein. Eine Liste wird durch eine Nummer gekennzeichnet und verfügt über eine bestimmte Anzahl von Plätzen.

b) Relationenmodell
Ausgehend von der ER-Darstellung ist ein Relationenschema in dritter Normalform (3. NF) zu entwerfen. Wie gewohnt, werden dabei Primärschlüssel durch Unterstreichen, Fremdschlüssel durch Überstreichen kenntlich gemacht.

Fortsetzung nächste Seite!

6. SQL

Szenario 1: Tischreservierung

Für die Verwaltung der Reservierungen in einer Datenbank verwendet ein kleines Gourmetrestaurant die folgenden Relationen:

GAST (GNR, Name, Vorname, Adresse)
RESERVIERUNG (GNR, TNR, Datum, Anzahl)
TISCH (TNR, Position, Plätze, Kinderplätze)

Die Primärschlüssel der Relationen sind unterstrichen, **GNR** in **RESERVIERUNG** ist Fremdschlüssel zu **GNR** in **GAST**, **TNR** in **RESERVIERUNG** ist Fremdschlüssel zu **TNR** in **TISCH**. Formulieren Sie die folgenden Datenbankoperationen in SQL!

a) Geben Sie den Namen und Vornamen aller Gäste aus, die einen Tisch für den 31.12.2005 reserviert haben!
b) Geben Sie eine Liste (**TNR** und **Position**) aller Tische aus, die für mindestens drei Personen am 31.12.2005 reserviert sind!
c) Fügen Sie eine neue Reservierung mit folgenden Daten ein:
   - Gastnummer (**GNR**): 17
   - Tischnummer (**TNR**): 3
   - Datum: 31.12.2005
   - Anzahl: 4 Personen
d) Ändern Sie die Anzahl der reservierten Plätze in folgender Reservierung von 4 auf 3:
   - Gastnummer (**GNR**): 13
   - Tischnummer (**TNR**): 9
   - Datum: 31.12.2005
e) Löschen Sie den Gast mit der Nummer (**GNR**) 27 und *alle* zugehörigen Reservierungen!

Szenario 2: Beim Paketdienst

Der internationale Paketdienst IPS (International Parcel Service) muss täglich eine Vielzahl von Postsendungen ausliefern und verwendet dazu folgende Relationen in einer relationalen Datenbank:

POSTKUNDE (KNR, Name, Vorname, Straße, PLZ)
SENDUNG (SID, AbsenderNR, EmpfängerNR, Porto, Einschreiben)
TOUR (TNR, Fahrer, Startzeit, Dauer)
LADELISTE (TNR, SendungsID, Position)

Es existieren folgende Fremdschlüsselbeziehungen: Das Attribut **SendungsID** der Relation **LADELISTE** ist ein Fremdschlüssel auf das Attribut **SID** der Relation **SENDUNG**. Die Attribute **AbsenderNR** und **EmpfängerNR** der Relation **SENDUNG** sind Fremdschlüssel auf das Attribut **KNR** der Relation **POSTKUNDE**. Die Attribute **KNR** und **PLZ** der Relation **POSTKUNDE**, das Attribut **SID** der Relation **SENDUNG**, sowie das Attribut **TNR** der Relation **TOUR** sind Ganzzahlen. Bei dem Attribut Startzeit der Relation **TOUR** handelt es sich um ein Datum, das Attribut **Dauer** wird als Ganzzahl in Minuten angegeben. Bei dem Attribut **Einschreiben** der Relation **SENDUNG** handelt es sich um einen boolschen Wert. Die Datentypen der weiteren Attribute sind dem Kontext zu entnehmen.

Fortsetzung nächste Seite!

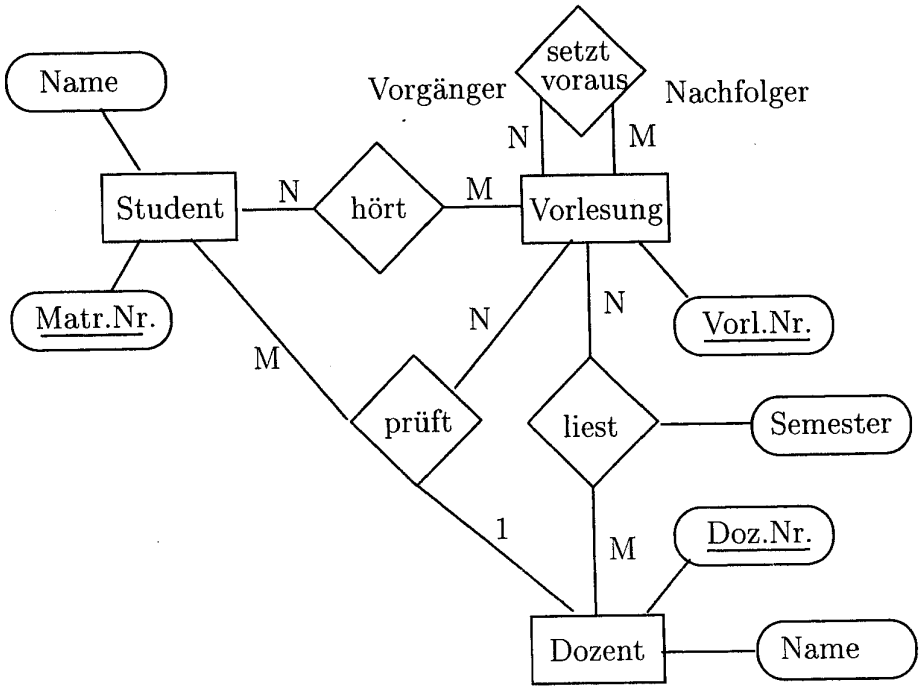
Geben Sie SQL-Anweisungen für folgende Problemstellungen an:

a) Erzeugung des beschriebenen Relationenschemas mit Tabellen, Primary Key Constraints und Foreign Key Constraints.
b) Geben Sie die Namen, Vornamen und Straßen aller Postkunden im Postleitzahlenbereich (**PLZ**) 30999 bis 31999 aus!
c) Geben Sie eine Liste aller Touren, bestehend aus Fahrer, Startzeit und Dauer, aufsteigend geordnet nach Tournummer (**TNR**) aus!
d) Geben Sie für die Tour, mit der Tournummer (**TNR**) 4227 eine Liste aller SendungsIDs (**SID**), die als Einschreiben auszuliefern sind, aufsteigend sortiert nach ihrer Position in der Ladeliste aus!
e) Berechnen Sie das durchschnittliche Porto aller Sendungen der Postkundin mit dem Vornamen „Pippi" und dem Nachnamen „Langstrumpf"!
f) Geben Sie eine Liste aller Touren (**TNR, Fahrer**) mit der Anzahl ihrer Positionen/Sendungen, geordnet nach der Dauer der Tour aus!
g) Fehlsendungen: Geben Sie eine Liste aller Sendungen, jeweils mit dem Vor- und Nachnamen des zugehörigen Postkunden aus, deren Sender und Empfänger gleich sind!

Aufgabe 2:

1. Datenmodellierung: ER-Modell, SQL

Gegeben sei folgender Ausschnitt aus dem ER-Modell einer Universitätsdatenbank:



Fortsetzung nächste Seite!

a) Übersetzen Sie das ER-Modell in das Relationenmodell!
b) Geben Sie für die Tabellen `prüft`, `liest` und `hört` CREATE TABLE-Statements an, welche auch die nötigen Schlüssel- sowie Fremdschlüsselbedingungen enthalten!
c) Ergänzen Sie das ER-Modell um einen schwachen Entity-Typen `Übungsgruppe` mit den Attributen `Gruppennummer` und `Raum`, der in Verbindung zur Tabelle `Vorlesung` stehen soll!

2. Datenbankanfragen in SQL

Gegeben seien die folgenden Tabellen:

- Student(Mat.Nr., Name, Alter, Fachbereich, Fachsemester).
- hört(Mat.Nr., Vorlesung, Semester, Note).

Erstellen Sie in SQL folgende Anfragen:

a) Bestimmen Sie alle Studierenden, die älter sind als 23 Jahre!
b) Bestimmen Sie das durchschnittliche Fachsemester der Studierenden des Fachbereichs "`Informatik`"!
c) Bestimmen Sie für jeden Studierenden - gegeben durch die Matrikelnummer - und jedes Semester die Durchschnittsnote über alle gehörten Vorlesungen!
d) Bestimmen Sie für jeden Studierenden - gegeben durch den Namen - und jedes Semester die Anzahl der Vorlesungen, die er in diesem Semester hört, vorausgesetzt, dass der Studierende im betrachteten Semester mindestens fünf Vorlesungen hört!
e) Bestimmen Sie alle Studierenden - gegeben durch den Namen - die dieselbe Vorlesung (mindestens) in zwei verschiedenen Semestern gehört haben!

3. Funktionale Abhängigkeiten und Zerlegungen

Gegeben sei das Relationenschema $R = (U, F)$ mit der Attributmenge $U = \{A, B, C, D, E\}$ und folgender Menge F von funktionalen Abhängigkeiten:

$$F = \{A \to B, AC \to BD, BC \to A\}$$

a) Bestimmen Sie die Attributhüllen $\{A\}_F^+$ und $\{B, C\}_F^+$.
b) Geben Sie alle Schlüssel für R an!
c) Zerlegen Sie R mittels des Synthesealgorithmus in ein 3NF-Datenbankschema!
d) Ist die Zerlegung von R in die beiden Relationenschemata $R_i = (U_i, F_i), i = 1, 2$, mit $U_1 = \{A, C, D\}$ und $U_2 = \{A, B, C\}$ und entsprechenden Mengen $F_i = \Pi_{U_i}(F^+)$ von funktionalen Abhängigkeiten verlustfrei? Ist sie in BCNF?

Themenschwerpunkt B
(Betriebssysteme)

Aufgabe 1:

1. Speicherverwaltung

Gegeben sei folgendes C-Programm:

```
char feld[5000];
int main() {
    int  i;
    for (i=0; i<10000; i++) { feld[i] = 'x'; }
}
```

Dieses Programm wird für einen 16-Bit-Prozessor mit MMU für Seitenadressierung übersetzt und gebunden. Die Seitengröße sei 4096 Byte. Der logische Adressraum sei so organisiert, dass die erste Seite immer ungenutzt bleibt (um Fehlzugriffe auf NULL-Zeiger zu erkennen). Für den Programmcode wird eine Seite benutzt, die ggf. auch von mehreren Prozessen gemeinsam genutzt werden kann. Die Daten liegen dahinter, der Stack wie üblich am Ende des Adressraums. Für den Stack wird die minimal mögliche Speichermenge angelegt. Die MMU unterstützt Zugriffsschutz (lesen, schreiben, ausführen) für die einzelnen Seiten, Auslagerung von Seiten, die Bildung von Freiseitenpuffern und einen Clock (Second-Chance)-Algorithmus.

a) Beschreiben Sie den Aufbau eines Seitendeskriptors einer solchen MMU (Skizze mit Erläuterung der verschiedenen Daten und ihrem Verwendungszweck)!
b) In einer Liste freier Kacheln (Seitenrahmen) des Hauptspeichers sind aufgeführt (jeweils die Anfangsadresse der Kachel): 0x4000, 0x7000, 0x2000, 0x9000, 0x6000, 0xa000, 0x3000.
Obiges Programm wird nun geladen. Skizzieren Sie den Aufbau der kompletten Seiten-Kachel-Tabelle des Prozesses! (Lassen Sie in Ihrer Zeichnung um die Tabelle herum Platz für die Erweiterungen der Skizze in den Teilaufgaben c) und d)!)
c) Wenn der Prozess in den Zustand laufend versetzt wird, muss die MMU in der Lage sein, die entsprechenden Adressumsetzungen vorzunehmen. Was muss das Betriebssystem beim Dispatching tun, damit dies funktioniert? Erweitern Sie Ihre Skizze um die hierfür erforderlichen Elemente und markieren Sie diese mit c)!
d) Die Adresse &feld[0] sei 0x1008. Beschreiben Sie, wie die Adressumsetzung bei der Anweisung feld[0] = 'x'; erfolgt, so dass der Wert 'x' an eine Stelle im Hauptspeicher des Rechners geschrieben wird! Ergänzen Sie auch Ihre Skizze entsprechend und markieren Sie die Elemente mit d)!
e) In welcher Seite liegt die Variable i?
f) Die for-Schleife überschreitet das Ende des Feldes feld. Was hat dies für Folgen? Bei welchem Schleifendurchlauf treten diese Folgen auf?

Fortsetzung nächste Seite!

2. Dateisystem

a) Hinter der hierarchischen Sicht auf ein Dateisystem, wie es typischerweise unter UNIX oder WindowsNT eingesetzt wird, verbirgt sich eigentlich ein einfach strukturierter flacher Namensraum (die "eigentlichen" Namen sind die Inode-Nummern unter UNIX bzw. die Indizes in die Master-File-Table bei WindowsNT).
Beschreiben Sie für ein solches Dateisystem Ihrer Wahl, wie das Betriebssystem es organisiert, dass der Inhalt der Datei /home/user1/datei1 bzw. \home\user1\datei1 (Dateigröße sei ca. 1 MByte) aufgefunden werden kann!
Skizzieren Sie hierzu alle Datenstrukturen, die hierfür auf dem Hintergrundspeicher angelegt sind, und beschreiben Sie für jeden Typ einer solchen Datenstruktur, wofür sie konzeptionell vorgesehen ist!

b) Zu jeder Datei verwaltet das Betriebssystem eine Menge von Attributen. Wo werden diese Attribute verwaltet?

c) Beispiele für solche Dateiattribute sind:
Dateityp,
Zugriffsrechte,
Dateigröße,
Zeit der letzten Änderung,
Zeit des letzten Zugriffs,
Link-Zähler und
Eigentümer.
Beschreiben Sie für jedes dieser Attribute jeweils möglichst ausführlich, wofür das Betriebssystem das Attribut verwaltet (was sagt das Attribut im Detail aus, in welchen Situationen ändert es sich ggf., auf welche Operationen hat es Auswirkungen etc.)!

d) UNIX- und auch neuere Versionen von WindowsNT-Dateisystemen unterscheiden zwischen "Hard-Links" und "Symbolic-Links". Was ist der Unterschied und warum benötigt man diese unterschiedlichen Arten von Links?

3. Prozesse

a) Nach welchen Kriterien kann man Auswahlstrategien zur Prozessorvergabe (Scheduling) klassifizieren?

b) Welche Strategie würden Sie für das Scheduling in einem Betriebssystem einsetzen, das für die Überwachung und Steuerung von Abläufen in einem Kernkraftwerk eingesetzt ist? Welche Parameter sind hierbei für die Schedulingstrategie primär entscheidend und wie werden sie in der von Ihnen gewählten Strategie umgesetzt?

c) Welche Strategie wird heutzutage üblicherweise in Mehrbenutzer-/Mehrprogramm-Betriebssystemen wie z. B. UNIX eingesetzt?
Welche Ziele werden mit solch einer Strategie verfolgt?
Bei solch einer Strategie spielen mehrere Parameter eine Rolle - teilweise haben diese Parameter sehr gegensätzliche Wirkungen. Beschreiben Sie die Parameter und ihre Wirkung auf das Systemverhalten!

Fortsetzung nächste Seite!

4. Threads und Koordinierung

Zwei Threads implementieren ein klassisches Erzeuger-Verbraucher-System:

Thread T1 liest von einem Eingabekanal (blockierendes Lesen, falls dort gerade keine Daten anstehen) und schreibt die Daten in einen Ringpuffer.
Thread T2 entnimmt die Daten aus dem Ringpuffer und gibt sie auf einem Ausgabekanal aus.

a) Wählen Sie zur Synchronisation der beiden Threads ein geeignetes Verfahren und begründen Sie diese Wahl kurz!
b) Skizzieren Sie in einer Programmiersprache Ihrer Wahl die Funktion producer(), die die Aufgabe von Thread T1 realisiert, sowie die Funktion consumer(), die die Aufgabe von Thread T2 realisiert!
c) Warum funktioniert die beschriebene Aufgabe mit User-Level-Threads nicht einwandfrei, sondern nur mit Kernel-Level-Threads? Beschreiben Sie hierzu kurz den grundsätzlichen Unterschied zwischen beiden Kategorien und arbeiten Sie dabei heraus, welcher Umstand das korrekte Verhalten des beschriebenen Erzeuger-Verbraucher-Szenarios verhindert!

Aufgabe 2:

Teilaufgabe 1

Die folgenden drei Jobs stehen zu den angegebenen Zeiten (in Minuten) zur Ausführung an:

Job	Ankunftszeit	Bearbeitungsdauer
1	0	20
2	4	10
3	10	6

1. Stellen Sie die zeitlichen Abläufe in Gantt-Diagrammen (Balkendiagramme über der Zeitachse) dar und berechnen Sie dazu den Turnaround eines jeden Jobs, sowie den Average Turnaround bei den folgenden Scheduling-Strategien:
   - First-Come-First-Served (FCFS)
   - Shortest-Job-First (SJF)
   - Preemptive-Shortest-Job-First (Preemptive SJF)
   - Round-Robin (RR) mit Zeitquantum q = 8

2. Zusätzlich zu den obigen Jobs 1 bis 3 (von der Kategorie Batch) sollen zu den Zeitpunkten 6 und 12 die Jobs Nummer 4 und 5 (von der Kategorie Interactive) mit jeweils einer Laufzeit von 10 Zeiteinheiten zur Bearbeitung anstehen. Nun soll eine Multi-Level-Scheduling-Strategie verfolgt werden derart, dass jeweils eine Warteschlange für die Kategorien Interactive und Batch existiert, wobei das Scheduling bei interaktiven Jobs mit dem Round-Robin-Verfahren mit Zeitquantum q = 4 und bei den Batch-Jobs nach FCFS abläuft. Die Warteschlange der interaktiven Jobs hat Priorität über die Warteschlange der Batch-Jobs, d. h. falls gerade ein Batch-Job bearbeitet wird und es kommt ein neuer interaktiver Job an, so wird der Batch-Job solange angehalten, bis keine interaktiven Jobs mehr im System sind. Geben Sie die Wartezeit eines jeden Jobs an!

Fortsetzung nächste Seite!

Teilaufgabe 2

Sie sind gerade an einen neuen Wohnort gezogen und wollen dies nun dort bei der zuständigen Meldebehörde anzeigen. Bei dieser Meldebehörde gibt es einen Warteraum, der maximal 20 Personen fasst. Sobald Sie in den Warteraum eintreten können, ziehen Sie aus einem Automaten eine fortlaufend vergebene Nummer, mit der Sie später von einem der drei Schalter aus aufgerufen werden. Ein Schalterbeamter wählt jeweils aus den (maximal 20) wartenden Personen diejenige mit der niedrigsten Nummer aus und bearbeitet diese Anmeldung vollständig.

Implementieren Sie Ihr obiges Vorgehen zur Anmeldung und die Arbeit eines Schalterbeamten in Pseudo-Code! Die Synchronisation der beteiligten Prozesse soll ausschließlich über allgemeine Semaphore (Counting Semaphore) geschehen. Hinweis: Die obige Aufgabenstellung beinhaltet im Wesentlichen ein Erzeuger-Verbraucher-Schema mit beschränktem Puffer.

Teilaufgabe 3

In der Freispeicherliste zu einem Arbeitsspeicher existieren die freien Blöcke in der angegebenen Reihenfolge mit den folgenden Größen:

10KB, 4KB, 20KB, 9KB, 12KB

Geben Sie für jeden Vergabeschritt die verbleibenden freien Speicherblöcke für die folgenden Anfragen an (ebenfalls in dieser Reihenfolge) nach Blöcken der Größe:

12KB, 10KB, 9KB

Verwenden Sie dabei die folgenden Speichervergabe-Verfahren:

a) First-Fit
b) Best-Fit
c) Worst-Fit

66114

Datenbank- und Betriebssysteme (vertieft)

Frühjahr 2008

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____	**Frühjahr 2008**	**66114**
Kennwort: _____		
Arbeitsplatz-Nr.: _____		

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

— Prüfungsaufgaben —

Fach: **Informatik (vertieft studiert)**

Einzelprüfung: **Datenbank- und Betriebssysteme**

Anzahl der gestellten Themen (Aufgaben): **4** Aufgaben, von denen zwei gemäß untenstehender Auswahlregeln zu bearbeiten sind!

Anzahl der Druckseiten dieser Vorlage: **16**

Zu den zwei Themenschwerpunkten A (Datenbanksysteme) und B (Betriebssysteme) ist jeweils entweder die Aufgabe 1 oder 2 zu wählen. Auf der Vorderseite des Kopfbogens sind im Feld „Gewähltes Thema: Nr." die Nummern der beiden gewählten Aufgaben anzugeben (z. B. A1, B2)!

Bitte wenden!

Themenschwerpunkt A (Datenbanksysteme)

AUFGABE 1

I. Allgemeinwissen

1. Bewerten Sie die folgenden Aussagen!
   Richtig oder falsch? Geben Sie für jede Aussage an, ob diese richtig oder falsch ist! Begründen Sie Ihre Aussage in jedem Fall!
   a) Für den Datenaustausch zwischen Datenbank und Anwendungsprogrammen ist das Datenbankmanagementsystem verantwortlich.
   b) Das Datenbankmanagementsystem erlaubt auch unkontrollierten Zugriff auf den Datenbestand.
   c) Jeder Benutzer/ jedes Anwendungsprogramm eines Datenbanksystems muss für den Datenzugriff die eigentliche Organisation der Datenbank kennen.
   d) Das Drei-Schichten-Modell nach ANSI/SPARC besteht aus der externen, der logischen und der physischen Schicht.
   e) Der Zugriff auf sehr große Datenbestände ist per Datenbanksystem effizient möglich.
   f) Integrität und Redundanzfreiheit wird mit Hilfe der mengenorientierten Datenmanipulation durch relationale Operatoren gewährleistet.
   g) Das relationale Datenmodell diente als Grundlage zur Entwicklung des hierarchischen Modells und des Netzwerkmodells.
   h) Die Ordnung der Tupel wird durch die Primärschlüsselwerte festgelegt.
   i) Sichten dienen der Vermeidung von redundanter Speicherung.
   j) Fremdschlüssel sind immer einmalig.

2. Beschreiben Sie kurz folgende Begriffe bzw. Abkürzungen im Kontext von Datenbanksystemen
   a) Primärschlüssel
   b) DML
   c) Datenbankmanagementsystem
   d) Satz

II. Relationale Algebra

Die relationale Algebra wird aufgebaut über einer Grundmenge von mengenwertigen Operationen. Diese Grundoperationen können auf Relationen angewendet werden und erzeugen als Ergebnis wieder eine Relation.

Notation:

π = Projektion;	σ = Selektion;	$\bowtie$ = Join;
x = kartesisches Produkt;	$\setminus$ = Mengendifferenz;	
$\cap$ = Schnittmenge;	$\cup$ = Vereinigungsmenge;	
ρ = Umbenennen;	$\div$ = relationale Division	

1. Welche Operationen können sowohl auf Mengen als auch auf Relationen angewandt werden?

2. Definieren Sie die Operation „Division"! Auf welche Grundoperation kann die Division zurückgeführt werden und wie? Ein Beweis ist nicht erforderlich.

3. Eine Relation A mit den Attributen u, v, w, x habe 5 Tupel, eine Relation B mit den Attributen x, y, z habe 3 Tupel.
   a) Wie viele Tupel und Attribute hat das kartesische Produkt aus A und B?
   b) Wie viele Tupel und Attribute hat der Natural Join aus A und B?

Fortsetzung nächste Seite!

III. Datenbankentwurf

1. Wozu dient das Drei-Schichten-Modell nach ANSI/Sparc?

2. Geben Sie ein selbst gewähltes Beispielszenario, an dem Sie die Vorteile des Vorgehens nach ANSI/Sparc erläutern!

IV. Normalformenlehre

Die Normalformenlehre befasst sich mit der Fragestellung, wie „schlechte" Relationenschemata erkannt und gegebenenfalls verbessert werden können.

1. Nennen Sie die drei möglichen grundlegenden Anomalien, die durch Normalisierung vermieden werden, beim Namen!

2. Welche Aussagen sind für funktionale Abhängigkeiten richtig (W, X, Y und Z sind beliebige Attributmengen)?
   a) $X \rightarrow Y \quad \Rightarrow \quad Y \subseteq X$
   b) $X \rightarrow Y$ und $Z \subseteq Y \quad \Rightarrow \quad X \rightarrow Z$
   c) $X \rightarrow Y$ und $YW \rightarrow Z \quad \Rightarrow \quad XW \rightarrow Z$
   d) $X \rightarrow Y$ und $Z \rightarrow Y \quad \Rightarrow \quad Z \rightarrow X$

3. Warum ist folgende Relation nicht in zweiter Normalform? Und warum werden Adressen dennoch häufig so gespeichert?
   `Adresse{Straße, Hausnummer, Postleitzahl, Wohnort}`

4. Wie muss folgende Relation mit weiteren funktionalen Abhängigkeiten $B \rightarrow D$ und $D \rightarrow E$ zerlegt werden, damit sie in dritter Normalform vorliegt?
   `R{A, B, C, D, E}`

V. Transaktionen und Transaktionsverarbeitung

1. Welche grundlegenden Eigenschaften müssen Datenbanktransaktionen heute garantieren? Benennen Sie jede Eigenschaft und beschreiben Sie jede mit max. einem Satz!

2. Können zwei parallel ablaufende Transaktionen (eine lesend, die andere schreibend), unter Garantie der oben definierten Eigenschaften, gleichzeitig auf dasselbe Datenbankobjekt zugreifen? Begründen Sie ihre Antwort in max. drei Sätzen!

3. Geben Sie an, welche Operationen auf Transaktionsebene Sie kennen, und beschreiben Sie jede mit max. zwei Sätzen!

4. Sie haben das so genannte „2 Phasen-Commit-Protokoll" kennen gelernt. Skizzieren Sie das Szenario und erklären Sie die unterschiedlichen Abläufe, die entweder zu einem erfolgreichen oder nicht erfolgreichen Abschluss führen!

Fortsetzung nächste Seite!

VI. ER-Modellierung und Relationenmodell

1. ER-Modellierung

Erstellen Sie das ER-Diagramm einer fiktiven Zimmerverwaltung eines Krankenhauses! Wo möglich bzw. sinnvoll, sollen 3-fache Beziehungen und Generalisierung/Spezialisierung verwendet werden. Attribute von Entitäten und Beziehungen sind anzugeben; Schlüsselattribute werden durch Unterstreichen gekennzeichnet. Die Kardinalitäten von Beziehungen und - falls nötig - Rollennamen sollen ins Diagramm aufgenommen werden. Verwenden Sie zur Angabe der Kardinalität die (Min, Max)-Notation! Führen Sie Surrogatschlüssel nur ein, falls es nötig ist!

„Im Krankenhaus"

Zimmer werden durch eine eindeutige Zimmernummer identifiziert. Weiterhin sind noch die Anzahl der Betten und die Größe in qm des Zimmers angegeben.

Das Krankenhaus beschäftigt **Personal**, wobei es zwei besondere Personengruppen gibt: **Ärzte** und **Schwestern**. Jede Person hat eine Personalnummer, Vorname und Nachname. Schwestern haben zusätzlich eine vereinbarte Wochenarbeitszeit und Ärzte ihr Approbationsjahr und die -behörde gespeichert.

Das Personal ist in verschiedenen **Stationen** (also Krankenhausabteilungen) beschäftigt. Jede Station wird durch genau eine Schwester geleitet. Außerdem gibt es zu jeder Station genau einen Arzt, der zu ihr in Chefarzt-Beziehung steht. Es gibt im Krankenhaus Ärzte, die nicht Chefarzt einer Station sind.

Jeder **Dienstplan** hat ein Beginn- und Enddatum. Durch den Dienstplan wird festgelegt, welche Schwester welches Zimmer betreuen muss. Es gibt keine leeren Dienstpläne und auch keine untätigen Schwestern; manche der Zimmer werden allerdings nicht von Schwestern betreut, da sie nicht der Patientenunterbringung dienen.

2. Relationenmodell

Ausgehend von Ihrem ER-Diagramm-Entwurf aus Aufgabenteil 1 ist ein Relationenschema in dritter Normalform (3. NF) zu entwerfen. Primärschlüssel werden dabei durch Unterstreichen, Fremdschlüssel durch Nennung der referenzierten Relation in eckigen Klammern hinter dem Attributnamen kenntlich gemacht. Attribute, die nicht Primärschlüssel sind, aber dennoch „NOT NULL" bzw. „UNIQUE" sein müssen, sind zu kennzeichnen.

Folgende willkürliche Beispiele dienen als Hinweis auf die Notation:

```
Haus (Straße, OrtId[Ort])
Ort (OrtId, PLZ, Name)
Mitarbeiter (PersNr, Vorname, Nachname, SozialversNr)
SozialversNr: UNIQUE
```

VII. SQL

Beachten Sie:

Primärschlüssel werden durch Unterstreichen, Fremdschlüssel durch Nennung der referenzierten Relation in eckigen Klammern hinter dem Attributnamen kenntlich gemacht.

Beispiel:

```
Haus (Straße, OrtId[Ort])
Ort (OrtId, PLZ, Name)
```

Fortsetzung nächste Seite!

Interpretation:
Das Attribut `OrtId` der Relation `Haus` verweist als Fremdschlüssel auf das Attribut `OrtId` der Relation `Ort`.

„Kundenverwaltung"

Ein Unternehmen verwendet folgendes Datenbankschema für seine Verwaltung:

```
Kunde(KNr, Name, Vorname)
Bestellung(BNr, Datum)
Artikel(ANr, Bezeichnung, KatNr[Kategorie])
Best_K_A(BNr[Bestellung], ANr[Artikel], KNr[Kunde], Anzahl)
Kategorie(KatNr, Bezeichnung)
```

In `Best_K_A` sind Teilbestellungen gespeichert.

Formulieren Sie folgende Datenbankoperation in SQL:

Einfache Anfragen:

a) Bestellungen sollen als Ganzes geliefert werden. Für Verhandlungen mit einem Lieferdienst soll ermittelt werden, wie viele Artikel eine Bestellung durchschnittlich enthält.
b) Es soll eine Kundenliste erstellt werden und dabei ausgegeben werden, wann der Kunde zuletzt eine Bestellung aufgegeben hat. Darzustellen sind KNr, Name, Vorname und Datum der letzten Bestellung. Das Ergebnis soll nach Datum absteigend sortiert sein.

Datenänderungen:

c) Ein neuer Kunde ist einzufügen. Er heißt „`Hans Mueller`" und er soll die Kundennummer bekommen, die 1 größer ist als die bisher größte!
d) Einige Artikel wurden noch keiner Kategorie zugeordnet (`Artikel.KatNr` ist `NULL`). Diese Artikel sind der Kategorie zuzuordnen, die die Bezeichnung 'Sonstiges' trägt.
e) Um zukünftige Produktverwechslungen bei Bestellungen auszuschließen, sollen alle Artikel mit der Bezeichnung Waage aus der Kategorie `Kuechengeraet` in `Kuechenwaage` umbenannt werden.
f) Im Datenbestand ist ein Fehler aufgetreten. Natürlich ergeben Teilbestellungen keinen Sinn, wenn `Anzahl=0` ist. Daher sind alle Einträge aus `Best_K_A` zu löschen, deren `Anzahl=0` ist.

Komplexe Anfragen:

(Es dürfen kein TOP(n) oder LIMIT verwendet werden! Sichten sind ausdrücklich erlaubt!):

g) Es sollen die fünf Kunden mit der größten Anzahl an bestellten Artikeln (Summe) ermittelt werden. Auszugeben sind Vorname, Name und Gesamtzahl der verkauften Artikel sortiert nach Name aufsteigend.
h) Es sollen die fünf am meisten (nicht am häufigsten) bestellten Artikel (Summe der Anzahl) ermittelt werden. Auszugeben sind Artikelbezeichnung und Gesamtzahl sortiert nach Artikelbezeichnung aufsteigend.
   Achtung: Unterschiedliche Artikel (unterschiedliche Artikelnummer) können die gleiche Bezeichnung tragen!

Fortsetzung nächste Seite!

AUFGABE 2

I. Das Entity-Relationship Modell

Das Fremdenverkehrsamt will sich einen besseren Überblick über Zirkusse verschaffen. In einer Datenbank sollen dazu die Zirkusse, die angebotenen Vorstellungen, die einzelnen Darbietungen in einer Vorstellung sowie die zugehörigen Dompteure und Tiere verwaltet werden.

Ein Zirkus wird eindeutig durch seinen Namen gekennzeichnet und hat einen Besitzer. Vorstellungen haben eine `VorstellungsID` und ein Datum. Darbietungen haben neben der eindeutigen `ProgrammNr` eine Uhrzeit. Ein Dompteur hat eine eindeutige `AngestelltenNr` sowie einen Künstlernamen. Tiere sind eindeutig durch eine `TierNr` bestimmt und haben außerdem eine Bezeichnung der Tierart.

Vorstellungen werden von genau einem Zirkus angeboten. Ein Zirkus bietet mehrere Vorstellungen an und stellt mehrere Dompteure an. Ein Dompteur ist genau bei einem bestimmten Zirkus angestellt. Eine Darbietung findet in einer bestimmten Vorstellung statt. In einer Vorstellung finden mehrere Darbietungen statt. Des Weiteren trainiert ein Dompteur mehrere Tiere, ein Tier kann allerdings auch von mehreren Dompteuren trainiert werden. In einer Darbietung tritt genau ein Dompteur mit mindestens einem Tier auf.

a) Erstellen Sie ein Entity-Relationship-Diagramm für obige Datenbank!
b) Setzen Sie das in a) erstellte Entity-Relationship-Diagramm in ein Relationenschema um! Dabei sind Relationships mit einer möglichst geringen Anzahl von Relationen zu realisieren, wobei unnötige Redundanzen vermieden werden sollen. Ein Relationenschema ist in folgender Form anzugeben: `Relation(Attribut1, Attribut2, . . .)`. Schlüsselattribute sind dabei zu unterstreichen. Achten Sie bei der Wahl des Schlüssels auf Eindeutigkeit und Minimalität!

II. Normalisierungstheorie

Gegeben sei das Relationenschema $R(\underline{A}, B, C, D, E, F, G)$. Die Attribute seien atomar, d. h. R ist in 1. Normalform. Neben $\{A\}$ gibt es keine weiteren Schlüsselkandidaten.
Zusätzlich zu den durch die Schlüsseleigenschaft geltenden funktionalen Abhängigkeiten gelten die folgenden funktionalen Abhängigkeiten: $D \rightarrow E$, $D \rightarrow F$, $F \rightarrow G$.

a) Ist R in 2. Normalform? Begründen Sie Ihre Antwort!
b) Ist R in 3. Normalform? Begründen Sie Ihre Antwort!
c) Zerlegen Sie R in mehrere Relationen, die alle in 3. Normalform sind (wenden Sie ein Verfahren Ihrer Wahl an)! Geben Sie für jede der neuen Relationen einen Schlüssel an!

Gegeben sei das Relationenschema $S(\underline{A}, \underline{B}, C, D, E, F)$. Die Attribute seien atomar, d. h. S ist in 1. Normalform. Neben $\{A, B\}$ gibt es keine weiteren Schlüsselkandidaten.
Zusätzlich zu den durch die Schlüsseleigenschaft geltenden funktionalen Abhängigkeiten gelten die folgenden funktionalen Abhängigkeiten: $A \rightarrow C$, $A \rightarrow D$; $B \rightarrow E$.

d) Ist S in 2. Normalform? Begründen Sie Ihre Antwort!
e) Ist S in 3. Normalform? Begründen Sie Ihre Antwort!
f) Zerlegen Sie S in mehrere Relationen, die alle in 3. Normalform sind (wenden Sie ein Verfahren Ihrer Wahl an)! Geben Sie für jede der neuen Relationen einen Schlüssel an!

Fortsetzung nächste Seite!

III. SQL

Gegeben sei das folgende relationale Datenbankschema (Schlüsselattribute sind jeweils unterstrichen):

```
Personal (PNR, Name, Geburtsdatum, Gehalt, ANR)
Abteilung (ANR, Bezeichnung, LeiterPNR, FNR)
Filiale (FNR, Ort)
```

In der Datenbank sind die Daten von Mitarbeitern, Abteilungen und Filialen eines Handelsunternehmens gespeichert. Jeder Mitarbeiter hat eine Personalnummer (PNR), einen Namen, ein Geburtsdatum, ein Gehalt und ist in einer bestimmten Abteilung beschäftigt. Jede Abteilung hat eine Abteilungsnummer (ANR), eine Bezeichnung, die Personalnummer des Abteilungsleiters (LeiterPNR) und gehört zu einer bestimmten Filiale. Jede Filiale hat eine Filialnummer (FNR) und ist in einem Ort beheimatet.

Formulieren Sie die folgenden Anfragen in SQL:

a) Geben Sie für jede Abteilung die Nummer und die Bezeichnung der Abteilung zusammen mit der Personalnummer und dem Namen des Abteilungsleiters aus! Das Ergebnis soll aufsteigend nach der Abteilungsnummer sortiert werden.

b) Geben Sie diejenigen Orte aus, in denen mehr als zwei Filialen beheimatet sind!

c) Finden Sie die Personalnummern derjenigen Mitarbeiter, für die ein anderer Mitarbeiter mit dem gleichen Geburtsdatum existiert!

Themenschwerpunkt B
(Betriebssysteme)

AUFGABE 1

I. Speicherverwaltung

In einem System mit Seitenadressierung (paged address space), Adresslänge = 16 Bit, Seitengröße = 4 KByte, Hauptspeichergröße = 64 KByte wird ein Programm durch einen Prozess ausgeführt *(zur Erinnerung: $4096_{10} = 1000_{16}$).*

Die erste Seite des virtuellen Adressraums wird nicht genutzt. Das Textsegment des Prozesses umfasst 2 Seiten, das Datensegment umfasst ebenfalls 2 Seiten, direkt im Anschluss daran. Das Stack-Segment umfasst eine Seite ganz am Ende des virtuellen Adressraums.

Die Seiten des Textsegments liegen im Hauptspeicher in den Seitenrahmen (= Kacheln) auf Adresse 0x4000 und 0x7000, die erste Seite des Datensegments liegt in Seitenrahmen 0x6000, die zweite Seite ist ausgelagert und liegt auf der Platte im Block 0x37000. Die Seite des Stacks liegt in Seitenrahmen 0x9000. Das (sehr kleine) Betriebssystem belegt die ersten 4 Seitenrahmen.

Die Seitenrahmen von 0xa000 bis 0xf000 sind durch einen anderen Prozess belegt.

1. Skizzieren Sie den Aufbau des virtuellen Adressraums des Prozesses sowie den Aufbau und die Belegung des Hauptspeichers und die Abbildungen dazwischen!

2. Die Seiten des Textsegments seien read-only in den Adressraum abgebildet, alle anderen Seiten zum Lesen und Schreiben. Die Ausführung von Maschinenbefehlen aus dem Daten- oder Stack-Segment ist nicht zulässig.
   Skizzieren Sie die Datenstrukturen, die die MMU für die Umsetzung des beschriebenen virtuellen Adressraums benötigt und wie dabei die logische Adresse 0x3240 umgesetzt wird!

3. In dem Programm stehen folgende Anweisungen:
```
void f1() {
    static int *p = (int *)0x4018;
    int i;
    ...
    *p = 41;
    i = 5;
    ...
```
   Beschreiben Sie, was bei der Ausführung des Prozesses passiert, wenn er die beiden obigen Zuweisungen ausführt:
   a) Welche Aktivitäten laufen der Reihe nach in der Anwendung und im Betriebssystem ab? (Nummerieren Sie diese Schritte!)
   b) Welche Wechsel des Prozesszustands finden hierbei bei welchen Schritten statt?
   c) Tragen Sie in Ihre Skizzen zu den Teilaufgaben 3a) und 3b) die Änderungen ein, die im Laufe der Ausführung der Anweisungen erfolgen, und markieren Sie die entsprechenden Stellen mit c)! Tragen Sie dabei auch die Orte ein (ungefähr), an denen die Werte 41 und 5 im Hauptspeicher hingeschrieben werden!

Fortsetzung nächste Seite!

II. Verklemmungen

1. Welche Bedingungen müssen gegeben sein, damit eine Verklemmung auftreten kann?

2. Welche drei grundsätzlichen Verfahren gibt es, um mit der Verklemmungsproblematik umzugehen?
   a) Beschreiben Sie jedes Verfahren! Wie ist jeweils die grundsätzliche Vorgehensweise?
   b) Geben Sie ein Beispiel für einen Algorithmus an, der bei einem dieser Verfahren zum Einsatz kommt und beschreiben Sie den Algorithmus!

Für die folgenden Teilaufgaben III., IV. und V. wird das Datensicherungs- und Archivsystem eines Unternehmens betrachtet. Hierzu besitzt das unternehmenseigene Rechenzentrum einen Bandroboter mit fünf Bandlaufwerken und einem Regallager für die Bänder. Das System wird zur täglichen Datensicherung sowie zur gezielten Archivierung von Datenbeständen genutzt.

III. Dateisystem

1. Wie würden Sie den Dateibestand eines großen Softwareentwicklungsprojekts auf der Festplatte organisieren, damit die Programmquellen (mehrere Anwendungsprogramme sowie Funktionsbibliotheken), die Dokumentation sowie mehrere, aktuell an Kunden ausgelieferte Binärversionen der Software sowohl als Ganzes, als auch in den beschriebenen Teilen einfach archiviert werden können? Erläutern Sie Ihre Entscheidung am besten anhand einer Skizze!

2. Welche Informationen über Dateien bzw. über Kataloge würden Sie archivieren? Wozu werden welche Informationen im Archiv bzw. bei oder nach einem Restaurieren des Archivs benötigt?

3. Skizzieren Sie in einer Programmiersprachen-ähnlichen Form den Aufbau eines einfachen Archivierungsprogramms, das eine Archivierung entsprechend 1. und 2. realisiert! Gehen Sie dabei vereinfachend davon aus, dass der gesamte Bandroboter über die Datei "/dev/roboter" angesprochen werden kann und die Auswahl des Bandes und des Laufwerks (in dieser Teilaufgabe) automatisch erfolgt!

IV. Prozesse und Threads

Da von modernen Plattensystemen schneller gelesen wird, als auf ein Bandgerät geschrieben werden kann, sollen die Daten auf mehrere Bänder parallel geschrieben werden. Hierzu soll das Archivierungsprogramm folgendermassen organisiert werden:

- Ein Prozess P_0 leistet die oben in Teilaufgabe 3 beschriebene Funktionalität und schreibt die Daten in einen großen Speicherbereich (10 MB), der in Blöcken zu je 1 MB organisiert ist und als Ringpuffer betrieben wird.
- Für jedes Bandgerät i (i = 1...5) gibt es einen Prozess P_i, der jeweils den nächsten verfügbaren Block aus dem Ringpuffer entnimmt und auf das Bandgerät schreibt.

1. Wie kann der Ringpuffer als gemeinsamer Speicher in einem Betriebssystem mit Seitenadressierung realisiert werden? In welchen Datenstrukturen müssen hierzu welche Einträge erfolgen?

2. Skizzieren Sie die logischen Adressräume der Prozesse P_0 und P_1 und den Zusammenhang zu dem physikalischen Hauptspeicher!

Fortsetzung nächste Seite!

3. Die beschriebene Realisierung mit Hilfe von Prozessen ist teuer. Wodurch entstehen diese Kosten (Speicher, Laufzeit)?

4. Eine Realisierung mit Hilfe von Threads wäre günstiger. Welche Arten von Threads kennen Sie, wodurch unterscheiden sie sich, was ist der Unterschied zu Prozessen?

5. Welche Thread-Art ist für die beschriebene Aufgabe geeignet und sollte statt einer Realisierung mit Prozessen genutzt werden? Welche Kosten spart man dadurch ein, welche Abläufe werden effizienter?

6. Warum ist die andere Thread-Art für die Aufgabe nicht geeignet?

V. Koordinierung

Unter der Annahme, dass mehrere Archivierungen bzw. Restaurierungen gleichzeitig anfallen können und dabei jeweils parallel zwei bis fünf Laufwerke genutzt werden, gibt es eine ganze Reihe von Koordinierungsproblemen:

- Zugriff auf einzelne Bänder
- Zugriff auf Bandlaufwerke
- Zugriff auf Ringpuffer

1. Welche Art von Koordinierungsproblem liegt in den obigen drei Situationen jeweils vor?

2. Die Archivierung von vier unterschiedlichen Softwareprojekten wird fast gleichzeitig gestartet. Entsprechend der Aufgabe „IV. Prozesse und Threads" soll jeweils auf drei Laufwerken parallel geschrieben werden. Welche Problematik hinsichtlich Verklemmungen existiert hier?

3. Nennen Sie zwei unterschiedliche Maßnahmen zur Verklemmungsvorbeugung im obigen (in Teilaufgabe 2 beschriebenen) Szenario! Welche der für Verklemmungen notwendigen Bedingungen werden dadurch jeweils entkräftet?

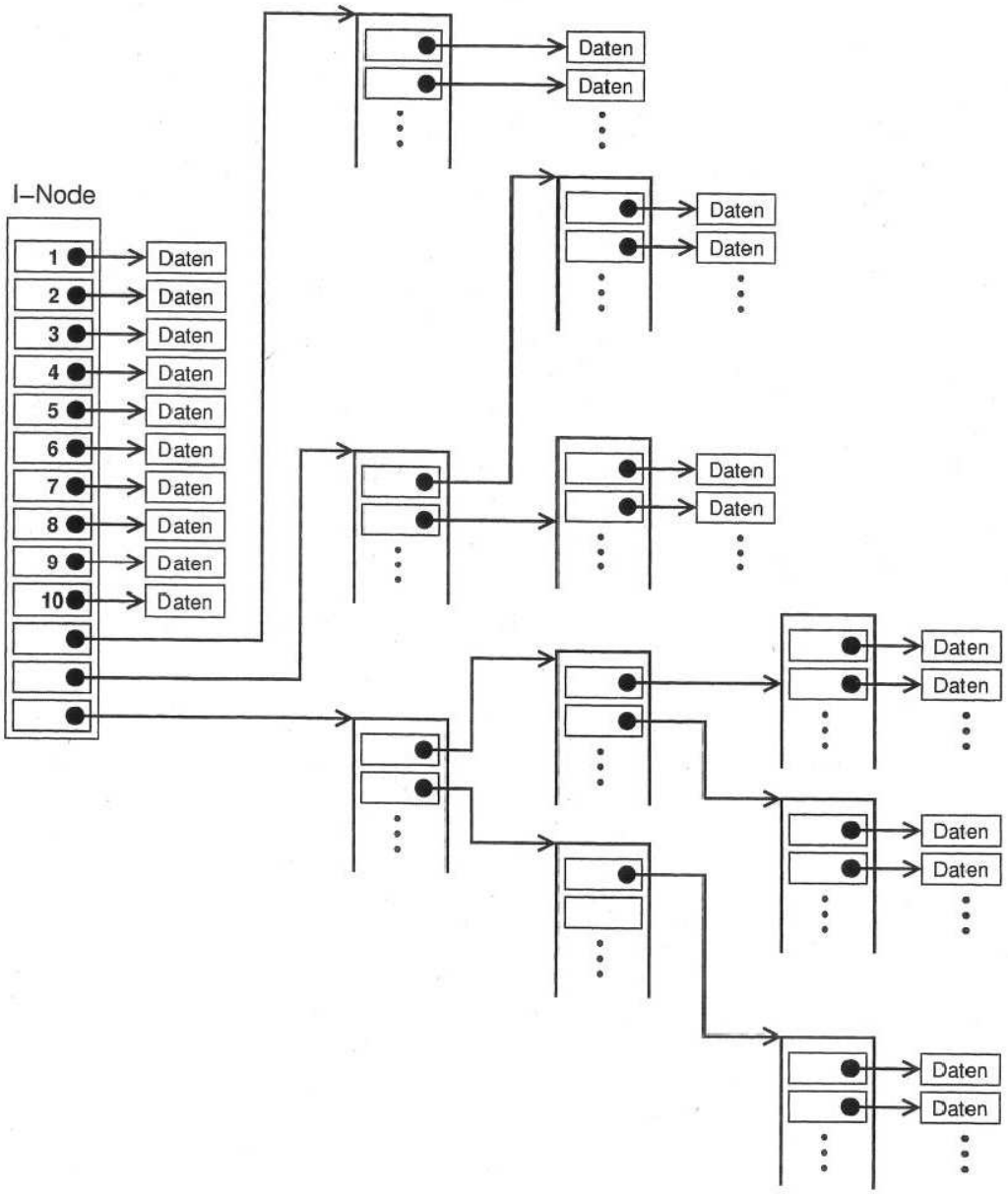
4. Skizzieren Sie in einer Programmiersprachen-ähnlichen Form zwei Funktionen `put` und `get`, die den Zugriff auf den Ringpuffer realisieren!

Fortsetzung nächste Seite!

AUFGABE 2

I. Dateisysteme

Unix-Filesysteme implementieren das Konzept der I-Nodes. Dabei findet eine strikte Trennung des eigentlichen Dateiinhalts von den Verwaltungsinformationen (z. B. Dateigröße, Dateityp, Eigentümer, Gruppe, Zugriffsrechte) statt. Die Anzahl der I-Nodes sowie die Größe der Blöcke werden bei der Formatierung des Dateisystems festgelegt.



Gehen Sie in dieser Aufgabe von folgenden Annahmen aus:

- Jeder I-Node enthält 10 Felder für direkte Adressen auf Datenblöcke der Datei, sowie jeweils einen Verweis auf den ersten, zweiten und dritten Indirektionsblock - insgesamt also 13 Adressfelder (siehe Grafik).
- Der erste Indirektionsblock (Indirektionsblock erster Stufe) enthält Adressen von Datenblöcken. Der zweite Indirektionsblock (Indirektionsblock zweiter Stufe) enthält Adressen von weiteren Indirektionsblöcken erster Stufe, die ihrerseits Adressen von Datenblöcken enthalten. Analog für den dritten Indirektionsblock mit wieder einer zusätzlichen Stufe.
- Alle Adressen sind 4 Bytes lang.

Fortsetzung nächste Seite!

- Die Blockgröße beträgt einheitlich 4 KBytes (gilt sowohl für Daten- als auch für Indirektionsblöcke)

Beantworten Sie die folgenden Fragen!

a) Wie groß ist der Inhalt einer Datei **mindestens** (genaue Angabe in Bytes), wenn ihr I-Node auf genau zwei Datenblöcke direkt verweist?
b) Wie groß ist der Inhalt einer Datei **höchstens** (genaue Angabe in Bytes), wenn ihr I-Node auf genau zwei Datenblöcke direkt verweist?
c) Es soll auf die Datei `/baum/zweig/ast` zugriffen werden. Wie viele I-Nodes und Datenblöcke müssen dazu mindestens gelesen werden?
d) Warum ist es sinnvoll, den Dateityp im I-Node festzuhalten?
e) Mit Hilfe des Befehls `ln` lassen sich in Unix Verknüpfungen (Links) auf eine Datei erstellen. Dabei wird zwischen Hardlinks (`ln quellDatei zielDatei`) und symbolischen Links (`ln -s quellDatei zielDatei`) unterschieden. Wodurch unterscheiden sich diese beiden Varianten?

Auf einem Rechner wird das Dateisystem Ext2 (ohne Journaling) mit einer Blockgröße von 4 KBytes verwendet. Folgende Situation ist gegeben:

- Der I-Node mit der Nummer 332 adressiert die Datenblöcke mit den Adressen *0xA0A8AFA0*, *0xA0A8AFA4* und *0xA0A8AFA8* direkt. Die verbleibenden sieben Felder für direkte Adressen, sowie die drei Felder für die Adressierung von Indirektionsblöcken enthalten keine Adressen (NULL). Der Referenzzähler (Linkzähler) steht auf 1.
- Der Datenblock *0x8124AAA0* beschreibt das Verzeichnis studies, welches unter anderem die Datei notes.txt enthält. Der entsprechende Verzeichniseintrag verweist auf den I-Node 332.
- Der Datenblock *0x011100B0* beschreibt das Verzeichnis info, welches unter anderem die Datei notes_copy.txt enthält. Der entsprechende Verzeichniseintrag verweist ebenfalls auf den I-Node 332.

f) Worin besteht in diesem Beispiel die Inkonsistenz?
g) Können die Dateien `notes.txt` und `notes_copy.txt` in den jeweiligen Verzeichnissen vom Nutzer gelesen werden, auch wenn die Inkonsistenz nicht behoben wird? Geben Sie an, ob keine, eine oder beide Dateien gelesen werden können und begründen Sie Ihre Antwort!
h) Was passiert mit dem I-Node 332, wenn der Nutzer nur die Verknüpfung `notes_copy.txt` aus dem Verzeichnis `info` entfernt (`rm notes_copy.txt`)? (Mit Begründung.)

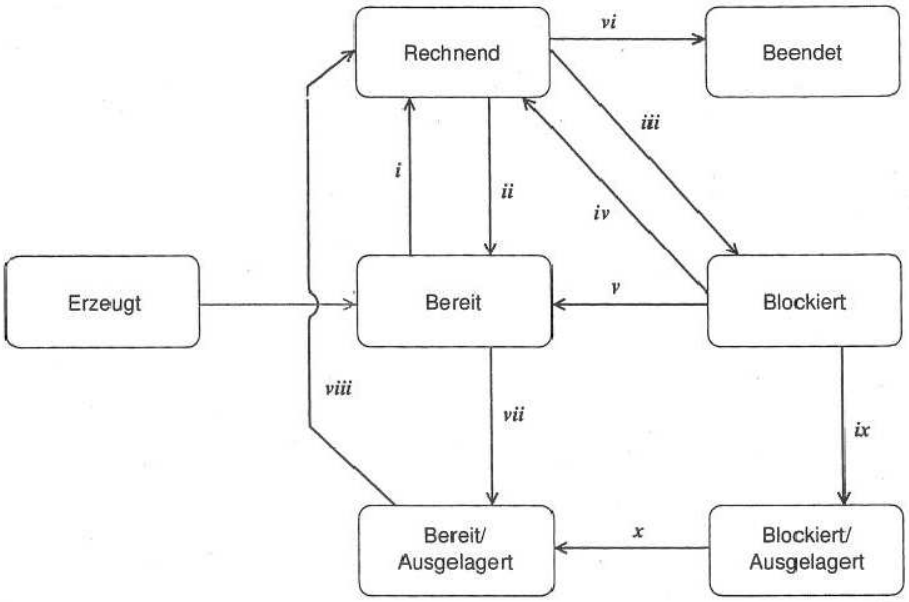
II. Zustands-Prozessmodelle

Diese Aufgabe beschäftigt sich mit dem gängigen 7-Zustands-Modell zur Beschreibung möglicher Ausführungszustände eines Prozesses sowie deren Übergänge. Die fünf Grundzustände sind Erzeugt, Bereit, Rechnend, Blockiert und Beendet, wobei in den beiden Zuständen Bereit und Blockiert jeweils unterschieden wird, ob der jeweilige Prozess sich aktuell im Hauptspeicher befindet (resident) oder ausgelagert wurde (swapped/suspended).

a) Geben Sie für jede der folgenden Transitionen an, ob es sich um einen **gültigen Übergang** handelt! Falls ja, geben Sie ein Beispiel an, welches den jeweiligen Zustandswechsel auslösen könnte! Falls nein, begründen Sie, warum dieser Zustandsübergang nicht sinnvoll ist!

Fortsetzung nächste Seite!

Beispiel:
Erzeugt → Bereit: Gültig; Beispiel: Das Prozess-Image liegt vollständig im Speicher, und der Prozess ist nun rechenbereit. Er wartet darauf, vom Scheduler zur Ausführung ausgewählt zu werden.



i) Bereit → Rechnend
ii) Rechnend → Bereit
iii) Rechnend → Blockiert
iv) Blockiert → Rechnend
v) Blockiert → Bereit
vi) Rechnend → Beendet
vii) Bereit → Bereit-Ausgelagert
viii) Bereit-Ausgelagert → Rechnend
ix) Blockiert → Blockiert-Ausgelagert
x) Blockiert-Ausgelagert → Bereit-Ausgelagert

b) Welche(r) der Zustände Bereit, Rechnend und Blockiert könnte(n) eingespart werden, wenn das Betriebssystem in reinem Batch-Betrieb (d. h. kein Multiprogramming) arbeitet? (Mit Begründung.)

III. Round-Robin-Scheduling und Prozesswechsel

Beim Scheduling nach der Round-Robin-Strategie wird eine FIFO-Warteschlange von Prozessen verwaltet, wobei jeweils dem in der Schlange ersten Prozess für eine feste Zeitdauer Q (die Länge der sog. Zeitscheibe) die CPU zugeteilt wird. Ein Prozess, dessen Zeitscheibe abgelaufen ist, wird am Ende der Warteschlange wieder eingereiht. Die mittlere Bedienzeit T eines Prozesses bezeichne die Zeitspanne, für die ein Prozess im Mittel die CPU benötigt, bevor er das Warteschlangensystem aufgrund einer angestoßenen E/A-Operation bzw. bei Prozessende freiwillig verlässt. Es handelt sich bei T also um die reine Rechenzeit des Prozesses. Die mittlere Verweilzeit V hingegen bezeichne die gesamte Zeitspanne, die ein Prozess im Mittel im System verweilt (ggf. wechselnd zwischen den Zuständen Bereit und Rechnend). Die mittlere Wartezeit W sei die durchschnittliche Summe der Zeiträume, in denen ein Prozess zwar im System verweilt, aber nicht rechnet. Ein Prozesswechsel benötige eine Zeitdauer S, die als Overhead verloren geht, aber unvermeidbar ist.

Zusammenfassend ergeben sich die folgenden Zusammenhänge:

- Größe der Zeitscheibe: Q
- Mittlere Bedienzeit (reine Rechenzeit) eines Prozesses: T
- Mittlere Verweilzeit eines Prozesses: V, $V \geq T$
- Mittlere Wartezeit eines Prozesses: W, $W = V - T$
- Zeit für Prozesswechsel: S

Fortsetzung nächste Seite!

a) Beschreiben Sie kurz, was beim Prozesswechsel passiert und welche Aufgabe das Betriebssystem dabei erfüllt! Erklären Sie in diesem Zusammenhang in kurzen Stichworten den Begriff des Prozesskontextes!

b) Die Zeitdauer S wird also wesentlich von der Komplexität des Prozesswechsels beeinflusst. Wovon hängt diese Komplexität maßgeblich ab?

c) Gegeben sind die Prozesse P_1 bis P_4 mit den in der folgenden Tabelle angegebenen Ankunfts- und Bedienzeiten. Es gelte: $Q = 2$ und vereinfachend $S = 0$ (also vernachlässigbar klein). Eine Zeitscheibe muss nicht voll ausgenutzt werden. Terminiert ein Prozess vor Ablauf von Q, so wird diese eine Zeitscheibe entsprechend verkürzt und der nächste Prozess kann sofort aktiviert werden.

Prozess	Ankunftszeitpunkt	Bedienzeit
P_1	0	4
P_2	1	5
P_3	3	2
P_4	6	4

i) Ermitteln Sie für jeden Prozess die **individuelle Verweil- und Wartezeit**! Erstellen Sie eine Tabelle nach folgendem Muster!
Tipp: Machen Sie sich dazu die Ausführungsreihenfolge der Prozesse durch ein Diagramm (z. B. Gantt-Chart) klar, und zeichnen Sie auch die Warteschlange (separates Blatt)! Für die Bewertung spielen die Zeichnungen allerdings keine Rolle.

Prozess	Verweilzeit	Wartezeit
P_1		
P_2		
P_3		
P_4		

ii) Berechnen Sie die **mittlere Verweil- und Wartezeit** für dieses System!

d) Unter der Annahme, dass in einem System stets rechenbereite Prozesse zur Verfügung stehen, lässt sich die CPU-Auslastung ρ wie folgt definieren: $\rho = \frac{T}{T+n \cdot S}$, wobei n die Anzahl der pro Prozess durchschnittlich anfallenden Prozesswechsel ist. Dabei soll gelten: Ein Prozesswechsel gehört zu dem Prozess, der zuletzt (also unmittelbar **vor** dem Prozesswechsel) gerechnet hat. Terminiert ein Prozess, so gehört der anschließende Wechsel zu einem rechenbereiten Prozess also auch noch zu dem gerade terminierten Prozess.

i) Wie wirkt sich ein hoher Wert für S auf die CPU-Auslastung aus?

ii) Geben Sie für n eine Formel in Abhängigkeit von T und Q an!

iii) Welche CPU-Auslastung ergibt sich, wenn gilt: $Q > T$? Gegen welche Ihnen bekannte Scheduling-Strategie konvergiert Round Robin in diesem Fall?

iv) Welche CPU-Auslastung ergibt sich, wenn gilt: $Q = S$?

IV. Seitenersetzung

In dieser Aufgabe sollen verschiedene Seitenersetzungsstrategien (Paging-Strategien) am konkreten Beispiel verglichen werden. Dabei sei die Menge der Seiten gegeben durch $N = \{0, 1, 2, 3, 4\}$. Die Menge der Seitenrahmen, die für die Speicherung der Seiten im Arbeitsspeicher zur Verfügung stehen, sei gegeben durch $F = \{f_0, f_1, f_2\}$. Auf die fünf Seiten der Menge N werde in folgender Reihenfolge zugegriffen:

$$w = 0\ 4\ 3\ 4\ 3\ 1\ 3\ 4\ 0\ 3\ 2\ 1\ 0$$

Fortsetzung nächste Seite!

Ein Seitenfehler liegt immer dann vor, wenn sich eine referenzierte Seite nicht im Arbeitsspeicher befindet. Dieser ist zu Beginn leer.

a) Ermitteln Sie die Anzahl der Seitenfehler für die Paging-Strategie LRU (Least Recently Used), indem Sie den Zustand des Speichers nach jedem Zugriff in einer Tabelle nach folgendem Muster dokumentieren! Informationen, die aufgrund der Paging-Strategie zusätzlich benötigt werden, tragen Sie ebenfalls ein!

Referenzierte Seiten	f_0	f_1	f_2	*Summe Seitenfehler*
0				
4				
3				
4				
3				
1				
3				
4				
0				
3				
2				
1				
0				

b) Ermitteln Sie nun die Anzahl der Seitenfehler für die Paging-Strategie LFU (Least Frequently Used), indem Sie den Zustand des Speichers nach jedem Zugriff in einer Tabelle nach folgendem Muster dokumentieren! Informationen, die aufgrund der Paging-Strategie zusätzlich benötigt werden, tragen Sie ebenfalls wieder ein!

Referenzierte Seiten	f_0	f_1	f_2	*Summe Seitenfehler*
0				
4				
3				
4				
3				
1				
3				
4				
0				
3				
2				
1				
0				

c) In der Praxis ergibt sich ein signifikantes Problem beim Einsatz der zweiten Strategie (Least Frequently Used). Welches? Wie könnte man LFU modifizieren, um dieses Problem zu minimieren? Tipp: Überlegen Sie sich, was passiert, wenn auf eine Seite zunächst über einen langen Zeitraum häufig zugegriffen wird, später dann die Seite nicht mehr benötigt wird (da z. B. der zugehörige Prozess terminiert wurde)!

Fortsetzung nächste Seite!

V. Verklemmungen (Deadlocks)

Gegeben sind die drei Prozesse P_1, P_2 und P_3 sowie der vier Betriebsmittel R_1 bis R_4. Zum betrachteten Zeitpunkt *t* gilt:

- P_1 nutzt R_3 und fordert R_1 an.
- P_2 nutzt R_4 und fordert R_3 an.
- P_3 nutzt R_1 und fordert R_2 und R_4 an.

a) Zeichnen Sie den **Resource-Allocation-Graph (Betriebsmittelgraph)** für den Zeitpunkt *t* unter Berücksichtigung aller vorhandenen Betriebsmittel!

b) Welche Betriebsmittelanforderung(en) ist (sind) sofort erfüllbar?

c) Entscheiden und begründen Sie anhand des Resource-Allocation-Graphen, ob nach Erfüllung der erfüllbaren Anforderung(en) ein Deadlock vorliegt!

66114

Datenbank- und Betriebssysteme (vertieft)

Herbst 2008

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____	**Herbst 2008**	**66114**
Kennwort: _____		
Arbeitsplatz-Nr.: _____		

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

— Prüfungsaufgaben —

Fach: **Informatik (vertieft studiert)**

Einzelprüfung: **Datenbank- und Betriebssysteme**

Anzahl der gestellten Themen (Aufgaben): **4** Aufgaben, von denen zwei gemäß untenstehender Auswahlregeln zu bearbeiten sind

Anzahl der Druckseiten dieser Vorlage: **15**

Zu den zwei Themenschwerpunkten A (Datenbanksysteme) und B (Betriebssysteme) ist jeweils entweder die Aufgabe 1 oder 2 zu wählen. Auf der Vorderseite des Kopfbogens sind im Feld „Gewähltes Thema: Nr." die Nummern der beiden gewählten Aufgaben anzugeben (z. B. A1, B2)!

Bitte wenden!

THEMENSCHWERPUNKT A
(Datenbanksysteme)

AUFGABE 1

1. Modellierung

Eine Fluggesellschaft möchte in einer relationalen Datenbank Informationen über ihre Flugzeuge und Flugrouten speichern:

Die Flugzeuge werden anhand einer eindeutigen Nummer (FNr) identifiziert. Zusätzlich werden Reichweite und Reisegeschwindigkeit gespeichert. Die Flugzeuge teilen sich in zwei disjunkte Gruppen: Passagierflieger und Transportflieger. Bei den Passagierfliegern sollen die jeweils in den Klassen (First, Dynasty, Economy) vorhandenen Sitzplätze gespeichert werden. Bei den Transportfliegern sollen maximales Gewicht, maximale Höhe, maximale Breite und maximale Tiefe der Nutzlast gespeichert werden.
Die Flugzeuge werden verwendet, um vorher festgelegte Touren zu fliegen. Wenn ein Flugzeug eine bestimmte Tour fliegen soll, müssen Abflugzeitpunkt bei Tourbeginn und Ankunftszeitpunkt bei Tourende gespeichert werden. Für die Touren werden jeweils eine eindeutige Nummer (TNr) und die Gesamtlänge der Tour gespeichert.
Jede Tour unterteilt sich in mehrere Teilstrecken. Die Teilstrecken können nur als Bestandteile von Touren auftreten. Für die Teilstrecken jeder Tour werden jeweils aufsteigende Nummern (ab „1") vergeben um die Flugreihenfolge festzulegen. Für die einzelnen Teilstrecken sollen zusätzlich der Abflugort, Abflugzeitpunkt, Ankunftsort, Ankunftszeitpunkt und die Länge der Teilstrecke gespeichert werden.
Die Orte werden anhand einer eindeutigen Nummer (ONr) identifiziert. Als Zusatzinformationen werden für die Orte Name, Postleitzahl und Land gespeichert.

a) Entwerfen Sie für das beschriebene Szenario ein ER-Diagramm. Bestimmen Sie hierzu:

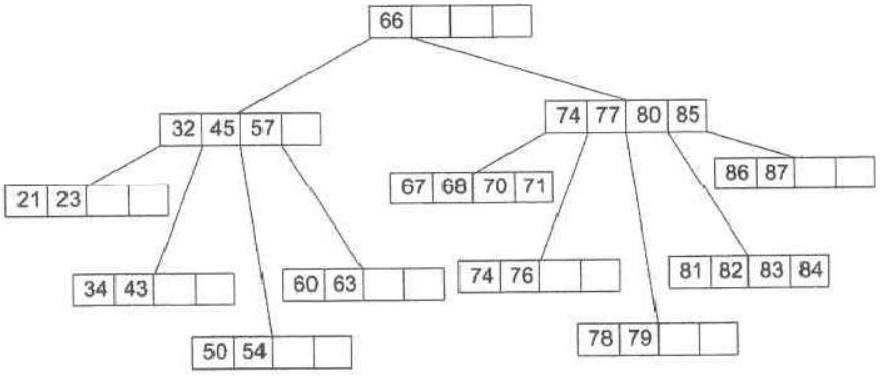
- die Entity-Typen, die Relationship-Typen und jeweils deren Attribute;
- ein passendes ER-Diagramm;
- die Primärschlüssel der Entity-Typen, welche Sie anschließend in das ER-Diagramm eintragen;
- die Funktionalitäten der Relationship-Typen, welche Sie ebenfalls in das ER-Diagramm eintragen.

b) Überführen Sie das ER-Modell aus Aufgabe a) in ein verfeinertes relationales Modell. Geben Sie hierfür die verallgemeinerten Relationenschemata an.

Fortsetzung nächste Seite!

2. Indexstrukturen

Als Indexstruktur einer Datenbank sei folgender B-Baum ($k = 2$) gegeben:



Führen Sie nacheinander die folgenden Operationen aus. Geben Sie die auftretenden Zwischenergebnisse an. Teilbäume, die sich in einem Schritt nicht verändern, müssen nicht erneut gezeichnet werden. Sollten Wahlmöglichkeiten auftreten, so sind größere Schlüsselwerte bzw. weiter rechts liegende Knoten zu bevorzugen.

a) Einfügen von Wert 72

b) Löschen von Wert 32

3. Zerlegungen

Gegeben sei ein Schema $R = (A, B, C, D, E, G)$ und eine FD-Menge

$F := \{E \rightarrow C, C \rightarrow B, CE \rightarrow G, B \rightarrow A\}$.

1. In welcher Normalform ist R bzgl. F?

2. Bestimmen Sie eine 3NF-Zerlegung von R bzgl. F.

3. Ermitteln Sie eine BCNF-Zerlegung von R bzgl. F.

Fortsetzung nächste Seite!

4. SQL

Betrachten Sie die folgenden Relationenschemata zur Verwaltung eines Weinlagers:

Wein:	{ID: integer, Art: string, Lage: string, Jahr: integer, Alc: float}
Weinlager:	{ID: integer, V_Preis: float, Bestand: integer, Min_Bestand: integer}
Weinempfehlung:	{Gericht: string, WeinID: integer}
Weinsorte:	{Art: string, Farbe: string}

Formulieren Sie die folgenden Anfragen in Standard-SQL. Zur Vereinfachung dürfen Sie dabei beliebige Views definieren.

a) Finden Sie alle Gerichte, die zu dem Wein mit dem größten Bestand passen.

b) Geben Sie Art und Durchschnittspreis der Weinsorten aus, deren Durchschnittspreis unter 10 € liegt.

c) Welche Weine werden nie empfohlen?

5. Transaktionen

Ein strikter 2-Phasen-Sperr-Protokoll-Scheduler erhält folgenden, aus den elementaren Operationen der Transaktionen T_1 und T_2 bestehenden Input-Schedule:

$s_{in} = r_1(A)r_1(B)r_2(B)w_2(A)w_1(A)c_1c_2$

Bestimmen Sie einen möglichen Output-Schedule s_{out}, den der Scheduler erzeugt. Beachten Sie hierbei, dass der Output-Schedule die erzeugten *lock*- und *unlock*-Befehle enthalten muss.

AUFGABE 2

1. Transaktionen und Transaktionsverarbeitung

1.1 Erklären Sie die grundlegenden Eigenschaften einer Transaktion (ACID-Prinzip).

1.2 Skizzieren Sie kurz ein Beispiel für das Lost-Update-Problem.

1.3 Was geschieht in den beiden Phasen des 2-Phasen-Sperrprotokolls?

2. ER-Modellierung und Relationenmodell

2.1 ER-Modellierung

Erstellen Sie das Modell eines fiktiven Auftragsverwaltungssystems in E/R-Notation. Attribute von Entitäten und Beziehungen sind anzugeben; Schlüsselattribute werden durch Unterstreichen gekennzeichnet. Die Kardinalitäten von Beziehungen sollen ins Diagramm aufgenommen werden. Führen Sie Surrogatschlüssel nur ein, falls es nötig ist.

„Auftragverwaltungssystem"

In unserem Minisystem müssen Rechnungen mit Rechnungsnummer, Datum, Name und Adresse des Kunden gespeichert werden. Eine Rechnung besteht dabei aus einer oder mehreren Rechnungspositionen.

Kunden werden unterteilt in Privatkunden und Geschäftskunden. Die Zuordnung ist eindeutig und vollständig, d.h. jeder Kunde gehört genau einer der beiden Gruppen an. Bei Privatkunden wird für die leichtere Abrechnung noch die Bankverbindung gespeichert, Geschäftskunden erhalten eine Nummer für ein eigenes Abrechnungskonto.

Eine Rechnungsposition besteht aus der Artikelnummer, der Menge, dem aktuellen Verkaufspreis, der Stückzahl und einer innerhalb einer Rechnung eindeutigen Rechnungspositionsnummer.

Ebenfalls sollen die Artikel verwaltet werden können, die durch eine Artikelnummer, einen Namen und eine Beschreibung gekennzeichnet sind. Zusätzlich wird der Einkaufspreis der Artikel gespeichert.

Artikel können entweder so genannte Einzelartikel sein oder zu Paketartikeln gebündelt werden. Paketartikel beinhalten dabei mehrere Einzelartikel und sollen über eine eigene Artikelnummer ausgewählt werden können. Auch die Einzelartikel in einem Paketartikel müssen separat in eine Rechnung aufgenommen werden können.

Fortsetzung nächste Seite!

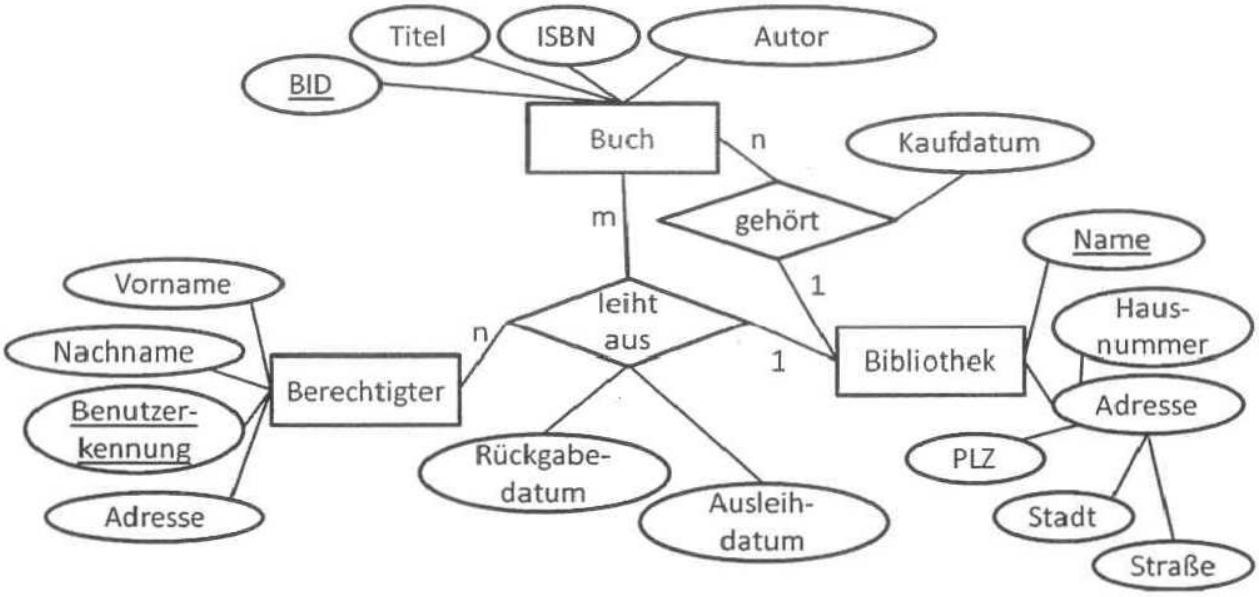
2.2 Relationenmodell: „Bibliothek"

Ausgehend von der gegebenen ER-Darstellung ist ein Relationenschema in dritter Normalform (3. NF) zu entwerfen.

Primärschlüssel werden dabei durch Unterstreichen, Fremdschlüssel durch Nennung der referenzierten Relation inklusive Attribut(e) in eckigen Klammern hinter dem Attributsnamen kenntlich gemacht, z. B.:

`Haus (Straße, OrtId[Ort(OrtId)])`
`Ort (OrtId, PLZ, Name)`

Das Attribut `OrtId` der Relation `Haus` verweist als Fremdschlüssel auf das Attribut `OrtId` der Relation `Ort`.



Fortsetzung nächste Seite!

3. SQL

Bitte beachten Sie: Primärschlüssel werden durch Unterstreichen, Fremdschlüssel durch Nennung der referenzierten Relation inklusive Attribut(e) in eckigen Klammern hinter dem Attributsnamen kenntlich gemacht, z.B.:

```
Haus (Straße, OrtId[Ort(OrtId)])
Ort (OrtId, PLZ, Name)
```

Das Attribut `OrtId` der Relation `Haus` verweist als Fremdschlüssel auf das Attribut `OrtId` der Relation `Ort`.

3.1 Szenario 1: „Online-Movie-Store"

Die Filme eines Online-Movie-Stores sind in folgendem Schema gespeichert:

Online-Movie-Store

```
FILM            (FID, Titel, Erscheinungsjahr, Laufzeit,
                 Drehort[ORT(OID)])

SCHAUSPIELER    (SID, Name, Geburtsjahr)

ORT             (OID, Land, Stadt)

SPIELT_MIT      (Schauspieler[SCHAUSPIELER(SID)], Film[FILM(FID)])
```

Bemerkung: Die Laufzeit eines Films ist als Zahl in Minuten angegeben.

Formulieren Sie die folgenden Anfragen an das oben stehende Schema in SQL:

i) Geben Sie die Namen aller Schauspieler aus, die in der Datenbank enthalten sind. Sortieren Sie die Ausgabe in aufsteigender Reihenfolge.

ii) Welche im Jahr 1996 erschienenen Filme spielen am Drehort Berlin (Deutschland)? Geben Sie Titel und Laufzeit an.

iii) Geben Sie Titel und Erscheinungsjahr aller Filme, in denen Heinz mitgewirkt hat, sortiert nach dem Erscheinungsjahr aus.

iv) Gesucht sind alle Filme, die in einem Jahr erschienen sind, in welchem insgesamt mehr als 4500 Minuten Film produziert worden sind. Außerdem sollen die Filme alle in New York (USA) gespielt haben. Die Titel dieser Filme sind mit Jahr und Gesamtlaufzeit des Jahres auszugeben.

v) In welchen Jahren wurden Filme gedreht, die Harrison und Tommy als Darsteller beinhalten?

Fortsetzung nächste Seite!

3.2 Szenario 2: „Schule modelliert!"

Die Schulverwaltungsdatenbank eines Gymnasiums wird gerade neu aufgebaut und die Mitglieder des Kollegiums erkunden die neuen Möglichkeiten der Datenbank.

```
Fach (Name, Beschreibung)
Schueler (SID, Name, Vorname, Geburtsdatum, Klasse)
Lehrer (LID, Name, Besoldungsstufe, Fachkombination)
Unterricht (LID[Lehrer(LID)], Klasse, Name[Fach(Name)], Tag,
Uhrzeit)
Raum (RID, Raumnummer, Gebaeude, AnzahlPlaetze)
```

Es existieren folgende Fremdschlüsselbeziehungen: Das Attribut `Name` der Relation `Unterricht` bezieht sich auf das Attribut `Name` der Relation Fach. Das Attribut `LID` der Relation `Unterricht` bezieht sich auf das Attribut `LID` der Relation `Lehrer`.
Die Attribute `SID` der Relation `Schueler` und `LID` der Relation `Lehrer` sind Ganzzahlen. Das Attribut `Name` der Relation Fach ist ein Text. Bei dem Attribut `Tag` der Relation `Unterricht` handelt es sich um ein Datum. Das Attribut `Uhrzeit` wird als fünfstellige Zeichenfolge im Format hh:mm angegeben. Das Attribut `AnzahlPlaetze` der Relation `Raum` ist eine dreistellige Ganzzahl. Das Attribut `Geburtsdatum` der Relation `Schueler` wird als Datum gespeichert. Die Datentypen der restlichen Attribute sind dem Kontext zu entnehmen.
Das Attribut `Klasse` der Relation `Unterricht` und das Attribut `Klasse` der Relation `Schueler` haben denselben Wertebereich.
Geben Sie SQL-Anweisungen für folgende Aufgaben an:

i) Erzeugung des beschriebenen Relationenschemas mit Tabellen, Primary Key Constraints und Foreign Key Constraints.

ii) Der Schüler „Kurt Binsen" (Geburtsdatum: 13.02.1991) hat die Schule gewechselt und soll der Tabelle `Schueler` hinzugefügt werden (Klasse: 10a).

iii) Die Klasse „7b" soll von der Lehrerin „Frau Schmidt" (LID: 15) montags um 09:30 Uhr im Raum „02.124" (RID: 124) unterrichtet werden. „Frau Schmidt" und Raum „02.124" sind in den entsprechenden Tabellen bereits eingetragen.

THEMENSCHWERPUNKT B
(Betriebssysteme)

AUFGABE 1

1.1 Die folgenden vier Jobs kommen in einem System zu den angegebenen Zeitpunkten (in Minuten) an:

Jobnummer	Ankunftszeit	Bearbeitungszeit
1	0	8
2	2	2
3	3	5
4	4	2

(a) Zeichnen Sie die Gantt-Diagramme für die Scheduling-Strategien First-Come-First-Served (FCFS), Round Robin (RR mit Zeitquantum=1) und Shortest-Job-First (SJF).

(b) Für jeden Job und jede der in (a) genannten Scheduling-Strategien: Berechnen Sie die Turnaround-Zeit. Berechnen Sie für jede Strategie die Average-Turnaround-Zeit.

1.2 In einem Demand-Paging System mit vier Kacheln sei für einen Prozess die Seitenreferenz-Folge zu bearbeiten:

0, 1, 0, 2, 3, 0, 4, 5, 0, 1

Ermitteln Sie die Anzahl der Seitenfehler, jeweils für die Ersetzungsstrategien

(a) First-In-First-Out,

(b) Optimale Ersetzung (OPT, „Belady's Strategie"), und

(c) Least-Recently-Used (LRU).

Notieren Sie zeilenweise nach jeder Seitenreferenz die komplette Speicherbelegung des Systems und markieren Sie die Zeilen, die nach einem Seitenfehler entstanden. Das erste Laden einer Seite zählt dabei ebenfalls als Seitenfehler.

Fortsetzung nächste Seite!

1.3 In einem Arbeitsspeicher existieren die folgenden nicht-belegten Blöcke in der angegebenen Reihenfolge mit den Größen:

60KB, 36KB, 100KB, 56KB, 72KB

Geben Sie schrittweise die verbleibenden freien Speicherblöcke für die folgenden Anforderungen an (ebenfalls in dieser Reihenfolge) nach Blöcken der Größe:

48KB, 58KB, 64KB, 44KB

Verwenden Sie dabei die Vergabeverfahren:

(a) First-Fit

(b) Best-Fit

(c) Worst-Fit

1.4 Der Dino-Park T-Rex besteht aus einem Dinosaurier-Museum und einem Erlebnispark mit Safari-Rundfahrt. Es gibt im Museum Platz für maximal $m=500$ Besucher und für die Rundfahrt stehen $n=100$ Safari-Wagen zur Verfügung, die jeweils nur einen Besucher aufnehmen können.

Die Besucher werden erst in das Museum eingelassen, falls dort die Maximalzahl m zurzeit nicht überschritten ist, sonst müssen sie vor dem Museum warten. Je nach Interesse verbringen die Besucher einige Zeit im Museum und stellen sich dann an, um einen Safari-Wagen zu bekommen.

Wenn ein leerer Wagen verfügbar ist, lässt dieser den am längsten wartenden Besucher einsteigen und fährt ihn durch den Erlebnispark und dann zum Ausgang des Dino-Parks. Wenn alle n Wagen mit Besuchern unterwegs sind, müssen weitere Besucher am Museumsausgang warten. Falls ein Wagen bereitsteht, einen Besucher aufzunehmen, jedoch kein Besucher wartet, so muss auch der Wagen warten.

Zur Lösung dieser Synchronisierungsprobleme soll jeder Besucher und jeder Safari-Wagen als ein eigener Prozess modelliert werden und es sollen ausschließlich Counting-Semaphoren benutzt werden.

(a) Schildern Sie zunächst, welche Fehler bei Verzicht auf Synchronisierungsmaßnahmen oder bei fehlerhafter Verwendung der Semaphor-Operationen im Dino-Park auftreten können.

Fortsetzung nächste Seite!

(b) Entwerfen Sie eine Lösung in Pseudocode für einen
`process visitor()` und einen
`process car()`

mit den folgenden globalen Semaphoren und den angegebenen Initialisierungen:

```
counting_sema museum_space = 500;
                // Besucherkapazität des Museums
counting_sema cars_waiting = 100;
                // Anzahl der verfügbaren Safari-Wagen
counting_sema visitors_waiting = 0;
                // Anzahl der auf Safari-Wagen wartenden Besucher
```

AUFGABE 2

1. Scheduling-Algorithmen

a) Beschreiben Sie zunächst in ein bis zwei Sätzen die Unterschiede in der Funktion von präemptiven und nicht-präemptiven Schedulingalgorithmen.

b) Betrachten Sie die folgenden Prozesse:

Prozess	Ankunftszeit	benötigte Rechenzeit
P0	0	6
P1	1	4
P2	2	2
P3	3	5
P4	4	3

Führen Sie das Scheduling für dieses Beispiel mit den Algorithmen *(Non-Preemptive) Shortest-Job-First* und *Preemptive-Shortest-Job-First* (auch bekannt als *"Shortest-Remaining-Time-First")* durch.

Stellen Sie den Gesamtablauf je Algorithmus tabellarisch dar, indem Sie für jeden Zeitpunkt den jeweils aktiven Prozess, die getroffene Scheduling-Entscheidung sowie den Zustand der "Ready-Queue" *nach* der Scheduling-Entscheidung angeben. Sie dürfen Zeilen auslassen, in denen sich im Vergleich zur vorigen Zeile nichts ändert.

Gehen Sie dabei von folgender Annahme aus: Sollte das Scheduling-Kriterium keine eindeutige Scheduling-Entscheidung zwischen zwei oder mehreren Prozessen zulassen, so wird von diesen der Prozess mit der *frühesten Ankunfszeit* bevorzugt.

c) Berechnen Sie für obiges Beispiel und NUR für *Preemptive-Shortest-Job First* die durchschnittliche Wartezeit und den Durchsatz. Geben Sie die Durchlaufzeit von Prozess P0 an.

Fortsetzung nächste Seite!

2. Verklemmungen

a) Geben Sie die Bedingungen an, die erfüllt sein müssen, damit eine Verklemmung auftreten kann. Erläutern Sie dabei jede Bedingung etwa in einem Satz.

b) Beschreiben Sie, welche dieser Bedingungen durch geeignete Verfahren verhindert werden können und diskutieren Sie diese Verfahren, indem Sie folgende Fragen für jedes angegebene Verfahren beantworten: Wie ist jeweils die grundlegende Vorgehensweise? Welche Nachteile ergeben sich ggf. durch den Einsatz eines solchen Verfahrens in einem Betriebssystem?

c) Betrachten Sie das "Dining Philosophers" Problem:

Fünf Philosophen sitzen an einem rundem Tisch. Zwischen benachbarten Philosophen liegt jeweils ein Essstäbchen, in der Mitte steht eine große Reisschüssel. Jeder Philosoph wechselt seine Tätigkeit zwischen Essen und Denken ab. Ein Philosoph benötigt zwei Essstäbchen um essen zu können. Um mit Essen zu beginnen nimmt ein Philosoph zunächst das rechts von ihm liegende Essstäbchen, dann das links neben ihm liegende. Nach dem Essen legt der Philosoph beide Essstäbchen ab und denkt wieder. Wenn ein Essstäbchen gerade nicht da liegt, so wartet der Philosoph darauf, dass sein Nachbar es ablegt.

Diskutieren Sie, ob bei dieser Konstellation eine Verklemmung auftreten kann, indem Sie sich auf die o. g. Bedingungen beziehen und zeigen Sie, ob und wie diese Bedingungen im Dining Philosophers Beispiel zutreffen.

d) Die Philosophen aus dem o. g. Beispiel ändern nun nach leidvoller Erfahrung den Ablauf ihrer Sitzungen so, dass der älteste von ihnen immer das linke Stäbchen zuerst nimmt und dann das rechte. Diskutieren Sie in etwa 2-3 Sätzen, welche Auswirkungen dies ggf. auf die Möglichkeit des Auftretens von Verklemmungen hat.

3. Synchronisation

Ein Professor führt Studienberatungen für Studierende durch. Das Vorzimmer des Professors hat Sitzplätze für drei Studierende, die Studienberatung selbst findet immer als Einzelgespräch im Büro des Professors statt. Sind keine beratungsbedürftigen Studierenden anwesend, widmet der Professor sich seiner Forschung. Forscht der Professor gerade, so muss er benachrichtigt werden, wenn ein beratungsbedürftiger Student das Vorzimmer betritt. Wenn im Vorzimmer kein Sitzplatz mehr frei ist, müssen Studierende vor der Tür warten.

Fortsetzung nächste Seite!

Ihnen stehen blockierende (BinarySemaphore) und zählende Semaphore (CountingSemaphore) zur Verfügung, um eine Software zur Synchronisierung der Studienberatung zu implementieren:

```
public interface Semaphore{
   public void wait();
   public void signal();
}

public class BinarySemaphore  implements Semaphore{
    public BinarySemaphore(){...}

}

public class CountingSemaphore implements Semaphore{

   public CountingSemaphore(int N){...}

}
```

Vervollständigen Sie in den folgenden Teilaufgaben a) - c) die Codeschablonen für die Initialisierung, für die Klasse Student und für die Klasse Professor.

An den mit ... gekennzeichneten Stellen können beliebig viele Codezeilen eingefügt werden. Es soll ein möglichst hoher Grad an Parallelität erreicht werden.

a) Initialisierung

Codeschablone für die Initialisierung:

```
// Kontrolliert Zugang zum Vorzimmer des Professors
Semaphore vorzimmer =   ...;

// Der Professor bei der Beratungstätigkeit
Semaphore professor =   ...;

// Die auf den Sitzplätzen wartenden Studierenden
Semaphore studierende =  ...;
```

Fortsetzung nächste Seite!

b) Klasse Student

Codeschablone für die Klasse Student:

```
public class Student extends Thread {
  public void run() {

    ...

    < Student betritt das Vorzimmer des Professors >

    ...
  }
}
```

c) Klasse Professor

Codeschablone für die Klasse Professor:

```
public class Professor extends Thread {
  public void run() {

    ...

    < Professor forscht >

    ...

    < Professor berät einen Studenten >

    ...

  }
}
```

66114

Datenbank- und Betriebssysteme (vertieft)

Frühjahr 2009

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____	**Frühjahr**	**66114**
Kennwort: _____	**2009**	
Arbeitsplatz-Nr.: _____		

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

— Prüfungsaufgaben —

Fach:	**Informatik (vertieft studiert)**
Einzelprüfung:	**Datenbank- und Betriebssysteme**
Anzahl der gestellten Themen (Aufgaben):	4 Aufgaben, von denen 2 gemäß untenstehender Auswahlregel zu bearbeiten sind
Anzahl der Druckseiten dieser Vorlage:	20

Zu den zwei Themenschwerpunkten A (Datenbanksysteme) und B (Betriebssysteme) ist jeweils entweder die Aufgabe 1 oder 2 zu wählen. Auf der Vorderseite des Kopfbogens sind im Feld „Gewähltes Thema: Nr.“ die Nummern der beiden gewählten Aufgaben anzugeben (z. B. A2, B1)!

Bitte wenden!

Themenschwerpunkt A

Datenbanksysteme

AUFGABE 1

1. Modellierung

Ein Pizzadienst möchte sein Bestellsystem mit einer relationalen Datenbank abwickeln:

Die Datenbank enthält allgemein Personen, die sich immer in Kunden und Personal unterteilen lassen. Zu jeder Person gibt es eine Nummer (PNr), den Namen und den Vornamen. Zu Kunden wird auch die Adresse (enthält Ort, PLZ, Straße und Hausnummer) gespeichert, zum Personal der Lohn. Die Mitglieder des Personals können aber auch eigene Bestellungen aufgeben. In so einem Fall werden zusätzlich auch die Kundeninformationen gespeichert.
Eine Bestellung wird immer einem Kunden zugeordnet. Die Bestellung besitzt eine eindeutige Bestellungsnummer (BNr) und sowohl Datum als auch Uhrzeit der Bestellung. Verknüpft mit der Bestellung werden beliebig viele, aber mindestens eine Bestellzeile gespeichert, welche die bestellten Gerichte aufführen. Die Zeilen einer Bestellung werden angefangen bei "1" durchnummeriert. Zu jeder Bestellzeile wird exakt 1 Gericht verknüpft. Der korrespondierende Preis wird in der Bestellzeile gespeichert.
Gerichte besitzen eine eindeutige Nummer (GNr), einen Namen und eine optionale Beschreibung. Gerichte unterteilen sich in die Typen Nudeln, Pizzen und Muscheln. Die Unterteilung geschieht natürlich überschneidungsfrei. Den Pizzen sollen insbesondere die enthaltenen Zutaten zugeordnet werden, welche eigens gespeichert werden. Jede Pizza muss aber mindestens eine Zutat enthalten. Die Zutaten können anhand ihres Namens unterschieden werden.
Es kommt immer wieder vor, dass Sonderwünsche an Speisen bestellt werden. Diese sind dann mit eigener Nummer als Gericht zu speichern. Weiter ist es möglich für Pizzen beliebig viele zusätzliche Zutaten zu ordern. Die Aufzählung dieser Zutaten erfolgt durch Erweiterung der Verknüpfung zwischen Bestellzeile und Gericht.

a) Entwerfen Sie für das beschriebene Szenario ein ER-Diagramm. Bestimmen Sie hierzu:

- die Entity-Typen, die Relationship-Typen und jeweils deren Attribute;
- ein passendes ER-Diagramm;
- die Primärschlüssel der Entity-Typen, welche Sie anschließend in das ER-Diagramm eintragen;
- die Funktionalitäten der Relationship-Typen, welche Sie ebenfalls in das ER-Diagramm eintragen.

Fortsetzung nächste Seite!

b) Überführen Sie das ER-Modell aus Aufgabe a) in ein verfeinertes relationales Modell. Geben Sie hierfür die verallgemeinerten Relationenschemata an.

c) Laut Aufgabenstellung ist es nur bei Pizzen erlaubt, optionale Zusatzzutaten zu bestellen. Das bisherige Relationenmodell gestattet es aber, Zusatzzutaten für jedes Gericht anzugeben. Dies soll in der Datenbank anhand von Integritätsbedingungen verhindert werden.

   i) Welche Arten von Integritätsbedingungen lassen sich allgemein unterscheiden?

   ii) Welche Ansätze bieten relationale Datenbanken zur Berücksichtigung der unterschiedlichen Arten von Integritätsbedingungen? Nennen Sie pro Art aus Teilaufgabe (i) einen möglichen Ansatz mit kurzer Begründung.

Fortsetzung nächste Seite!

2. Normalformen

Gegeben sei folgende Datenbasis:

ANr	Datum	**Zeile**	ProduktID	Hersteller	KNr	Kunde	SB	Kommentar	PreisEuro
1	2.3.2004	1	PC160X	Semins	356	Meierhof 1976	SL3	Interesse an RAID1	900
1	2.3.2004	2	LCD17A	Sitech	356	Meierhof 1976	HH5	-	120
1	2.3.2004	3	THZ99DB	Heifi	356	Meierhof 1976	HH5	Anbindung zu LCD17A	250
2	1.7.2005	1	HIFI2000	Datavolt	235	Hemebach 1982	HH5	Erweiterung für DDSS 8.9	1200
3	5.7.2005	1	SDU23X8	Sitech	56	Craiton 1964	SL3	-	1500
...	...	...	...						

Für jedes Angebot wird eine eindeutige Angebotsnummer (ANr) vergeben und das Datum des Angebots gespeichert. Ein Angebot wird immer für genau einen Kunden angelegt. Zu jedem Kunden werden Name und Geburtsjahr gespeichert. Ein Angebot umfasst mehrere Angebotszeilen (Zeile). Jede Zeile repräsentiert ein angebotenes Produkt, zu dem jeweils auch der Hersteller gespeichert wird. Für jede Angebotszeile wird ein(e) SachbearbeiterIn angegeben, der/die für den jeweiligen Teil des Angebotes zuständig ist. Jede(r) SachbearbeiterIn hat die Möglichkeit einen eigenen Kommentar zur Angebotszeile abzugeben, welcher für die interne Verwendung im Unternehmen gedacht ist. Zusätzlich kann der Preis jeder Angebotszeile einzeln festgelegt werden. Der Preis einer Angebotszeile muss nicht mit dem Preis des angebotenen Produktes übereinstimmen, sondern kann davon abweichen. Der/die SachbearbeiterIn kann den Preis stattdessen nach eigenem Ermessen anpassen.

a) Wo verletzt die Datenbasis bereits die 1. Normalform?

b) Welche Probleme (Anomalien) treten bei dieser Datenbasis auf, die sich durch eine Überführung in die Boyce-Codd-Normalform beheben lassen?

c) Beschreiben Sie obige Anforderungen mit einer Menge von nichttrivialen funktionalen Abhängigkeiten für die Datenbasis.

d) Formen Sie das gegebene Schema in die 3. Normalform um. Geben Sie hierfür zunächst ein Schema in 1. Normalform an, welches Sie anschließend in die 3. Normalform überführen.

Fortsetzung nächste Seite!

3. Relationale Algebra und SQL

Gegeben sei folgendes Schema der relationalen Datenbank eines Transportunternehmens:

$Ladung\{\underline{LNr}, GewichtKG, VolumenM, StreckeKM\}$

$Fahrzeug\{\underline{FzNr}, Hersteller, MaxLastKG, MaxVolumenM, Stillgelegt\}$

$Fahrer\{\underline{FrNr}, Vorname, Nachname, Ort\}$

$Transport\{\underline{LNr}, \underline{TNr}, FzNr, FrNr, VonOrt, VonZeit, NachOrt, NachZeit, LaengeKM\}$

Jede beauftragte Ladung wird mit einer Nummer (LNr) versehen. Neben Gewicht (in Kilogramm) und Volumen (in Kubikmeter) wird auch die Gesamtlänge der Transportstrecke (in Kilometer) einer Ladung eingetragen.
Für den Transport stehen verschiedene Fahrzeuge und Fahrer zur Verfügung. Fahrzeuge besitzen eine Maximalbeschränkung bezüglich Last und Volumen ihrer Ladung. Es ist prinzipiell möglich verschiedene Ladungen in einem Fahrzeug gleichzeitig zu transportieren, so lange die Gesamtkapazität des Fahrzeuges nicht überschritten wird.
Der eigentliche Transport einer Ladung kann in einem Stück bis zum Zielort erfolgen oder in mehrere Teilstrecken zerlegt werden. Wird eine Zerlegung vorgenommen, so können die einzelnen Teilstrecken von unterschiedlichen Fahrzeugen und/oder Fahrern gefahren werden. Die Nummern für die einzelnen Teilstrecken (TNr) werden dabei fortlaufend ab 1 vergeben.

a) Formulieren Sie die Anfragen in den folgenden Teilaufgaben (i - ii) in der Anfragesprache SQL. Zur Vereinfachung dürfen Sie sich beliebige Sichten (Views) definieren.

   i) Geben Sie Vorname und Nachname der fünf Fahrer mit der größten Anzahl gefahrener Kilometer aus.

   ii) Geben Sie gruppiert nach Hersteller die durchschnittlich je Fahrzeug transportierte Last aus.

b) Formulieren Sie in relationaler Algebra die Anfragen in den folgenden Teilaufgaben (iii - iv).

   iii) An welchen Ort wurde die Ladung mit der Nummer "341" als letztes geliefert (Zielort)? Sie können dazu annehmen, dass Start- und Zielort der Ladung verschiedene Namen besitzen.

   iv) Welche Fahrer hatten mindestens eine Fahrt mit Stopp im Ort "Rom"? Geben Sie Vorname, Nachname und Hersteller des verwendeten Fahrzeuges aus.

Fortsetzung nächste Seite!

4. Transaktionen

Gegeben sei folgender, aus elementaren Operationen der Transaktionen T_1, T_2 und T_3 bestehender Schedule:

$$s_{in} = r_2(A)w_3(A)w_2(C)w_1(C)w_3(A)r_1(A)r_3(B)w_2(B)c_2c_3c_1$$

a) Geben Sie den Konfliktgraphen für s_{in} an. Wie kommen die Knoten und Kanten zu Stande?

b) Welche Aussage lässt sich aus dem Konfliktgraphen von Teilaufgabe a) über den zugrunde liegenden Schedule treffen? Begründen Sie Ihre Antwort.

c) Bestimmen Sie einen möglichen Output-Schedule s_{out}, den ein Scheduler erzeugt, der dem strikten 2-Phasen-Sperr-Protokoll folgt. Geben Sie im Output-Schedule insbesondere auch die erzeugten *lock*– und *unlock*–Befehle an.

Aufgabe 2

1. Normalformenlehre

a) Wann ist ein Superschlüssel Schlüsselkandidat?

b) Welche Aussagen sind für funktionale Abhängigkeiten richtig? Begründen Sie Ihre Antwort kurz.

i) $X \rightarrow Y \Rightarrow XZ \rightarrow YZ$

ii) $X \rightarrow Y$ und $Y \rightarrow Z \Rightarrow X \rightarrow Z$

iii) $X \rightarrow Y$ und $Z \rightarrow Y \Rightarrow X \rightarrow Z$ oder $Z \rightarrow X$

iv) $W \rightarrow X$ und $Y \rightarrow Z \Rightarrow WY \rightarrow XZ$

c) Warum ist folgende Relation nicht in dritter Normalform?
`Adresse(Straße, Hausnummer, Postleitzahl, Wohnort)`
Warum werden Adressen dennoch häufig so gespeichert?

d) Wie muss Relation
`R(A, B, C, D, E)`
mit weiteren funktionalen Abhängigkeiten $B \rightarrow D$ und $C \rightarrow E$ zerlegt werden, damit sie in zweiter Normalform vorliegt? Wie muss sie weiter zerlegt werden, so dass sie in dritter Normalform vorliegt?

Fortsetzung nächste Seite!

2. ER-Modellierung und Relationenmodell

a) **ER-Modellierung**
Erstellen Sie das Modell eines fiktiven Krankenhauses in ER-Notation. Wo möglich bzw. sinnvoll, sollen 3-fache Beziehungen und Generalisierung/Spezialisierung verwendet werden. Attribute von Entitäten und Beziehungen sind anzugeben; Schlüsselattribute werden durch Unterstreichen gekennzeichnet. Die Kardinalitäten von Beziehungen und (falls nötig) Rollennamen sollen ins Diagramm aufgenommen werden. Verwenden Sie zur Angabe der Kardinalität die (Min, Max)-Notation. Führen Sie Surrogatschlüssel nur ein, falls es nötig ist.

„Im Krankenhaus"

Zimmer werden durch eine eindeutige Zimmernummer identifiziert. Weiterhin ist noch die Anzahl der Betten im Zimmer angegeben.
Stationäre **Patienten** liegen in maximal einem Zimmer, ambulante Patienten dagegen liegen nicht in einem Zimmer. Zimmer können leer sein oder bis zu vier Patienten beherbergen. Patienten haben einen Vornamen, Nachnamen und ein Geburtsdatum.
Behandlungen werden durch ein Datum und eine Uhrzeit beschrieben. An einer Behandlung nimmt genau ein Patient teil. Jeder Patient wird mind. einer Behandlung unterzogen.
Ärzte werden durch Vorname, Nachname und eine eindeutige Piepsernummer beschrieben.
Fachgebiete werden durch einen Namen beschrieben und durch eine Abkürzung eindeutig identifiziert.
Behandlung, Fachgebiet und Arzt stehen in einer engen Beziehung: Im Rahmen einer Behandlung beherrscht jeder Arzt genau ein Fachgebiet und es gibt für jedes Fachgebiet genau einen Arzt. Natürlich kann ein Arzt ein Fachgebiet nicht nur im Rahmen einer einzigen, sondern bei mehreren Behandlungen vertreten.

b) **Relationenmodell**
Ausgehend von der ER-Darstellung ist ein Relationenschema in dritter Normalform (3. NF) zu entwerfen. **Primärschlüssel werden dabei durch Unterstreichen, Fremdschlüssel durch Nennung der referenzierten Relation in eckigen Klammern hinter dem Attributsnamen kenntlich gemacht, z.B.:**
`Haus (Straße, OrtId[Ort])`
`Ort (OrtId, PLZ, Name)`
Das Attribut `OrtId` der Relation `Haus` verweist als Fremdschlüssel auf das Attribut `OrtId` der Relation `Ort`.

Fortsetzung nächste Seite!

3. SQL

Bitte beachten Sie: Primärschlüssel werden dabei durch Unterstreichen, Fremdschlüssel durch Nennung der referenzierten Relation in eckigen Klammern hinter dem Attributsnamen kenntlich gemacht.

Beispiel:
`Haus (Straße, OrtId[Ort])`
`Ort (OrtId, PLZ, Name)`

Interpretation: Das Attribut `OrtId` der Relation `Haus` verweist als Fremdschlüssel auf das Attribut `OrtId` der Relation `Ort`.

a) **Szenario 1: „Musikverwaltung“**
Eine Musikliebhaberin verwendet folgendes einfache Datenbankschema für die Verwaltung ihrer CD-Sammlung:

`Künstler (KId, Name, Musikrichtung, Webseite)`
`CD (CDId, Name, KId[Künstler], Spielzeit, VöD)`
`Track (TId, CDId[CD], Nummer, Name, Spielzeit)`

Jeder Künstler veröffentlicht CDs, die wiederum aus mehreren Tracks bestehen können. `VöD` in der Tabelle `CD` bezeichnet das Veröffentlichungsdatum. Primärschlüssel sind unterstrichen. `KId` in `CD` ist Fremdschlüssel zu `KId` in `Künstler`, `CDId` in `Track` ist Fremdschlüssel zu `CDId` in `CD`.

Formulieren Sie bitte folgende Datenbankoperation in SQL:

i) Geben Sie die Namen und Spielzeiten aller Tracks der CD mit der CDId „42“ aus!

ii) Geben Sie die Namen der CDs mit der längsten Spielzeit aus!

iii) Geben Sie eine Liste der Namen aller CDs des Künstlers „Robbie Williams“, aufsteigend sortiert nach VöD, aus!

iv) Geben Sie die durchschnittliche Spielzeit aller Tracks der CDs des Künstlers „Sting“ aus!

Die Musikliebhaberin möchte in Zukunft neben CDs auch andere Medientypen, wie z.B. DVDs oder Kassetten verwalten.

v) Beschreiben Sie in jeweils max. 3 Sätzen, zwei grundlegend unterschiedliche Methoden zur Erweiterung des Datenbankschemas um die zusätzlichen Medientypen.

vi) Geben Sie für jede der beiden genannten Alternativen genau einen Vor- und einen Nachteil an!

vii) Geben Sie jeweils komplett die beiden alternativen Datenbankschemata nach der Berücksichtigung der Medientypen an!

Fortsetzung nächste Seite!

b) **Szenario 2: „Bei der Feuerwehr“**

Die Kreisfeuerwehr Eggolshausen wird unter Obereinsatzleiter Mayerle modernisiert. Die Rettungskräfte haben zur Einsatzverwaltung eine eigene kleine Datenbank entwickelt. Diese ist schon seit einigen Wochen in Betrieb und muss derzeit noch per Hand mit SQL-Befehlen gepflegt werden.

```
Anschrift (AID, Straße, Nr, PLZ, Stockwerk)
Feuerwehrmann (PID, Name, Vorname, EID[Einsatzfahrzeug])
Einsatzfahrzeug (EID, Kennzeichen, Plätze, Ausrüstung)
Einsatz (AID[Anschrift], EID[Einsatzfahrzeug], Datum, Ursache)
```

Es existieren folgende Fremdschlüsselbeziehungen: Das Attribut `AID` der Relation `Einsatz` bezieht sich auf das Attribut `AID` der Relation `Anschrift`. Das Attribut `EID` der Relation `Einsatz` bezieht sich auf das Attribut `EID` der Relation `Einsatzfahrzeug`. Die Attribute `AID` und `Nr` der Relation `Anschrift`, `PID` der Relation `Feuerwehrmann` und `EID` der Relation `Einsatzfahrzeug` sind Ganzzahlen. Bei dem Attribut `Datum` der Relation `Einsatz` handelt es sich um ein Datum. Das Attribut `PLZ` der Relation `Anschrift` ist eine fünfstellige Ganzzahl. Die Attribute `Kennzeichen` und `Ausrüstung` der Relation `Einsatzfahrzeug` werden als Text gespeichert. Die Datentypen der restlichen Attribute sind dem Kontext zu entnehmen.

Geben Sie SQL-Anweisungen für folgende Problemstellungen an:

i) Erzeugung des beschriebenen Relationenschemas mit Tabellen, Primary Key Constraints und Foreign Key Constraints!

ii) Einem Haus in der Heinestr. 4 in 99099 Eggolshausen wurde ein neues drittes Stockwerk aufgesetzt. Nun soll das neue Stockwerk eingetragen werden.

iii) Nachdem der Bungalow in der Musterstr. 3 in 99099 Eggolshausen bereits mehrfach gebrannt hat und auch Löscheinsätze gefahren wurden, wird er jetzt abgerissen. Löschen Sie den Bungalow aus der Tabelle `Anschrift` und berücksichtigen Sie auch ggf. davon abhängige Datensätze.

iv) Ein Einsatz wurde gefahren: Es brannte erstmalig in der Weidenstr. 8 im 4. Stock in 99099 Eggolshausen am 15.07.2006. Das Löschfahrzeug mit dem Kennzeichen EGG-UM 0815 und der `EID` 7 wurde zum Einsatz beordert. Tragen Sie den Einsatz und ggf. weitere notwendige Datensätze in die Datenbank ein.

v) Geben Sie eine Liste aller Einsatzursachen, sortiert nach Häufigkeit der Einsatzursache aus.

Themenschwerpunkt B

Betriebssysteme

AUFGABE 1

1. Prozess-Scheduling

a) Skizzieren Sie in einem Diagramm, welche Zustände Prozesse in einem System annehmen können und welche Zustandsübergänge existieren. Beschriften Sie die Knoten des Diagramms mit den Zustandsbezeichnungen und die Kanten des Diagramms mit den Ereignissen, die den entsprechenden Zustandsübergang auslösen.

b) Betrachten Sie die folgenden Prozesse:

Prozess	Ankunftszeit	Benötigte Rechenzeit
P0	0	8
P1	1	6
P2	2	3
P3	3	9
P4	4	4

Führen Sie das Scheduling für dieses Beispiel mit den Algorithmen *First Come First Served* und *Preemptive Shortest Job First* (auch bekannt als "*Shortest Remaining Time First*") durch.

Stellen Sie den Gesamtablauf je Algorithmus tabellarisch dar, indem Sie für jeden Zeitpunkt den jeweils aktiven Prozess, die getroffene Scheduling-Entscheidung sowie den Zustand der "Ready-Queue" *nach* der Scheduling-Entscheidung angeben. Sie dürfen Zeilen auslassen, in denen sich im Vergleich zur vorigen Zeile nichts ändert.

Gehen Sie dabei von folgender Annahme aus: Sollte das Scheduling-Kriterium keine eindeutige Scheduling-Entscheidung zwischen zwei oder mehreren Prozessen zulassen, so wird von diesen der Prozess mit der *frühesten Ankunfszeit* bevorzugt.

c) Berechnen Sie für obiges Beispiel und NUR für *Preemptive Shortest Job First* die durchschnittliche Wartezeit und den Durchsatz. Geben Sie die Durchlaufzeit von Prozess P1 an.

Fortsetzung nächste Seite!

2. Verklemmungen

a) Geben Sie (ohne weitere Erläuterung) die Bedingungen an, die erfüllt sein müssen, damit eine Verklemmung auftreten kann.

b) Eine einfache Möglichkeit zur Verhinderung von Verklemmungen oder *Deadlocks* ist der Einsatz des Zeitstempelverfahrens *wound-wait*, welches nach folgendem Prinzip arbeitet:

- Jedem Prozess P wird vom Betriebssystem ein eindeutiger Zeitstempel $TS(P)$ zugeordnet, so dass $TS(P) \neq TS(Q)$ für $P \neq Q$.
- Die Zeitstempel werden streng monoton wachsend vergeben, so dass jeder Prozess einen kleineren Zeitstempel besitzt als alle jüngeren Prozesse. Für zwei Prozesse P und Q gilt:
  P ist *älter* als Q genau dann, wenn $TS(P) < TS(Q)$.
  P ist *jünger* als Q genau dann, wenn $TS(P) > TS(Q)$.
- Wenn ein Prozess P ein Betriebsmittel anfordert, das erst von einem Prozess Q freigegeben werden müsste, wird nach folgender Strategie verfahren:
  - Ist P älter als Q, wird Q abgebrochen, so dass P weiterlaufen kann.
  - Ist P jünger als Q, wartet P auf die Freigabe des Betriebsmittels.

i) Wenden Sie das wound-wait Verfahren auf folgende Situation an:

Gegeben sind zwei Prozesse P_1 und P_2 mit Zeitstempel $TS(P_1) = 11$ und $TS(P_2) = 12$. P_1 hält Betriebsmittel A und B. P_2 hält Betriebsmittel C. Zum Zeitpunkt 20 fordert P_2 Betriebsmittel A an, das P_1 hält. Zum Zeitpunkt 21 fordert P_1 Betriebsmittel C an, das P_2 hält.

Beschreiben Sie kurz, wie sich das System zu den Zeitpunkten 20 und 21 verhält, wenn das wound-wait Verfahrens angewendet wird.

ii) Begründen Sie, warum beim wound-wait Verfahren kein Deadlock zwischen *zwei* verschiedenen Prozessen auftreten kann.

iii) Beweisen Sie für eine *beliebige* endliche Menge von n Prozessen $\{P_1, ..., P_n\}$, dass beim Einsatz des wound-wait Verfahrens kein Deadlock auftreten kann. Nehmen Sie dazu an, dass sich k Prozesse, $k \in \{2, ..., n\}$, in einem Wartezyklus befinden. Zeigen Sie im Anschluss, dass diese Situation unter Anwendung des wound-wait Verfahren nicht auftreten kann.

Fortsetzung nächste Seite!

3. Synchronisation

Synchronisieren Sie mit Hilfe von Semaphoren den Parkvorgang von Fahrzeugen, die nebenläufig in einem Parkhaus ankommen und abfahren. Das Parkhaus besitzt genau eine Einfahrt, genau eine Ausfahrt, zwei Kassenautomaten und 100 Stellplätze.

Bei der Einfahrt in das Parkhaus fordert der Fahrzeugführer ein Parkticket an. Sobald Stellplätze frei sind, öffnet sich die Einfahrtsschranke und das Fahrzeug parkt an einem freien Stellplatz.

Beim Verlassen des Parkhauses wird der Parkpreis an einem der beiden Kassenautomaten entrichtet und anschließend das Fahrzeug durch die Ausfahrt bewegt.

Der Parkvorgang soll durch die Prozesse `Ankunft` und `Abfahrt` simuliert werden, welche die kontinuierliche nebenläufige Ankunft bzw. Abfahrt von Fahrzeugen aus dem Parkhaus abbilden. Zur Synchronisation stehen Ihnen binäre (`BinarySemaphore`) und zählende Semaphore (`CountingSemaphore`) zur Verfügung.

```
public interface Semaphore{
  public void wait();
  public void signal();
}

public class BinarySemaphore implements Semaphore{
  public BinarySemaphore(){...}
  ...
}

public class CountingSemaphore implements Semaphore{
  public CountingSemaphore(int N){...}
  ...
}
```

Vervollständigen Sie die Codeschablonen auf den nächsten Seiten, indem Sie die Kommentare der Form `/* Ergänzung` $N.M$ `*/` durch Code für die Initialisierung der verwendeten Semaphore bzw. für die Synchronisation von Threads der Klassen `Ankunft` und `Abfahrt` ersetzen. Sie können an den mit `/* Ergänzung` $N.M$ `*/` gekennzeichneten Stellen beliebig viele Codezeilen oder auch keine einfügen. Schreiben Sie Ihre Lösung in folgender Form auf:

Ergänzung	Code
1.1	`new BinarySemaphore()`
1.2	...
1.3	...
...	...
3.3	...

Fortsetzung nächste Seite!

Ihre Lösung soll einen möglichst hohen Grad an Parallelität ermöglichen. Sorgen Sie dafür, dass nur dann Fahrzeuge das Parkhaus verlassen können, wenn sich welche im Parkhaus befinden. Zu Beginn der Simulation sollen 20 der 100 Stellplätze des Parkhauses bereits belegt sein.

Codeschablone für die Initialisierung:

```
// Einfahrt des Parkhauses
Semaphore einfahrt = /* Ergänzung 1.1 */;

// Ausfahrt des Parkhauses
Semaphore ausfahrt = /* Ergänzung 1.2 */;

// Kassenautomaten
Semaphore kassenautomat = /* Ergänzung 1.3 */;

// freie Stellplaetze des Parkhauses
Semaphore freierStellplatz = /* Ergänzung 1.4 */;

// belegte Stellplaetze des Parkhauses
Semaphore belegterStellplatz = /* Ergänzung 1.5 */;
```

Codeschablone für die Klasse `Ankunft`:

```
public class Ankunft extends Thread {
  public void run() {

    while(true) {

    /* Ergänzung 2.1 */

    // Parkticket an der Parkhauseinfahrt entnehmen

    /* Ergänzung 2.2 */

    // In das Parkhaus einfahren

    /* Ergänzung 2.3 */

    // Parken

    /* Ergänzung 2.4 */

    }
  }
}
```

Fortsetzung nächste Seite!

Codeschablone für die Klasse `Abfahrt` (10 Punkte):

```
public class Abfahrt extends Thread {
  public void run() {

    while(true) {

    /* Ergänzung 3.1 */

    // Parkpreis am Automaten entrichten

    /* Ergänzung 3.2 */

    // Aus dem Parkhaus ausfahren

    /* Ergänzung 3.3 */
    }
  }
}
```

AUFGABE 2

1. Prozesse

a) Welche vier Prozesszustände gibt es? Erklären Sie die Prozesszustände jeweils in einem Satz!

b) Die zwei folgenden Programme A und B laufen parallel auf einer Einprozessormaschine mit RoundRobin-Scheduling-Strategie. Gehen Sie davon aus, dass die Abarbeitung einer nummerierten Programmzeile immer genau eine ganze Zeitscheibe in Anspruch nimmt. Nehmen Sie die Eingabe `a.readData()` als blockierend an. Bei Deblockierung wird der Befehl wiederholt. Gehen Sie außerdem davon aus, dass zu Beginn die Variable `count=0` ist! Etwa in der Mitte von Zeitscheibe 6 kommt einmalig ein Datum für A.

```
Program A:
while (true) {                (A1)
  a.readData();               (A2)
  array[count]=a;             (A3)
  count=count+1;              (A4)
}
```

```
Program B:
while (true) {                (B1)
  if (count>0) {              (B2)
    count=count-1;            (B3)
    array[count].print();     (B4)
  }
}
```

Schreiben Sie die Abarbeitungsreihenfolge als Folge von Programmzeilennummern (A1, B1, ...) beginnend mit Programm A in Zeitscheibe 1 bis zur Zeitscheibe 20 nieder!

Muster:

Zeit	***1***	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***	***11***	***12***	***13***	***14***	***15***	***16***	***17***	***18***	***19***	***20***
	A1	B1	...																	

Was sind die Prozesszustände der beiden Prozesse jeweils kurz vor Ende der Zeitscheiben 4, 7 und 10?

Fortsetzung nächste Seite!

2. Schedulingstrategien

a) Definieren Sie *Fairness* für eine Schedulingstrategie aus Sicht eines Benutzers! Was ist *nicht-preemptives Scheduling*? Inwieweit ist nicht-preemptives Scheduling fair?

b) Gegeben sind folgende Prozesse mit ihren Bereitzeitpunkten und ihrem Rechenzeitbedarf. Führen Sie für eine Einprozessormaschine das nicht-preemptive Scheduling gemäß Shortest Processing Time First (SPF) und das preemptive Scheduling gemäß Shortest Remaining Time First (SRTF) mit Zeitscheibenlänge 1 durch.

Prozess	***Bereitzeit***	***Rechenzeit***
A	0	5
B	0	4
C	1	2
D	4	1
E	6	2

Zeichnen Sie die entsprechenden Gantt-Diagramme und berechnen Sie jeweils die mittlere Verweilzeit!

Muster:

	1	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***	***11***	***12***	***13***	***Mittlere Verweilzeit***
SPF	...													
SRTF	...													

Welche Gefahr bergen diese beiden Strategien SPF und SRTF generell in Bezug auf Prozesse mit hohem Rechenzeitbedarf?

Nennen Sie je eine preemptive und eine nicht-preemptive Schedulingstrategie, die diesen Nachteil grundsätzlich nicht haben! Begründen Sie Ihre Antworten!

c) Gegeben sind folgende periodische, unterbrechbare Prozesse A, B und C auf einer Einprozessormaschine. A benötigt alle 15 Zeiteinheiten eine Zeiteinheit Rechenzeit, B benötigt alle 5 Zeiteinheiten 3 Zeiteinheiten Rechenzeit und C benötigt alle 6 Zeiteinheiten 2 Zeiteinheiten Rechenzeit.

Berechnen Sie die Prozessorauslastung!

Interpretieren Sie die periodischen Prozesse jeweils als Folge von einzelnen Prozessen und führen Sie das Scheduling gemäß preemptiven Shortest Remaining Time First mit Zeitscheibenlänge 1 durch und zeichnen Sie das entsprechende Gantt-Diagramm. Bei gleichen Fristen wird A gegenüber B und C bevorzugt und B wird gegenüber C bevorzugt.

Muster:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
A																														
B																														
C																														

Fortsetzung nächste Seite!

3. Synchronisation

a) Nennen Sie die vier notwendigen und hinreichenden Bedingungen für Verklemmungen und jeweils eine passende Maßnahme um Verklemmungen auszuschließen!

b) Betrachten Sie die folgenden Prozeduren `P(s)` und `V(s)`!

```
void P(int s)
{
   while (s<=0) noop;
   s=s-1;
}
```

```
void V(int s)
{
   s=s+1
}
```

Welche Grundannahme ist gefordert, damit diese beiden Prozeduren zu einer Implementierung der Semaphoroperationen `P(s)` und `V(s)` für ein Semaphor `s` werden und wie kann man diese Annahme erfüllen?

Welche zwei Nachteile hat die hier gezeigte Implementierungsvariante für Semaphore?

Geben Sie eine Implementierungsvariante für `P(s)` und `V(s)` ohne diese Nachteile an! Geben Sie explizit an, welche Datenstrukturen für `s` benötigt werden und wozu!

Fortsetzung nächste Seite!

4. Virtueller Speicher

a) Wie viel Speicher wird bei einer Seitengröße von *p* Bytes in einem System „verschwendet", in dem jeder Prozess durchschnittlich *s* Bytes belegt und ein Eintrag in der Seitentabelle *e* Bytes Platz verbraucht? Begründen Sie Ihre Antwort!

b) Gegeben sei ein Rechner mit 5 Bit physikalischem und virtuellem Adressraum und 5 Bit breiten Worten. Der Inhalt des physikalischen Speichers ist unten angegeben. Es gelten folgenden Annahmen:

- Die Seiten sind jeweils 4 Worte groß.
- Die Basisadresse der Seitentabelle ist 00000.
- Ein Seitentabelleneintrag besteht aus 5 Bit, die ersten (=höherwertigsten) 3 Bits geben die physikalische Seitenrahmennummer, das 4. Bit sei das present- bzw. mapped-Bit und das 5. Bit sei hier nicht von Bedeutung.
- Bei der zweistufigen Adressierung werden 2 Bit zur Indizierung der Seitentabellen der zweiten Stufe verwendet.

Adresse	***Inhalt***	***Adresse***	***Inhalt***	***Adresse***	***Inhalt***	***Adresse***	***Inhalt***
00000	10111	01000	01110	10000	11111	11000	00110
00001	10011	01001	00011	10001	11010	11001	01010
00010	11011	01010	10010	10010	01011	11010	00010
00011	11101	01011	00000	10011	01101	11011	10001
00100	01000	01100	11000	10100	00101	11100	10100
00101	00100	01101	10000	10101	11100	11101	01001
00110	01111	01110	11110	10110	00111	11110	10110
00111	11001	01111	01100	10111	10101	11111	00001

Welche Daten stehen an der Adresse `11001`, wenn Sie diese Adresse als physikalische Adressen, als 1-stufige und als 2-stufige virtuelle Adresse verwenden?

Was passiert wenn Sie auf die einstufige virtuelle Adresse `11101` zugreifen?

Fortsetzung nächste Seite!

5. Seitenverwaltung

a) Beschreiben Sie der Reihe nach, welche Schritte zur Behebung eines Seitenfehlers nötig sind!

b) Zu jedem Zeitpunkt der „Abarbeitung" einer Referenzkette durch eine hier nicht näher spezifizierte Verdrängungsstrategie seien die Seiten in einem als Keller organisierten Speicher gegeben. Bestimmen Sie die entsprechende Distanzkette!

1	A	A	B	C	B	D	E	F	G	D	H	H	A	H	B	F	H	D	A	F	F	E	B	E
2			A	B	C	B	D	E	F	G	D	D	H	A	H	B	F	H	E	A	A	F	E	B
3				A	A	C	B	D	E	F	G	G	D	D	A	H	B	F	H	E	E	A	F	F
4						A	C	B	D	E	F	F	G	G	D	A	A	B	F	H	H	H	A	A
5							A	C	B	B	E	E	F	F	G	D	D	A	B	B	B	B	H	H
6								A	C	C	B	B	E	E	F	G	G	G	G	G	G	G	G	G
7									A	A	C	C	B	B	E	E	E	E	E	E	E	E	E	E
8											A	A	C	C	C	C	C	C	C	C	C	C	C	C

66114

Datenbank- und Betriebssysteme (vertieft)

Herbst 2009

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____	**Herbst 2009**	**66114**
Kennwort: _____		
Arbeitsplatz-Nr.: _____		

Fach: **Informatik (vertieft studiert)**

Einzelprüfung: **Datenbank- und Betriebssysteme**

Anzahl der gestellten Themen (Aufgaben): **4 Aufgaben, von denen zwei gemäß untenstehender Auswahlregeln zu bearbeiten sind!**

Anzahl der Druckseiten dieser Vorlage: **14**

Zu den zwei Themenschwerpunkten A (Datenbanksysteme) und B (Betriebssysteme) ist jeweils entweder die Aufgabe 1 oder 2 zu wählen. Auf der Vorderseite des Kopfbogens sind im Feld „gewähltes Thema: Nr." die Nummern der beiden gewählten Aufgaben anzugeben (z. B. A2, B1)!

Bitte wenden!

Themenschwerpunkt A
(Datenbanksysteme)

Aufgabe 1

1. ER-Modellierung

Sie wurden beauftragt, eine Internetanwendung zu entwickeln, die deren Nutzern folgende Möglichkeiten geben soll:

1. Ein Benutzer soll sich am Portal anmelden können. Der Zugang ist passwortgeschützt. Vorname und Name sowie die Emailadresse sollen im Profil gespeichert, aber veränderbar sein.
2. Ein Benutzer kann die Utensilien seiner Küche, also Töpfe, Pfannen, Kochlöffel usw. in seinem Profil speichern.
3. Ein Benutzer soll Gerichte finden können. Jedes Gericht soll über ein Foto, eine Geschichte und eine Anleitung, dem Kochrezept, verfügen. Weiterhin soll bekannt sein, welche Utensilien zur Zubereitung benötigt werden. So können auf Wunsch auch nur die Gerichte angezeigt werden, die mit dem Besitz des Kochfreundes zubereitet werden können.
4. Bei Wahl eines Gerichtes soll auf Wunsch eine Zutatenliste ausgegeben werden können.

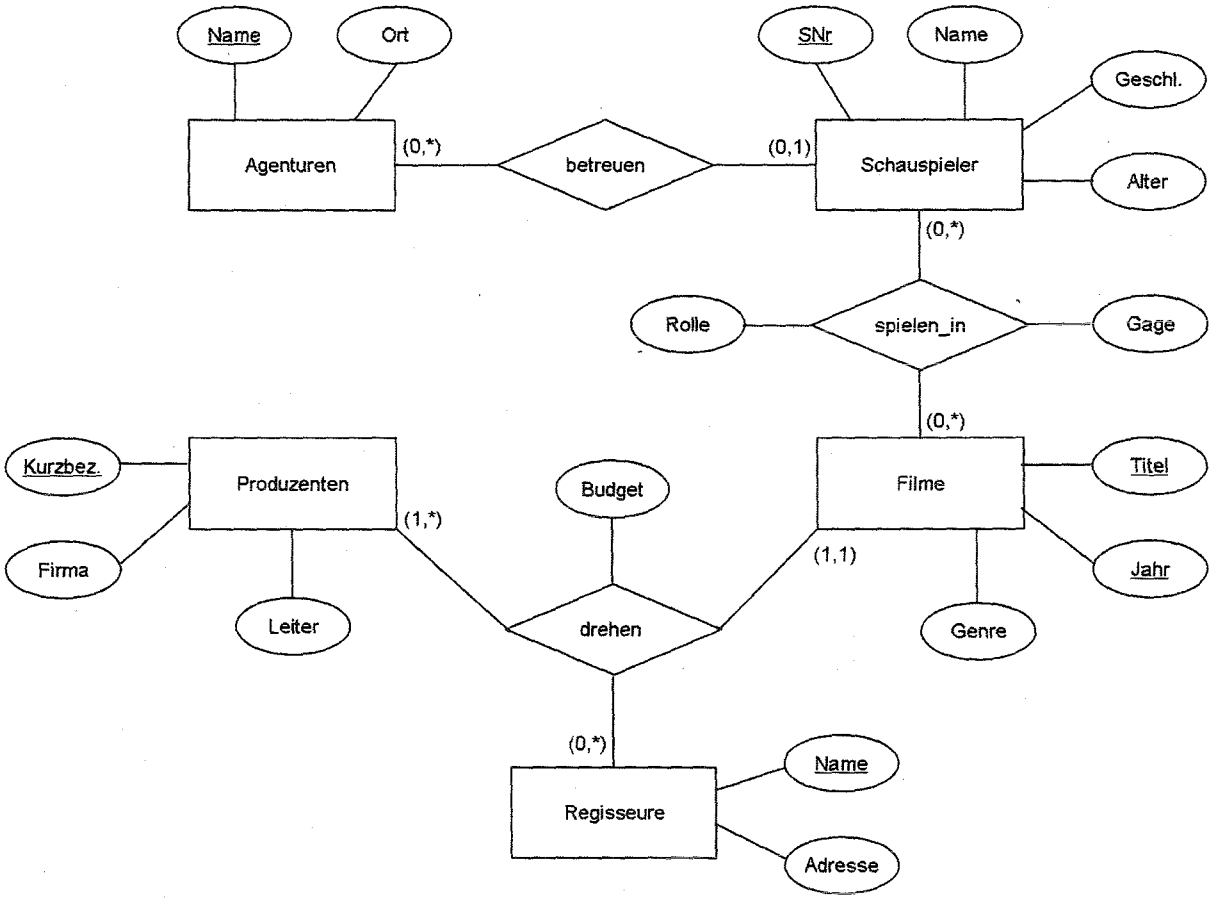
Erstellen Sie ein minimales E/R-Diagramm für die zugrunde liegende Datenbank eines fiktiven Webportals für Kochfreunde!

Attribute von Beziehungen und Entitäten sind anzugeben. Kennzeichnen Sie die Schlüsselattribute durch Unterstreichen. Die Kardinalitäten von Beziehungen sind anzugeben. Verwenden Sie zur Angabe der Kardinalität die (Min, Max)-Notation.
Modellieren Sie die Zutaten als Entität, die Menge und Maß (Prise, Gramm, Liter,...) enthält.

2. Relationenmodell

Überführen Sie folgendes ER-Diagramm in ein Relationenschema in dritter Normalform (3. NF)! Wie gewohnt werden dabei Primärschlüssel durch Unterstreichen, Fremdschlüssel durch Überstreichen kenntlich gemacht!

Fortsetzung nächste Seite!



3. SQL

Gegeben sei folgende Datenbank, die das Ausleihwesen einer Bibliothek unterstützt:

LESER(LsNr, Name, Wohnort, GebDat)
BUCH(ISBN, Titel, Seitenzahl, Verlag, Erscheinungsjahr, Anzahl-Exemplare)
VERLAG(Verlag, Verlagsort, Adresse)
EXEMPLAR(ISBN, ExpNr, InventarNr, Standort)
AUSLEIHE(ISBN, ExpNr, LsNr, Ausleihdatum)

Datenänderungen:

a) Geben Sie eine SQL-Anweisung an, um einen neuen Datensatz für den Leser Bernd Bücherwurm aus München anzulegen, der sein Geburtsdatum vergessen hat. Die zugewiesene Leser-Nummer (LsNr) soll um 1 größer als die bisher größte sein.

b) Formulieren Sie folgenden SQL-Code in natürlicher Sprache:

```
DELETE FROM Ausleihe
WHERE Ausleihdatum < 16.8.2008 AND LsNr = (SELECT LsNr FROM Leser WHERE
                                           Name='Bernd Bücherwurm')
```

Bemerkung: Es gibt nur einen Leser mit diesem Namen.

Fortsetzung nächste Seite!

Anfragen:

c) Welche Leser (Name) haben ein Buch mit dem Titel „Der Zauberberg“ ausgeliehen?

d) Welche Leser (LsNr) haben mehr als ein Exemplar desselben Buchs ausgeliehen?

e) An wie vielen Orten sind die Verlage ansässig, von denen die Bibliothek Bücher hat?

f) Wie viele Buchexemplare gibt es in der Bibliothek von Büchern, die zwischen 1978 und 1983 (Jahreszahlen jeweils inklusive) erschienen sind und die im Titel die Wörter „Informatik“ oder „Mathematik“ enthalten, und wie hoch ist ihre durchschnittliche Seitenzahl?

Fortsetzung nächste Seite!

Aufgabe 2

1 Datenbanksysteme und Modelle

Die Entwicklung der ersten Datenbanksysteme begann in den frühen 60er Jahren. Seitdem haben sie die Datenverwaltung auf der Grundlage einfacher Betriebssystemdateien auf vielen Gebieten abgelöst.

a) Definieren Sie den Begriff Datenbanksystem und grenzen Sie die Funktionalitäten eines Datenbanksystems von denen eines Dateisystems ab!

b) Nennen Sie die Phasen des Datenbankentwurfsprozesses und charakterisieren Sie diese kurz!

c) Entity-Relationship-Modell - Erläutern Sie die nachfolgenden Konzepte eines ER-Modells!
- Entität
- Relationship/Beziehungstyp
- Attribut
- Kardinalität

2 Konsistenz in Datenbanksystemen

Transaktionen sind ein wichtiges Konzept in Datenbanksystemen. Sie führen stets eine Datenbank von einem konsistenten Zustand in einen konsistenten Zustand über.

a) Was bedeutet Konsistenz im Zusammenhang mit Datenbanken?

b) Was versteht man unter dem ACID-Prinzip? Welche Konsequenzen im Verhalten des Datenbanksystems hat das ACID-Prinzip für den Anwender und Entwickler?

c) Darf innerhalb einer Transaktion ein inkonsistenter Zustand vorliegen? Begründen Sie die Antwort!

d) Welche Arten von Integritätsbedingungen können Sie mit SQL darstellen? Geben Sie Beispiele!

e) Was ist ein Trigger und wie funktioniert er?

Fortsetzung nächste Seite!

3 ER-Modellierung und Relationenmodell

3.1 ER-Modellierung

Eine Stadtbücherei möchte ein neues Ausleihsystem entwickeln. Erstellen Sie ein Modell der beschriebenen Stadtbücherei in ER-Notation. Wo möglich bzw. sinnvoll, sollen ternäre Beziehungen, schwache Entitätstypen und Generalisierung/Spezialisierung verwendet werden. Attribute von Entitäten und Beziehungen sind anzugeben; Schlüsselattribute werden durch Unterstreichen gekennzeichnet. Die Kardinalitäten von Beziehungen und - falls nötig – Rollennamen sollen ins Diagramm aufgenommen werden.

Personen und Einrichtungen der Bibliothek werden wie folgt beschrieben:

- Für **Personen** werden folgende Informationen aufgezeichnet: Ausweisnummer, Vorname, Nachname, Adresse (Land, PLZ, Ort, Straße, Hausnummer), Geburtstag und eine Liste von Telefonnummern.
- **Sachbearbeiter** sind eine Gruppe von Personen, die durch zusätzliche Eigenschaften charakterisiert sind: Personalnummer, Fachrichtung, Zimmernummer, Gebäude, Anstellungsdatum und Gehalt.
- Einige Mitarbeiter sind **Führungskräfte** (Sachgebietsleiter, Spezialbibliotheksleiter oder Büchereileiter), für die zu den Angaben der Sachbearbeiter noch weitere Charakteristika hinzukommen: Sachgebiet (Musikbibliothek, alte Schriften etc.) und Gehaltsstufe (A16, B1 etc.)
- Darüber hinaus gibt es eine weitere Personengruppe, die **Bürger**. Zusätzlich zu den Eigenschaften der Personen besitzen sie folgende Attribute: Bibliotheksausweisenummer, Beruf (Student, Rentner etc.) und Datum der Ausweiserstellung.
- Jedes Sachgebiet wird durch seinen Namen, der übergeordneten Bibliotheksabteilung, zu der es gehört, und die Anzahl seiner Planstellen beschrieben.
- Die Bibliothek hat **Medien**, die sie verleiht. Jedes Medium hat eine eindeutige Nummer, aus der das zugeordnete Sachgebiet abgeleitet werden kann, Tag der Anschaffung und im Fall von Büchern eine ISBN-Nummer. In den übrigen Fällen wird das Medium durch Hersteller und Produktname beschrieben. Büchermedien sind Büchern über ISBN-Nummern zuzuordnen.
- **Bücher** werden durch ISBN-Nummer, Titel, Jahr der Veröffentlichung, Auflage, mehrere Suchbegriffe, Seitenzahl und eine Autorenliste beschrieben.
- **Buchverlage** sollen durch Namen und Ort repräsentiert werden.
- **Autoren** werden lediglich durch Vor- und Nachname identifiziert. Eine weitere Identifikation braucht nicht zu erfolgen.
- Mitarbeiter und Bürger können Bücher bei der Bibliothek ausleihen. Die Ausleihfrist beträgt jeweils 14 Tage. Bei einer Überschreitung dieser Frist werden Gebühren (5 EUR) erhoben. Überschreiten die kumulierten Gebühren, die eine Person zu entrichten hat, eine vorgegebene Grenze (20 EUR), so kann die betreffende Person keine weiteren Bücher ausleihen.
- Einige Führungskräfte sind Sachgebietsleiter; für jedes Sachgebiet gibt es aber höchstens einen Leiter.
- Bücher werden von Verlagen herausgegeben und von Autoren geschrieben.
- Medien sind immer einem Sachgebiet zugeordnet. Im Fall von Büchern wird ein Medium auch immer einem Buch zugeordnet.

3.2 Relationenmodell

Entwerfen Sie ausgehend von der ER-Darstellung ein Relationenschema in dritter Normalform (3. NF). Primärschlüssel werden durch Unterstreichen, Fremdschlüssel durch die Nennung der entsprechenden Relation kenntlich gemacht.

Beispiel:
Abteilung(<u>Abteilungsnummer</u>, Abteilungsname)
Mitarbeiter(<u>Mitarbeiternummer</u>, Abteilung[Abteilung])

Fortsetzung nächste Seite!

4 SQL

Gegeben ist folgende Datenbank:

SQL> select * from buecher;

BUCH_NR	TITEL	AUTOR
1	Oracle9i	G.Stuerner
2	Informix 6.0/7.1	D.Petkovic
3	SYBASE	D.McGoveran

SQL> select * from leser;

LESER_NR	NAME
1	G. Schroeder
2	H. Kohl
3	A. Merkel

SQL> select * from ausleihe;

BUCH_NR	LESER_NR	EIN_AUS	DATUM
3	3	AUS	09. Jun 08
2	2	AUS	11. Jun 08
1	1	AUS	12. Jun 08
3	3	EIN	13. Jun 08
1	1	EIN	15. Jun 08
1	2	AUS	17. Jun 08
2	3	EIN	27. Jun 08
2	2	EIN	18. Jun 08
2	3	AUS	19. Jun 08
1	2	EIN	22. Jun 08
2	3	EIN	07. Jun 08
3	1	EIN	30. Jun 08
3	1	AUS	19. Jun 95
3	1	AUS	01. Jul 95
3	1	EIN	03. Jul 95

Fortsetzung nächste Seite!

In `buecher` ist `BUCH_NR` Primärschlüssel. In `leser` ist `LESER_NR` Primärschlüssel. In `ausleihe` besteht der Primärschlüssel aus `BUCH_NR` und `LESER_NR`. `EIN_AUS` beschreibt, ob ein Buch entliehen wurde („AUS") oder zurückgebracht wurde („EIN").

4.1 Erstellung der Tabellen

a) Schreiben Sie die nötigen SQL-Befehle, um die oben beschriebenen Tabellen zu erstellen.

b) Wie müssten die Tabellen angelegt werden, damit dasselbe Buch vom selben Leser mehr als einmal ausgeliehen werden kann?

c) Wie kann sichergestellt werden, dass das Rückgabedatum stets größer ist als das Entleihdatum und dass nur Bücher entliehen werden können, die auch zurückgegeben wurden (beim Einfügen von Rows in Table `ausleihe`)?

4.2 SQL-Anfragen

Geben Sie zu folgenden Anfragen das SQL-Statement an:

a) Welche Bücher wurden noch nie entliehen?

b) Welche Bücher sind im Moment entliehen?

c) Wie viele Bücher hat A. Merkel im Moment entliehen und noch nicht zurückgebracht?

d) In einer Statistik sollen alle abgeschlossenen Ausleihvorgänge erfasst werden (TITEL, NAME, AUSGANG, EINGANG, DAUER).

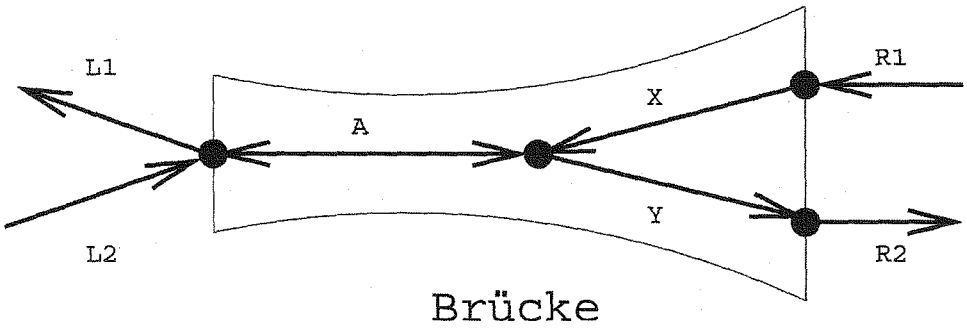
e) In einer weiteren Statistik soll die durchschnittliche Ausleihzeit pro Buch bestimmt werden (TITEL, DURCHSCHNITTSDAUER).

Themenschwerpunkt B
(Betriebssysteme)

Aufgabe 1:

1. Synchronisation mit Semaphoren

Gegeben sei folgendes Szenario: Fahrzeuge befahren aus zwei Richtungen eine Brücke mit Engstelle:



Die Brücke darf nur von zwei Fahrzeugen gleichzeitig befahren werden. Die Streckenabschnitte A, X und Y können nur jeweils ein Fahrzeug aufnehmen. Die Fahrzeuge aus den beiden Richtungen können folgendermaßen formalisiert werden:

```
R-Fahrzeug
{
  while (TRUE)
  {
    <Anfahrt bis R1>;
    <von R1 nach X>;
    <von X nach A>;
    <von A nach L1>;
    <Abfahrt von L1>;
  }
}
```

```
L-Fahrzeug
{
  while (TRUE)
  {
    <Anfahrt bis L2>;
    <von L2 nach A>;
    <von A nach Y>;
    <von Y nach R2>;
    <Abfahrt von R2>;
  }
}
```

Es sind beliebig viele R-Fahrzeuge und L-Fahrzeuge möglich.

Synchronisieren Sie die Fahrzeuge auf der Brücke, indem Sie in geeigneter Weise Semaphore deklarieren und Semaphor-Operationen einfügen. Geben Sie zu den verwendeten Semaphoren die Startwerte an. Folgende Bedingungen müssen erfüllt sein:

Fortsetzung nächste Seite!

- Maximal zwei Fahrzeuge dürfen gleichzeitig die Brücke, also die Streckenabschnitte A, X und Y, befahren.
- Es dürfen keine Kollisionen auf der Brücke auftreten. Auch Auffahrunfälle, d. h. Kollisionen zwischen Fahrzeugen aus der selben Richtung, sind zu verhindern.
- Sperrphasen sind möglichst kurz zu halten.
- Es treten keine Verklemmungen auf.

Zu Beginn sind alle Streckenabschnitte frei.
Die Streckenabschnitte L1, L2, R1 und R2 müssen nicht synchronisiert werden.

2. Process Scheduling

a) Untersuchen Sie die Schedulingverfahren first-come-first-served, shortest-jobs-first, round robin und statische Priorität bezüglich folgender Kriterien:

- unterbrechend oder nicht unterbrechend
- Fairness
- Echtzeitfähigkeit

b) Welche Konsequenzen hat der Aspekt, ob ein Scheduling-Verfahren unterbrechend oder nicht unterbrechend ist, auf die Kriterien Fairness und Echtzeitfähigkeit? Welchen generellen Vorteil hat ein nicht unterbrechendes Verfahren?

c) Welches Verfahren hat den höchsten Durchsatz? Wieso wird dieses Verfahren trotzdem kaum implementiert?

d) Berechnen Sie den Durchsatz für die Prozessfolge P1 (Dauer 50 ms), P2 (Dauer 70 ms), P3 (Dauer 10 ms), P4 (Dauer 30 ms) für die Schedulingverfahren shortest-jobs-first und round robin (Zeitscheibe 20 ms). Es wird angenommen, dass alle Prozesse gleichzeitig starten und ein Prozesswechsel 2ms dauert.

Fortsetzung nächste Seite!

3. Memory Management

a) Sei eine Menge von Seiten (pages) N = {0, 1, 2, 3, 4, 5, 6} und eine Menge von Seitenrahmen (frames) F = { f_1, f_2, f_3 } gegeben. Nun wird in folgender Reihenfolge auf die Seiten zugegriffen:

ω = 6 2 5 6 4 4 6 1 3 1

Vollziehen Sie die Seitenersetzungs-Strategien LRU (Least Recently Used) und FIFO (First In First Out) schrittweise nach, indem Sie zwei Tabellen nach folgendem Muster anlegen und ausfüllen!

An-forderung	f_1	f_2	f_3	Page-Fault j / n
6				
2				
5				
6				
4				
4				
6				
1				
3				
1				

b) Welches Verfahren hat im Beispiel die bessere Performanz?

c) Nennen Sie für jedes der beiden Verfahren einen Vorteil und einen Nachteil!

Fortsetzung nächste Seite!

4. Deadlocks

a) Nennen und erläutern Sie kurz die 4 Deadlock-Bedingungen (Coffman-Bedingungen)!

b) Nennen und erläutern Sie kurz 4 Möglichkeiten der Verklemmungs-Verhinderung (Deadlock-Prevention)!

c) Verklemmungs-Vermeidung (Deadlock-Avoidance):

Wir gehen aus von 3 Prozessen (p_1, p_2, p_3), die um 3 Sorten von Ressourcen (Sorte 1: DVD-Brenner, Sorte 2: Grafikkarten, Sorte 3: TV-Karten) im System konkurrieren.
Es seien die Zahlen der existierenden Instanzen der verschiedenen Ressourcen-Sorten durch das Tupel (e_1, e_2, e_3) gegeben und die Zahlen der im Moment noch freien Instanzen der Ressourcen-Sorten durch das Tupel (a_1, a_2, a_3). Die aktuelle Belegung von Instanzen der verschiedenen Ressourcen-Sorten durch Prozesse sei durch die 3 × 3 Matrix c und die aktuelle maximale Rest-Anforderung von Instanzen der Ressourcen-Sorten durch Prozesse durch die 3 × 3 Matrix r gegeben:

$$c = \begin{pmatrix} 4 & 0 & 3 \\ 0 & 3 & 0 \\ 3 & 2 & 3 \end{pmatrix}$$

$$r = \begin{pmatrix} 4 & 3 & 0 \\ 0 & 3 & 3 \\ 0 & 0 & 3 \end{pmatrix}$$

$$e = (9, 6, 8)$$

$$a = (2, 1, 2)$$

In den Matrizen c und r seien die Zeilen den Prozessen und die Spalten den Ressourcen-Sorten zugeordnet. (Beispiel zur Verdeutlichung: Dem Prozess 3 entspricht in den Matrizen c und r jeweils die dritte Zeile. Das heißt, Prozess p_3 hat im Moment 3 Instanzen der Ressourcen-Sorte 1 (also 3 DVD-Brenner), 2 Instanzen der Ressourcen-Sorte 2 (also zwei Grafikkarten) und 3 Instanzen der Ressourcen-Sorte 3 (also 3 TV-Karten). Prozess 3 möchte in Zukunft bspw. maximal noch 0 DVD-Brenner, 0 Grafikkarten und 3 TV-Karten haben.)
Sollte im aktuellen Zustand eine Anforderung eines DVD-Brenners durch den Prozess p_1 vom Betriebssystem erfüllt werden? Beantworten Sie die Frage durch Anwendung des Banker-Algorithmus! Stellen Sie alle Ihre Ausführungsschritte so dar, dass die Ausführung des Algorithmus nachzuvollziehen ist!

Fortsetzung nächste Seite!

Aufgabe 2:

Ein Verwaltungsprogramm für ein Drucksystem mit mehreren Druckern arbeite folgendermaßen:

i. Druckaufträge werden in Form von Auftragsdateien in einem Dateiverzeichnis "/var/spool/printq" abgelegt. Der Name einer Auftragsdatei ist folgendermaßen aufgebaut: cpid . pid ist ist dabei die Prozess-Id des Prozesses, der den Druckauftrag erzeugt (z. B. Auftragsdatei „c4711" für einen Auftrag des Prozesses mit der pid 4711).

ii. Das Verwaltungsprogramm liest beim Start das Dateiverzeichnis durch und erzeugt für jeden Druckauftrag eine Verwaltungsstruktur, in der der Name der Auftragsdatei und der Zeitpunkt der Auftragserstellung eingetragen werden. Diese Datenstrukturen werden in eine Warteschlange eingetragen.

iii. Für jeden vorhandenen Drucker wird nun ein Prozess erzeugt, der Aufträge aus der Warteschlange entnimmt und den Inhalt der zugehörigen Auftragsdatei jeweils auf „seinem" Drucker ausdruckt.

iv. Das Verwaltungsprogramm blockiert sich nach dem initialen Einrichten der Warteschlange und wird „geweckt", sobald ein Benutzerprozess einen neuen Auftrag in die Warteschlange eingetragen hat.

1 Prozesse, Dateien

a) Da Prozess-Ids in vielen Betriebssystemen wiederverwendet werden, kann es theoretisch passieren, dass es eine Auftragsdatei mit der zu vergebenden Nummer bereits gibt. Wie kann ein Programm, das einen Auftrag absetzen möchte, solch einen drohenden Konflikt erkennen und was könnte man zur Behebung des Konflikts tun?

b) Die Prozesse, die Aufträge absetzen, laufen unter der normalen Berechtigung ihrer Benutzer. Was muss man hinsichtlich der Zugriffsrechte der Auftragsdateien und des Dateiverzeichnisses für die Auftragsdateien berücksichtigen, damit

(1) solche Aufträge überhaupt erzeugt werden können,
(2) es nicht möglich ist, dass ein böswilliger Benutzer fremde Aufträge löscht, und
(3) niemand fremde Druckaufträge lesen kann?

c) Skizzieren Sie in einer Programmiersprachen-ähnlichen Form den Ablauf vom Start des Verwaltungsprogramms bis zur vollständigen Einrichtung der Warteschlange mit den Datenstrukturen.

d) Wie könnte man das Aufwecken des Verwaltungsprogramms in Punkt iv realisieren?

Fortsetzung nächste Seite!

2 Speicher

a) Wie kann der Speicherbereich für die Warteschlange aus Punkt ii als gemeinsamer Speicher in einem Betriebssystem mit Seitenadressierung realisiert werden? In welchen Datenstrukturen des Betriebssystems müssen hierzu welche Einträge erfolgen?

b) Skizzieren Sie die logischen Adressräume von zwei Drucker-Prozessen und den Zusammenhang zu dem physikalischen Hauptspeicher.

c) Worauf müsste man achten, wenn man die Warteschlange als verkettete Liste der Datenstrukturen realisieren wollte?

d) Warum ist die Realisierung in Form eines Feldes einfacher?

e) Wie könnte man mit der Situation umgehen, in der die Größe des Feldes nicht ausreicht, um alle anstehenden Druckaufträge aufnehmen zu können?

3 Prozesse, Threads

a) Die beschriebene Realisierung mit Hilfe von Prozessen ist Ressourcen-aufwändig. Warum?

b) Eine Realisierung mit Hilfe von Threads wäre günstiger. Welche Arten von Threads kennen Sie, wodurch unterscheiden sie sich, was ist der Unterschied zu Prozessen?

c) Welche Thread-Art ist für die beschriebene Aufgabe geeignet und sollte statt einer Realisierung mit Prozessen genutzt werden? Welche Kosten spart man dadurch ein, welche Abläufe werden effizienter?

d) Warum ist die andere Thread-Art für die Aufgabe nicht geeignet?

4 Koordinierung

Beschreiben Sie nun die Koordinierung der Warteschlange mit den Auftragsverwaltungsstrukturen zwischen dem Prozess, der das Verwaltungsprogramm ausführt, und den Drucker-Prozessen:

a) Welche Zugriffskonflikte bestehen potentiell?

b) Welche Warte-/Aufwecksituationen gibt es?

c) Welche Koordinierungsmittel sind jeweils dafür geeignet?

d) Skizzieren Sie die Koordinierung auf der Seite des Verwaltungsprogramms und auf der Seite der Drucker-Prozesse jeweils in Programmiersprachen-ähnlicher Form.

66114

Datenbank- und Betriebssysteme (vertieft)

Frühjahr 2010

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____	**Frühjahr**	
Kennwort: _____	**2010**	**66114**
Arbeitsplatz-Nr.: _____		

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

— Prüfungsaufgaben —

Fach:	**Informatik (vertieft studiert)**
Einzelprüfung:	**Datenbank und Betriebssysteme**
Anzahl der gestellten Themen (Aufgaben):	4 Aufgaben, von denen 2 gemäß untenstehender Auswahlregel zu bearbeiten sind
Anzahl der Druckseiten dieser Vorlage:	17

Zu den zwei Themenschwerpunkten A (Datenbanksysteme) und B (Betriebssysteme) ist jeweils entweder die Aufgabe 1 oder 2 zu wählen. Auf der Vorderseite des Kopfbogens sind im Feld „Gewähltes Thema: Nr.“ die Nummern der beiden gewählten Aufgaben anzugeben (z. B. A2, B1)!

Bitte wenden!

Themenschwerpunkt A

(Datenbanksysteme)

AUFGABE 1

1. Relationale Algebra

Die relationale Algebra wird aufgebaut über einer Grundmenge von mengenwertigen Operationen. Diese Grundoperationen können auf Relationen angewendet werden und erzeugen als Ergebnis wieder eine Relation.

Notation: π = Projektion; σ = Selektion; $\bowtie$ = Join; x = kartesisches Produkt; \ = Mengendifferenz; $\cap$ = Schnittmenge; $\cup$ = Vereinigungsmenge; ρ = Umbenennen; $\div$ = relationale Division

a) Beschreiben Sie ein vollständiges SQL-Statement (oder mehrere) und geben Sie an, welche Klausel mit welcher Operation aus der relationalen Algebra korrespondiert (alle Operationen müssen verwendet werden).

b) Definieren Sie die mengenwertige Operation „Division". Auf welche Grundoperation kann die Division zurückgeführt werden und wie? Ein Beweis ist nicht erforderlich.

2. Transaktionen und Transaktionsverarbeitung

a) Nennen und definieren Sie die vier wesentlichen Merkmale einer Datenbanktransaktion.

b) Erklären Sie steal, no-steal, force, no-force im Zusammenhang mit der Fehlerbehandlung.

c) Fassen Sie die vier Kombinationen von force/no-force, steal/no-steal hinsichtlich der Anforderungen an die Redo- und Undo-Recovery auf folgende Weise zusammen:

	Force	No-Force
Steal	• Kein Redo • Undo	• •
No-Steal	• •	• •

Übertragen Sie die Tabelle und ergänzen Sie diese.

Fortsetzung nächste Seite!

3. ER-Modellierung und Relationenmodell

Szenario: Sie sollen ein System zur Verwaltung von Firmen entwerfen.

- Eine Firma ist eindeutig bezeichnet durch ihre Handelsregisternummer. Daneben hat sie einen Namen und eine Adresse, die sich aus Straße, Hausnummer, Ort und Postleitzahl zusammensetzt.
- Firmen bestehen aus Abteilungen. Eine Firma kann dabei aus mehreren Abteilungen bestehen, eine Abteilung kann jedoch immer nur zu einer Firma gehören. Es kann Firmen geben, die nicht in Abteilungen eingeteilt sind, aber jede Abteilung gehört zu einer Firma. Abteilungen haben eine eindeutige AbteilungsID und einen Namen.
- Firmen beschäftigen Mitarbeiter. Ein Mitarbeiter kann nur für eine Firma arbeiten, eine Firma aber beliebig viele Mitarbeiter haben. Ein Mitarbeiter hat eine eindeutige Sozialversicherungsnummer (SVN), einen Namen (bestehend aus Name und Vorname), ein Geburtsdatum und ein Alter, das sich aus dem Geburtsdatum errechnet. Daneben hat ein Mitarbeiter eines oder mehrere Spezialgebiete, auf denen er Fachmann ist.
- Mitarbeiter können Angehörige haben. Zu diesen werden Geburtstag und Name gespeichert. Ein Mitarbeiter kann keine zwei Angehörigen mit gleichem Namen haben.
- Abteilungen bearbeiten Projekte. Ein Projekt ist gekennzeichnet durch seine ProjektID. Daneben besteht es aus einem Namen und einem Kunden. Ein Projekt kann von mehreren Abteilungen bearbeitet werden und eine Abteilung kann an mehreren Projekten arbeiten. Um nicht den Überblick zu verlieren, soll der Zeitraum gespeichert werden, in dem eine Abteilung ein Projekt bearbeitet. Dieser Zeitraum gliedert sich in die Felder „von“ und „bis“.

a) ER-Modellierung:

Erstellen Sie ein ER-Modell des oben beschriebenen Szenarios. Geben Sie Kardinalitäten von Beziehungen sowie Rollennamen an. Verwenden Sie auch abgeleitete, zusammengesetzte und mehrwertige Attribute, mehrstellige Beziehungstypen und schwache Entitätstypen, wo es sinnvoll ist.

b) Relationenmodell:

Überführen Sie das ER-Modell in ein Relationenschema.

Notation: Unterstreichen Sie Primärschlüssel; kennzeichnen Sie Fremdschlüssel durch Angabe der referenzierten Relation in eckigen Klammern.

Beispiel:
Autor(<u>AID</u>, Name)
Buch(<u>ISBN</u>, Schriftsteller[Autor])

Fortsetzung nächste Seite!

4. SQL

Ein Internetauktionshaus nutzt eine relationale Datenbank zum Speichern seiner relevanten Daten. Angenommen ist folgender Auszug aus dem Tabellenschema zu den Benutzern, Auktionen und Geboten:

User(UId, Name,...)
Auction(AId, Name, Begin_Auction, End_Auction, Seller[User], ...)
Bid(UId, AId, Bid_Time, Amount)

Das Auktionshaus will ein Bewertungssystem einführen, um die Risiken für Verkäufer eindämmen zu können.
Folgende Tabelle soll zu den existierenden Tabellen hinzugefügt werden:

Review(RId, AId[Auction], BuyerID[User], Rating, Rating_Time)

RId, AId und BuyerID sind beliebige ganzzahlige Datentypen, Rating ist ein ganzzahliger Datentyp, der die Werte 1,2,3,4,5 annehmen kann, und Rating_Time ist ein timestamp. RId ist Primärschlüssel und die beiden Fremdschlüssel dürfen nie leer sein.

a) Geben Sie das vollständige CREATE-Statement zum Anlegen von Review an, das möglichst viele Constraints erfüllt.

b) Erstellen Sie eine Sicht mit dem Namen Auction_Review mit den Attributen (AId, UId, Durchschnittsrating):
Zu jeder Auktion (gekennzeichnet mit AId) soll das Durchschnittsrating jedes Nutzers (UId) angegeben werden, der auf diese Auktion geboten hat. Die Durchschnittsratings sollen auf allen Ratings des Nutzers basieren, die zeitlich vor Ende der betrachteten Auktion vergeben wurden.

c) Erstellen Sie eine Sicht mit dem Namen Auction_Max_Amount mit den Attributen (AId, MaxAmount):
Zu jeder abgeschlossenen Auktion (aktueller Zeitpunkt mit sysdate()) soll der größte gebotene Betrag angegeben werden.

d) Ausgehend von der Sicht Auction_Max_Amount sollen die 5 TOP-Verkäufer (Umsatz) aus dem Jahr 2005 bestimmt werden (Auktionsende in 2005). Geben Sie die SQL-Query an.

Fortsetzung nächste Seite!

Aufgabe 2

1. Integritätsbedingungen

Eine Aufgabe von Datenbankmanagementsystemen ist die Erhaltung der Datenintegrität.

a) Welche Arten von Integritätsbedingungen gibt es?

b) Geben Sie für jede Art der Integritätsbedingungen je ein Beispiel an.

2. Modellierung

Der Veranstalter der Ausstellung "Becit" möchte diese in einer relationalen Datenbank verwalten:

Zu jedem Ausstellungsraum gibt es eine eindeutige Raumnummer (RNr). Ein Raum hat eine bestimmte Fläche (in m^2), eine Höhe (in m) und zusätzliche Sonderausstattungen wie Beamer, Fernseher, Stühle, Bänke, usw. Die Ausstellungsstände werden in Räumen aufgestellt. Es hat sich eingebürgert, den Ständen innerhalb von Räumen Buchstaben zuzuordnen (SId). So ist beispielsweise Stand "3B" der 2. Stand in Raum 3. Zu Ständen ist bekannt, wieviel Fläche (in m^2) sie benötigen, die Anzahl benötigter Tische und Stühle und möglicherweise benötigte Sonderausstattungen des entsprechenden Raumes.
Die Stände werden dann den Anbietern zugeordnet. Für die Anbieter werden eindeutige Ids vergeben. Zusätzlich sind von diesen Name, Adresse und Telefonnummer bekannt.
Das Personal kümmert sich um den reibungslosen Ablauf der Ausstellung. Jedes Mitglied des Personals besitzt eine Personalnummer (PNr). Ebenfalls bekannt ist der Lohn des Personals.
Das Personal unterteilt sich in Auf-/Abbau-Personal, Sicherheitspersonal und Reinigungspersonal. Das Auf-/Abbau-Personal bekommt bestimmte Stände zugewiesen, die es auf- und abbaut. Das Sicherheitspersonal kümmert sich um die Sicherheit der Ausstellung. Die einzelnen Mitglieder des Sicherheitspersonals patroullieren hierfür in den ihnen zugewiesen Ausstellungsräumen. Sie können über ein Funkgerät verfügen und ggf. auch bewaffnet sein. Das Reinigungspersonal kümmert sich um die Endreinigung der Ausstellungsräume. Da hier in verschiedenen Räumen unterschiedlich viel Arbeit anfallen kann, können vorab keine Räume zugeordnet werden.
Zu beachten ist insbesondere, dass sämtliche Personen des Personals auch mehrere Aufgaben übernehmen können, d.h. sie können beim Aufbau mithelfen, zusätzlich während der Ausstellung für die Sicherheit sorgen und am Ende ggf. die Räume mit reinigen.

a) Entwerfen Sie für das beschriebene Szenario ein ER-Diagramm. Bestimmen Sie hierzu:

- die Entity-Typen, die Relationship-Typen und jeweils deren Attribute;
- ein passendes ER-Diagramm;
- die Primärschlüssel der Entity-Typen, welche Sie anschließend in das ER-Diagramm eintragen;
- die Kardinalitäten der Relationship-Typen, welche Sie ebenfalls in das ER-Diagramm eintragen.

Fortsetzung nächste Seite!

b) Überführen Sie das ER-Modell aus Aufgabe a) in ein verfeinertes relationales Modell. Geben Sie hierfür die verallgemeinerten Relationenschemata an.

c) Bei den Zuordnungen der Sonderausstattungen kann es zur Verletzung einer Integritätsbedingung kommen.

   i) Begründen Sie, woran dies liegt.

   ii) Wie lässt sich das Problem innerhalb einer Datenbank verhindern?

3. Normalformen

Gegeben sei das folgende Relationenschema

Kaeufe: {[*ArtikelId, ArtikelBezeichnung, EinkaufsPreis, VerkaufsPreis, Datum, TagesAktion, KundeName, KundeVorname, KundeGebdatum, Filiale, Verkaeufer*]}

und eine Menge F von funktionalen Abhängigkeiten. **Hinweis:** Die Abkürzungen innerhalb der Menge F beziehen sich auf die Großbuchstaben in den Attributnamen der Relation $Kaeufe$.

$$
\begin{aligned}
F := \{ & \\
& AI \rightarrow AB \\
& AI, AB \rightarrow EP \\
& KN, KV \rightarrow KG \\
& AI, D, KN, KV \rightarrow V \\
& AI, AB, KN, KV, D, V \rightarrow F, VP \\
& D \rightarrow TA \\
& V \rightarrow F \\
\} &
\end{aligned}
$$

a) Ermitteln Sie sämtliche Schlüsselkandidaten von $Kaeufe$.

b) Zeigen Sie, dass das Relationenschema $Kaeufe$ nicht in dritter Normalform ist.

c) Bestimmen Sie eine kanonische Überdeckung F_c zu F.

d) Bringen Sie $Kaeufe$ mit dem Synthesealgorithmus in 3NF.

4. SQL und Relationale Algebra

Gegeben sei folgendes Schema der relationalen Datenbank eines Aktienhändlers:

$Aktie\{\underline{ANr}, Name, Firma[Firma], Ausgabedatum, Ausgabepreis\}$

$Aktienkurs\{\underline{ANr[Aktie]}, \underline{Datum}, \underline{Zeit}, Wert\}$

$Firma\{\underline{Name}, Adresse, Telefon\}$

$Kunde\{\underline{KndNr}, Name, Geburtsdatum, Telefon\}$

$Konto\{\underline{KntNr}, KndNr[Kunde], Eroeffnungsdatum, Saldierungsdatum\}$

Fortsetzung nächste Seite!

$$Buchung\{\underline{KntNr[Konto]}, \underline{BNr}, ANr[Aktie], Aktienzahl, EKdatum, EKzeit, EKpreis, VKdatum, VKzeit, VKpreis\}$$

Hinweis: Für Fremdschlüssel wird nach dem Format `<Attr> [<RefTable>]` die referenzierte Tabelle angegeben. Beispielsweise ist $KndNr[Kunde]$ in der Relation `Konto` ein Fremdschlüssel der Relation `Kunde`.

Sämtliche verfügbaren Aktien sind in der Relation `Aktie` enthalten. Die zugehörigen Kurse werden in der Relation `Aktienkurs` festgehalten. Zu jedem (im Rahmen dieser Aufgabe notwendigen) Zeitpunkt existiert ein korrespondierendes Tupel. Die Aktiengesellschaften (also Firmen, welche Aktien ausgeben) sind in der Relation `Firma` gespeichert. Die Kunden des Aktienhändlers werden in der Relation `Kunde` verwaltet. Jeder Kunde kann beliebig viele Konti eröffnen. Diese werden in der Relation `Konto` aufgeführt und dem Kunden direkt zugeordnet. Aktienkäufe und Aktienverkäufe erfolgen paketweise. Kauft ein Kunde ein Paket einer Aktie, so wird ein neues Tupel in der Relation `Buchung` erstellt. Die Buchungsnummern werden für jedes Konto jeweils in aufsteigender Reihenfolge beginnend bei 1 vergeben. Innerhalb eines neuen Tupels werden Umfang des Pakets (Aktienzahl), Datum und Zeitpunkt des Kaufs und insbesondere der Preis des Aktienpakets festgehalten. Die entsprechenden Attribute für einen Verkauf werden zu $NULL$ initialisiert. Wird später ein Verkauf des Aktienpakets durchgeführt, so werden die Werte für `VKdatum`, `VKzeit` und `VKpreis` entsprechend gesetzt. Aktuell sei der Zeitpunkt 13:00:00 Uhr und das Datum 2008/07/20. Für Datumsangaben gelte insbesondere, dass eine Verschiebung des Datums um t Tage durch folgende Addition durchgeführt werden kann: $2008/07/20 + t$. Beispielsweise ergibt $2008/07/20 + 20 = 2008/08/09$.

a) Formulieren Sie die Anfragen in den folgenden Teilaufgaben (i - ii) in der Anfragesprache SQL. Zur Vereinfachung dürfen Sie sich beliebige Sichten (Views) definieren.

   i) Gesucht sind alle Aktienanteile, welche Kunde "Müller" aktuell hält. Geben Sie für diese Kontonummer, Buchungsnummer, Aktiennummer und die zugehörigen Firmennamen aus.

   ii) Nehmen Sie an, das Ergebnis aus Teilaufgabe i) sei als View *AktuelleAktienMueller* verfügbar.
   Geben Sie nun gruppiert nach den Firmennamen die Gewinne/Verluste aus, welche Müller beim Verkauf seiner aktuellen Aktienanteile machen würde.

b) Formulieren Sie folgende Anfrage in der relationalen Algebra:

   Für welche Firmen hat sich der Aktienkurs im 1. Jahr nach der Ausgabe mindestens verdoppelt?

Fortsetzung nächste Seite!

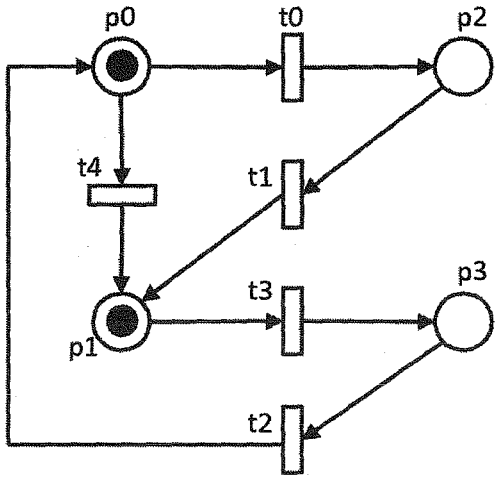
Themenschwerpunkt B

(Betriebssysteme)

AUFGABE 1

1. Petrinetze

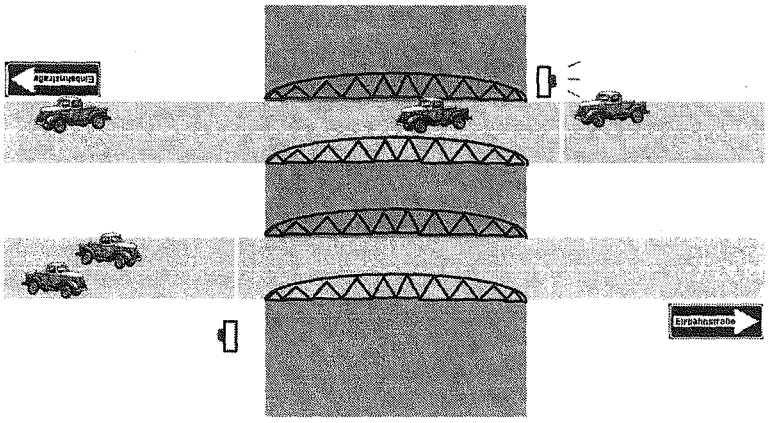
a) Gegeben sei folgendes boolsches Petri-Netz (boolsch bedeutet: Alle Stellen haben Kapazität 1):



i) Geben Sie den Erreichbarkeitsgraphen für dieses Petrinetz an!

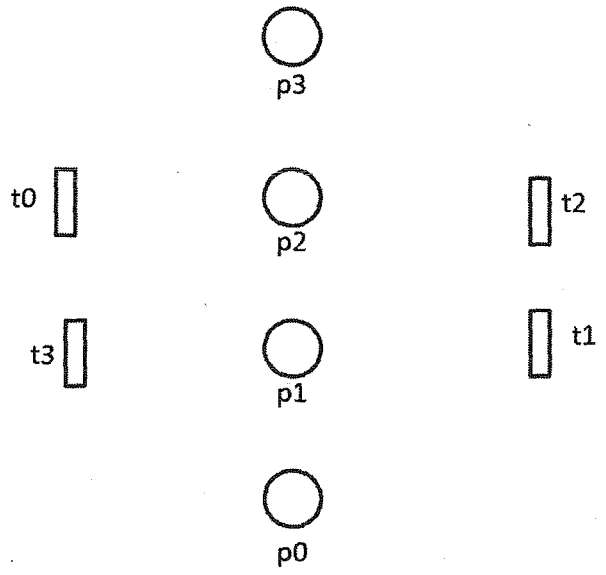
ii) Argumentieren Sie anhand des Erreichbarkeitsgraphen, ob das Netz verklemmt ist!

b) Zwei Ufer sind durch zwei Brücken mit nur je einer Fahrtrichtung verbunden. Eine Brücke kann jedoch immer nur ein Fahrzeug tragen.



Gegeben sei folgende unvollständige Skizze für ein Petrinetz, das diese Bedingung garantiert. Die Stellen haben jeweils die Kapazität ∞.

Fortsetzung nächste Seite!



i) Vervollständigen Sie die Skizze indem Sie geeignete Anfangsmarkierungen, Übergänge und Benennungen für Stellen und Transitionen eintragen!

ii) Begründen Sie, warum Ihr Ansatz das Gewünschte leistet!

2. Wechselseitiger Ausschluss, Deadlocks

a) Betrachten Sie folgende Ansätze zur Garantie des wechselseitigen Ausschlusses (Assembler-artiger Code):

Ansatz A:

```
enterCriticalRegion:
   LOAD REGISTER, LOCK
   MOVE LOCK, #1
   CMP REGISTER, #0
   JNE enterCriticalRegion
   RET
leaveCriticalRegion:
   MOVE LOCK, #0
   RET
```

Ansatz B:

```
enterCriticalRegion:
  TSL REGISTER, LOCK
  CMP REGISTER, #0
  JNE enterCriticalRegion
  RET
leaveCriticalRegion:
  MOVE LOCK, #0
  RET
```

i) Welcher der beiden Ansätze garantiert den wechselseitigen Ausschluss? Begründen Sie Ihre Antwort!

ii) Was ist ein Nachteil beider Ansätze?

b) In Java kann der wechselseitige Ausschluss unter Verwendung des Schlüsselwortes `synchronized` realisiert werden. Diese Konstrukte realisieren das Monitor-Konzept zum wechselseitigen Ausschluss. Dies funktioniert intern mit *intrinsic locks*. Jedem Objekt ist ein *intrinsic lock* zugeordnet. Threads, die eine `synchronized` Methode eines Objekts betreten, erwerben das *intrinsic lock* des Objekts. Nun kann kein anderer Prozess eine `synchronized` Methode des Objekts mehr betreten.

Erklären Sie warum es in folgendem Code zu einem Deadlock kommen kann! Überprüfen Sie die vier Deadlock-Bedingungen (Cuffman-Bedingungen) an diesem Beispiel!

```
public class Duchess {
    private final String name;
    public Duchess(String name) {
        this.name = name;
    }
    public String getName() {
        return this.name;
    }
    public synchronized void kiss(Duchess kisser) {
        System.out.println(this.name + ": " +  kisser.getName() + " has kissed me!");
        kisser.kissBack(this);
    }
    public synchronized void kissBack(Duchess kisser) {
        System.out.println(this.name + ": " +  kisser.getName() + " has kissed me back!");
    }
}
public class Deadlock{
  public static void main(String[] args) {
      final Duchess josephine = new Duchess("Josephine");
      final Duchess emily = new Duchess("Emily");
      new Thread(new Runnable() {
          public void run() { emily.kiss(josephine); }
      }).start();
      new Thread(new Runnable() {
          public void run() { josephine.kiss(emily); }
      }).start();
  }
}
```

Fortsetzung nächste Seite!

3. Prozess-Scheduling

Thema dieser Aufgabe sind verschiedene Prozessorverwaltungs-Strategien (Scheduling-Strategien). Als Modell sei dazu ein einfaches Ein-Bediener-System gegeben. Das heißt wir betrachten Aufträge mit Bedienzeiten (Rechenzeiten) sowie Ankunftszeiten (der Zeitpunkt, ab dem der Auftrag im System vorhanden ist und berechnet werden kann) und einen Prozessor, auf dem jeweils immer nur einer der Aufträge bearbeitet werden kann. Wir nehmen vereinfachend an, dass ein Prozesswechsel keine Zeit kostet.

Die Ausführung eines solchen Systems kann man sehr gut anhand von Scheduling-Diagrammen visualisieren, bei dem auf der x-Achse die Zeiteinheiten und auf der y-Achse die verschiedenen Aufträge angetragen sind. Dabei ist jeder Auftrag durch eine Linie von seinem Ankunftszeitpunkt bis zum Zeitpunkt seiner abgeschlossenen Berechnung eingetragen, wobei die Linie gestrichelt ist solange er wartet und durchgezogen, solange ihm der Prozessor zugeteilt ist.

In Abb. 1 ist ein solches Diagramm zur Verdeutlichung für eine fiktive Scheduling-Strategie dargestellt.

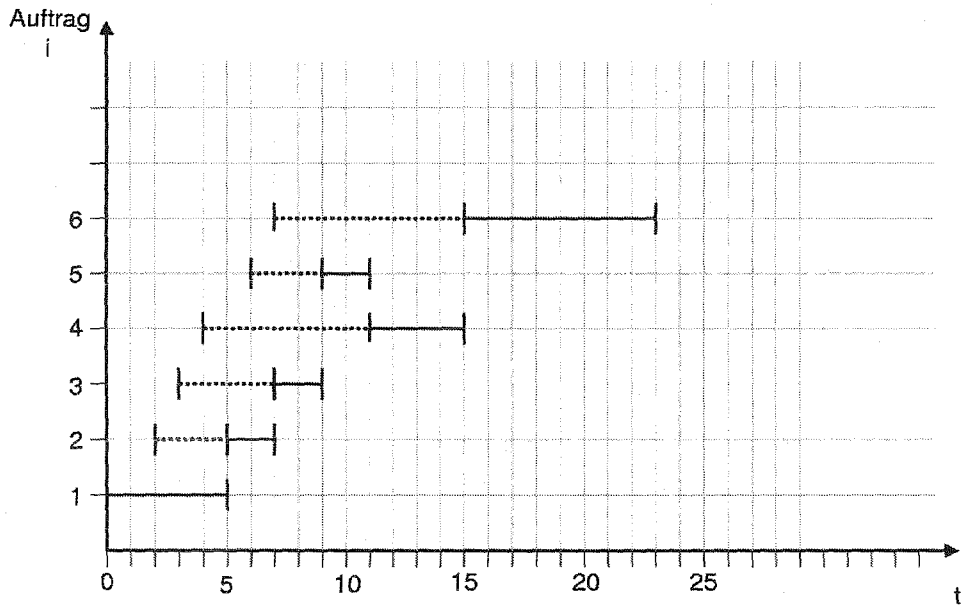


Abbildung 1: Scheduling Diagramm (fiktive Strategie)

Im Modell-Bediensystem existieren sechs Aufträge $A_1, \ldots, A_6$ mit den Bedienzeiten $b = (4, 3, 2, 3, 2, 6)$ (bedeutet A_1 benötigt 4 Zeiteinheiten, A_2 3 usw.). Der Vektor der Ankunftszeiten sei durch $a = (2, 3, 5, 6, 7, 8)$ (bedeutet A_1 ist ab Zeitpunkt 2, A_2 ab 3 usw. in der Warteschlange) gegeben. Vor dem Zeitpunkt $t = 0$ sei das Bedienungs-System leer.

a) aa) Zeichnen Sie entsprechende Diagramme für das angegebene Beispiel-Bedienersystem für folgende Scheduling-Strategien:

i) FCFS (First Come First Served): FCFS bedeutet, dass die Aufträge in der Reihenfolge ihrer Ankunft ausgeführt werden. Auftragsunterbrechung ist nicht möglich.

ii) SRPT (Shortest Remaining Processing Time): SRPT bedeutet, dass jeweils der Auftrag mit kürzester Restbedienzeit ausgeführt wird. Auftragswechsel sind nur bei

Fortsetzung nächste Seite!

Ankunftszeitpunkten neuer Aufträge möglich.

Wir legen für unser Bediensystem fest: Haben zwei Prozesse gleiche Restbedienzeiten und rechnet einer davon gerade, wird zunächst dem gerade rechnenden Prozess der Vorrang gegeben. Haben ansonsten zwei Prozesse gleiche Restbedienzeiten wird dem Prozess Vorrang gegeben, der zuerst im System angekommen war.

bb) Warum ist SRPT schwierig umzusetzen? (1 Satz)

cc) Warum ist es sinnvoll, im Falle von realen (nicht-idealisierten) Systemen bei gleichen Restbedienzeiten dem gerade rechnenden Prozess den Vorrang zu geben? (1 Satz)

b) aa) Berechnen Sie die *mittlere Verweilzeit* $\overline{V}$ und die *mittlere Wartezeit* $\overline{W}$ für die beiden Strategien der vorangehenden Teilaufgabe. Nutzen Sie zur Berechnung folgende Formeln:

- c_i bezeichnet die **Abgangszeit** (der Zeitpunkt seiner fertigen Berechnung) des Auftrags A_i, a_i seine Ankunftszeit und b_i seine Bedienzeit;
- $v_i = c_i - a_i$ bezeichnet die **Verweilzeit** des Auftrags A_i;
- $w_i = v_i - b_i = c_i - a_i - b_i$ bezeichnet die **Wartezeit** des Auftrags A_i;
- Gibt es $n \in N$ Aufträge, so gilt für die **mittlere Verweilzeit**:

$$\overline{V} = \frac{1}{n}\sum_{i=1}^{n} v_i$$

- Für die **mittlere Wartezeit** gilt:

$$\overline{W} = \frac{1}{n}\sum_{i=1}^{n} w_i$$

bb) Welche Ziele sollen beim Prozess-Scheduling in Batch-Systemen optimiert werden?

cc) Welches Ziel ist in interaktiven Systemen besonders wichtig?

c) aa) Wie funktionieren Zeitscheiben-basierte Scheduling Verfahren prinzipiell?

bb) Warum führt man hier zusätzlich oft Prioritäten ein?

cc) Nennen und erläutern Sie kurz einen Vorteil und einen Nachteil kleiner Zeit-Quanten!

Fortsetzung nächste Seite!

4. Memory Management

a) Sei eine Menge von Seiten (pages) $N = \{0, 1, 2, 3, 4, 5, 6\}$ und eine Menge von Seitenrahmen (frames) $F = \{f_1, f_2, f_3\}$ gegeben. Nun wird in folgender Reihenfolge auf die Seiten zugegriffen:

$w = 4\ 2\ 4\ 0\ 5\ 5\ 6\ 3\ 1\ 3\ 2\ 6\ 4$

Vollziehen Sie die Seitenersetzungs-Strategien LRU (Least Recently Used) und FIFO (First In First Out) schrittweise nach, indem Sie die entsprechenden folgenden Tabellen auf Ihr Blatt übertragen und vervollständigen!

Strategie LRU:

An-forderung	f_1	f_2	f_3	Page-Fault j / n
4			4	j
2		2	4	j
4		2	4	n
0	0	2	4	j
5				
5				
6				
3				
1				
3				
2				
6				
4				

Strategie FIFO:

An-forderung	f_1	f_2	f_3	Page-Fault j / n
4			4	j
2		2	4	j
4		2	4	n
0	0	2	4	j
5				
5				
6				
3				
1				
3				
2				
6				
4				

Fortsetzung nächste Seite!

b) Gegeben sei ein virtueller Adressraum von 24 Bit und eine Seiten- und Kachelgröße von je 4096 Byte. Der Speicher sei Byte-weise adressierbar. Weiterhin sei eine Seiten-Kacheltabelle gegeben, die u.a. folgende Einträge hat (Format: erste Spalte: Dezimal; zweite Spalte: Binär):

129	001101
130	101111
131	110100
132	000101
133	001100
134	101110
135	010110
136	110111
...	...

Ermitteln Sie zu der virtuellen Adresse

$v = (0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,1\,1\,0\,0\,0\,0\,1\,1\,1\,0\,0\,0\,1\,1)$

die physikalische Adresse!

Fortsetzung nächste Seite!

AUFGABE 2

Teilaufgabe 1

Die folgenden vier Jobs stehen zu den angegebenen Zeiten (in Minuten) zur Ausführung an:

Job	Ankunftszeit	Bearbeitungsdauer
A	0	15
B	1	8
C	6	10
D	10	1

a) Stellen Sie die zeitlichen Abläufe in Gantt-Diagrammen (Balkendiagramme über der Zeitachse) dar und berechnen Sie dazu den Turnaround eines jeden Jobs, sowie den Average Turnaround bei den folgenden Scheduling-Strategien:

1. First-Come-First-Served (FCFS)
2. Shortest-Job-First (SJF)
3. Preemptive-Shortest-Job-First (Preemptive SJF=Shortest Remaining Time First)
4. Round-Robin (RR) mit Zeitquantum q = 8

b) Nun soll eine „2-Level-Feedback-Strategie“ verfolgt werden derart, dass je eine Warteschlange für die Kategorien „Kurzläufer“ und „Langläufer“ existiert, wobei das Scheduling bei den Kurzläufern mit Round-Robin und dem Zeitquantum q = 1 und bei den Langläufern nach FCFS abläuft.

Jeder neu ankommende Job wird zunächst als Kurzläufer eingestuft. Erst wenn ein neuer Job eine Bearbeitungszeit von 2 Zeitquanten überschreitet, muss er in die Kategorie der Langläufer wechseln. Die Warteschlange der Kurzläufer hat Priorität über die Warteschlange der Langläufer, d.h. falls gerade ein Langläufer bearbeitet wird und es kommt ein neuer Job an, so wird der aktive Langläufer-Job solange angehalten, bis keine Kurzläufer mehr im System sind. Geben Sie jeweils die Wartezeit eines jeden Jobs an.

Fortsetzung nächste Seite!

Teilaufgabe 2

a) Vor einem Supermarkt sind die verfügbaren 100 Einkaufswagen alle in einer Reihe **A** aufgestellt. Jeder Wagen ist durch eine kleine Kette mit dem nächsten Wagen gekoppelt. Der erste Wagen der Reihe wird mit einer Kette an einen Stahlträger gekoppelt. Jeder Kunde des Supermarkts benötigt für seinen Einkauf genau einen Einkaufswagen und er hält hierfür eine 1€-Münze parat, mit der die Kette zum derzeit letzten Einkaufswagen der Reihe gelöst werden kann. Sind alle 100 Einkaufswagen belegt, so wartet ein neuer Kunde darauf, dass ein anderer Kunde seinen Einkaufswagen wieder zurückbringt.

b) Implementieren Sie das Vorgehen der Kunden und die Synchronisation bei der Nutzung der Einkaufswagen in (Java-ähnlichem) Pseudo-Code unter ausschließlicher Verwendung binärer Semaphore.

c) Nach einem Ausbau des Supermarkts werden für die Kunden zusätzlich zur vorhandenen Reihe **A** für die 100 Einkaufswagen an einer anderen Stelle **B** des Parkplatzes weitere 50 Einkaufswagen desselben Typs als Reserve angeboten. Damit die Kunden nicht unnötig zu **A** oder **B** gehen müssen, hat die Leitung des Supermarkts große, von überall sichtbare, elektronische Anzeigetafeln anbringen lassen, auf denen jeweils die aktuelle Zahl verfügbarer Einkaufswagen bei **A** und **B** angeschrieben ist. Beschreiben Sie diese kundenfreundlichere Lösung mit „Counting Semaphores“ und kommentieren Sie mögliche Sonderfälle von „Race Conditions“.

Fortsetzung nächste Seite!

Teilaufgabe 3

a) Wie kann man seitens eines Betriebssystems für die aktuell vorhandenen Prozesse und Betriebsmittel erkennen, ob zu einem bestimmten Zeitpunkt gerade ein Deadlock vorliegt? Erläutern Sie knapp das Vorgehen.

b) Für den Fall, dass für eine aktuell vorhandene Gruppe von Prozessen und Betriebsmitteln zu einem bestimmten Zeitpunkt ein Deadlock erkannt wurde, wie sollte das Betriebssystem in solch einer Situation reagieren? Erläutern Sie knapp mögliche Strategien sowie deren jeweilige Vor- und Nachteile.

c) Der exklusive „Verein zur Pflege philosophischer Diners" hat über einen Aufnahmeantrag eines neuen Bewerbers zu entscheiden. Dieser führt in seinem Antrag aus, dass mit seiner Aufnahme die Gefahr von Deadlocks bei den traditionellen Diners beseitigt würde, da er Linkshänder sei, während alle bisherigen fünf Vereinsmitglieder Rechtshänder seien. Bei diesen Diners am runden Tisch sind für die fünf Philosophen je ein Teller und ein Essstäbchen vorhanden sowie für alle in der Tischmitte eine Schüssel mit Reis. Um essen zu können, benötigt ein Philosoph die beiden Essstäbchen, die links und rechts seines Tellers liegen.
In der Tat hatte es in der Vereinsgeschichte schon Deadlocks gegeben, bei der die Philosophen, die bisher ausnahmslos Rechtshänder waren, grundsätzlich zuerst das Essstäbchen rechts des jeweiligen Tellers ergriffen und dann geduldig darauf warteten, dass das Essstäbchen links ebenfalls verfügbar wird.

Zeigen Sie mit Hilfe eines Widerspruchsbeweises über die notwendigen Bedingungen bei Deadlocks, dass der neue Bewerber als Linkshänder, der immer zuerst nach dem Essstäbchen zur Linken seines Tellers greifen wird, tatsächlich die Deadlock-Gefahr für den Verein beseitigen würde.

Teilaufgabe 4

In einem Demand-Paging System mit drei Kacheln sei die folgende Seitenreferenz-Folge (von links nach rechts gelesen) zu bearbeiten:

0, 1, 2, 3, 0, 2, 1, 3

a) Notieren Sie zeilenweise nach jeder Seitenreferenz die komplette Speicherbelegung des Systems und markieren Sie die Zeilen, die nach einem Seitenfehler (page fault) entstanden. Das erste Laden einer Seite zählt dabei ebenfalls als ein Seitenfehler.

b) Ermitteln Sie die Anzahlen der Seitenfehler für die beiden Ersetzungsstrategien

Optimale Ersetzung („Belady's Strategy")

Second Chance („Clock Policy")

66114

Datenbank- und Betriebssysteme (vertieft)

Herbst 2010

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____	**Herbst 2010**	**66114**
Kennwort: _____		
Arbeitsplatz-Nr.: _____		

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

— Prüfungsaufgaben —

Fach: **Informatik (vertieft studiert)**

Einzelprüfung: **Datenbank- und Betriebssysteme**

Anzahl der gestellten Themen (Aufgaben): **4 Aufgaben, von denen zwei gemäß untenstehender Auswahlregel zu bearbeiten sind!**

Anzahl der Druckseiten dieser Vorlage: **21**

Zu den zwei Themenschwerpunkten A (Datenbanksysteme) und B (Betriebssysteme) ist jeweils entweder die Aufgabe 1 oder 2 zu wählen. Auf der Vorderseite des Kopfbogens sind im Feld „gewähltes Thema: Nr." die Nummern der beiden gewählten Aufgaben anzugeben (z. B. A2, B1)!

Bitte wenden!

Themenschwerpunkt A
(Datenbanksysteme)

Teilaufgabe 1:

1. Entity-Relationship-Modell

Ein Handelsunternehmen möchte seine Struktur verbessern und ein Datenbanksystem zur Verwaltung seiner Filialen, angebotenen Waren und Kunden erstellen.

Die Basis dieses Systems bilden die Filialen des Unternehmens. Jede Filiale ist eindeutig durch ihre Filialnummer gekennzeichnet und befindet sich in einer Stadt. Außerdem hat jede Filiale einen Filialleiter.

Zu jeder Filiale gehört genau ein Lager mit einer eindeutigen Lagernummer und ebenfalls einem Leiter. Jedes Lager verfügt über eine bestimmte Menge an verschiedenen Waren. Jede Ware kann in mehreren Lagern vorrätig sein und ist über eine Nummer, einen Namen und einen Preis gekennzeichnet.

Ein Kunde kann in einer Filiale des Unternehmens Bestellungen aufgeben. Der Kunde hat eine Kundennummer, einen Namen und eine Adresse. Eine Bestellung enthält dabei jeweils einen Warenartikel, dessen gewünschte Menge und das Datum, an dem die Bestellung abgeholt wird.

(a) Erstellen Sie ein Entity-Relationship-Diagramm für obige Datenbank.

(b) Setzen Sie das in Teilaufgabe a) erstellte Entity-Relationship-Diagramm in ein Relationenschema um. Relationships sollen mit einer möglichst geringen Anzahl von Relationen realisiert werden. Dabei sind unnötige Redundanzen zu vermeiden. Ein Relationenschema ist in folgender Form anzugeben: `Relation (Attribut1, Attribut2, ...)`. Schlüsselattribute sind dabei zu unterstreichen.
Achten Sie bei der Wahl des Schlüssels auf Eindeutigkeit und Minimalität.

Fortsetzung nächste Seite!

2. Normalisierung

Gegeben sei folgende Datenbank für Wareneingänge eines Warenlagers. Die Primärschlüssel-Attribute sind unterstrichen.

ZulieferungsNr	ArtikelNr	Datum	Artikelname	Menge
1	1	01.01.2009	Handschuhe	5
1	2	01.01.2009	Mütze	10
2	3	05.01.2009	Schal	2
2	1	05.01.2009	Handschuhe	8
3	4	06.01.2009	Jacke	2

(a) Erläutern Sie, inwiefern obiges Schema die 3. Normalform verletzt.

(b) Geben Sie für obige Datenbank alle vollen funktionalen Abhängigkeiten (einschließlich der transitiven) an.

(c) Überführen Sie das obige Relationenschema in die 3. Normalform.
Erläutern Sie die dazu durchzuführenden Schritte jeweils kurz.

3. SQL

Gegeben sei das folgende Relationenschema:

```
Fahrzeug(MNR:int(3), FZGNR:char(12), Baujahr:int(4),
         KMStand:int(5),Preis:int(5))
Modell(MNR:int(3), HNR:int(3), Typ:char(20), Neupreis:int(5),
         ps:int(3))
Hersteller(HNR:int(3), Name:char(20))
```

Dabei sind die Schlüsselattribute jeweils unterstrichen und zusätzlich für alle Attribute die Typen angegeben. Formulieren Sie die folgenden Anfragen bzw. Anweisungen in SQL.

(a) Geben Sie die Anweisungen in SQL-DDL an, die notwendig sind, um die Relationen „Fahrzeug“, „Modell“ und „Hersteller“ zu erzeugen. Achten Sie dabei darauf, die Primärschlüssel der Relationen zu kennzeichnen.

(b) Bestimmen Sie die Typen aller Modelle des Herstellers mit Namen BMW.

(c) Bestimmen Sie den Mindestpreis, bezogen auf das Attribut „Preis“, der Fahrzeuge eines jeden Herstellers.

(d) Bestimmen Sie die Namen der Hersteller, für die von jedem ihrer Modelle mindestens ein Fahrzeug in der Datenbank gespeichert ist.

(e) Bestimmen Sie die Namen aller Hersteller, von denen mindestens fünf Fahrzeuge eines beliebigen Modells in der Datenbank gespeichert sind.

Fortsetzung nächste Seite!

Teilaufgabe 2:

1. Datenbankanfragen in SQL

Wir betrachten die Datenbank FUSSBALL-EUROPAMEISTERSCHAFT mit den folgenden Tabellen:

- Spieler(Rückennummer, Land, Name, Vorname, Länderspiele, Verein, Land_V):
  Spielerinformation, einschließlich des Heimatlandes (Land), der Anzahl der Länderspiele, dem Verein und dem Land des Vereins (Land_V)

- Nationen(Land, Hauptstadt):
  Informationen zu den teilnehmenden Nationen

- nimmt_teil_an(Rückennummer, Land, Spiel_ID):
  Beteiligung eines Spielers an Spielen

- Spiele(Spiel_ID, Land_Heim, Land_Gast, Datum, Stadion_ID, Tore_Heim, Tore_Gast):
  Informationen zu den Spielen, wobei Land_Heim und Land_Gast die beiden an einem Spiel beteiligten Nationen sind

- Stadion(Stadion_ID, Stadt, Land, Zuschauerkapazität):
  Informationen zu den Stadien

Erstellen Sie in SQL folgende Anfragen:

(a) Bestimmen Sie alle teilnehmenden Spieler, die beim "FC Bayern München" (Verein) spielen.

(b) Bestimmen Sie alle Spiele, die im Heimatland von einer der beiden beteiligten Nationen stattfinden.

(c) An wie vielen Spielen nimmt "Bastian Schweinsteiger" teil?

(d) Bestimmen Sie für jede Nation den Spieler mit den meisten Länderspielen. Dazu können Sie eine Unteranfrage oder eine Sicht (View) verwenden, die in der Hauptanfrage aufgerufen wird.

Fortsetzung nächste Seite!

2. Datenmodellierung und Datenbankaufbau

Wir betrachten die Datenbank FUSSBALL-EUROPAMEISTERSCHAFT aus Aufgabe 1.

(a) Definieren Sie die Relationenschemata „Spieler" und „nimmt_teil_an" in SQL mittels CREATE TABLE und geben Sie dabei geeignete Schlüssel- und Fremdschlüsselbedingungen an. Nehmen Sie dabei an, dass die anderen Tabellen schon erzeugt sind.

(b) Fügen Sie in SQL mittels INSERT in jede der beiden Relationen „Spieler" und „nimmt_teil_an" jeweils mindestens ein Tupel ein, so dass die natürlichen Schlüssel- und Fremdschlüssel bedingungen aus Teil a) erfüllt sind. Eventuelle weitere Fremdschlüsselbedingungen auf andere Tabellen müssen nicht berücksichtigt werden.

(c) Geben Sie ein ER-Diagramm mit Funktionalitätsbedingungen für das Datenbankschema an, in dem möglichst viele Relationen als Relationships modelliert sind.

3. Funktionale Abhängigkeiten

Gegeben sei das Relationenschema $R = (U, F)$ mit der Attributmenge

$$U = \{A,B,C,D,E,G\}$$

und der folgenden Menge F von funktionalen Abhängigkeiten:

$$F = \{ A \rightarrow B,\ AB \rightarrow CD,\ CD \rightarrow E\}.$$

a) Bestimmen Sie die Attributhüllen $\{A\}_F^+$ und $\{C,D\}_F^+$.

b) Erklären Sie, warum die funktionale Abhängigkeit $CD \rightarrow E$ zu Update-Anomalien (d. h. beim Einfügen, Löschen und Ändern) führt.

c) Bestimmen Sie alle Schlüssel von R.

d) Bestimmen Sie eine Basis und eine 3NF-Zerlegung von R.

e) Zerlegen Sie R bezüglich der BCNF-verletzenden funktionalen Abhängigkeit $CD \longrightarrow E$.

Themenschwerpunkt B
(Betriebssysteme)

Teilaufgabe 1:

1. Verklemmungen

a) Nennen Sie alle Bedingungen für das Auftreten von Verklemmungen (Deadlocks).

b) Gegeben sei folgender Systemzustand:

	maximum					allocation					available			
	A	B	C	D		A	B	C	D		A	B	C	D
P0	2	5	8	7	P0	0	0	8	4		1	0	3	1
P1	0	3	3	1	P1	0	1	0	0					
P2	1	2	2	2	P2	0	2	0	1					
P3	8	5	2	2	P3	2	2	1	1					
P4	5	3	3	3	P4	5	0	0	3					

Überprüfen Sie mit Hilfe des Bankier-Algorithmus, ob sich das System in einem sicheren Zustand befindet. Geben Sie im positiven Fall eine sichere Prozessfolge an und im negativen Fall den Zustand, in dem keinem weiteren Prozess die benötigten Betriebsmittel zugeteilt werden können.

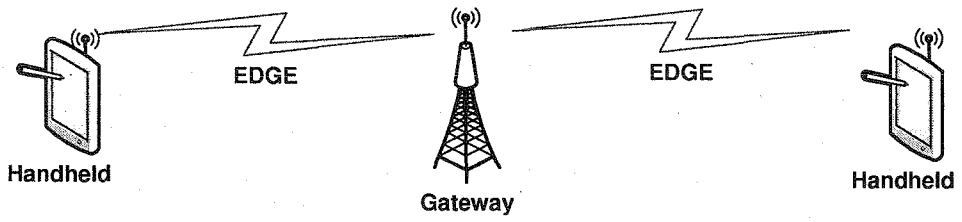
c) Wann kann es in einem System, in dem von Beginn an ein Verklemmungsvermeidungsalgorithmus läuft, trotzdem zu einer Verklemmung (einem Deadlock) kommen? Begründen Sie kurz Ihre Antwort.

Fortsetzung nächste Seite!

2. Flusskontrolle

Ein Student möchte mit seinem Handheld ein Musikstück auf das Handheld seines Freundes schicken. Hierfür verbindet er sich über eine EDGE-Datenverbindung mit einem Gateway, welches seinerseits eine EDGE-Datenverbindung mit dem Handheld des Freundes aufbaut. Die Datenverbindung besteht für die Dauer des Musikstückaustauschs.

Die folgende Abbildung illustriert diesen Zusammenhang:



Nehmen Sie an, die EDGE-Verbindung hat eine Bandbreite von 470 kBit/s und die Datenübertragung von einem Handheld zum anderen Handheld dauert 250 ms. Das Gateway benötigt 20 ms für die Umsetzung eines Frames zwischen den EDGE-Verbindungen.

Bearbeiten Sie nun die folgenden Teilaufgaben:

a) Nehmen Sie an, dass zur Ende-zu-Ende Flusskontrolle ein Sliding-Window-Verfahren eingesetzt wird. Die Framegröße ist jeweils auf genau 1024 Bytes normiert. Welche Fenstergröße (Anzahl der speicherbaren Pakete) ist mindestens erforderlich, um für die EDGE-Verbindungen eine Auslastung von 80% zu erreichen?

b) Betrachten Sie nun die Anwendung der Slow-Start Methode, wobei keine Überlastung vorliegt. Das Empfangsfenster am Gateway ist 40 kByte und die maximale Framegröße beträgt 2 kByte. Wie lange dauert es mindestens, bis das erste volle Fenster vom Handheld gesendet werden kann?

Fortsetzung nächste Seite!

3. ISO/OSI-Modell

a) Geben Sie die Schichten des ISO/OSI-Modells an. Achten Sie auf die richtige Reihenfolge der Schichten.

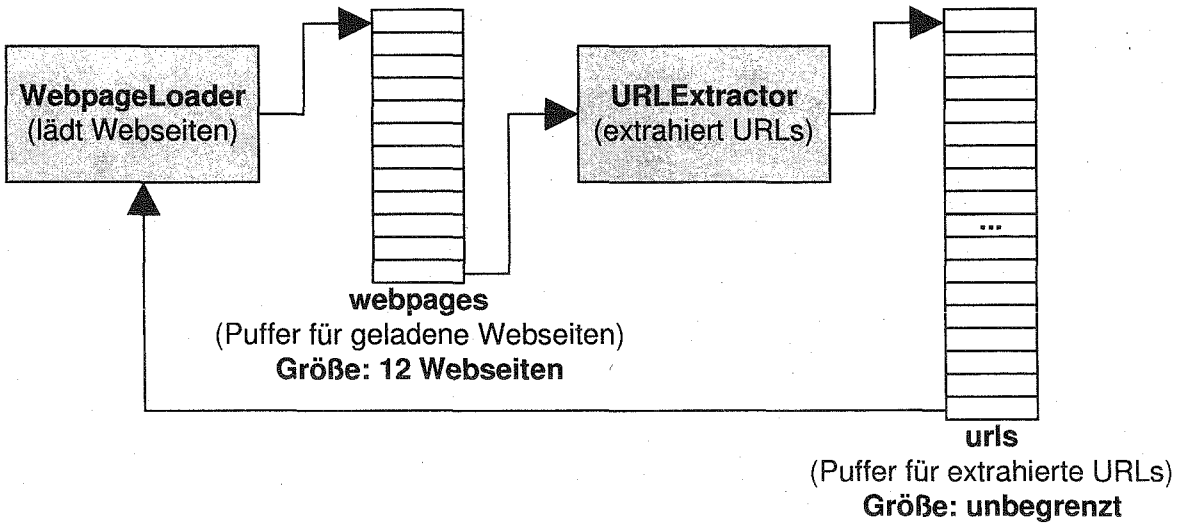
b) Führen Sie kurz (1-2 Sätze pro Schichtübergang) aus, wie eine HTTP-Anfrage im ISO-OSI Modell abgehandelt wird. Beginnen Sie hierbei bei der obersten Schicht am Web-Client („Klick auf den Link im Browser") bis hinunter zur Leitung und wieder hoch bei dem aufgerufenen Web-Server.

Fortsetzung nächste Seite!

4. Producer-Consumer-Problem

Ein Webcrawler oder Spider ist ein Computerprogramm, das automatisch das World Wide Web durchsucht und Webseiten analysiert. Webcrawler werden vor allem von Suchmaschinen eingesetzt. Wie beim Internetsurfen gelangt ein Webcrawler über Hyperlinks von einer Webseite zu weiteren URLs. Dabei werden alle aufgefundenen Adressen gespeichert und der Reihe nach besucht.

In dieser Aufgabe wird ein vereinfachter Webcrawler betrachtet. Er besteht aus zwei Komponenten. Der `WebpageLoader` entnimmt eine URL aus dem Puffer `urls`, lädt die Webseite dieser URL und legt sie im Puffer `webpages` ab. Der `URLExtractor` entfernt eine Webseite aus dem Puffer `webpages`, extrahiert alle URLs, die in dieser Webseite vorkommen, und speichert diese im Puffer `urls`. Dabei sei der Puffer für Webseiten, `webpages`, auf zwölf Elemente beschränkt. Der Puffer für URLs, `urls`, habe dagegen keine Begrenzung.



Die beiden Komponenten, der `WebpageLoader` und der `URLExtractor`, sollen in eigenen Threads laufen. Außerdem soll es möglich sein, dass mehrere Instanzen der Klassen `WebpageLoader` bzw. `URLExtractor` parallel arbeiten. Dafür ist eine adäquate Synchronisation des Zugriffs auf die beiden Puffer, `webpages` und `urls`, erforderlich.

Zur Synchronisation stehen Ihnen binäre Semaphore (`BinarySemaphore`) und zählende Semaphore (`CountingSemaphore`) zur Verfügung:

```
public interface Semaphore {
    public void wait();
    public void signal();
}

public class BinarySemaphore implements Semaphore {
    public BinarySemaphore() {
        [...]
    }
    [...]
}
```

Fortsetzung nächste Seite!

```
public class CountingSemaphore implements Semaphore {
    public CountingSemaphore(int initialCount){
        [...]
    }
    [...]
}
```

Vervollständigen Sie die Codeschablonen in den Teilaufgaben a), b) und c), indem Sie die Kommentare der Form `// Ergänzung X` durch Code für die Initialisierung der verwendeten Semaphore bzw. für die Synchronisation der Threads der Klassen `WebpageLoader` und `LinkExtractor` ersetzen. Sie können an den mit `// Ergänzung X` gekennzeichneten Stellen beliebig viele Codezeilen oder auch keine einfügen. Schreiben Sie Ihre Lösung in folgender Form auf:

Ergänzung	Code
1	`new BinarySemaphore()`
2	...
3	...
...	...

Ihre Lösung soll einen möglichst hohen Grad an Parallelität ermöglichen.

a) Vervollständigen Sie die folgenden Initialisierungen der Semaphore.

```
// Semaphore für Zugriffsschutz auf den Puffer für Webseiten
Semaphore webpagesAccess = // Ergänzung 1

// Semaphore, um bei leerem Puffer auf eine Webseite zu warten
Semaphore webpagesCounter = // Ergänzung 2

// Semaphore, um zu warten, wenn der Webseitenpuffer voll ist
Semaphore webpagesBuffer = // Ergänzung 3

// Semaphore für den Zugriffsschutz auf den Puffer für URLs
Semaphore urlsAccess = // Ergänzung 4

// Semaphore, um bei leerem Puffer auf eine URL zu warten
Semaphore urlsCounter = // Ergänzung 5
```

b) Ergänzen Sie die folgende Vorlage für die Klasse `WebpageLoader`.

```
public WebpageLoader extends Thread {
    public void run() {
        while (active) {
            load();
        }
    }
    private void load() {
        // Ergänzung 6
        URL url = urls.removeFirst();
        // Ergänzung 7
        Webpage webpage = loadWebpage(url);
        // Ergänzung 8
        webpages.add(webpage);
```

Fortsetzung nächste Seite!

```
        // Ergänzung 9
    }
    [...]
}
```

c) Ergänzen Sie die folgende Vorlage für die Klasse `URLExtractor`.

```
public URLExtractor extends Thread {
    public void run() {
        while (active) {
            load();
        }
    }
    private void load() {
        // Ergänzung 10
        Webpage webpage = webpages.removeFirst();
        // Ergänzung 11
        List<URL> newURLs = extractURLs(webpage);
        for (URL newURL: newURLs) {
            // Ergänzung 12
           urls.add(newURL);
            // Ergänzung 13
        }
    }
    [...]
}
```

d) Der Puffer für Webseiten kann bis zu 12 Webseiten fassen. Die Größe des Puffers für extrahierte URLs ist dagegen bisher nicht begrenzt. Was kann passieren, wenn man die Größe dieses Puffers ebenfalls beschränkt (z.B. auf 100 URLs)?

Fortsetzung nächste Seite!

5. Maschinennahe Programmierung

Betrachten Sie folgende stark vereinfachte Maschine:

- Es gibt 8 Register R1 - R8 mit je 1 Byte Breite.
- Einziges Datenformat ist eine 1 Byte lange Ganzzahl.
- Folgende Befehle sind verfügbar:

Befehl	Pseudocode	Erläuterung
`MOV Rx, Ry`	`Rx := Ry`	Kopiert den Inhalt von Register Ry in Register Rx.
`MOV Rx, i`	`Rx := i`	Lädt die Zahl i in Register Rx. Die Zahl i kann auch negativ sein.
`MOV Rx, [Ry]`	`Rx := Hauptspeicher[Ry]`	Lädt den Wert der Hauptspeicheradresse, die in Register Ry steht, in Register Rx.
`MOV [Rx], Ry`	`Hauptspeicher[Rx] := Ry`	Kopiert den Wert von Ry in den Hauptspeicher an die Adresse, die in Rx steht.
`ADD Rx, Ry, Rz`	`Rx := Ry + Rz`	Weist dem Register Rx die Summe der Werte der Register Ry und Rz zu.
`ADD Rx, Ry, i`	`Rx := Ry + i`	Weist dem Register Rx die Summe des Wertes des Registers Ry und der Zahl i zu. Die Zahl i kann auch negativ sein.
`marke:`	`marke:`	Definiert eine Marke, zu der gesprungen werden kann.
`JMP marke`	`goto marke`	Springt zu einer definierten Marke *marke*.
`JZ Rx, marke`	`if Rx = 0 then goto marke`	Springt zu einer Marke *marke*, wenn das Register Rx den Wert 0 enthält.
`JLE Rx, Ry, marke`	`if Rx <= Ry then goto marke`	Springt zu einer Marke *marke*, wenn der Wert des Registers Rx kleiner oder gleich dem Wert des Registers Ry ist.
`JG Rx, Ry, marke`	`if Rx > Ry then goto marke`	Springt zu einer Marke *marke*, wenn der Wert des Registers Rx größer als der Wert des Registers Ry ist.

- Jede Zeile eines Programms enthält genau einen Befehl.
- Jede Zeile kann am Anfang eine Marke gefolgt von einem Doppelpunkt enthalten, um als Sprungziel zu dienen.
- Beispiel für ein Programm:

```
        MOV R1, 100
        MOV R2, [R1]
        MOV R3, 3
        JG R2, R3, mind4
        MOV R2, 4
mind4:  MOV [R1], R2
```

Fortsetzung nächste Seite!

In dieser Aufgabe werden Sie in vier Schritten den Bubblesort-Sortieralgorithmus in Assembler programmieren.

a) Schreiben Sie ein Assemblerprogramm, welches das Datum von der Hauptspeicheradresse R1 + R4 in Register R5 und das Datum von der Hauptspeicheradresse R1 + R4 + 1 in Register R6 lädt. Die Werte von R1 und R4 dürfen nicht verändert werden. Verwenden Sie R7 als Hilfsregister für die Adressberechnung.

Pseudocode:

```
R5 := Hauptspeicher[R1 + R4]
R6 := Hauptspeicher[R1 + R4 + 1]
```

b) Schreiben Sie ein Assemblerprogramm, welches den Wert von R5 an Adresse R1 + R4 in den Hauptspeicher schreibt und den Wert von R6 an Adresse R1 + R4 + 1. Die Werte von R1 und R4 dürfen nicht verändert werden. Verwenden Sie R7 als Hilfsregister für die Adressberechnung.

Pseudocode:

```
Hauptspeicher[R1 + R4] := R5
Hauptspeicher[R1 + R4 + 1] := R6
```

c) Schreiben Sie ein Assemblerprogramm, welches, falls der Wert von R5 größer ist als der Wert von R6, die Werte der Register R5 und R6 vertauscht und R3 auf 1 setzt. Verwenden Sie R7 als Hilfsregister für die Vertauschung.

Pseudocode:

```
if (R5 > R6) then
    swap(R5, R6)
    R3 := 1
endif
```

d) Schreiben Sie ein Assemblerprogramm, welches in einer Schleife den Wert von R4 von 0 bis zum Wert von R2 - 2 hochzählt.
In der Schleife wird der Code aus (a), (c) und (b) ausgeführt. Kopieren Sie dabei nicht den bereits geschriebenen Code, sondern verwenden die Platzhalter (a), (c) und (b).

Pseudocode:

```
for R4 := 0 to R2 - 2 do
  (a)
  (c)
  (b)
endfor
```

Fortsetzung nächste Seite!

Den geschriebenen Assemblercode können wir nun folgendermaßen zum Bubblesort-Algorithmus vervollständigen.

Pseudocode:

```
R1 := <Startadresse des Arrays im Hauptspeicher>
R2 := <Länge des Arrays>
R3 := 1
while R2 != 0 and R3 != 0 do
    R3 := 0
    (d)
    R2 := R2 - 1
endwhile
```

Assemblerprogramm:

```
          MOV R1, <Startadresse des Arrays im Hauptspeicher>
          MOV R2, <Länge des Arrays>
          MOV R3, 1
while:    JZ R2, endwhile
          JZ R3, endwhile
          MOV R3, 0
          (d)
          ADD R2, R2, -1
          JMP while
endwhile: MOV R7, 0
```

Fortsetzung nächste Seite!

Teilaufgabe 2:

1. **Speichermanagement**

a) **Virtueller Speicher - Seiten und Kacheln**

Ein 32 *Bit*-System habe einen 1 *Gigabyte* großen physischen Speicher. Eine Kachel beziehungsweise Seite habe eine Größe von je 4 *Kilobyte*.

i) In wieviele Kacheln ist der physische Speicher unterteilt?

ii) In wieviele Seiten ist der virtuelle Speicher unterteilt?

b) **Virtueller Speicher - Seitenkacheltabellen**

Betrachtet wird wiederum dasselbe 32 *Bit*-System mit entsprechend großem virtuellen Adressraum und einer Seitengröße von 4 *Kilobyte*.

i) Wie groß ist die Seitenkacheltabelle für jeden Prozess, wenn pro Eintrag 8 *Byte* benötigt werden?

Um Speicherplatz zu sparen, soll eine zweistufige Hierarchie von Seitenkacheltabellen eingesetzt werden. Für das *Page Directory* werden nun 7 *Bit* verwendet.

ii) Wieviele Einträge hat dann jede Seitenkacheltabelle der zweiten Stufe?

iii) Wieviel Speicher wird für die Verwaltung des virtuellen Adressraums *mindestens* benötigt, das heißt wenn von einem Prozess nur eine einzige Seite belegt würde? Die Größe eines jeden Eintrags sei wiederum 8 *Byte*.

c) **Seitenersetzung - Optimale Strategie**

Für die notwendige Ersetzung von Seiten in den Kacheln des physischen Speichers gibt es verschiedene Strategien, so zum Beispiel *First In First Out*, *Least Frequently Used* oder *Least Recently Used* etc. Was wäre die optimale Strategie und findet diese in der Praxis ihren Einsatz?

d) **Seitenersetzung - Second Chance-Strategie**

Außer den vorgenannten Strategien zur Seitenersetzung gibt es noch die sogenannte *Second Chance*-Strategie. Das Betriebssystem macht sich hierbei diejenigen zwei Bits eines jeden Seitenkacheltabellen-Eintrags zunutze, die angeben, ob lesend (*Read-Bit*) oder schreibend (*Dirty-Bit*) auf eine Seite zugegriffen wurde, um das Working Set der Seiten eines Prozesses abzuschätzen.

Während letzteres Bit benötigt wird, um zu bestimmen, ob eine Seite auf den Hintergrundspeicher geschrieben werden muss, bevor sie ausgetauscht werden kann, wird aufgrund des ersteren entschieden, ob eine Seite eine zweite Chance bekommen und somit vorerst noch nicht ersetzt werden soll.

Im Folgenden wird ein System mit 7 Seiten und 4 Kacheln betrachtet. Beim Einlagern sowie bei Zugriffen auf eine Seite wird deren *Read-Bit* gesetzt, im Falle einer zweiten Chance aber gelöscht.

Fortsetzung nächste Seite!

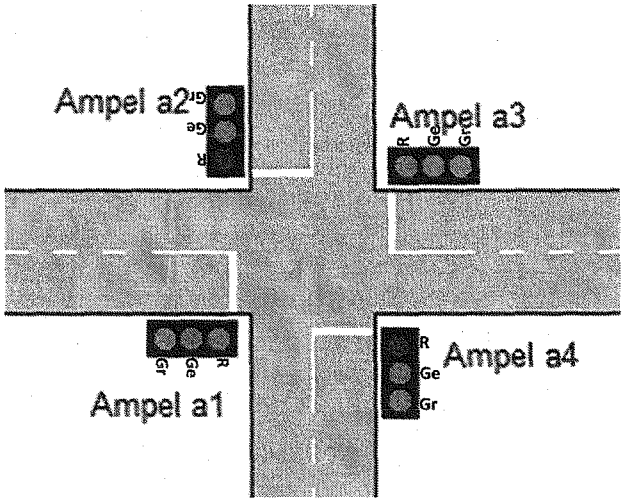
i) Zeigen Sie den Verlauf der Second Chance-Strategie anhand der folgenden *Lese*-Zugriffe in der angegebenen Reihenfolge. Notieren Sie das *Read-Bit* als Superskript, so zum Beispiel als 2^1, falls das Bit für die Seite 2 gesetzt ist, und als 2^0, falls dies nicht der Fall sein sollte.

	Kacheln				Page Queue				
Zugriff	f_1	f_2	f_3	f_4	1	2	3	4	Pagefault
1	1	-	-	-	1^1	-	-	-	1
3	1	3	-	-	3^1	1^1	-	-	1
7	1	3	7	-	7^1	3^1	1^1	-	1
4									
3									
5									
2									
7									
3									
2									
5									

ii) Allgemein könnte man annehmen, dass bei der Verwendung von mehr Kacheln im Gegenzug weniger Seitenzugriffsfehler auftreten. Ist das wirklich immer so? Begründen Sie Ihre Antwort kurz!

Fortsetzung nächste Seite!

2. Petrinetze



Zur Regelung des Verkehrs an einer Kreuzung werden vier Ampeln eingesetzt. Schräg gegenüberliegende Ampeln (d.h. Ampel a1 und a3 sowie a2 und a4) haben stets identische Farbbilder. Die Signalabfolge jeder Ampel lautet:

- Grün: Die Einfahrt ist freigegeben.
- Gelb: Achtung, es wird gleich Rot-Kreuzung freigeben.
- Rot: Die Einfahrt ist verboten.
- Rot-Gelb: Übergang von Rot nach Grün. Gleich wird die Erlaubnis zur Fahrt gegeben.

a) **Petrinetz**

Zeichnen Sie ein Petri-Netz, das die Signalabfolgen der Ampeln a1 und a2 modelliert! Stellen Sie durch entsprechende Modellierung sicher, dass die Ampeln a1 und a2 nur bei Signalbild Rot den selben Zustand haben! Zudem soll das Netz derart aufgebaut sein, dass dieselbe Ampel mehrere Male die komplette Signalfolge durchlaufen kann, ohne dass inzwischen die andere Ampel schalten muss. Vergessen Sie nicht, eine Anfangsmarkierung anzugeben! Erläutern Sie kurz die Funktion der verwendeten Stellen und Transitionen!

b) **Kritischer Bereich**

Kennzeichnen Sie diejenigen Stellen in Ihrem Petrinetz, die den kritischen (exklusiven) Bereich darstellen!

c) **Erreichbarkeitsgraph**

Geben Sie den Erreichbarkeitsgraphen für Ihr Petrinetz an!

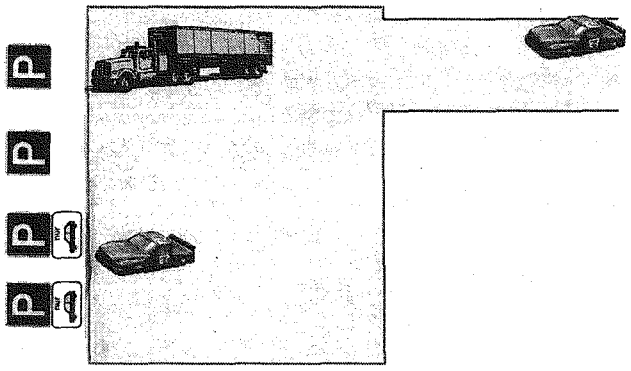
d) **Verklemmungen**

Diskutieren Sie mit Hilfe des Erreichbarkeitsgraphen, ob es im Netz zu Verklemmungen kommen kann!

Fortsetzung nächste Seite!

3. Semaphore

In dieser Aufgabe geht es um das Konzept der Semaphore am Beispiel eines Parkhauses. Fahrzeuge (im Falle von Betriebssystemen: Prozesse) fahren in ein Parkhaus mit 4 Parkplätzen ein und aus. Die Ein- und Ausfahrt ist so schmal, dass sie nur von einem Fahrzeug passiert werden kann. Dabei muss immer gelten, dass sich maximal so viele Fahrzeuge im Parkhaus befinden (inklusive eines Fahrzeuges in der Ein- und Ausfahrt), wie Stellplätze vorhanden sind. Ein Fahrzeug darf die Einfahrt also nicht rückwärts wieder verlassen, weil kein Stellplatz frei ist. Von den 4 Stellplätzen des Parkhauses sind nur 2 groß genug, dass auch LKWs dort abgestellt werden können. Generell darf auf einem Stellplatz nur ein Fahrzeug stehen. Die folgende Abbildung zeigt einen Beispielzustand des Parkhauses mit 2 belegten Stellplätzen.



a) **Semaphore - Funktionsweise**

Geben Sie informell die Funktionsweise eines Semaphors an!

b) **Deadlockbedingungen**

Nennen Sie die 4 Deadlock-Bedingungen! Geben Sie für jede Bedingung ein Beispiel, inwiefern sie in dem Parkhausszenario erfüllt wird!

c) **Synchronisation mit Semaphoren**

Für sich alleine betrachtet laufen die beteiligten Prozesse folgendermaßen ab:

```
Process PKW
{
 while (TRUE)
 {
  //PKW vor Parkhaus
  <durch Ein- und Ausfahrt fahren>;
  <Parken>;
  <durch Ein- und Ausfahrt fahren>;
  //PKW vor Parkhaus
 }
}
```

```
Process LKW
{
 while (TRUE)
 {
  //LKW vor Parkhaus
  <durch Ein- und Ausfahrt fahren>;
  <Parken>;
  <durch Ein- und Ausfahrt fahren>;
  //LKW vor Parkhaus
   }
 }
```

Die beiden Prozesse sollen nun synchronisiert werden, indem (in Pseudocode-Notation) in geeigneter Weise Semaphore deklariert und Semaphor-Operationen eingefügt werden.

Fortsetzung nächste Seite!

Die Pseudocode-Operation zum Deklarieren eines Semaphors mit Namen `name` mit Startwert `n` lautet:

```
name(n);
```

Die Pseudocode-Operationen zum Aufrufen von P und V auf dem Semaphore mit Namen `name` lauten:

```
P(name)
V(name)
```

Des weiteren sei ein boolescher Wert `großerStellplatz` gegeben, über den stets bestimmt werden kann, ob ein Fahrzeug im Begriff ist, sich auf einen großen Stellplatz zu stellen, beziehungsweise ob es auf einem großen Stellplatz steht:

if (`großerStellplatz`) then *Pseudocode-Operation*

i) Zu Beginn sei das Parkhaus leer. Listen Sie alle benötigten Semaphoren auf und erklären Sie ihren Zweck beziehungsweise ihre Pragmatik!

ii) Fügen Sie in den oben aufgelisteten Pseudocode an geeigneter Stelle entsprechende Deklarationen sowie Aufrufe von P und V Operationen ein!

iii) Erläutern Sie kurz, warum Ihre Lösung die Bedingung der *Mutual Exclusion* erfüllt!

Fortsetzung nächste Seite!

4. Dateisysteme

a) Kleine Blockgrößen

Beim Anlegen eines Dateisystems kann die zu verwendende Blockgröße unterschiedlich gewählt werden. Geben Sie je einen Vor- und Nachteil von kleinen Blockgrößen an!

b) Unix Calls

Nennen und beschreiben Sie insgesamt vier UNIX-Systemaufrufe zur Manipulation von Dateisystemen, Dateien oder Verzeichnissen!

c) Baumstruktur und Links

Üblicherweise sind UNIX-basierte Dateisysteme hierarchisch in Baumstrukturen gegliedert. Diese Baumstrukturen bilden gerichtete azyklische Graphen. Leider werden diese Strukturen durch Links zerstört, d. h. die Eigenschaft der Zyklenfreiheit geht verloren. Warum wird dies dennoch durchgeführt? Nennen Sie je einen Vor- und Nachteil, der durch das Zulassen von Links entsteht!

d) Dateinamen

Wo speichert ein NTFS-basiertes Dateisystem einen Dateinamen? Wo ein UNIX-basiertes Dateisystem?

e) Inode-Struktur

Skizzieren Sie graphisch eine UNIX-inode-Struktur! Gehen Sie davon aus, dass ein inode zehn Blöcke direkt adressieren kann, über je einen Eintrag für einfach indirekte, zweifach indirekte und dreifach indirekte Adressen verfügt. Metadaten müssen nicht vollständig aufgezählt werden – drei Beispiele genügen. Machen Sie in Ihrer Skizze den Unterschied zwischen direkten, einfach indirekten, zweifach indirekten und dreifach indirekten Adressen deutlich und erläutern Sie, wozu diese verwendet werden!

Fortsetzung nächste Seite!

5. Prozesse, Threads und wechselseitiger Ausschluss

a) Bedingungen an WA-Lösung

Nennen und erläutern Sie kurz vier Bedingungen, die eine gute Lösung zur Garantie des wechselseitigen Ausschlusses für Prozesse erfüllen sollte!

b) Thread ↔ Prozess

i) Erläutern Sie den Unterschied zwischen Thread und Prozess! Gehen Sie dabei auf deren Beziehung zueinander ein!

ii) Nennen Sie 2 Beispiele für Informationen bzw. Datenstruktur-Instanzen, die für jeden (User-)Thread spezifisch sind!

iii) Nennen Sie 2 Beispiele für Informationen bzw. Datenstruktur-Instanzen, die für jeden Prozess spezifisch sind (die also (User-)Threads gemeinsam nutzen müssen)!

c) Gründe für Threads

Nennen Sie vier Gründe für die Einführung von Threads!

d) Wechselseitiger Ausschluss

Sind folgende Ansätze zum wechselseitigen Ausschluss von Prozessen wirksam? Begründen Sie kurz! Wenn Sie wirksam sind, welche Nachteile haben Sie?

- Abschalten (Disabling) von Interrupts
- Einfache Lock Variable ohne TSL

P1 and P2

```
while (TRUE){
    if(lock == 0){
       lock = 1;
       critical_region();
       lock = 0;
    }
    non_critical_region();
}
```

- Lock Variable mit TSL

P1 and P2

```
enterCriticalRegion:
        TSL REGISTER, LOCK          copy lock to register and set lock to 1
        CMP REGISTER, #0            was lock zero?
        JNE enterCriticalRegion     if it was not zero (lock was set): loop
        RET                         return

leaveCriticalRegion:
        MOVE LOCK, #0               store 0 in lock
        RET                         return
```

- Strict Alteration

P1

```
while (TRUE){
    while(turn!=0);
    critical_region();
    turn = 1;
    non_critical_region();
}
```

P2

```
while (TRUE){
    while(turn!=1);
    critical_region();
    turn = 0;
    non_critical_region();
}
```

66114

Datenbank- und Betriebssysteme (vertieft)

Frühjahr 2011

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____ Kennwort: _____ Arbeitsplatz-Nr.: _____	**Frühjahr 2011**	**66114**

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

— Prüfungsaufgaben —

Fach: **Informatik (vertieft studiert)**

Einzelprüfung: **Datenbank- und Betriebssysteme**

Anzahl der gestellten Themen (Aufgaben): **4 Aufgaben, von denen zwei gemäß untenstehender Auswahlregel zu bearbeiten sind!**

Anzahl der Druckseiten dieser Vorlage: **23**

Zu den zwei Themenschwerpunkten A (Datenbanksysteme) und B (Betriebssysteme) ist jeweils entweder die Teilaufgabe 1 oder 2 zu wählen. Auf der Vorderseite des Kopfbogens sind im Feld „gewähltes Thema: Nr." die Nummern der beiden gewählten Aufgaben anzugeben (z. B. A2, B1)!

Bitte wenden!

Themenschwerpunkt A
(Datenbanksysteme)

Teilaufgabe 1:

1. ER-Modellierung

Erstellen Sie ein ER-Modell für die Datenbank einer Firmenkantine:

- Die Kantine hat eine reichhaltige Auswahl an Gerichten. Jedes Gericht hat eine Nummer, einen Namen und einen Preis. Gerichte, die sich in ihrer Zusammenstellung nur leicht (z. B. aufgrund einer Beilage) unterscheiden, können den gleichen Namen haben.

- Die Kantine bietet täglich aus ihrer Auswahl mehrere Gerichte an. Sie möchte Informationen darüber haben, welches Gericht an welchem Tag wie oft verkauft wurde.

- Jedes Gericht besteht aus mehreren Bestandteilen: genau einem Hauptbestandteil (z.B. Fleisch) und mehreren Beilagen (z.B. Suppe, Salat, Nachspeise). Es ist wichtig zu erkennen, ob es sich bei einem Bestandteil um einen Hauptbestandteil oder um eine Beilage handelt. Die Beilagen sind fest und können nicht ausgewählt werden. Die Zusammensetzung eines Gerichts ist nur für die Vorbereitung von Bedeutung, da jeder Bestandteil einzeln zubereitet wird. Jeder Bestandteil kann in verschiedenen Gerichten verwendet werden, wobei Redundanzen vermieden werden sollen.

- Für jeden Bestandteil eines Gerichts ist ein Rezept vorhanden.

- Ein Bestandteil besteht aus mehreren Zutaten (z.B. Zucker, Mehl, ...). Jede Zutat hat einen Namen und eine Kalorienanzahl (pro 100g).

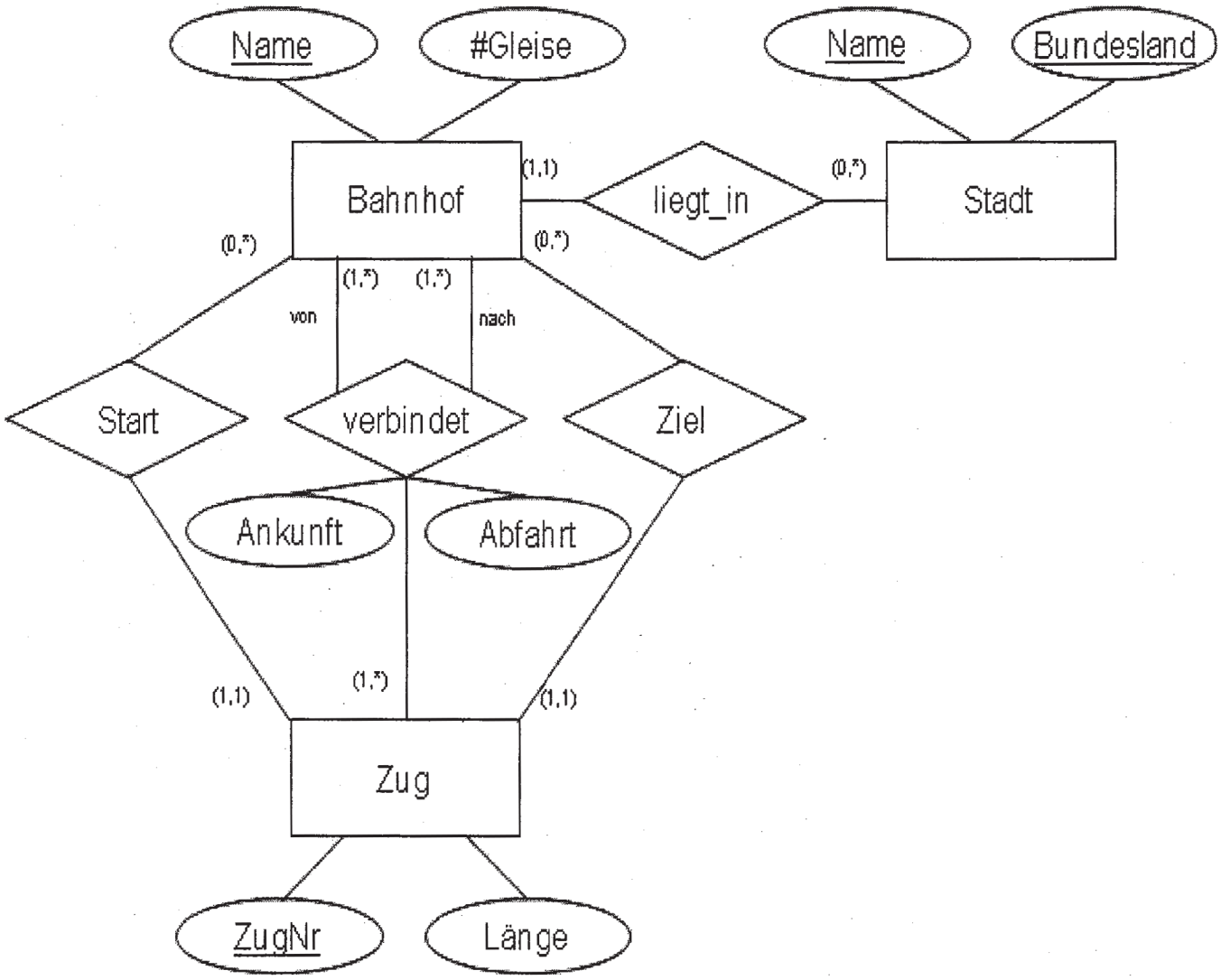
- Zu guter Letzt soll die Menge einer Zutat, die in einem Bestandteil eines Gerichts verwendet wird, in der Datenbank erfasst werden.

Schlüsselattribute werden durch Unterstreichen gekennzeichnet. Kardinalitäten von Beziehungen sollen in (min, max)-Notation in das Diagramm mit aufgenommen werden.

Fortsetzung nächste Seite!

2. Relationenmodell

Gegeben sei im Folgenden eine ER-Modellierung von Zugverbindungen. Übertragen Sie das ER-Modell in ein relationales Schema und kennzeichnen Sie durch Unterstreichen die gewählten Primärschlüssel. Verfeinern Sie schließlich das relationale Schema soweit wie möglich durch Eliminierung von Relationen.



Fortsetzung nächste Seite!

3. Relationale Anfragen

Gegeben sei die folgende relationale Datenbank des Online-Blumenhändlers „Flowers4you“:

Das Warenangebot an Blumenbouquets findet sich in der Tabelle „Angebot“. Jedes Produkt hat eine Produktnummer (PNr), einen Produktnamen (PName), eine Produktinformation (PInfo) und einen Preis (Preis) in Euro:

Angebot			
PNr	*PName*	*PInfo*	*Preis*
1	Wolkenlos	12 Mini-Sonnenblumen	19,99
2	Sonnenstrahl	20 gelbe Rosen	25,99
3	Romantico	12 rote Rosen	19,99
4	Leuchtfeuer	5 weiße Rosen, 5 rote Rosen, 5 gelbe Rosen	22,99
...	...	...	...

Die Kundendaten Kundennummer (KNr), Name (Name), Adresse (Adr), Stadt (Stadt), Postleitzahl (PLZ), Land (Land), Kreditkarte (KK), Kreditkartennummer (KKNr) und das Ablaufdatum der Kreditkarte (AD) werden in der Tabelle „Kunde“ gespeichert:

Kunde								
KNr	*Name*	*Adr*	*Stadt*	*PLZ*	*Land*	*KK*	*KKNr*	*AD*
1001	Robert Rosenthal	Blumenstr. 12	München	81929	DE	Visa	78574	09/11
2035	Willibald Wiese	Nelkenstr. 23	Wiesenthal	09484	DE	Visa	38521	05/10
3023	Florence Fleur	Rue des fleurs 2	Lyon	76849	FR	American Express	80765	08/10
...	...	...	...	...	...	...	...	...

Kundenaufträge werden in der Tabelle „Bestellung“ verwaltet. Jeder Kundenauftrag besteht aus einer Bestellnummer (BstNr), einer Produktnummer (PNr), einer Kundennummer (KNr), einem Bestelldatum (Datum) und den Adressinformationen des Empfängers der Bestellung (EName, EAdr, EStadt, EPLZ, ELand):

Bestellung								
BstNr	*PNr*	*KNr*	*Datum*	*EName*	*EAdr*	*EStadt*	*EPLZ*	*ELand*
00001	3	3023	09.05.2010	Florence Fleur	Rue des fleurs 2	Lyon	76849	FR
00002	3	1001	02.03.2010	Nina Wiese	Blumenfeld 3	München	81929	DE
00003	1	1001	23.10.2010	Otto Lilienthal	Grasweg 7	Nürnberg	90001	DE
...	...	...	...	...	...	...	...	...

Fortsetzung nächste Seite!

a) Formulieren Sie die folgenden Anfragen in voller Allgemeinheit (d.h. nicht in Bezug auf die obigen Beispieldaten) in **relationaler Algebra**:

i. Alle Produkte, die über 30 Euro kosten.
ii. Die Empfängernamen und Empfängeradressen aller Bestellungen ab dem 01.01.2010, die nach München geliefert werden.
iii. Den Namen und die Stadt aller Kunden, die das Produkt mit der Produktnummer 12 bestellt haben.
iv. Die Nummern aller Bestellungen, bei denen der Besteller und der Empfänger in derselben Stadt wohnen.

Was ist der Fehler in folgender relationaler Anfrage? Erklären Sie diesen kurz!

$$\pi_{PInfo}(Kunde \ |\times| \ Bestellung)$$

b) Formulieren Sie die folgenden Anfragen in voller Allgemeinheit (d.h. nicht in Bezug auf die obigen Beispieldaten) in **SQL**:

i. Die Empfängernamen und Empfängeradressen aller Bestellungen vom 01.01.2010, die nach Nürnberg geliefert werden.
ii. Die Bestellnummer und die zugehörigen Informationen (EName, EStadt, ELand) aller Bestellungen, die über 40 Euro kosten.
iii. Die Produktnummern der Produkte, die unter dem Durchschnittspreis aller angebotenen Produkte liegen.
iv. Die Anzahl an registrierten Kunden pro Stadt.
v. Die Anzahl an Bestellungen pro Stammkunde (Kunden mit mehr als 5 Bestellungen) mit dem jeweils **im Durchschnitt** ausgegebenen Geldbetrag.

Fortsetzung nächste Seite!

4. Normalformen

Gegeben sei der folgende Ausschnitt aus einer relationalen Flugbuchungs-Datenbank:

Buchung								
KNr	*KName*	*KOrt*	*Abflug*	*Flugnr*	*Abflugort*	*Zielort*	*Gesellschaft*	*Gezahlt*
18	Hans Meier	Berlin	10.10.2010/ 12:25	LH723	Frankfurt	New York	Lufthansa	VISA
42	Markus Baum	München	23.5.2010/ 9:30	BA534	München	Paris	British-Airways	Bar
48	Johann Hartl	Hamburg	26.9.2010/ 17:45	BA744	Hamburg	London	British-Airways	Konto
...	...	...	...	...	...	...	...	...

Wobei die folgenden funktionalen Abhängigkeiten gelten sollen:

KNr → KName, KOrt
Flugnr → Abflugort, Zielort, Gesellschaft

a) Geben Sie sämtliche Schlüsselkandidaten der Relation Buchung an!
b) Welchen Normalformen genügt die Relation Buchung? Begründung!
c) Zeigen Sie anhand von Beispielen die drei möglichen Anomalien auf, die bei der Verwendung der Relation Buchung auftreten können!
d) Erklären Sie, wie man die Relation „Buchung" verändern müsste, damit sie in dritter Normalform vorliegt! Geben Sie auch eventuell neue Relationen oder Attribute an, die dafür nötig sein könnten!

5. Transaktionsverwaltung und Mehrbenutzersynchronisation

a) Nennen und beschreiben Sie die vier grundlegenden Eigenschaften einer Datenbanktransaktion!
b) Geben Sie zwei Fehler an, die im unkontrollierten (und nicht synchronisierten) Mehrbenutzerbetrieb auftreten können und beschreiben Sie diese jeweils anhand eines Beispiels!

Teilaufgabe 2:

1. E/R Modellierung, SQL-DDL

Für den Buchhandel Dubenhugel soll ein System zur Verwaltung entworfen werden:

- Das Unternehmen besitzt mehrere Filialen, welche eindeutig durch die Filialen-ID gekennzeichnet sind. Jede Filiale hat eine Adresse (Strasse, Hausnummer, Ort und Postleitzahl) und einen Filialleiter. Daneben ist jede Filiale in mehrere Abteilungen thematisch untergliedert.
- Jede Abteilung hat eine eindeutige Abteilungs-ID und einen Abteilungsleiter.
- Jeder Mitarbeiter von Dubenhugel ist mindestens einer Abteilung zugeordnet. Für jeden Mitarbeiter sollen neben dem Namen auch die Adresse (gleiche Attribute wie Filialadresse), die Telefonnummer und das Geburtsdatum abgespeichert werden.
- Außerdem sollen in dem System die momentan vorhandenen Bücher gespeichert werden. Ein Buch hat eine ISBN-Nummer und einen Titel. Jedes Buch wurde von mindestens einem Autor verfasst. Zu jedem Buch soll bekannt sein, in welcher Filiale und Abteilung es sich befindet.
- Zu jeder Abteilung soll gespeichert werden, welche Bücher in welcher Stückzahl vorhanden sind.
- Zu den Autoren in der Datenbank soll der Name, das Geburtsdatum und der Geburtsort abgelegt werden.
- Zusätzlich sind zu jedem Buch ähnliche Bücher bekannt. Auch diese Information soll in das System einfliessen.

a) Erstellen Sie ein Entity-Relationship-Diagramm für obige Datenbank.

b) Setzen Sie das gegebene E/R-Diagramm in ein entsprechendes relationales Datenbankschema um. Identifizieren Sie dazu zunächst weitere Attribute, die in obigem Diagramm noch nicht enthalten sind und die gegebenen Entities in sinnvoller Weise beschreiben. Es sollten mindestens zwei Attribute pro Entity angegeben werden. Geben Sie die resultierenden Relationenschemata in folgender Schreibweise an:

```
Relation (Attribut1, Attribut2, ..., AttributN)
```

Identifizieren Sie für jede Relation einen Primärschlüssel und unterstreichen Sie diesen. Achten Sie auf eine geeignete Modellierung der Beziehungen (Relationships).

Fortsetzung nächste Seite!

2. Relationale Algebra

Gegeben sei das folgende relationale Datenbank-Schema zur Verwaltung von Zehnkampf-Wettbewerben. Die Primärschlüssel-Attribute sind jeweils unterstrichen.

"Athlet":

ANr	Name	Land
A1	Roman Sebrle	Tschechien
A2	Bryan Clay	USA
A3	Tomas Dvorak	Tschechien
A4	Dan O'Brien	USA
A5	Dimitri Karpow	Kasachstan

"Disziplin":

DNr	Bezeichnung	Art
D1	100m	Lauf
D2	Weitsprung	Sprung
D3	Kugelstossen	Wurf
D4	Hochsprung	Sprung
D5	400m	Lauf
D6	110m Hürden	Lauf
D7	Diskuswerfen	Wurf
D8	Stabhochsprung	Sprung
D9	Speerwerfen	Wurf
D10	1500m	Lauf

"Wettkampf":

WNr	Bezeichnung	Jahr
W1	Olympische Spiele	2004
W2	Olympische Spiele	2008
W3	Weltmeisterschaft	2004
W4	Weltmeisterschaft	2006
W5	Weltmeisterschaft	2008

"Teilnahme":

ANr	DNr	WNr	Platzierung
A1	D2	W1	1
A1	D10	W1	4
A1	D7	W4	2
A2	D2	W1	3
A2	D2	W2	1
A3	D10	W1	2
A4	D7	W1	5
A4	D2	W3	2
A5	D3	W1	4
A5	D10	W4	3

Formulieren Sie die folgenden Anfragen in relationaler Algebra:

a) Finden Sie die Namen aller Athleten aus Tschechien.

b) Geben Sie Bezeichnungen und Jahre aller Wettkämpfe aus, an denen Athleten aus den USA teilgenommen haben.

c) Gesucht sind die Bezeichnungen aller Disziplinen, für die Ergebnisse für die Weltmeisterschaft aus dem Jahr 2004 vorliegen.

Geben Sie das Ergebnis der folgenden Ausdrücke der relationalen Algebra als Tabellen an:

d) $\pi_{DNr}(Teilnahme) \bowtie Disziplin$

e) $\pi_{ANr,DNr,Platzierung}(\sigma_{Platzierung<4}(Teilnahme)) \bowtie \pi_{DNr,Art}(Disziplin)$

Fortsetzung nächste Seite!

3. SQL

Gegeben sei das relationale Datenbank-Schema aus Aufgabe 2.

"Athlet":

ANr	Name	Land
A1	Roman Sebrle	Tschechien
A2	Bryan Clay	USA
A3	Tomas Dvorak	Tschechien
A4	Dan O'Brien	USA
A5	Dimitri Karpow	Kasachstan

"Disziplin":

DNr	Bezeichnung	Art
D1	100m	Lauf
D2	Weitsprung	Sprung
D3	Kugelstossen	Wurf
D4	Hochsprung	Sprung
D5	400m	Lauf
D6	110m Hürden	Lauf
D7	Diskuswerfen	Wurf
D8	Stabhochsprung	Sprung
D9	Speerwerfen	Wurf
D10	1500m	Lauf

"Wettkampf":

WNr	Bezeichnung	Jahr
W1	Olympische Spiele	2004
W2	Olympische Spiele	2008
W3	Weltmeisterschaft	2004
W4	Weltmeisterschaft	2006
W5	Weltmeisterschaft	2008

"Teilnahme":

ANr	DNr	WNr	Platzierung
A1	D2	W1	1
A1	D10	W1	4
A1	D7	W4	2
A2	D2	W1	3
A2	D2	W2	1
A3	D10	W1	2
A4	D7	W1	5
A4	D2	W3	2
A5	D3	W1	4
A5	D10	W4	3

Formulieren Sie die folgenden Anfragen in SQL:

a) Finden Sie die Namen und Herkunftsländer aller Athleten, die bei der Teilnahme an einem Wettkampf den ersten Platz erreicht haben.

b) Geben Sie alle Attribute der Wettkämpfe aus, an denen kein Athlet teilgenommen hat.

c) Geben Sie die Nummern und Namen aller Athleten zusammen mit der Anzahl der verschiedenen Disziplinen, in der jeder Athlet an einem Wettkampf teilgenommen hat. Sortieren Sie das Ergebnis absteigend nach dieser Anzahl.

Wie sieht die Ergebnisrelation zu folgenden Anfragen aus?

d)
```
select a.ANr, b.Anr from Athlet a, Athlet b where a.Land = b.Land and a.Anr
<> b.ANr;
```

e)
```
select Bezeichnung from Disziplin where DNr in (select DNr from Teilnahme
where ANr not in (select ANr from Athlet where Land <> 'Tschechien'
and Land <> 'USA'));
```

Fortsetzung nächste Seite!

4. Entwurfstheorie

Gegeben sei die nachfolgende relationale Datenbank mit unterstrichenen Schlüsselattributen. Sie enthält die Daten von Bestellungen bei einem Versandhaus.

Relation "Bestellung":

Waren ID	Kunden ID	Zustelldienst ID	Kunden Name	Kunden Wohnort	Waren Bezeichnung	Waren Preis	Zustelldienst Name
1	1	3	Müller	München	Blumentopf	10	DHL
2	1	2	Müller	München	Decke	12	UPS
3	2	1	Huber	Nürnberg	Rasierer	26	FedEx
2	2	3	Huber	Nürnberg	Decke	12	DHL
5	3	2	Meier	Hamburg	Gläser	26	UPS
6	4	1	Meier	München	Block	8	FedEx

a) Beschreiben Sie kurz, welche Redundanzen in der Datenbank vorhanden sind und welche Anomalien auftreten können. Geben Sie ein Beispiel für jede Anomalie an.

b) Geben Sie für das obige Datenbankschema alle funktionalen Abhängigkeiten (inkl. der transitiven) an.

c) Überführen Sie das obige Relationenschema in die dritte Normalform.

d) Erläutern Sie kurz, welchen Nachteil Normalisierung allgemein für die Anfragebearbeitung hat.

Themenschwerpunkt B
(Betriebssysteme)

Teilaufgabe 1:

1. Verklemmung

a) Nennen Sie alle Bedingungen für das Auftreten von Verklemmungen (Deadlocks).

b) Fünf Philosophen haben in ihrer wöchentlichen Diskussionsrunde ein regelmäßiges Problem. Sie sitzen an einem runden Tisch, auf dem eine Schüssel mit Spaghetti steht. Zwischen jeweils zwei Philosophen liegt eine Gabel. Somit liegen insgesamt fünf Gabeln auf dem Tisch. Wenn ein Philosoph hungrig ist, greift er zuerst die Gabel links von seinem Teller und dann die Gabel rechts. Danach beginnt er zu essen. Ein Philosoph braucht immer zwei Gabeln zum Essen. Wenn er satt ist, legt er seine Gabeln wieder zurück. Danach diskutiert er mit den anderen Philosophen weiter. Liegt eine Gabel nicht an ihrem Platz, wartet ein Philosoph solange, bis die Gabel wieder verfügbar ist.

   i. Welches Problem tritt regelmäßig auf? Begründen Sie ihre Antwort in 3-4 Sätzen.

   ii. An manchen Abenden kann ein einzelner Philosoph nicht kommen. Somit sitzen die anderen nur zu viert an dem Tisch, der für fünf Philosophen gedeckt ist. Tritt das Problem aus a) an diesen Abenden auch auf? Begründen Sie ihre Antwort in 2-3 Sätzen.

   iii. Die fünf Philosophen haben sich nach längerer Diskussion auf folgendes Verfahren geeinigt. Sie installieren eine Tischglocke, welche alle fünf Minuten klingelt. Beim Klingeln muss der Philosoph, welcher am längsten seine Gabel(n) hält, diese auf den Tisch zurücklegen. Beweisen Sie in Form eines Widerspruchsbeweises, dass dieses Verfahren das Problem der Philosophen löst.

Fortsetzung nächste Seite!

2. Synchronisation

Für die Führungen in einem Museum soll ein neues Besucher-Warte-System eingeführt werden. Das Museum hat genau einen Museumsführer und einen Warteraum für maximal 10 Besucher. Wenn ein Besucher zum Museum gelangt, kann er nur dann den Warteraum betreten, wenn ein Platz frei ist. Der Zugang zum Warteraum ist ein langer schmaler Gang. Durch diesen kann immer nur in einer Richtung gegangen werden. Der Museumsführer nimmt immer genau einen Besucher bei einer Führung mit. Wenn gerade kein Besucher im Warteraum ist, legt sich der Museumsführer schlafen.

```
public interface Semaphore {
    public void wait();
    public void signal();
}

public class BinarySemaphore implements Semaphore {
    public BinarySemaphore() {
        [...]
    }
    [...]
}

public class CountingSemaphore implements Semaphore {
    public CountingSemaphore(int initialCount){
        [...]
    }
    [...]
}
```

Vervollständigen Sie die Codeschablonen in den Teilaufgaben a), b) und c), indem Sie die Kommentare der Form `// Ergänzung X` durch Code für die Initialisierung der verwendeten Semaphore bzw. für die Synchronisation der Threads der Klassen `Besucher` und `Museumsfuehrer` ersetzen. Sie können an den mit `// Ergänzung X` gekennzeichneten Stellen beliebig viele Codezeilen oder auch keine einfügen. Schreiben Sie Ihre Lösung in folgender Form auf:

Ergänzung	Code
1	`new BinarySemaphore()`
2	...
3	...
...	...

Ihre Lösung soll einen möglichst hohen Grad an Parallelität ermöglichen.

a) Vervollständigen Sie die folgenden Initialisierungen der Semaphore.

```
Codeschablone für die Initialisierung:
// Zugriff auf den Warteraum des Museums
Semaphore mutex = //  Ergänzung 1 ;
// Zählt die Besucher im Warteraum des Museums
Semaphore vistor = //  Ergänzung 2 ;
// Zählt die freien Plätze im Warteraum des Museums
Semaphore empty = //  Ergänzung 3 ;
```

Fortsetzung nächste Seite!

b) Ergänzen Sie die folgende Vorlage für die Klasse `Besucher`.

```
public Besucher extends Thread {
    public void run() {
        while (active) {
            museumsbesuch();
        }
    }
    private void museumsbesuch() {

        //  Ergänzung 4

        // Besucher nimmt im Warteraum Platz

        //  Ergänzung 5

    }
    [...]
}
```

c) Ergänzen Sie die folgende Vorlage für die Klasse `Museumsfuehrer`.

```
public Museumsfuehrer extends Thread {
    public void run() {
        while (active) {
            fuehren();
        }
    }
    private void fuehren() {

        //  Ergänzung 6

        // Besucher aus dem Warteraum holen.

        //  Ergänzung 7

        // Besucher durch das Museum führen.

        //  Ergänzung 8

    }
    [...]
}
```

Fortsetzung nächste Seite!

3. Prozess-Scheduling

a) Nennen Sie vier Optimierungsziele eines CPU-Schedulers.

Gegeben sind die folgenden Prozesse:

Prozess	P0	P1	P2	P3	P4
Ankunftszeit	0	1	2	3	4
Benötigte Rechenzeit	7	5	10	2	6

b) Führen Sie das Scheduling für die gegebenen Prozesse mit den Algorithmen *First Come First Served (FCFS)*, *Round Robin (RR)* (Zeitscheibe der Größe 2) und *präemptivem Shortest Remaining Time Next (SRTN, preemptive)* durch. Stellen Sie den Ablauf inklusive der Zustände der Prozesse (Wartend, Rechnend) als Gantt-Diagramm (Balkendiagramm) dar.
Hinweis: Bei Round Robin kann es vorkommmen, dass zum gleichen Zeitpunkt ein Prozess eintrifft und einem anderen Prozess die CPU entzogen wird. Gehen Sie in diesem Fall davon aus, dass der ankommende Prozess zuerst in die Queue eingeordnet wird. Gehen Sie desweiteren bei SRTN davon aus, dass für Prozesse mit gleichen verbleibenden Rechenzeiten FCFS verwendet wird.

c) Geben Sie für *Round Robin* die durchschnittliche Wartezeit und den Durchsatz an.

Fortsetzung nächste Seite!

4. Speicherverwaltung

In einem Betriebssystem ist eine Seitenrahmentabelle vorhanden, welche für maximal vier Seiten ausgelegt ist.

Gegeben ist eine Folge von Seitenzugriffen:

Seitenzugriffe P0 P1 P2 P3 P4 P2 P0 P1

Führen Sie die Seitenzugriffe unter Verwendung des Seitenersetzungsalgorithmus *Least Recently Used (LRU)* durch. Stellen Sie die Einträge in der Seitenrahmentabelle für jeden Zugriff dar. Geben Sie außerdem an, ob ein Seitenfehler aufgetreten ist.

Sie können Ihre Lösung zum Beispiel in Tabellenform angeben.

Seite	(Seite 1)	(Seite 2)	(Seite 3)	(Seite 4)	Seitenfehler
P0					
P1					
P2					
P3					
P4					
P2					
P0					
P1					

Teilaufgabe 2:

1. Seitenersetzung

In dieser Aufgabe sollen verschiedene Seitenersetzungsstrategien (Paging-Strategien) am konkreten Beispiel verglichen werden. Dabei sei die Menge der Seiten gegeben durch $N = \{A, B, C, D, E\}$. Die Menge der Seitenrahmen, die für die Speicherung der Seiten im Arbeitsspeicher zur Verfügung stehen, sei gegeben durch $F = \{f_0, f_1, f_2\}$. Auf die fünf Seiten der Menge N werde in folgender Reihenfolge zugegriffen:

{w = A E D E D B D E A D C B A E }

Der Arbeitsspeicher ist zu Beginn leer.

a) Ermitteln Sie die Anzahl der Seitenfehler für die Paging-Strategie LFU (Least Frequently Used), indem Sie alle Veränderungen im Speicher in der folgenden Tabelle dokumentieren. Tragen Sie außerdem alle Informationen in die Tabelle ein, die aufgrund der Paging-Strategie zusätzlich benötigt werden.

Referenzierte Seiten	f_0	f_1	f_2	Summe Seitenfehler
A				
E				
D				
E				
D				
B				
D				
E				
A				
D				
C				
B				
A				
E				

b) Nehmen Sie an, dass Sie nun die Anzahl der Seitenrahmen im Arbeitsspeicher erhöhen. Wie verhält sich dann die Anzahl der Seitenfehler? Begründen Sie ohne eine neue Tabelle anzufertigen.

Fortsetzung nächste Seite!

2. Prozesse

a) Welche drei Arten von Informationen enthält der Prozesskontrollblock (PCB)?

b) Was ist der Unterschied zwischen PCB und Prozess-Image?

c) Welche Vorteile ergeben sich, wenn ein Prozessmodell zwischen vielen verschiedenen Prozesszuständen differenziert?

d) Welche Nachteile ergeben sich, wenn ein Prozessmodell zwischen vielen verschiedenen Prozesszuständen differenziert?

e) In den folgenden Fragen soll das 5–Zustands–Prozessmodell betrachtet werden.

   i) Skizzieren Sie das 5–Zustands–Prozessmodell. Kennzeichnen Sie alle Übergänge durch Pfeile und benennen Sie die Übergänge.

   ii) Nummerieren Sie die Übergänge in Ihrer Skizze. Geben Sie für jeden Übergang ein Beispiel an.

Fortsetzung nächste Seite!

3. Multilevel Feedback Queuing (MLFQ)

In dieser Aufgabe soll die Scheduling-Strategie Multilevel Feedback Queuing genauer untersucht werden.

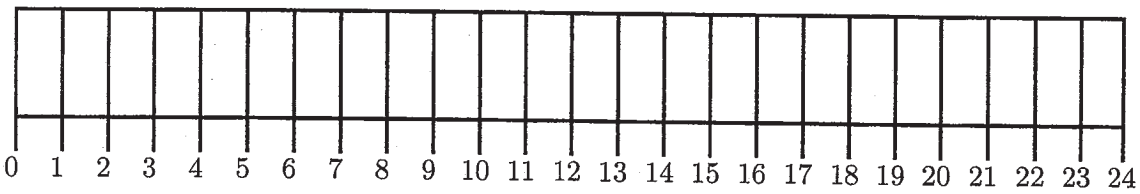
Gegeben seien die Prozesse A bis E mit den folgenden Ankunfts- und Rechenzeiten:

Prozess	Ankunftszeitpunkt	Rechenzeit
A	0	9
B	1	5
C	2	1
D	5	5
E	10	2

Es gilt: Der Ankunftszeitpunkt t bezeichnet auch den Zeitpunkt, zu dem der Prozess in die Warteschlange eingereiht wird (ohne jede Verzögerung). Wird ein Prozess vor seinem Terminieren zum Zeitpunkt t' unterbrochen, so reiht er sich in die entsprechende Warteschlange mit Ankunftszeit t' wieder ein. Sollten zwei Prozesse zum gleichen Zeitpunkt in die Warteschlange eingeordnet werden, so wird die lexikalische Ordnung auf den Prozessnamen herangezogen (Prozess A vor Prozess B, usw.).

a) Geben Sie für die **Strategie Multilevel Feedback Queueing (MLFQ)** an, wann welchem Prozess Rechenzeit zugeteilt wird und wann die Prozesse jeweils terminieren, indem Sie das nachfolgende Gantt-Chart vervollständigen. Gehen Sie von drei Prioritätsklassen (0 ist die höchste Priorität) mit folgenden Werten aus:

Priorität	Verfahren
0	Round-Robin mit Quantum 1
1	Round-Robin mit Quantum 3
2	First Come First Served (FCFS)



Fortsetzung nächste Seite!

b) Gegeben sei folgendes Gantt-Chart:

	A	A	A	B	B	C	C	B	B	D	D	E	E	B	B	D	D	E	E	B

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

Die Ankunftszeiten und Rechenzeiten der Prozesse sind dabei wie folgt gegeben:

Prozess	Ankunftszeitpunkt	Rechenzeit
A	0	3
B	2	7
C	2	2
D	5	4
E	7	4

Hinweis: es handelt sich dabei **nicht** um ein Gantt-Chart, das sich aus den gegebenen Prozessen nach Anwendung der vorhin beschriebenen MLFQ-Strategie ergibt.

Berechnen Sie für alle hier angegebenen Prozesse jeweils die Verweilzeit und die Wartezeit sowie (als Dezimalzahl mit einer Nachkommastelle) die mittlere Verweilzeit und die mittlere Wartezeit.

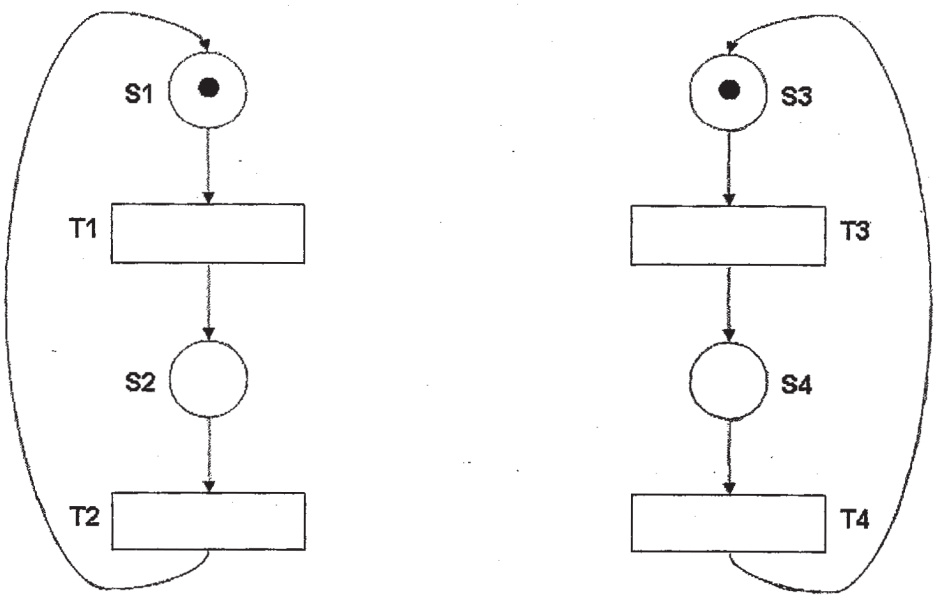
Fortsetzung nächste Seite!

4. Synchronisation nebenläufiger Prozesse

Wenn Prozesse nebenläufig ausgeführt werden, macht das eine geeignete Synchronisation dieser Prozesse erforderlich. Um den wechselseitigen Ausschluss der Zugriffe auf kritische Bereiche zu realisieren, können Binärsemaphore wie folgt eingesetzt werden:

```
BinarySemaphore mutex;
init(mutex, 1);

P:
REPEAT
  <rechne im unkritischen Bereich>
  wait(mutex);
  <rechne im kritischen Bereich>
  signal(mutex);
  <rechne im unkritischen Bereich>
UNTIL FALSE;
```

Beispielhaft für **zwei** Prozesse P_1 und P_2 soll dieses Synchronisationsproblem durch ein Petrinetz modelliert werden. Die Stellen können als Zustände interpretiert werden. Dabei entspricht S_1 (S_3) dem Zustand "P_1 (P_2) rechnet im unkritischen Bereich", und S_2 (S_4) dem Zustand "P_1 (P_2) rechnet im kritischen Bereich".



Fortsetzung nächste Seite!

a) Übertragen Sie dieses Petrinetz auf Ihr Klausurpapier und ergänzen Sie dieses um weitere Stellen, Transitionen und Kanten (sofern jeweils erforderlich), sodass zu jedem Zeitpunkt höchstens ein Prozess in seinem kritischen Bereich rechnen kann. Verwenden Sie hierfür **keine** Inhibitorkanten. Zeichnen Sie direkt in das obige Petrinetz.

b) Geben Sie den Erreichbarkeitsgraphen für Ihr Petrinetz an, und beschriften Sie darin alle Übergänge mit den Namen der jeweiligen Transitionen. Begründen Sie, ob das Petrinetz verklemmt ist oder nicht.

c) Das Beispiel soll nun zu einem Erzeuger/Verbraucher-Szenario erweitert werden. Der Prozess P_1 erzeugt Daten und schreibt diese in einen gemeinsamen Speicher mit 5 Plätzen. Der Prozess P_2 liest diese Daten. Dazu werden die beiden Zählsemaphoren `platz` und `bestand` eingeführt und wie folgt initialisiert:

```
Semaphore platz, bestand;
init(platz, 5);
init(bestand, 0);
```

Ergänzen Sie die folgenden Prozessdefinitionen unter Verwendung dieser Semaphoren so, dass P_1 nicht in den vollen Speicher schreiben und P_2 nicht aus einem leeren Speicher lesen kann.

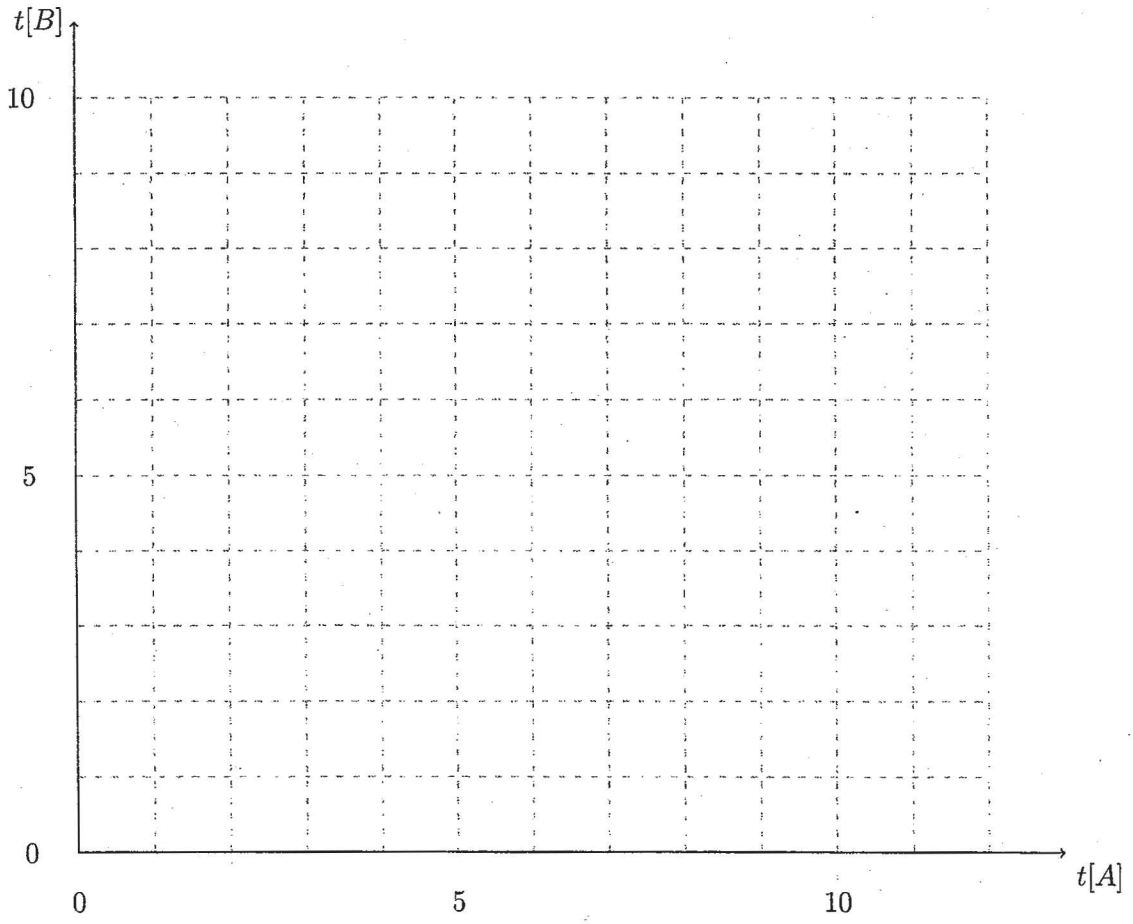
```
P1:                                  P2:
<rechne im unkritischen Bereich>     <rechne im unkritischen Bereich>

wait(mutex);                         wait(mutex);



<erzeuge Element>                    <lies Element>



signal(mutex);                       signal(mutex);



<rechne im unkritischen Bereich>     <rechne im unkritischen Bereich>
```

Fortsetzung nächste Seite!

5. Prozessfortschrittsdiagramm

Gegeben seien zwei Prozesse A und B. A benötigt zu seiner Ausführung zwölf Zeiteinheiten, B zehn Zeiteinheiten. Es stehen 3 Betriebsmittel (BM) zur Verfügung, die von den Prozessen während ihrer Ausführung benötigt werden. A benötigt BM1 im Zeitraum $2-6$, BM2 in $4-7$, BM3 in $8-10$ und Prozess B benötigt BM1 in $5-8$, BM2 in $4-7$ und BM3 in $1-3$.

a) Skizzieren Sie im unten angegebene Diagramm das Prozessfortschrittsdiagramm! Zeichnen Sie alle unmöglichen und unsicheren Bereiche ein und beschriften Sie diese entsprechend! Wo genau tritt ggf. der Deadlock ein?



b) Kann es bei nicht-preemptivem Scheduling zu einem Deadlock kommen? Begründen Sie Ihre Antwort und zeichnen Sie für nicht-preemptives Scheduling alle *prinzipiell* (!) verschiedenen Möglichkeiten der Abarbeitung von Prozess A und B in ihrer Abbildung aus Aufgabe a) ein!

c) Wie viele verschiedenen Möglichkeiten gibt es *insgesamt* (!) bei preemptivem Scheduling die Prozesse erfolgreich terminieren zu lassen?

Fortsetzung nächste Seite!

6. Deadlock-Vermeidung

Wenn man eine der Bedingungen für die Entstehung eines Deadlocks bei der Konzeption eines Betriebssystems von Anfang an ausschliesst, können Deadlocks gänzlich vermieden werden. Benennen und erklären Sie die vier Bedingungen, die zur Entstehung eines Deadlocks führen.

66114

Datenbank- und Betriebssysteme (vertieft)

Herbst 2011

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____	**Herbst 2011**	**66114**
Kennwort: _____		
Arbeitsplatz-Nr.: _____		

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

— Prüfungsaufgaben —

Fach: **Informatik (vertieft studiert)**

Einzelprüfung: **Datenbank- und Betriebssysteme**

Anzahl der gestellten Themen (Aufgaben): **4 Aufgaben, von denen zwei gemäß untenstehender Auswahlregel zu bearbeiten sind!**

Anzahl der Druckseiten dieser Vorlage: **13**

Zu den zwei Themenschwerpunkten A (Datenbanksysteme) und B (Betriebssysteme) ist jeweils entweder die Aufgabe 1 oder 2 zu wählen. Auf der Vorderseite des Kopfbogens sind im Feld „gewähltes Thema: Nr." die Nummern der beiden gewählten Aufgaben anzugeben (z. B. A2, B1)!

Bitte wenden!

Themenschwerpunkt A

Aufgabe A1

Datenbanksysteme

1. SQL

Gegeben sei folgendes Schema, welches Zoos, Tiere, die in ihnen eine Zeitlang leben, und die Tierarten der Tiere speichert:

zoos:	{[*zooid, name, stadt*]}	*Zoos*
tiere:	{[*tierid, name, geburtstag, todestag*]}	*Tiere*
lebt_in:	{[*zooid, tierid, von, bis*]}	*Aufenthalte der Tiere in verschiedenen Zoos*
tierarten:	{[*tierartid, name*]}	*Tierarten*
hat_tierart:	{[*tierid, tierartid*]}	*Zuordnung von Tierarten zu Tieren*
tierartenhierarchie	{[*oberartid, unterartid*]}	*Zuordnung von direkten Untertierarten zu Tierarten*

Beachten Sie bei der Bearbeitung dieser Aufgabe folgende Hinweise:

- *Bei allen lebenden Tieren steht im Todestag der Wert NULL.*
- *Ein Tier kann nicht in mehreren Zoos gleichzeitig leben.*
- *Es kann sein, dass ein Tier eine Zeit seines Lebens in keinem Zoo lebt. Dies ist zum Beispiel der Fall, wenn es in Freiheit geboren wurde.*
- *Das Attribut* bis *der Relation* lebt_in *kann den Wert* NULL *besitzen, wenn das Tier mit dem Schlüssel* tierid *zurzeit im Zoo mit dem Schlüssel* zooid *lebt. Es kann aber auch sein, dass bereits ein Datum festgelegt wurde, an dem das Tier den Zoo verlassen wird. Eventuell ist sogar der Verbleib des Tieres in verschiedenen Zoos auf Jahre hinaus in die Zukunft geplant.*
- *Eine Tierart kann eine beliebige Anzahl Untertierarten haben.*
- *Eine Tierart kann eine beliebige Anzahl Obertierarten haben.*

a) Formulieren Sie eine SQL-Anfrage, die *tierid* und *name* aller Tiere zurückgibt, die eine Zeit im Zoo mit der *zooid* 42 gelebt haben, zurzeit in diesem Zoo leben oder voraussichtlich in Zukunft in diesem Zoo leben werden. Stellen Sie sicher, dass Ihre Anfrage keine Duplikate erhält. Sortieren Sie das Ergebnis nach den Namen der Tiere.

b) Formulieren Sie eine SQL-Anfrage, die für jeden Zoo *zooid*, *name* und die Anzahl der Tiere, die zurzeit in diesem Zoo leben, zurückliefert! Zoos, in denen zurzeit keine Tiere leben, sollen nicht im Ergebnis auftauchen. Sortieren Sie das Ergebnis nach der Anzahl der Tiere, die zurzeit im Zoo leben.
Hinweis: CURRENT_DATE liefert in SQL das heutige Datum.

Fortsetzung nächste Seite!

c) Formulieren Sie eine SQL-Anfrage, die *tierid* und *name* aller Tierarten zurückgibt, die einen Umzug von Berlin nach München erlebt haben oder voraussichtlich erleben werden.

Hinweis:

- *Mithilfe des Minus-Operators kann in SQL die Differenz in Tagen zwischen zwei Daten berechnet werden.*
  Der Ausdruck DATE('2012-02-02') – DATE('2011-02-02') liefert beispielsweise das Ergebnis 365.
- *Der Umzug eines Tieres von einem Zoo in einen anderen lässt sich dadurch erkennen, dass das Tier an einem Tag im Ausgangszoo und am darauffolgenden Tag im Zielzoo lebt.*

d) Formulieren Sie eine SQL-Anfrage, die angibt, wie viele Schildkröten am 01.01.1900 in Münchner Zoos gelebt haben.

Hinweis:

- *Sie dürfen Views verwenden, um Teilergebnisse auszudrücken!*
- *DATE('1900-01-01') liefert das Datum 01.01.1900 in SQL.*
- *Es kann sein, dass Schildkröten von damals heute noch leben.*
- *Auch Unterarten von Schildkröten sind Schildkröten (z. B. Meeresschildkröten). Dasselbe gilt für die Unterarten der Unterarten von Schildkröten (z. B. Bastardschildkröten) und so weiter (z. B. Olivbastardschildkröte, Atlantikbastardschildkröte). Folglich ist zur Beantwortung dieser Anfrage eine rekursive (temporäre) View bzw. Common Table Expression zwingend erforderlich.*

2. Modellierung

Für eine Autovermietung wird ein geeignetes ER-Modell zur Verwaltung von Mietwagenreservierungen benötigt.

- Jedes Fahrzeug besitzt eine eindeutige Fahrzeugidentifikationsnummer sowie eine Farbe. Zusätzlich wird für jedes Fahrzeug gespeichert, ob dieses mit einem Navigationsgerät und/oder einer Klimaanlage ausgestattet ist.
- Jedes Fahrzeug gehört genau einem Fahrzeugtyp an. Ein Fahrzeugtyp wird durch die Kombination einer Typschlüsselnummer (TSN) und Herstellerschlüsselnummer (HSN) identifiziert. Zu einem Fahrzeugtyp sind dessen Modellname (z. B. Golf) und dessen Motorbezeichnung (z. B. 2.0 TDI) zu speichern.
- Bei einem Fahrzeugtyp kann es sich um eine Limousine, einen Kombi oder einen Van handeln. Für Vans ist zusätzlich die Anzahl der Sitze zu speichern, für Kombis wird das Kofferraumvolumen und für Limousinen die Anzahl der Türen angegeben. Ein Fahrzeugtyp kann höchstens zu einer dieser Karosserieformen zählen. Es kann jedoch Fahrzeugtypen geben, die keiner der drei genannten Formen entsprechen.

Fortsetzung nächste Seite!

- Ein Fahrzeugtyp gehört mindestens einer Fahrzeugklasse an, die eindeutig durch ihre Klassenbezeichnung (bspw. Mittelklasse, Kompaktklasse, etc.) identifiziert wird. Es kann Fahrzeugtypen geben, die sich nicht eindeutig einer Klasse zuordnen lassen, und daher mehreren Klassen angehören.
- Von Kunden der Autovermietung wird deren Vorname, Nachname sowie deren Geburtsdatum und Anschrift gespeichert. Außerdem erhält jeder Kunde initial eine eindeutige Kundennummer.
- Ein Kunde kann eine Reservierung aufgeben. Dabei kann der Kunde zwischen zwei Arten von Reservierungen wählen. Nutzt er eine Spezialreservierung, so kann er den gewünschten Fahrzeugtyp wählen (z. B. Golf mit 2.0 TSI). Wählt er dagegen die Standardreservierung, so kann er sich lediglich auf eine Fahrzeugklasse (z. B. Kompaktklasse) festlegen. In beiden Fällen gibt eine Reservierung zudem den Zeitraum an, in dem das Auto gemietet werden soll. Eine Reservierung besitzt eine Reservierungsnummer, die jedoch nur bezüglich des jeweiligen reservierenden Kunden eindeutig ist.

a) Entwerfen Sie ein ER-Modell, das die oben beschriebenen Punkte möglichst präzise modelliert.
Entwickeln Sie dazu geeignete Entity-Typen mit entsprechenden Attributen sowie Relationship-Typen, um den logischen Zusammenhang zwischen diesen aufzuzeigen.
Machen Sie Gebrauch von Generalisierungen/Spezialisierung, sofern sinnvoll.
Kennzeichnen Sie die Primärschlüssel im Modell.
Fügen Sie zudem Funktionalitäten in das ER-Diagramm ein.

b) Überführen Sie das in Teilaufgabe a) konstruierte ER-Diagramm in ein verfeinertes Relationsmodell. Geben Sie hierfür die verallgemeinerten Relationenschemata an. Vermeiden Sie dabei redundante Attribute.

3. Normalformen

Gegeben sei folgende Relation

R: {ID, Vorname, Nachname, Geburtsdatum,
Breitengrad, Längengrad, Länge, Breite, Fläche},

die ausgewiesenen Parzellen des Mondes ihre jeweiligen Eigentümer zuweist.

- Ein Eigentümer einer Mondparzelle erhält mit dem erstmaligen Erwerb einer Parzelle eine eindeutige *ID* zugewiesen. Zur Vereinfachung wird hier zu einem Eigentümer lediglich dessen *Vorname, Nachname* und *Geburtsdatum* verwaltet.
- Parzellen sind stets von rechteckiger Gestalt und paralleler Ausrichtung zum Mondäquator und können damit stets eindeutig durch deren südöstlichste Koordinate identifiziert werden. Diese wird in Form des Paares *(Breitengrad, Längengrad)* modelliert.
- Eine jede Parzelle besitzt eine *Länge* (in Ost-West-Richtung) sowie eine *Breite* (in Nord-Süd-Richtung). Zudem wird für eine Parzelle deren *Fläche* gespeichert, die aus der *Länge* und *Breite* der Parzelle resultiert.
- Hier wird davon ausgegangen, dass eine Parzelle stets nur maximal einen Eigentümer haben kann.

Fortsetzung nächste Seite!

a) Geben Sie für das beschriebene Szenario eine Menge funktionaler Abhängigkeiten F an, aus der sämtliche funktionalen Abhängigkeiten der Relation R abgeleitet werden können.

b) Bestimmen Sie eine kanonische Überdeckung F_C für F.

c) Bestimmen Sie anhand der in Teilaufgabe b) entwickelten kanonischen Überdeckung F_C den oder die Schlüsselkandidaten der Relation R.

d) Erfüllt R die erste Normalform? Welches Kriterium muss dafür erfüllt sein?

e) Ist R in 2NF? Begründen Sie Ihre Entscheidung.

f) R erfüllt nicht die dritte Normalform. Wenden Sie daher den Synthesealgorithmus auf R an, um ein neues Relationenschema zu erhalten, das die dritte Normalform erfüllt.

Aufgabe A2

1. Datenmodellierung: ER-Modell, SQL

In einer *Brokering-Datenbank* sollen folgende Informationen gespeichert werden:

- Es gibt Büros an verschiedenen Standorten und jedes Büro hat eine Adresse (Schlüssel) sowie eine Telefonnummer.
- Jeder Broker arbeitet in genau einem Büro; es wird auch gespeichert, seit wann er dort arbeitet. Ein Broker hat verschiedene Telefonnummern, und er wird durch eine Broker-ID (Schlüssel) identifiziert. Jedes Büro wird von einem Broker geleitet; es wird auch gespeichert, seit wann dieser das Büro leitet.
- Ein Konto hat eine Kontonummer (Schlüssel), und es wird der Eröffnungstag und der Status des Kontos gespeichert. Ein Konto wird von einem Broker betreut.
- Ein Kunde wird durch eine Kunden-ID (Schlüssel) identifiziert, und es werden außerdem der Name und das Geburtsdatum des Kunden gespeichert. Ein Kunde kann verschiedene Konten haben, die von den Büros betreut werden.

Wenn das Konto eines Kunden von einem Büro betreut wird, dann soll der das Konto betreuende Broker in diesem Büro arbeiten. Diese Integritätsbedingung I kann im ER-Modell nicht abgebildet werden, sie kann aber im zugehörigen relationalen Modell überprüft und berücksichtigt werden.

a) Erstellen Sie ein ER-Modell, welches das oben dargestellte Szenario möglichst vollständig abbildet. Verwenden Sie wann immer möglich (binäre oder auch höherstellige) Relationships.

b) Übertragen Sie Ihr ER-Modell ins relationale Modell. Erstellen Sie dazu Tabellen mit Hilfe von CREATE TABLE-STATEMENTS in SQL. Berücksichtigen Sie die Fremdschlüsselbeziehungen.

c) Schreiben Sie ein SELECT-Statement, das die obige Integritätsbedingung I überprüft.

d) Geben Sie geeignete INSERT-Statements an, die in alle beteiligten Tabellen jeweils mindestens ein Tupel einfügt, so dass alle Integritätsbedingungen erfüllt sind, nachdem alle Einfügungen ausgeführt werden.

Fortsetzung nächste Seite!

2. Datenbankfragen in SQL

Gegeben seien die folgenden Tabellen aus einer *Fußball-Datenbank*:

- *Spieler (Verein, Rückennummer, Name, Alter, Position),*
- *Verein (Name, Stadt),*
- *Spiel (ID, Heim, Gast, Datum, Tore_Heim, Tore_Gast):*
  Für jedes Spiel werden der Heim- und der Gastverein sowie die von diesen jeweils erzielte Anzahl von Toren gespeichert.
- *Tor (SpielID, Spielzeit, Verein, Rückennummer):*
  Für jedes Tor werden die Spielzeit (eindeutiger Zeitstempel in Form einer reellen Zahl zwischen 0 und 90 zur Repräsentation der vergangenen Minuten) sowie der Spieler (gegeben durch Verein und Rückennummer) gespeichert; wir nehmen an, dass beide Halbzeiten genau 45 Minuten dauern.

Formulieren Sie auf den obigen Tabellen folgende SQL-Anfragen:

a) Bestimmen Sie alle Vereine aus *München*.

b) Bestimmen Sie alle Heimspiele, die der Verein *Bayern München* gewonnen hat.

c) Bestimmen Sie alle Tore, die in der ersten Halbzeit gefallen sind.

d) Bestimmen Sie für jeden Spieler die Anzahl seiner erzielten Tore.

e) Bestimmen Sie, nach Vereinen und Positionen gruppiert, die Anzahl der Spieler auf diesen Positionen.

f) Erstellen Sie mittels eines CREATE VIEW-Statements eine Sicht, die für jedes Spiel und jeden beteiligten Verein die Anzahl der erzielten Punkte enthält.

g) Füllen Sie die Sicht, indem Sie das Ergebnis geeigneter SELECT-Statements einfügen. Ein Sieg soll dabei 3 Punkte bringen, ein Unentschieden 1 Punkt.

h) Bestimmen Sie auf der Basis der Sicht für jeden Verein die Anzahl seiner erzielten Punkte.

3. Funktionale Abhängigkeiten und Zerlegungen

Gegeben sei das Relationenschema $R = (U, F)$ mit der Attributmenge

$$U = \{ A, B, C, D, E \}$$

und der folgenden Menge F von funktionalen Abhängigkeiten:

$$F = \{ AD \rightarrow BC,\ C \rightarrow AD,\ CD \rightarrow E \}$$

a) Geben Sie alle Schlüssel für R an – jeweils mit Begründung.

b) Ist R in 2NF, 3NF bzw. in BCNF? Geben Sie jeweils eine Begründung an.

c) Zerlegen Sie R mittels des Synthesealgorithmus in ein 3NF-Datenbankschema.

d) Zerlegen Sie R bezüglich der funktionalen Abhängigkeit $C \rightarrow AD$ verlustfrei in zwei Relationenschemata $R_i = (U_i, F_i)$, $i = 1, 2$.

Themenschwerpunkt B

Aufgabe B1

Betriebssysteme

1. Semaphore

In ein automatisches Flugbuchungssystem soll eine Gefährdungsanalyse integriert werden. Ein vom Nutzer eingegebener Datensatz mit Buchungsdaten wird hierfür einer automatisierten Gefährdungsanalyse (G) unterzogen, die entweder „bedenklich" oder „unbedenklich" als Antwort zurückliefert. Wird ein Datensatz als „unbedenklich" eingestuft, so wird die Buchung durch das elektronische Buchungssystem (B) durchgeführt. Liefert die Gefährdungsanalyse „bedenklich" als Antwort, so wird der Datensatz durch einen Sicherheitsbeauftragen (S) manuell überprüft; dieser kann je nach Ergebnis die Buchung entweder freischalten (dann wird sie durch dasselbe elektronische Buchungssystem durchgeführt) oder verweigern.

Für die Gefährdungsanalyse steht ein Rechenzentrum mit 50 Servern zur Verfügung. Diese Analyse ist somit für bis zu 50 Datensätze gleichzeitig möglich. Das elektronische Buchungssystem kann bis zu 100 Buchungen gleichzeitig verarbeiten und es steht genau ein Sicherheitsbeauftragter zur Verfügung.

Modellieren Sie den Buchungsprozess mit integrierter Gefährdungsanalyse unter Verwendung von Zähl-Semaphoren (Counting Semaphores), so dass möglichst viele Berechnungen parallel durchgeführt werden können:

a) Definieren Sie in einer objektorientierten Sprache Ihrer Wahl oder in Pseudocode die Schnittstellen (Konstruktor, nach außen sichtbare Methoden) einer Zähl-Semaphore SEM. Eine Implementierung der Methoden ist nicht notwendig.

b) Benutzen Sie Ihr Ergebnis aus dem Aufgabenteil a), um den folgenden Code zu vervollständigen. Ersetzen Sie dabei die Kommentare der Form `/ * Ergaenzung X */` durch geeignete Semaphor-Aufrufe. Einem Kommentar können dabei einer, mehrere oder keine Aufrufe entsprechen.

```
class Buchungssytem {
     Sem G = new SEM /* Ergaenzung 1 */
     Sem B = new SEM /* Ergaenzung 2 */
     Sem S = new SEM /* Ergaenzung 3 */

     public void flugbuchung(Datensatz d) {
          /* Ergaenzung 4 */
          if (gefaehrdungsanalyse(d) == unbedenklich) {
               /* Ergaenzung 5 */
               buchung(d);
          } else { /* if (gefaehrdungsanalyse(d) == bedenklich) */
               /* Ergaenzung 7 */
               if (freigeschaltet(d)) {
                  /* Freischaltung durch Sicherheitsbeauftragten */
                  /* Ergaenzung 8 */
                  buchung(d);
                  /* Ergaenzung 9 */
               }
               /* Ergaenzung 10 */
          }
          /* Ergaenzung 11 */
     }
```

Fortsetzung nächste Seite!

2. Virtueller Speicher

In dieser Aufgabe geht es um eine optimale Wahl der Seitengröße p eines Systems mit Seitenadressierung. Nehmen Sie an, dass der gesamte Speicher des Rechners genau ein Programm enthält. Die durchschnittliche Größe eines Programms sei l Bytes.

a) Was ist die interne Fragmentierung?

b) Im Schnitt bleibe die Hälfte der Seiten unbenutzt. Wie groß ist die durchschnittliche interne Fragmentierung, wenn $p = 1024$ Bytes ist?

c) Welche Information über eine Seite muss in der Seitentabelle mindestens (unabhängig von der gewählten Verdrängungsstrategie) vorgehalten werden? Wovon hängt die Anzahl i der für einen solchen minimalen Seitentabelleneintrag benötigten Bytes ab?

d) Welche Gründe sprechen für die Wahl eines kleinen bzw. großen Wertes von p?

e) Geben Sie die durchschnittliche Größe der Seitentabelle als Funktion von l, i und p an.

f) Sowohl für die interne Fragmentierung als auch für die Seitentabelle wird Speicherplatz verbraucht, der für Nutzdaten nicht zur Verfügung steht. Bestimmen Sie für gegebene i und l den Wert von p, für welchen die Summe aus durchschnittlicher interner Fragmentierung und durchschnittlicher Größe der Seitentabelle minimal ist.

g) Sei $l = 128$ Kilobyte. Für welchen Wert von i ergibt Ihre Lösung aus dem vorherigen Punkt die optimale Seitengröße $p = 1024$ Byte?

3. Deadlocks, Bankier-Algorithmus

a) Nennen Sie die vier Eigenschaften, die ein Betriebssystem besitzen muss, so dass ein Deadlock auftreten kann (ohne Begründung).

b) Beim Bankieralgorithmus wurde der Begriff des „sicheren Zustands" verwendet. Wie lautet die Definition eines sicheren Zustands?

c) Vier Prozesse (p_1, p_2, p_3, p_4) greifen auf fünf Ressourcenklassen (A, B, C, D, E) zu. Der Vektor verfügbarer Ressourcen ist $V = (6,9,9,7,9)$ und die Maximalanforderungsmatrix ist

$$M = \begin{pmatrix} 3 & 7 & 5 & 2 & 4 \\ 2 & 1 & 8 & 5 & 7 \\ 4 & 4 & 4 & 1 & 3 \\ 3 & 6 & 2 & 4 & 3 \end{pmatrix}$$

Geben Sie für die folgenden Zustände mittels des Bankiersalgorithmus an, ob sie sicher oder unsicher sind. Begründen Sie Ihre Aussage zwei Mal. Geben Sie für die nicht sicheren Zustände an, welche Prozesse an dem potentiellen Deadlock beteiligt sind.

i. Der Zustand Z_1 mit der folgenden Belegungsmatrix:

$$E = \begin{pmatrix} 1 & 3 & 2 & 2 & 0 \\ 1 & 0 & 1 & 0 & 4 \\ 2 & 2 & 1 & 0 & 1 \\ 0 & 2 & 2 & 4 & 2 \end{pmatrix}$$

ii. Der Zustand Z_2, der sich aus Zustand Z_1 ergibt, wenn p_1 eine weitere Ressource aus der Klasse B zugeteilt bekommt.

iii. Der Zustand Z_3, der sich aus Zustand Z_1 ergibt, wenn p_4 eine weitere Ressource aus der Klasse B zugeteilt bekommt.

d) Führt jeder unsichere Zustand unweigerlich in einen Deadlock? Begründen Sie Ihre Antwort kurz.

Fortsetzung nächste Seite!

4. Prozess-Scheduling

In dieser Aufgabe geht es um präemptives und nicht-präemptives Scheduling von Prozessen. Gegeben seien vier Prozesse *A, B, C* und *D* mit den folgenden Auskunfts- und Laufzeiten:

Prozess	Auskunftszeit	Laufzeit
A	3	3
B	1	4
C	2	2
D	2	7

Wird ein Prozess unterbrochen, so reiht sich der Restprozess mit dem Zeitpunkt der Unterbrechung als Ankunftszeit in die Warteschlange ein. Kommen zwei Prozesse zeitgleich an, so wird nach Prozessnamen priorisiert (*A* vor *B* vor *C* vor *D*). Geben Sie für die folgenden drei Scheduling-Strategien an, zu welchen Zeiteinheiten welcher Prozess Rechenzeit zugeteilt bekommt. Geben Sie ferner für jeden Prozess *P* die Reaktionszeit r_P := erster Rechenzeitpunkt – Ankunftszeitpunkt sowie die Antwort a_P := letzter Rechenzeitpunkt – Ankunftszeitpunkt an (vergessen Sie dabei nicht, die letzte Zeiteinheit zu berücksichtigen):

a) Shortest Job First ohne Präemption

b) Shortest Job First mit Präemption

c) Round Robin mit Zeitscheibe 2

Aufgabe B2

1. Scheduling

a) Nennen Sie jeweils einen Vor- und einen Nachteil bei der Bevorzugung kurzer bzw. langer Prozesse beim Scheduling!
Worin unterscheiden sich harte und weiche Echtzeitanforderungen?

b) Gegeben sind folgende Prozesse mit ihren Bereitzeitpunkten und ihrem Rechenzeitbedarf. Führen Sie für eine Einprozessormaschine das nicht-preemptive Scheduling gemäß *Highest Response Ratio Next* (HRN) und das preemptive Scheduling gemäß *Round-Robin* (RR) mit einer Zeitscheibenlänge von 1 durch. Der Overhead für den Prozesswechsel ist hier vernachlässigbar.

Prozess	***Bereitzeitpunkt***	***Benötigte Rechenzeit***
A	0	5
B	0	3
C	1	2
D	4	1
E	7	2
F	14	1

Zuerst werden hinzustoßende Prozesse ans Ende der (eventuell leeren) Warteschlange angehängt; erst dann wird der nächste Prozess gemäß der Scheduling-Strategie bestimmt, welcher dem Prozessor zugeteilt wird. Ein Prozess kann also bereits zum Zeitpunkt seiner Ankunft im System ausgeführt werden (z. B. *A* zum Zeitpunkt *0*).

Fortsetzung nächste Seite!

Übertragen Sie die Tabellen auf Ihr Blatt. Tragen Sie in der Zeile Ausführung jenen Prozess ein, der für die entsprechende Zeitscheibe dem Prozessor zugeteilt wird. Die restlichen Zeilen der Tabelle stellen die Ready-Queue dar. Tragen Sie hier die Prozesse ein, die in der Warteschlange auf Zuteilung warten. Geben Sie jeweils beim Prozesswechsel von HRN auch alle $h_i(t)$ an.

Highest Response Ratio Next:

0					5					10					15				
Ausf.																			
Ready-Queue																			

Round Robin:

0					5					10					15				
Ausf.																			
Ready-Queue																			

Geben Sie für jeden Prozess die Verweildauer im System an:

Prozess	A	B	C	D	E	F
HRN						
RR						

c) Gegeben sind folgende periodische, unterbrechbare Prozesse A, B und C auf einer Einprozessormaschine. A benötigt alle 5 Zeiteinheiten 2 Zeiteinheiten Rechenzeit, B benötigt alle 4 Zeiteinheiten 2 Zeiteinheiten Rechenzeit und C benötigt alle 12 Zeiteinheiten 1 Zeiteinheit Rechenzeit. Berechnen Sie die Prozessorauslastung.

Führen Sie das Scheduling gemäß preemptiven Rate-Monotonic-Scheduling (RMS) mit Zeitscheibenlänge 1 durch und zeichnen Sie das entsprechende Gantt-Diagramm soweit sinnvoll!

Orientieren Sie sich an folgendem Muster:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
A																														
B																														
C																														

	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
A																														
B																														
C																														

Was stellen Sie fest?

Fortsetzung nächste Seite!

2. Synchronisation

a) Nennen Sie die vier hinreichenden und notwendigen Bedingungen für Verklemmungen und jeweils eine passende Maßnahme, um diese auszuschließen.

Gegeben sind die folgenden Programme in Pseudocode.

```
(1) Init () {
(2)       max=100;
(3)       size=0;
(4) }

(1) Erzeuger () {
(2)        while(true) {
(3)              Element=baue_neues_Element();
(4)              while(size>=max) {}
(5)              Puffer[size]=Element;
(6)              size=size+1;
(7)        }
(8) }

(1) Verbraucher () {
(2)        while(true) {
(3)              if(size>0) {
(4)                  Element=Puffer[size];
(5)                  size=size-1
(6)                  verarbeite_Element(Element);
(7)              }
(8)        }
(9) }
```

Im System sind *n* Erzeuger vorhanden und produzieren Elemente, die in eine Liste der Größe *max* eingefügt werden. Der Verbraucher ist nur einmal im System vorhanden und entnimmt Elemente aus der Liste und verarbeitet sie.

b) Welche Probleme können bei der gegebenen Implementierung auftreten? Geben Sie jeweils ein Beispiel an.

c) Schützen Sie Erzeuger und Verbraucher vor den in b) genannten Problemen durch die Verwendung eines Monitors.

Geben Sie die **Schnittstelle** des Monitors mit allen benötigten Variablen und Funktionen in Pseudocode **inklusive Kommentaren** an und beschreiben Sie, wie der o. g. Quellcode für Erzeuger und Verbraucher angepasst werden muss, wenn dieser Monitor verwendet werden soll.

Hinweis: Es ist nicht notwendig, hier die Realisierung des Monitors anzugeben.

Fortsetzung nächste Seite!

3 Virtuelle Adressabbildung

a) Probleme der direkten Speicherbelegung:

Problem 1: Welches Problem kann durch die absolute Adressierung von Programmcode entstehen?

Problem 2: Welches Problem kann durch die Zuweisung eines festen Speichersegments an einen Prozess entstehen?

Problem 3: Welches Problem kann durch eine fehlende Regulierung der Speicherzugriffe verschiedener Prozesse entstehen?

Erklären Sie, wieso diese Probleme jeweils durch die Verwendung von virtuellem Speicher gelöst werden.

b) Gegeben sei ein Rechner mit 5 Bit physikalischem und virtuellem Adressraum und 5 Bit breiten Worten. Der Inhalt des physikalischen Speichers ist unten angegeben. Es gelten folgende Annahmen:

- Die Seiten sind jeweils 4 Worte groß.
- Die Basisadresse der Seitentabelle ist `10000`.
- Ein Seitentabelleneintrag besteht aus 5 Bit, die ersten (= höherwertigsten) 3 Bits geben die physikalische Seitenrahmennummer an, das 4. Bit ist das *present*- bzw. *mapped*-Bit und das 5. Bit ist hier nicht von Bedeutung.
- Bei der zweistufigen Adressierung werden 2 Bit zur Indizierung der Seitentabellen der zweiten Stufe verwendet.

Adresse	*Inhalt*	*Adresse*	*Inhalt*	*Adresse*	*Inhalt*	*Adresse*	*Inhalt*
00000	10111	01000	01010	10000	10111	11000	00110
00001	10011	01001	00011	10001	11010	11001	01010
00010	11011	01010	10010	10010	11100	11010	00010
00011	11101	01011	00001	10011	01101	11011	00011
00100	01000	01100	10000	10100	00101	11100	10110
00101	00110	01101	10010	10101	11110	11101	01111
00110	01111	01110	11110	10110	00101	11110	11010
00111	11000	01111	01101	10111	10001	11111	11100

Welche Daten stehen an der Adresse `01011`, wenn Sie diese Adresse als physikalische Adresse, als einstufige und als zweistufige virtuelle Adresse verwenden?

c) Geben Sie eine einstufige, virtuelle Adresse für die physikalische Adresse `10101` an.
Geben Sie eine einstufige, virtuelle Adresse an, welche zu einem Seitenfehler führt.

Fortsetzung nächste Seite!

4. Virtueller Speicher und Seitenverdrängung

a) Um was handelt es sich bei der so genannten Belady-Anomalie?
Welcher Zusammenhang besteht zwischen der Belady-Anomalie und der Keller-Eigenschaft?

b) Gegeben sei eine Maschine mit einem physikalischen Speicher für 4 Seitenrahmen und ein virtueller Adressraum mit 6 Seiten A bis F. Gegeben sei die folgende Referenzkette:

A B C A D E B F A C B E

Geben Sie für die folgenden Seitenverdrängungsstrategien an, welche Seiten sich zu jedem Zeitpunkt im realen Speicher befinden:

– Optimal (nach Belady)

– Least Frequently Used (Der Zugriffszähler einer Seite wird hier bei ihrer Verdrängung auf Null zurückgesetzt.)

Bei Zweideutigkeiten wird alphabetische Ordnung verwendet, das heißt, es wird die Seite entfernt, deren Bezeichner im Alphabet zuerst kommt.

Markieren Sie auch für jede Strategie die Anzahl der auftretenden Seitenfehler.

Orientieren Sie sich an folgendem Muster:

Belady:

A	*B*	*C*	*A*	*D*	*E*	*B*	*F*	*A*	*C*	*B*	*E*

Anzahl der Seitenfehler: _____

Least Frequently Used:

A	*B*	*C*	*A*	*D*	*E*	*B*	*F*	*A*	*C*	*B*	*E*

Anzahl der Seitenfehler: _____

Welcher Nachteil kann durch die Verwendung von LFU entstehen?

66114

Datenbank- und Betriebssysteme (vertieft)

Frühjahr 2012

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____	**Frühjahr 2012**	**66114**
Kennwort: _____		
Arbeitsplatz-Nr.: _____		

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

— Prüfungsaufgaben —

Fach: **Informatik (vertieft studiert)**

Einzelprüfung: **Datenbank- und Betriebssysteme**

Anzahl der gestellten Themen (Aufgaben): **4 Aufgaben, von denen zwei gemäß untenstehender Auswahlregel zu bearbeiten sind!**

Anzahl der Druckseiten dieser Vorlage: **14**

Zu den zwei Themenschwerpunkten A (Datenbanksysteme) und B(Betriebssysteme) ist jeweils entweder die Aufgabe 1 oder 2 zu wählen.
Auf der Vorderseite des Kopfbogens sind im Feld „gewähltes Thema: Nr." die Nummern der beiden gewählten Aufgaben anzugeben (z. B. A2, B1)!

Bitte wenden!

Themenschwerpunkt A
(Datenbanksysteme)

Teilaufgabe 1:

1. Modellierung

Sie sollen ein System zur Verwaltung von Pferderennen entwerfen.
Gehen Sie dabei von folgendem Szenario aus:

- Unternehmen werden über ihre eindeutige Unternehmens-Id identifiziert. Sie haben eine Adresse und besitzen Rennställe.
- Der Name eines Rennstalls ist nur innerhalb eines Unternehmens eindeutig. Für jeden Rennstall wird das Gründungsdatum gespeichert.
- Pferde gehören immer zu einem Rennstall. Pferdenamen werden in einem Rennstall nur jeweils maximal einmal vergeben.
- Jockeys sind in einem Rennstall beschäftigt. Jeder Rennstall vergibt seine eigenen Personalnummern. Für jeden Jockey werden sein Vorname und Name gespeichert.
- Rennen haben ein Datum, ein Preisgeld und einen Namen, über den sie identifiziert werden.
- Unternehmen unterstützen Rennen finanziell mit einem bestimmten Betrag.
- Jockeys nehmen mit Pferden an Rennen teil. Im Rennen erreichen sie einen bestimmten Platz. Die Kombination aus Jockey und Pferd ist nicht fest, bei unterschiedlichen Rennen können Jockeys verschiedene Pferde reiten. Jockeys können auch mit Rennpferden von fremden Rennställen, die anderen Unternehmen gehören können, an Rennen teilnehmen.

a) Entwerfen Sie für das beschriebene Szenario ein ER-Modell.
Bestimmen Sie hierzu:

- die Entity-Typen, die Relationship-Typen und jeweils deren Attribute,
- ein passendes ER-Diagramm,
- die Primärschlüssel der Entity-Typen, welche Sie anschließend in das ER-Diagramm eintragen, und
- die Funktionalitäten der Relationship-Typen, welche Sie ebenfalls in das ER-Diagramm eintragen.

b) Überführen Sie das ER-Modell aus Aufgabe a) in ein verfeinertes relationales Modell. Geben Sie hierfür die verallgemeinerten Relationenschemata an. Achten Sie dabei insbesondere darauf, dass die Relationenschemata keine redundanten Attribute enthalten.

Fortsetzung nächste Seite!

2. Anfragen

Zu einer Website, auf der Besucher im Browser Online-Spiele spielen können, liegt das folgende relationale Schema einer Datenbank vor:

$$\begin{aligned} \text{Team} &: \{[\underline{\text{TNr}}, \text{Name}, \text{Teamfarbe}]\} \\ \text{Spieler} &: \{[\underline{\text{SNr}}, \text{Name}, \text{Icon}, \text{TNr}, \text{EMail}]\} \\ \text{Minispiel} &: \{[\underline{\text{MNr}}, \text{Name}, \text{Kategorie}, \text{Schwierigkeit}]\} \\ \text{Wettkampf} &: \{[\underline{\text{WNr}}, \text{Sieger}, \text{Geschlagener}, \text{MNr}, \text{Dauer}]\} \end{aligned}$$

Auf der Website treten jeweils zwei Spieler gegeneinander in Minispielen an. In diesen ist es das Ziel, den gegnerischen Spieler in möglichst kurzer Zeit zu besiegen. Minispiele gibt es dabei in verschiedenen Schwierigkeitsstufen ('leicht', 'mittel', 'schwer', 'sehr schwer') und verschiedenen Kategorien ('Denkspiel', 'Geschicklichkeitsspiel', usw.). Die Attribute *Sieger* und *Geschlagener* sind jeweils Fremdschlüsselattribute, die auf das Attribut *SNr* der Relation *Spieler* verweisen. Beachten Sie, dass das *Dauer*-Attribut der Wettkampf-Relation die Dauer eines Wettkampfes in der Einheit Sekunden speichert.

a) Formulieren Sie geeignete Anfragen in relationaler Algebra für die folgenden Teilaufgaben:

   i. Geben Sie die Namen der Minispiele zurück, die zur Kategorie 'Denkspiele' zählen.

   ii. Finden Sie die Spieler, die mindestens einmal in weniger als 5 Minuten besiegt wurden.

   iii. Geben Sie die Namen und E-Mail-Adressen aller Spieler zurück, die in einem Minispiel des Typs 'Geschicklichkeitsspiel' gewonnen haben.

b) Formulieren Sie geeignete SQL-Anfragen für die folgenden Teilaufgaben. Beachten Sie dabei, dass Ihre Ergebnisrelationen keine Duplikate enthalten sollen.

   i. Geben Sie jede Spielekategorie aus, für die ein Minispiel der Schwierigkeitsstufe 'sehr schwer' vorhanden ist.

   ii. Geben Sie zu jedem Team seine *TNr* und die Anzahl der Mitglieder aus.

   iii. Geben Sie die Wettkämpfe aus, deren Dauer unter der durchschnittlichen Dauer der Wettkämpfe liegt.

   iv. Geben Sie für jeden Spieler seine *SNr*, seinen Namen, die Anzahl seiner Siege, die durchschnittliche Dauer seiner siegreichen Wettkämpfe und die Anzahl der Teams, aus denen er bereits mindestens einen Spieler besiegt hat, zurück.

   v. Finden Sie die Teams, die bereits mindestens 10 Wettkämpfe gewonnen haben. Geben Sie zu diesen Teams die Teamnamen und die Anzahl der siegreichen Wettkämpfe zurück. Sortieren Sie die Ausgabe abfallend nach der Anzahl der siegreichen Wettkämpfe.

   vi. Aktualisieren Sie den Wettkampf mit der *WNr* 42. Setzen Sie den Sieger auf den Spieler mit der *SNr* 5037 und den Verlierer auf den Spieler mit der *SNr* 13. Erhöhen Sie zusätzlich die Dauer dieses Wettkampfs um 2 Minuten.

Fortsetzung nächste Seite!

3. Entwurfstheorie

Gegeben ist der folgende Ausschnitt einer relationalen Datenbank:

Vereine							
UserId	ClubId	Vorname	Name	Login	ClubName	Funktion	Vorstand
1	1	Max	Muster	maxm	IT Kicker	Trainer	false
1	2	Max	Muster	maxm	CC Nerds	2. Vorsitzende(r)	true
2	2	Marta	Maier	maier1	CC Nerds	1. Vorsitzende(r)	true
3	1	Tim	Peters	timpeters	IT Kicker	1. Vorsitzende(r)	true
4	1	Peter	Huber	timpeters	IT Kicker	Spieler(in)	false
5	3	Tina	Müller	tinam	IT Club	Kassenprüfer(in)	true
...	...	...	...	...	...	...	...

In der Datenbank sollen folgende funktionale Abhängigkeiten gelten:

$UserId \rightarrow Vorname, Name, Login$
$Login \rightarrow Vorname, Name, UserId$
$UserId, ClubId, Name \rightarrow Funktion, ClubName$
$Login, ClubId, Name \rightarrow Funktion, ClubName$
$ClubId \rightarrow ClubName$
$Funktion \rightarrow Vorstand$

a) Geben Sie sämtliche Schlüsselkandidaten an.

b) Welcher Normalform genügt die Relation *Vereine*? Begründen Sie Ihre Antwort ausführlich.

c) Zeigen Sie anhand von Beispielen die drei möglichen Anomalien auf, die bei der Verwendung der Relation *Vereine* auftreten können!

d) Verwenden Sie den Synthesealgorithmus, um die Relation in die dritte Normalform zu überführen.

e) Befinden sich die neuen Relationen in Boyce-Codd-Normalform? Begründen Sie Ihre Antwort.

Fortsetzung nächste Seite!

Teilaufgabe 2:

1. Allgemeinwissen

1.1 Bewerten Sie die folgenden Aussagen. Geben Sie für die folgenden Aussagen an, ob sie richtig oder falsch sind. Begründen Sie ihre Antworten kurz!

a. B-Bäume verlieren ihre Vorteile gegenüber binären Bäumen in Hauptspeicherdatenbanken.
b. Eine Relation kann mehrere Primärschlüsselattribute besitzen.
c. Heutzutage werden zumeist hierarchische Datenbanksysteme eingesetzt.
d. Fremdschlüsselwerte dürfen nicht NULL sein.
e. Alle Metadaten werden in Tabellen gespeichert.

1.2 Isolation von Transaktionen: Erläutern Sie kurz die Anomalien und geben Sie jeweils ein Beispiel an!

a. Lost upgrade
b. Dirty read
c. Non-repeatable read
d. Phantom

2. Relationale Algebra

2.1 Nennen Sie die fünf Grundoperationen der relationalen Algebra.

2.2 Warum wird in gewöhnlichen Datenbanken die relationale Algebra nicht strikt eingehalten? Geben Sie ein Beispiel an und erklären Sie, wie man die relationale Eigenschaft erzwingen kann.

Fortsetzung nächste Seite!

3. ER-Modellierung und Relationenmodell

3.1 Nennen und erläutern Sie kurz vier Beispiele für Integritätsbedingungen.

3.2 Sie sollen die Datenbank eines Auktionshauses modellieren. Erstellen Sie ein ER-Diagramm bestehend aus Entitäts- und Beziehungstypen sowie Attributen. Geben Sie auch die Kardinalitäten mit an.

Verwenden Sie bei Entitäten und Attributen ausschließlich die beschriebenen, soweit dies möglich ist. Verwenden Sie auch keine künstlichen Schlüssel, es sei denn, diese sind der Aufgabenstellung eindeutig zu entnehmen. Die Kardinalitätseinschränkungen können Sie entweder in der (min,max)-Notation oder der einfachen Notation nach Chen (1:1, 1:N, M:N) angeben.

In der Formulierung der Beziehungen sind die Kardinalitätseinschränkungen genau zu beachten. Im Allgemeinen gilt: Alles was nicht explizit eingeschränkt ist, muss im Modell auch uneingeschränkt bleiben (keine Einschränkungen in eine Beschreibung hineininterpretieren). Alles was explizit eingeschränkt ist, darf im Modell nicht uneingeschränkt bleiben (alle formulierten Einschränkungen sollen erkennbar sein).

1. Im Zentrum des Systems stehen die Benutzer. Sie werden durch eine ID identifiziert und haben einen Benutzernamen, sowie eine Adresse, die sich aus Strasse, PLZ und Ort zusammensetzt.
2. Benutzer initiieren Auktionen. Jede Auktion ist genau einem Benutzer zugeordnet und wird ebenfalls über eine ID identifiziert. Daneben hat sie einen Namen, sowie Beginn- und Endezeitpunkt.
3. Benutzer geben Gebote für Auktionen ab. Zu jedem Gebot werden Betrag und Zeitpunkt des Gebotes gespeichert. Natürlich soll ein Benutzer auch mehrfach auf eine Auktion bieten können.
4. Zu guter Letzt existiert ein Bewertungssystem. Eine Bewertung wird von einem Benutzer zu einer bestimmten Auktion abgegeben. Zu ihr sollen die Bewertung selbst (ein numerischer Wert) und der Zeitpunkt der Bewertung gespeichert werden. Pro Benutzer und Auktion kann nur eine Bewertung abgegeben werden.

3.3 Erstellen Sie zu dem ER-Diagramm aus Aufgabe 3.2 ein Relationenschema. Seien Sie dabei so vollständig wie möglich – bspw. Berücksichtigung totaler Partizipationen. Vermeiden Sie unnötiges Ausprägen von Relationen bei allen Beziehungen.

Beispiel für die Notation:

```
Relationenname (Primärschlüssel, Attribut1, Attribut2, …,
Fremdschlüssel[AndereRelation],
(FKAttrA, FKAttrB)[DritteRelation])

Attribut2 NOT NULL
```

Fortsetzung nächste Seite!

4. Normalformen

4.1 Die Normalisierung von Relationenschemata dient der Vermeidung von Redundanzen und dadurch bedingter Anomalien. Diskutieren Sie anhand des folgenden Beispiels die Einfüge-, Änderungs- und Löschanomalien.

AbtNr	AbtName	AbtOrt	MitarbeiterNr	MitarbeiterName
1	Entwicklung	München	1	Schmidt, H.
2	Forschung	München	2	Müller, G.
1	Entwicklung	München	3	Kohl, H.
3	Fertigung	Nürnberg	4	Weizenbaum, L.
4	Logistik	Passau	5	Nienhof, R.
6	Vertrieb	Würzburg	6	Baier, V.

4.2 In welcher Normalform ist das Beispiel aus Aufgabe 4.1? Welche Bedingung der nächsthöheren Normalform ist verletzt? Berücksichtigen Sie, dass eine Relation nur dann in n-ter Formalform ist, wenn sie die Bedingungen aller m-ten Normalformen mit $m \leq n$ erfüllt.

Fortsetzung nächste Seite!

Themenschwerpunkt B
(Betriebssysteme)

Teilaufgabe 1:

1. Scheduling, Round-Robin, FCFS, SJF

a) Beschreiben Sie die Prozess-Scheduling-Strategien „Round-Robin", „First Come First Served" und „Shortest Job First" knapp (in jeweils 1–2 Sätzen). Welche der Strategien sind präemptiv, welche nicht?

b) „Round-Robin"-Scheduler benutzen üblicherweise eine Liste aller derzeit ausführbarer Prozesse, wobei jeder Prozess maximal einmal in der Liste vorkommt.

Was würde passieren, wenn ein Prozess zweimal in der Liste vorkäme? Was für einen Grund könnte es geben, das zu erlauben?

c) Es sollen fünf Prozesse mit verschiedenen Lauf- und Ankunftszeiten ausgeführt werden:

Prozess	Laufzeit	Ankunftszeit
p_1	5	2
p_2	4	3
p_3	4	0
p_4	2	9
p_5	3	6

- Geben Sie an, in welcher Reihenfolge ein Scheduler mit der Strategie „First Come First Served" die Prozesse ausführen würde.
- In welcher Reihenfolge würde ein Scheduler mit der Strategie „Shortest Job First" die Prozesse ausführen?
- Bestimmen Sie zu beiden Schedules die mittlere Durchlaufzeit.

d) Welche der drei Strategien ist beweisbar optimal? Unter welchen Voraussetzungen? (Die Angabe des Beweises ist nicht notwendig.)

Fortsetzung nächste Seite!

2. Verdrängungsstrategien

a) Paging-Verdrängungsstrategien treffen ihre Entscheidungen basierend auf Kennzahlen wie Anzahl der Zugriffe, Zeit des ersten Zugriffs, Zeit des nächsten Zugriffs oder Zeit des letzten Zugriffs. Ordnen Sie die vier oben genannten Kennzahlen den folgenden vier Strategien zu:

- First-In, First-Out (FIFO);
- (Hypothetische) optimale Strategie (OPT);
- Least Frequently Used (LFU);
- Least Recently Used (LRU).

b) Es stehen drei Seitenrahmen zur Verfügung, die anfangs leer sind. Wenden Sie die vier Strategien FIFO, LFU, LRU und OPT auf die Folge von Seitenanforderungen an:

$$2; 4; 2; 1; 1; 5; 1; 3; 1; 2.$$

Geben Sie für jede der Strategien die Belegung der drei Seitenrahmen nach jeder Seitenanforderung an. Notieren Sie jeweils in Klammern den Wert der relevanten Kennzahl aus dem Aufgabenteil a). Geben Sie für jede Strategie die Anzahl der vorgenommenen Verdrängungen an.

3. Deadlocks, Bankier-Algorithmus

a) Nennen Sie die vier Eigenschaften, die ein Betriebssystem besitzen muss, so dass ein Deadlock auftreten kann. Beschreiben Sie knapp die Eigenschaften. (Eine Begründung, warum diese Eigenschaften zu einem Deadlock führen können, ist nicht notwendig.)

b) Beim Bankieralgorithmus wird der Begriff des „sicheren Zustands“ verwendet. Wie lautet die Definition eines sicheren Zustands?

c) Führt jeder unsichere Zustand unweigerlich zu einen Deadlock? Begründen Sie Ihre Antwort kurz.

Fortsetzung nächste Seite!

4. Dateisysteme

a) Skizzieren Sie kurz (jeweils 1-2 Sätze) die Vor- und Nachteile der folgenden drei Konzepte zur Realisierung von Dateien in einem Dateisystem:

- Zusammenhängende Belegung
- Verkettete Listen
- I-Nodes

Betrachten Sie hierbei insbesondere die Effizenz beim wahlfreien Zugriff, den Verbrauch an Hauptspeicher und die Probleme beim Löschen und Vergrößern von Dateien.

b) Wie läuft ein wahlfreier Zugriff auf das Byte Nr. 50000 einer Datei beim I-Nodes Konzept ab?

Der entsprechende I-Node sei schon im Hauptspeicher vorhanden. Die Blockgröße sei 4 KB und die Zeigergröße sei 4 Byte. Die Anzahl der direkten Zeiger ist 10. Die Nummerierung der Bytes fängt mit der Nummer 0 an.

Bitte geben Sie an, welche Zeiger daran beteiligt sind, an welcher Position in den Blöcken diese zu finden sind und wohin sie zeigen.

Fortsetzung nächste Seite!

Teilaufgabe 2:

1 Systempuffer

1.1 Erläutern Sie die Schritte, die die Pufferverwaltung durchführen muss, wenn eine Seite angefordert wird.

1.2 Für einen Puffer stehen drei Frames im Hauptspeicher zur Verfügung. Zu Beginn seien alle Frames leer. Geben Sie an, welche Seiten zu welchen Zeitpunkten in den Frames des Prozesses eingelagert sind. Notieren Sie auch die Kontrollzustände für jede Belegung. Geben Sie auch die Anzahl der nötigen Seiteneinlagerungen an.

Verwenden Sie die Ersetzungsstrategie LRU.

Seitenreferenz		1	2	3	4	5	6	7	8	9	10	11	12
Referenzfolge		1	3	2	5	3	1	4	4	2	4	1	5
Hauptspeicher	Kachel 1												
	Kachel 2												
	Kachel 3												
Kontrollzustände (Zeit des letzten Zugriffs)	Kachel 1												
	Kachel 2												
	Kachel 3												

1.3 Nennen Sie zwei weitere Ersetzungsstrategien und charakterisieren Sie diese kurz!

Fortsetzung nächste Seite!

2. Programme, Prozesse, Threads

a) Was versteht man unter einem Programm, einem Prozess und einem Thread?

b) Ein Programm kann von mehreren Prozessen ausgeführt werden.
Wie kann diese Situation entstehen?
Was sind die Gemeinsamkeiten, was sind die Unterschiede zwischen den Prozessen?

c) Wie werden Prozesse vom Betriebssystem verwaltet?
Welche Eigenschaften, Attribute etc. werden für die Verwaltung benutzt?

d) Welche strategischen Entscheidungen muss das Betriebssystem bezüglich der Zuteilung von Ressourcen an Prozesse treffen?
Geben Sie Beispiele für Vorgehensweisen und konkrete Strategien bei den Ressourcen CPU (drei Beispiele) und Speicher (zwei Beispiele) an und erläutern Sie diese.

e) Beschreiben Sie die unterschiedlichen Möglichkeiten zur Realisierung von Threads.
Was sind jeweils die Vor- und Nachteile?

3. Speicher und Adressräume

a) Eine wesentliche Systemfunktion bei Mehrprogrammbetrieb ist Adressraumschutz.
Erläutern Sie, was man darunter versteht und warum Adressraumschutz essentiell ist.
Vor welchen Gefahren schützt er?

b) Erläutern Sie in Stichworten und jeweils anhand einer Skizze die Funktionsweise von "Adressraumschutz durch Eingrenzung" sowie von "Adressraumschutz durch Segmentierung". Erläutern Sie Vor- und Nachteile und geben Sie dabei auch an, zu welchem Zeitpunkt jeweils die Bindung von Programmadressen an Arbeitsspeicheradressen erfolgt, wer die Bindung besorgt und wie der Zugriffsschutz konkret funktioniert.

Fortsetzung nächste Seite!

4 Synchronisation

Ein Datensicherungsprogramm sei so realisiert, dass es in einer ersten Phase eine Reihe von *m* Dateikatalogen parallel auf unterschiedliche Bandlaufwerke sichert. Es stehen *n* Bandlaufwerke zur Verfügung, für jedes Laufwerk gibt es eine Datenstruktur, die alle für den Zugriff erforderlichen Daten enthält. Die Datenstrukturen der freien Geräte sind in einem Ringpuffer verwaltet.

Ein "Haupt-Thread" arbeitet die *m* Dateikataloge der Reihe nach ab. So lange freie Laufwerke verfügbar sind, erzeugt er für jeden Dateikatalog einen eigenen "Arbeiter-Thread", entnimmt die Datenstruktur eines freien Laufwerks aus dem Ringpuffer und übergibt sie zusammen mit der Information über den zu bearbeitenden Dateikatalog dem Arbeiter-Thread. Sind keine Laufwerke mehr verfügbar, wartet er, bis wieder ein Laufwerk frei wird, und erzeugt dann den nächsten Arbeiter-Thread.

Sobald ein Arbeiter-Thread mit der Sicherung seines Dateikatalogs fertig ist, trägt er die Datenstruktur seines Laufwerks wieder in den Ringpuffer ein und terminiert.

Der Haupt-Thread wartet schließlich, bis alle Arbeiter-Threads fertig sind, und erzeugt dann eine Abschlussmeldung.

a) An welchen Stellen ist zwischen den beteiligten Threads Synchronisation erforderlich?

b) Skizzieren Sie in einer Programmiersprache Ihrer Wahl die wesentlichen Datenstrukturen und Schritte im Programmablauf sowie die Synchronisation mit Hilfe von Semaphoren. Erläutern Sie kurz die Bedeutung der von Ihnen eingesetzten Semaphore und welche Werte sie initial haben müssen.

Fortsetzung nächste Seite!

5 SQL

Gegeben ist folgende Datenbank:
SQL> select * from angestellte;

PERS_NR	ABT_NR	NAME	GRUNDGEHALT	BONUS
1	1	Schmidt	40.000	4.000
2	1	Maier	45.000	6.000
3	3	Ewing	38.000	
...	...	...	...	...

SQL> select * from abteilung;

ABT_NR	NAME	TYP	UMSATZ
1	V Region Nord	VERTRIEB	3.000.000
2	V Region Süd	VERTRIEB	4.500.000
3	F&E	ENTWICKLUNG	
...	...	...	...

In `angestellte` sind `PERS_NR` Primärschlüssel und `ABT_NR` Fremdschlüssel.
In `abteilung` ist `ABT_NR` Primärschlüssel.

SQL-Anfragen
Schreiben Sie SQL-Statements für die folgenden Fragestellungen. Wenn Sie möchten, können Sie zur Vereinfachung Sichten anlegen.

a) Welche Abteilungen haben mehr als 10 Angestellte?

b) Welche Mitarbeiter verdienen mindestens 25% mehr als der Durchschnitt ihrer Abteilung? Maßgeblich ist die Summe aus Grundgehalt und Bonus.

c) Welches sind die drei Abteilungen vom Typ VERTRIEB, die das schlechteste Verhältnis von UMSATZ zu Personalkosten haben? Die Personalkosten bestimmen sich dabei aus der Summe der Gehälter der Angestellten der jeweiligen Abteilung (wiederum Grundgehalt plus Bonus). Geben Sie neben dem Abteilungsnamen auch das Verhältnis von Umsatz und Personalkosten aus.

66114

Datenbank- und Betriebssysteme (vertieft)

Herbst 2012

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____	**Herbst 2012**	**66114**
Kennwort: _____		
Arbeitsplatz-Nr.: _____		

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

— Prüfungsaufgaben —

Fach:	**Informatik (vertieft studiert)**
Einzelprüfung:	**Datenbank- und Betriebssysteme**
Anzahl der gestellten Themen (Aufgaben):	**4 Aufgaben, von denen zwei gemäß untenstehender Auswahlregel zu bearbeiten sind!**
Anzahl der Druckseiten dieser Vorlage:	**16**

Zu den zwei Themenschwerpunkten A (Datenbanksysteme) und B (Betriebssysteme) ist jeweils entweder die Aufgabe 1 oder 2 zu wählen. Auf der Vorderseite des Kopfbogens sind im Feld „gewähltes Thema: Nr." die Nummern der beiden gewählten Aufgaben anzugeben (z. B. A2, B1)!

Bitte wenden!

Themenschwerpunkt A
(Datenbanksysteme)

Teilaufgabe 1:

Aufgabe 1: Datenmodellierung: ER-Modell, SQL

In einer Flugdatenbank sollen folgende Informationen modelliert werden:

- Eine Fluggesellschaft hat einen Namen und mehrere Standorte.
- Ein Flughafen hat einen eindeutigen Identifikator (ID), und es werden die Stadt und das Land angegeben, in denen der Flughafen liegt.
- Ein Flug verbindet zwei Flughäfen; er wird durch eine Flugnummer und ein Startdatum identifiziert. Zusätzlich werden die Startzeit, das Landedatum, die Landezeit, die Flugstrecke und die Anzahlen der insgesamt verfügbaren sowie der bereits reservierten Plätze gespeichert.
- Zur Unterstützung von Datenbankanfragen über die Zeit soll es einen Tabelle geben, die jedem Datum den Tag, den Monat, das Jahr und die Jahreszeit (Frühjahr, Sommer, Herbst, Winter) als separate Attribute zuordnet. Eine weitere Tabelle soll jeder Flugnummer eine Fluggesellschaft zuordnen.

a) Modellieren Sie das oben dargestellte Szenario möglichst vollständig in einem ER-Modell! Verwenden Sie wann immer möglich (binäre oder auch höherstellige) Relationships!

b) Übertragen Sie Ihr ER-Modell ins relationale Datenmodell. Erstellen Sie dazu Tabellen mit Hilfe von CREATE TABLE-Statements in SQL. Berücksichtigen Sie die Fremdschlüsselbeziehungen!

c) Es soll die Integritätsbedingung eingehalten werden, dass die Anzahl der bereits reservierten Plätze eines Fluges höchstens so groß ist wie die Anzahl der insgesamt verfügbaren Plätze. Schreiben Sie ein SELECT-Statement, das diese Integritätsbedingung überprüft, indem es die verletzenden Flüge ausgibt.

d) Geben Sie geeignete INSERT-Statements an, die in alle beteiligten Tabellen jeweils mindestens ein Tupel einfügen, so dass alle Integritätsbedingungen erfüllt sind, nachdem alle Einfügungen ausgeführt wurden!

Fortsetzung nächste Seite!

Aufgabe 2: Datenbankanfragen und –änderungen in SQL

Formulieren Sie in SQL die folgenden Anfragen, Views bzw. Update-Statements an die Flugdatenbank aus Aufgabe 1:

a) Welche Flüge der Fluggesellschaft Lufthansa (LH) starten in Frankfurt am Main, Deutschland. Ausgegeben werden sollen Flugnummer und Startdatum!

b) Wie viele Flüge gibt es, gruppiert nach Startdatum, von Deutschland nach Frankreich?

c) Bestimmen Sie alle Verbindungen von München, Deutschland, nach Australien mit einmaligem Umsteigen. Der Anschlussflug soll an demselben Tag erfolgen, an dem der erste Flug beendet wird. Für Flug und Anschlussflug sollen jeweils Flugnummer und Startdatum ausgegeben werden. Bestimmen Sie dabei auch die Gesamtflugstrecke!

d) Bestimmen Sie die Anzahl der Reservierungen für die Flüge von Frankfurt am Main, Deutschland, nach London, Großbritannien, am 17.01.2012!

e) Erstellen Sie eine View, die die Anzahl der ausgebuchten Flüge gruppiert nach Jahr und Jahreszeit angibt!

f) Buchen Sie zwei Plätze des Fluges mit der Flugnummer DL007 am 18.01.2012 auf den Flug mit der Flugnummer LH007 am 19.01.2012 um - nehmen Sie an, dass es noch genügend viele freie Plätze gibt.

Aufgabe 3: Funktionale Abhängigkeiten und Zerlegungen

Gegeben sei das Relationenschema $R = (U, F)$ mit der Attributmenge

$$U = \{A,B,C,D,E\}$$

und der folgenden Menge F von funktionalen Abhängigkeiten:

$$F = \{ B \rightarrow C,\ C \rightarrow BD,\ ACD \rightarrow BE\}$$

a) Geben Sie alle Schlüssel für das Relationenschema R (jeweils mit Begründung) sowie die Nichtschlüsselattribute an!

b) Ist R in 3NF bzw. in BCNF ? Geben Sie jeweils eine Begründung an.

c) Geben Sie eine Basis G von F an! Zerlegen Sie R mittels des Synthesealgorithmus in ein 3NF-Datenbankschema.

d) Zerlegen Sie R bezüglich der funktionalen Abhängigkeit $C \rightarrow BD$ verlustfrei in zwei Relationenschemata $R_i = (U_i, F_i)$, $i = 1, 2$! Ist die Zerlegung unabhängig (mit Begründung)?

Teilaufgabe 2:

Aufgabe 1: Allgemeinwissen

1.1 Bewerten Sie die folgenden Aussagen.
Geben Sie für die folgenden Aussagen an, ob sie richtig oder falsch sind! Begründen Sie ihre Antworten kurz!

a) B-Bäume verlieren ihre Vorteile gegenüber binären Bäumen in Hauptspeicherdatenbanken.
b) Eine Relation kann mehrere Primärschlüsselattribute besitzen.
c) Heutzutage werden zumeist objektorientierte Datenbanksysteme eingesetzt.
d) Fremdschlüsselwerte dürfen nicht NULL sein.
e) Alle Metadaten werde in Tabellen gespeichert.

1.2 Nennen Sie kurz die vier wesentlichen Transaktionen!

Aufgabe 2: Normalformen

2.1 Die Normalisierung von Relationenschemata dient der Vermeidung von Redundanzen und dadurch bedingter Anomalien. Diskutieren Sie anhand des folgenden Beispiels die Einfüge-, Änderungs- und Löschanomalien.

AbtNr	AbtName	AbtOrt	MitarbeiterNr	MitarbeiterName
1	Entwicklung	München	1	Schmidt, H.
2	Forschung	München	2	Müller, G.
1	Entwicklung	München	3	Kohl, H.
3	Fertigung	Nürnberg	4	Weizenbaum, L.
4	Logistik	Passau	5	Nienhof, R.
6	Vertrieb	Würzburg	6	Baier, V.

2.2 In welcher Normalform ist das Beispiel aus Aufgabenteil 2.1? Zeigen Sie, dass alle Bedingungen für diese Normalform erfüllt sind! Welche Bedingung der nächsthöheren Normalform ist verletzt? Berücksichtigen Sie, dass eine Relation nur dann in n-ter Formalform ist, wenn sie die Bedingungen aller m-ten Normalformen mit $m \leq n$ erfüllt!

Fortsetzung nächste Seite!

Aufgabe 3: ER-Modellierung und Relationenmodell

3.1 Nennen und erläutern Sie kurz vier Beispiele für Integritätsbedingungen!

3.2 Erstellen Sie ein ER-Diagramm bestehend aus Entitäts- und Beziehungstypen sowie Attributen! Geben Sie auch die Kardinalitäten mit an!
Verwenden Sie bei Entitäten und Attributen ausschließlich die beschriebenen, soweit dies möglich ist. Verwenden Sie keine künstlichen Schlüssel, es sei denn, diese sind der Aufgabenstellung eindeutig zu entnehmen. Die Kardinalitätseinschränkungen können Sie entweder in der (min,max)-Notation oder der einfachen Notation nach Chen (1:1, 1:N, M:N) angeben.

Als Szenario dient ein **Auktionshaus**, das folgendermaßen beschrieben ist:

Hinweis:
In der Formulierung der Beziehungen sind die Kardinalitätseinschränkungen genau zu beachten. Im Allgemeinen gilt: alles was nicht explizit eingeschränkt ist, muss im Modell auch uneingeschränkt bleiben (keine Einschränkungen in eine Beschreibung hineininterpretieren). Alles was explizit eingeschränkt ist, darf im Modell nicht uneingeschränkt bleiben (alle formulierten Einschränkungen sind zu erkennen).

Szenario:

1. Im Zentrum des Systems stehen die Benutzer. Sie werden durch eine ID identifiziert und haben einen Benutzernamen, sowie eine Adresse, die sich aus Straße, PLZ und Ort zusammensetzt.
2. Benutzer initiieren Auktionen. Jede Auktion ist genau einem Benutzer zugeordnet und wird ebenfalls über eine ID identifiziert. Daneben hat sie einen Namen, sowie Beginn- und Endezeitpunkt.
3. Benutzer geben Gebote für Auktionen ab. Zu jedem Gebot werden Betrag und Zeitpunkt des Gebotes gespeichert. Natürlich soll ein Benutzer auch mehrfach auf eine Auktion bieten können.
4. Zu guter Letzt existiert ein Bewertungssystem. Eine Bewertung wird von einem Benutzer zu einer bestimmten Auktion abgegeben. Zu ihr sollen die Bewertung selbst (ein numerischer Wert) und der Zeitpunkt der Bewertung gespeichert werden. Pro Benutzer und Auktion kann nur eine Bewertung abgegeben werden.

3.3 Erstellen Sie zu dem ER-Diagramm aus Aufgabenteil 3.2 ein Relationenschema!
Seien Sie dabei so vollständig wie möglich – bspw. Berücksichtigung totaler Partizipationen. Vermeiden Sie unnötiges Ausprägen von Relationen bei allen Beziehungen!

Beispiel für die Notation:

```
Relationenname (Primärschlüssel, Attribut1, Attribut2, …,
     Fremdschlüssel[AndereRelation],
     (FKAttrA, FKAttrB)[DritteRelation])

     Attribut2 NOT NULL
```

Fortsetzung nächste Seite!

Aufgabe 4: Schlüsselzugriffe

4.1 Gegeben sei ein anfangs leerer B-Baum (maximale Anzahl der Einträge pro Knoten ist 2k, k=1). Die Zahlen 21, 25, 29, 19, 13, 23 und 20 sind in dieser vorgegebenen Reihenfolge in den B-Baum einzufügen. Löschen Sie dann den Schlüssel 23.

Zeichnen Sie die Zustände des B-Baums nach jeder Strukturänderung! Falls sich durch eine Operation die Struktur des Baumes nicht ändert, muss er nicht neu gezeichnet werden, sondern der einzufügende Wert kann einfach in den Baum eingetragen werden.
Wenn eine Operation mehrere Strukturänderungen erfordert, so ist der Baum nach jeder Änderung neu zu zeichnen.

4.2 Nennen Sie zwei Unterschiede zwischen B-Baum und B*-Baum sowie je einen Vor- und Nachteil beider Varianten!

Aufgabe 5: Anfrageverarbeitung

5.1 Zeichnen Sie für folgendes SQL-Statement einen nicht-optimierten Anfragegraphen.

```
SELECT a1.name AS aname, a1.price,
       artgroup.name AS gname,
       MAX (a2.price) AS max_price
  FROM article a1, article a2, artgroup
 WHERE a1.group_id = artgroup.id
   AND a2.group_id = a1.group_id
   AND artgroup.name LIKE 'Kabel %'
 GROUP BY a1.name, a1.price, artgroup.name
   HAVING MAX(a2.price) > 2*a1.price
```

5.2 Wie kann die relationale Algebra zur Optimierung von SQL-Anfragen eingesetzt werden? Geben Sie zwei Beispiele für heuristische Optimierungsregeln an.

Themenschwerpunkt B
(Betriebssysteme)

Teilaufgabe 1:

Aufgabe 1: Synchronisation mit Sempaphoren

1. Gegeben sei folgende informelle Definition der Operation P auf Semaphor s durch Pseudocode. Der Semaphor sei symbolisch durch int s dargestellt. Es wird als gegeben angenommen, dass die Operation atomar ist und wechselseitig ausgeschlossen ausgeführt wird.

```
public void P (int s) {
   s = s - 1;
   if (s < 0) { Prozess in die Menge der bezüglich s
     wartenden Prozesse einreihen }
}
```

2. Geben Sie eine analoge Definition der Operation V an.

Gegeben sei folgende Implementierung der Synchronisation des Erzeuger-Verbraucher-Problems in Pseudocode. Der Zugriff auf einen Puffer W erfolge durch einen Semaphor wa, der mit 1 initialisiert sei. Zusätzlich gibt es einen Semaphor voll (initalisiert mit 0), der die Elemente im Puffer zählt, sowie einen Semaphor leer (initalisiert mit der Anzahl der freien Pufferplätze), der Anzahl der freien Pufferplätze zählt.

```
Erzeuger E
{
 while (TRUE)
 {
  <produziere>;
  P(leer);
  P(wa);
  <schreibe nach W>;
  V(wa);
  V(voll);
 }
}
```

```
Verbraucher V
{
 while (TRUE)
 {
  P(wa);
  P(voll);
  <entnimm aus W, falls Element da; sonst warte>;
  V(wa);
  V(leer):
  <verarbeite>;
 }
}
```

Was ist das Problem bei dieser Implementierung? Begründen Sie Ihre Antwort kurz.

Fortsetzung nächste Seite!

3. Ein Möbelhändler hat eine Lagerhalle für Tische und Stühle. Beliefert wird er von einem Fabrikanten F, der bei jeder Lieferung einen Tisch und zwei Stühle bringt. Ein Ausfahrer T bringt je Fahrt einen Tisch zum Kunden, ein Ausfahrer S je Fahrt einen Stuhl. Zum Be- und Entladen muss eine Laderampe benutzt werden. Für sich alleine betrachtet laufen die beteiligten Prozesse folgendermaßen ab:

```
Process T
{
 while (TRUE)
 {
  <zur Rampe fahren>;
  <1 Tisch aufladen>;
  <Rampe verlassen>;
 }
}
```

```
Process S
{
 while (TRUE)
 {
  <zur Rampe fahren>;
  <1 Stuhl aufladen>;
  <Rampe verlassen>;
 }
}
```

```
Process F
{
 while (TRUE)
 {
  <zur Rampe fahren>;
  <1 Tisch entladen>;
  <2 Stühle entladen>;
  <Rampe verlassen>;
 }
}
```

Synchronisieren Sie diese drei Prozesse, indem Sie in Pseudocode-Notation in geeigneter Weise Semaphore deklarieren und Semaphor-Operationen einfügen.

Die Pseudocode-Operation zum Deklarieren eines Semaphors mit Namen `name` mit Startwert `n` laute:

```
name(n);
```

Die Pseudocode-Operationen zum Aufrufen von P und V auf der Semaphore mit Namen `name` lauten:

```
P(name)
V(name)
```

Sperrphasen sind möglichst kurz zu halten und folgende Bedingungen müssen erfüllt sein:

- Zur Laderampe kann jeweils nur ein Prozess fahren.
- Der Abholer T darf nur zur Rampe fahren, wenn noch ein Tisch im Lager ist.
- Der Abholer S darf nur zur Rampe fahren, wenn noch ein Stuhl im Lager ist.
- Die Lagerhalle hat beschränkte Kapazität, sie kann höchstens 12 Möbelstücke (Tische oder Stühle) aufnehmen.
- Der Fabrikant darf nur zur Rampe fahren, wenn er seine Lieferung vollständig abladen kann.
- Zu Beginn sei das Lager leer und die Rampe frei.

a) Listen Sie alle benötigten Semaphoren auf und erklären Sie ihren Zweck bzw. Ihre Pragmatik!

b) Fügen Sie in dem oben aufgelisteten Pseudocode des Ablaufs der beteiligten Prozesse an geeigneter Stelle entsprechende Aufrufe zum Deklarieren der Semaphoren und der Aufrufe von P und V ein! Übertragen Sie den Pseudocode dazu auf Ihr Arbeitsblatt.

Fortsetzung nächste Seite!

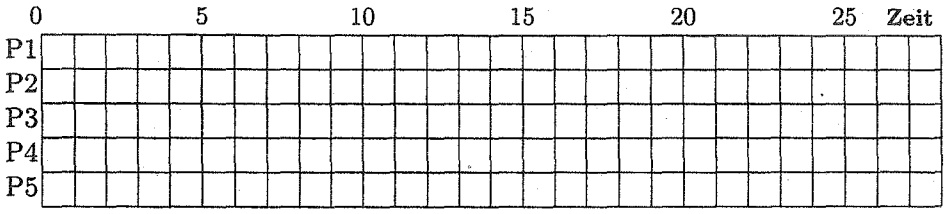
Aufgabe 2: Scheduling

1. Was versteht man allgemein unter **preemptive scheduling** sowie **nonpreemptive scheduling**? Erläutern Sie kurz den Unterschied zwischen beiden Strategien.
2. Gegeben seien fünf Prozesse mit ihren jeweiligen Ankunftszeiten a und Bedienzeiten b:

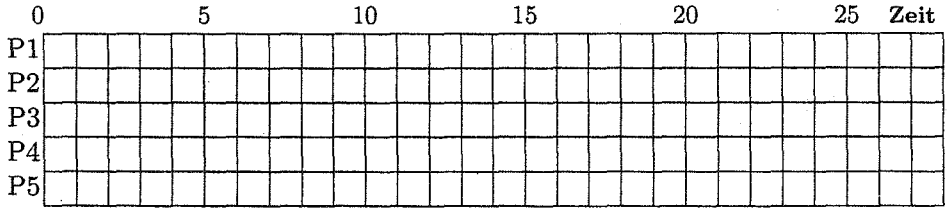
Prozess	*Ankunfszeit a*	*Bedienzeit b*
P1	0	8
P2	6	7
P3	3	5
P4	8	3
P5	5	3

Übertragen Sie die nachfolgenden Gantt-Diagramme und tragen Sie für die angegebenen Scheduling-Verfahren ein, welcher Prozess zu welcher Zeit die CPU erhält. Markieren Sie rechenwillige Prozesse mit "-" und rechnende Prozesse mit "X" in der jeweiligen Spalte.

(a) **First Come First Served (FCFS)**: FCFS bedeutet, dass die Prozesse in der Reihenfolge ihrer Ankunft ausgeführt werden. Prozessunterbrechung ist nicht möglich.



(b) **Shortest Remaining Processing Time (SRPT)**: SPRT bedeutet, dass jeweils der Prozess mit der kürzesten Restbedienzeit ausgeführt wird. Prozessunterbrechung ist nur bei Ankunftszeitpunkten neuer Prozesse möglich. Haben zwei Prozesse gleiche Restbedienzeiten, soll in unserem System dem Prozess Vorrang gegeben werden, der *zuletzt* im System angekommen war.



Fortsetzung nächste Seite!

3. Berechnen Sie die **mittlere Verweilzeit** $\overline{V}$ und die **mittlere Wartezeit** $\overline{W}$ für die beiden Strategien der vorangehenden Teilaufgabe. Nutzen Sie zur Berechnung folgende Formeln:

- c_i bezeichnet die **Abgangszeit** (der Zeitpunkt seiner fertigen Bearbeitung) des Prozesses Pi, a_i seine Ankunftszeit und b_i seine Bedienzeit;
- $v_i = c_i - a_i$ bezeichnet die **Verweilzeit** des Prozesses Pi;
- $w_i = v_i - b_i = c_i - a_i - b_i$ bezeichnet die **Wartezeit** des Prozesses Pi;
- Gibt es $n \in N$ Prozesse, so gilt für die **mittlere Verweilzeit**:

$$\overline{V} = \frac{1}{n} \sum_{i=1}^{n} v_i$$

- Für die **mittlere Wartezeit** gilt:

$$\overline{W} = \frac{1}{n} \sum_{i=1}^{n} w_i$$

4. Warum ist bei interaktiven Systemen weder FCFS noch SRPT besonders gut als Strategie geeignet?

Fortsetzung nächste Seite!

Aufgabe 3: Seitenersetzung

Bei der Ausführung eines Speicherzugriffs bei der virtuellen Speicherverwaltung kann es vorkommen, dass sich die referenzierte Seite nicht im Arbeitsspeicher befindet. Diese Situation wird Seitenfehler (engl. Page fault) genannt.

Die Behandlung eines Seitenfehlers erfordert Maßnahmen zur Ersetzung einer Seite im Arbeitsspeicher, um die gewünschte Seite in eine Kachel (frame) des Arbeitsspeichers einlagern zu können, muss zunächst eine andere Seite vom Arbeitsspeicher auf den Hintergrundspeicher ausgelagert werden oder eine unveränderte Seite kann verworfen werden. Bei der Auswahl der auszulagernden Seite können verschiedene Seitenersetzungs-Strategien angewendet werden.

1. Erklären Sie die Seitenersetzungs-Strategien **LIFO** und **LFU** jeweils knapp.
2. Sei eine Menge von Seiten $N = \{0, 1, 2, 3, 4, 5\}$ und eine Menge von Kacheln (frames) $F = \{f_1, f_2, f_3, f_4\}$ gegeben. Nun wird in folgender Reihenfolge auf die Seiten zugegriffen:

$$w = 1\ 3\ 5\ 4\ 4\ 2\ 3\ 2\ 1\ 0\ 5\ 3\ 5\ 0\ 4\ 3$$

Vollziehen Sie die Strategie **LRU (Least Recently Used)** schrittweise nach, indem Sie die folgende Tabelle auf Ihr Arbeitsblatt übertragen und vervollständigen! Notieren Sie in der betreffenden Spalte, ob ein Seitenfehler auftrat. Sie können optional die Spalte "Notizen" geeignet verwenden, wenn Sie möchten.

Anfrage	f_1	f_2	f_3	f_4	Seitenfehler	Notizen
1	1	-	-	-	ja	
3	1	3	-	-	ja	
5	1	3	5	-	ja	
4	1	3	5	4	ja	
4	1	3	5	4	nein	
2						
3						
2						
1						
0						
5						
3						
5						
0						
4						
3						

3. Wenn man die Menge der Kacheln vergrößert, sinkt dann zwangsläufig bei allen Seitenersetzungs-Strategien die Zahl der Seitenfehler? (Ohne Begründung)

Teilaufgabe 2:

Aufgabe 1:

Die folgenden drei Jobs stehen zu den angegebenen Zeiten (in Minuten) zur Ausführung an:

Job	Ankunftszeit	Bearbeitungsdauer
T1	0	20
T2	4	10
T3	10	6

(a) Stellen Sie die zeitlichen Abläufe in Gantt-Diagrammen (Balkendiagramme über der Zeitachse) dar und berechnen Sie dazu den Turnaround eines jeden Jobs, sowie den Average Turnaround bei den folgenden Scheduling-Strategien:

(1) First-Come-First-Served (FCFS)
(2) Shortest-Job-First (SJF)
(3) Preemptive-Shortest-Job-First (Preemptive SJF)
(4) Round-Robin (RR) mit Zeitquantum q = 8

(b) Zusätzlich zu den obigen Jobs T1, T2, T3 (der Kategorie Batch) sollen zu den Zeitpunkten 6 und 12 die Jobs T4 und T5 (der Kategorie Interactive) mit jeweils einer Laufzeit von 10 Zeiteinheiten zur Bearbeitung anstehen.

Nun soll eine Multi-Level-Scheduling-Strategie verfolgt werden derart, dass jeweils eine Warteschlange für die Kategorien Interactive und Batch existiert, wobei das Scheduling bei interaktiven Jobs mit dem Round-Robin-Verfahren mit Zeitquantum q = 4 und bei den Batch-Jobs nach FCFS abläuft.

Die Warteschlange der interaktiven Jobs hat Priorität über die Warteschlange der Batch-Jobs, d. h. falls gerade ein Batch-Job bearbeitet wird und es kommt ein neuer interaktiver Job an, so wird der Batch-Job solange angehalten, bis keine interaktiven Jobs mehr im System sind. Geben Sie die Wartezeit eines jeden Jobs an.

Fortsetzung nächste Seite!

Aufgabe 2:

Ein Rechnersystem mit vier Betriebsmitteltypen R_1, R_2, R_3, R_4 und fünf Prozessen P_1, P_2, P_3, P_4, P_5 befinde sich im unten gezeigten Zustand.

		R_1	R_2	R_3	R_4
Verfügbarkeitsvektor		0	2	2	5
Belegte Betriebsmittel	P_1	2	3	0	6
	P_2	0	0	2	0
	P_3	4	5	2	3
	P_4	2	1	0	0
	P_5	4	1	0	0
Maximale Anforderungen inkl. bereits belegter Betriebsmittel	P_1	2	5	0	6
	P_2	0	5	2	7
	P_3	6	5	4	3
	P_4	2	1	0	0
	P_5	6	5	0	6

(a) Erklären Sie informell die Vorgehensweise des Banker's-Algorithmus'!

(b) Prüfen Sie zunächst, ob sich das obige System in einem sicheren Zustand befindet!

(c) Prüfen Sie mit Hilfe des Banker's-Algorithmus', ob eine neu hinzukommende Ressourcen-Anforderung (0 2 0 4) des Prozesses P_4 erfüllt werden sollte!

Fortsetzung nächste Seite!

Aufgabe 3:

Bei einer Auktion mit einem Auktionator und 50 Interessenten werde der erzielbare Verkaufspreis des zu versteigernden Kunstwerks vom Auktionator folgendermaßen ermittelt:

Der Einstiegspreis von 1000,-- EUR wurde vom Auktionator festgesetzt. Zu diesem Preis sind auch garantiert alle Interessenten zahlungswillig.

In jeder Runde antwortet nun jeder der 50 Interessenten genau einmal, ob er bereit ist, den aktuellen Preis zu bezahlen. Die Antwortreihenfolge der Interessenten ist dabei beliebig.

Erst wenn sich alle 50 Interessenten erklärt haben, wird vom Auktionator geprüft, ob es mehr als einen zahlungswilligen Interessenten gibt.

- Ist dies der Fall, so setzt der Auktionator einen um 100,-- EUR höheren Preis für die nächste Runde fest.
- Findet sich für einen aktuellen Preis kein Zahlungswilliger mehr, so ist die Auktion beendet und der Auktionator kassiert den vorherigen, also den noch um 100,-- EUR niedrigeren Preis von einem beliebigen der zuletzt zahlungswilligen Interessenten.

Übertragen und ergänzen Sie die nachfolgenden Pseudocode-Fragmente für den Auktionator und die Interessenten-Prozesse zu einer korrekten Lösung bezüglich der oben formulierten Vorgaben.
Als Semaphor-Operationen sind wait(.) und signal(.) zu verwenden.
Kommentieren Sie dabei Ihre jeweiligen Verwendungen der wait(.)- und signal(.)-Operationen.

Fortsetzung nächste Seite!

Globale Initialisierungen:

```
/* wechselseitiger Ausschluss von Auktionator und der Gruppe aller Interessenten */
Semaphore mutexPreisRunde = new Semaphore(1);

/* zur Synchronisation der Interessenten */
Semaphore mutexInteressenten = new Semaphore(0);

int preis = 1000;
int antworten = 0;
int zahlungswillige = 50;
```

Auktionator:

```
while( Auktion nicht beendet ) {
        ..................................................
        ..................................................

        if (zahlungswillige > 1) {
                ..................................................
                ..................................................
                ..................................................
                ..................................................
                ..................................................
                ..................................................


        }
        else {
                Auktion beenden;
                Preis kassieren;
                Kunstwerk übergeben;
        }
}
```

Interessent:

```
while( Auktion nicht beendet ) {

        ..................................................
        ..................................................
        ..................................................
        ..................................................

        if (neuer Preis noch ok) {

                ..................................................
                ..................................................
                ..................................................
        }

        ..................................................
        ..................................................
        ..................................................
        ..................................................
        ..................................................
        ..................................................

}
```

Fortsetzung nächste Seite!

Aufgabe 4:

(a) Beschreiben Sie knapp die drei Speicherverwaltungsstrategien **First-Fit**, **Best-Fit** und **Worst-Fit**. Nennen Sie zu jeder Strategie mindestens einen Vorteil und einen Nachteil!

(b) Gegeben seien die folgenden freien Speicherblöcke in der Freispeicher-Liste in genau dieser Reihenfolge, wobei 1MB = 1024KB. Die Blöcke liegen jeweils nicht direkt nebeneinander im RAM.

1000KB	1MB	200KB	110KB	4MB

Es werden nun an das Betriebssystem in folgender Reihenfolge Speicheranforderungen gestellt:

1000KB	900KB	200KB	100KB	110KB	2MB

Wie teilt jede der Strategien **First-Fit**, **Best-Fit** und **Worst-Fit** den Speicher zu? Zur vollständigen Lösung gehört die Liste aller Frei-Blöcke nach jeder erfüllten Anforderung.

(c) Welches Verfahren schneidet für das obige Beispiel am schlechtesten ab bzgl. Erfüllung der Anfragen? Begründen Sie Ihre Antwort!

(d) Welches Verfahren arbeitet in obigem Beispiel am effizientesten bzgl. unerwünschter Fragmentierung des freien Speichers? Begründen Sie Ihre Antwort!

66114

Datenbank- und Betriebssysteme (vertieft)

Frühjahr 2013

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____	**Frühjahr 2013**	**66114**
Kennwort: _____		
Arbeitsplatz-Nr.: _____		

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

— Prüfungsaufgaben —

Fach: **Informatik (vertieft studiert)**

Einzelprüfung: **Datenbank- und Betriebssysteme**

Anzahl der gestellten Themen (Aufgaben): **4 Aufgaben, von denen zwei gemäß untenstehender Auswahlregel zu bearbeiten sind!**

Anzahl der Druckseiten dieser Vorlage: **22**

Zu den zwei Themenschwerpunkten A (Datenbanksysteme) und B(Betriebssysteme) ist jeweils entweder die Aufgabe 1 oder 2 zu wählen.
Auf der Vorderseite des Kopfbogens sind im Feld „gewähltes Thema: Nr." die Nummern der beiden gewählten Aufgaben anzugeben (z. B. A2, B1)!

Bitte wenden!

Themenschwerpunkt A
(Datenbanksysteme)

Teilaufgabe 1:

1. Allgemeinwissen

a) Definieren Sie den Begriff Datenbanksystem und grenzen Sie die Funktionalitäten eines Datenbanksystems von denen eines Dateisystems ab.
b) Nennen Sie die Phasen des Datenbankentwurfsprozesses und charakterisieren Sie diese kurz.
c) Nennen und beschreiben Sie kurz die vier wesentlichen Eigenschaften von Transaktionen.
d) Nennen und erläutern Sie kurz vier Beispiele für Integritätsbedingungen.

2. ER-Modellierung und Relationenmodell

a) Erstellen Sie ein ER-Diagramm bestehend aus Entitäts- und Beziehungstypen sowie Attributen für folgendes Szenario. Geben Sie auch die Kardinalitäten mit an. Verwenden Sie bei Entitäten und Attributen ausschließlich die beschriebenen, soweit dies möglich ist. Verwenden Sie keine künstlichen Schlüssel, es sei denn, diese sind der Aufgabenstellung eindeutig zu entnehmen. Die Kardinalitätseinschränkungen können Sie entweder in der (min,max)-Notation oder der einfachen Notation nach Chen (1:1, 1:N, M:N) angeben.

Als **Szenario** dient eine Mitarbeiterverwaltung, die folgendermaßen beschrieben ist:

1. Mitarbeiter werden über ihre Sozialversicherungsnummer identifiziert. Sie haben einen Namen, bestehend aus Vor- und Nachnamen, sowie ein Geburtsdatum und ein Alter, das aus dem Geburtsdatum errechnet wird.
2. Ein Mitarbeiter kann der Vorgesetzte anderer Mitarbeiter sein.
3. Jeder Mitarbeiter ist einer Abteilung zugeordnet. Eine Abteilung wird über ihren Namen und eine Nummer identifiziert (z.B. „Entwicklung", „3"). Zu einer Abteilung werden mehrere Orte gespeichert, an denen die Abteilung angesiedelt ist.
4. Mitarbeiter können Abteilungen leiten. Jede Abteilung wird von einem Mitarbeiter geleitet, aber nicht jeder Mitarbeiter muss eine Abteilung leiten.
5. Abteilungen können Projekte realisieren. Ein Projekt wird durch eine Nummer identifiziert und findet an einem Ort statt.
6. Jeder Mitarbeiter arbeitet an einem oder mehreren Projekten. An jedem Projekt muss mindestens ein Mitarbeiter arbeiten. Zusätzlich wird gespeichert, ab wann ein Mitarbeiter an einem Projekt arbeitet.

Hinweis: In der Formulierung der Beziehungen sind die Kardinalitätseinschränkungen genau zu beachten. Im Allgemeinen gilt: Alles was nicht explizit eingeschränkt ist, muss im Modell auch uneingeschränkt bleiben (keine Einschränkungen in eine Beschreibung hineininterpretieren). Alles was explizit eingeschränkt ist, darf im Modell nicht einfach uneingeschränkt bleiben (alle formulierten Einschränkungen sind zu erkennen).

Fortsetzung nächste Seite!

b) Erstellen Sie zu dem E/R-Diagramm aus Teil (a) ein Relationenschema. Seien Sie dabei so vollständig wie möglich – bspw. Berücksichtigung totaler Partizipationen. Vermeiden Sie unnötiges Ausprägen von Relationen bei allen Beziehungen.

Beispiel für die Notation:

```
Relationenname (Primärschlüssel, Attribut1, Attribut2, ...,
    Fremdschlüssel[AndereRelation],
    (FremdschlüsselattributA,
    FremdschlüsselattributB)[DritteRelation])
    Attribut2 NOT NULL
```

3. Normalformen

Gegeben sei die nachfolgende relationale Datenbank mit unterstrichenen Schlüsselattributen. Sie enthält Informationen über Opernaufführungen. Relation "Opernvorstellungen":

Oper	Komponist	Datum	Ort	Opernhaus	Plätze	Dirigent	Dauer	Solist
Rigoletto	Verdi	08.04.2013	Wien	Staatsoper	2298	Lopez-Cobos	2:00	Polenzani
Don Carlos	Verdi	09.04.2013	Wien	Staatsoper	2298	de Billy	4:00	Youn
Rigoletto	Verdi	11.04.2013	Dresden	Semperoper	1300	Fisch	2:00	Berrugi
Macbeth	Verdi	16.04.2013	Mailand	Scala	2000	Gergiev	2:45	Vassallo
Carmen	Bizet	20.05.2013	Wien	Staatsoper	2298	de Billy	2:45	Garanca
Carmen	Bizet	02.06.2013	Wien	Staatsoper	2298	de Billy	2:45	Alagna
Don Carlos	Verdi	12.10.2013	Mailand	Scala	2000	Luisi	4:00	Pape

Gegeben seien ebenso die folgenden funktionalen Abhängigkeiten:

FD1: Oper → Komponist, Dauer
FD2: Ort → Opernhaus, Plätze
FD3: Oper, Ort → Dirigent
FD4: Oper, Datum, Ort → Solist

a) Beschreiben Sie kurz, welche Redundanzen in der Datenbank vorhanden sind und welche Anomalien auftreten können. Geben Sie ein Beispiel für jede Anomalie an.

b) Welcher Normalform genügt das angegebene Relationenschema? Begründen Sie Ihre Entscheidung.

c) Erläutern Sie kurz, welchen Nachteil Normalisierung allgemein für die Anfragebearbeitung haben kann.

Fortsetzung nächste Seite!

4. Anfrageverarbeitung

a) Zeichnen Sie für folgendes SQL-Statement einen nicht-optimierten Anfragegraphen.

```
SELECT e1.name AS ename, e1.salary, dept.name AS dname,
       AVG (e2.salary) AS avg_salary
  FROM employee e1, employee e2, dept
 WHERE e1.dept_id = dept.id
   AND e2.dept_id = e1.dept_id
   AND dept.name LIKE 'Sales %'
 GROUP BY e1.name, e1.salary, dept.name
 HAVING AVG(e2.salary) > 2*e1.salary
```

b) Wie kann die relationale Algebra zur Optimierung von SQL-Anfragen verwendet werden?

Fortsetzung nächste Seite!

Teilaufgabe 2:

Aufgabe 1: Relationale Algebra und SQL

Gegeben sei folgendes Schema:

- Züge:
  {[ZugNr: integer, AnzahlWagons : integer]}
- Städte:
  {[StadtNr: integer, Name: string]}
- Bahnhöfe:
  {[BahnhofNr: integer, Name: string, LiegtInStadtNr: integer]}
- Verbindungen:
  {[ZugNr: integer, BahnhofNrVon: integer, BahnhofNrNach: integer, AbfahrtsZeit: date, AnkunftsZeit: date]}

Hinweis:
Ein Element $\in$ date hat der Einfachheit halber die Form: „5:32PM". Wir nehmen an, dass Verbindungen jeden Tag gelten.

Aufgabe 1.1:
Geben Sie tabellarisch eine Beispielausprägung mit mindestens zwei Zeilen pro Tabelle des angegeben Schemas an.

Aufgabe 1.2

1. Finden Sie die Züge, welche morgens um 5 Uhr von dem Bahnhof „Berlin Ostbahnhof" zu dem Bahnhof „München Hauptbahnhof" fahren **(in relationaler Algebra).**
2. Finden Sie die Städte, von denen mindestens ein Zug mit mehr als drei Wagons zu einem Bahnhof in Mannheim fährt **(in relationaler Algebra).**
3. In welcher Stadt liegt der Bahnhof „Coesfeld Schulzentrum"? **(in relationaler Algebra)**
4. Finden Sie die Namen der Städte an denen zur gleichen Uhrzeit mehr als ein Zug ankommt **(in relationaler Algebra).**

Fortsetzung nächste Seite!

Aufgabe 1.3:

Beantworten Sie folgende Fragen in SQL:

1. An welchen Uhrzeiten fahren Züge von „München Hauptbahnhof" nach „Berlin Ostbahnhof"?
2. Finden Sie den Zug oder die Züge mit den meisten Wagons.
3. Welche Bahnhöfe (BahnhofNr angeben) sind mit dem Bahnhof "München Hauptbahnhof" direkt verbunden? Geben Sie die Bahnhöfe duplikatfrei aus. „München Hauptbahnhof" selbst darf nicht im Ergebnis vorkommen.
4. Finden sie alle Bahnhöfe (BahnhofNr), die mit **genau zweimal** Umsteigen von dem Bahnhof „München Hauptbahnhof" aus erreicht werden können.
5. Geben Sie die Namen der Bahnhöfe an, die überdurchschnittlich oft als Ziel vorkommen. Geben Sie außerdem zu jedem der Bahnhöfe die Anzahl der ankommenden Züge an. Beachten Sie hierbei, dass auch Bahnhöfe, die von keinem Zug erreicht werden können, den Durchschnitt beeinflussen.

Aufgabe 1.4:

Gegeben sei folgende SQL Anfrage:

```
SELECT z.ZugNr FROM
        Zug z, Bahnhof b1, Bahnhof b2, Verbindung v
WHERE
        v.BahnhofNrVon = b1.BahnhofNr AND
        v.BahnhofNrNach = b2.BahnhofNr AND
        b1.Name = "München" AND
        b2.Name = "Berlin" AND
        z.ZugNr = v.ZugNr
```

Übertragen Sie das gegebene SQL Statement in die relationale Algebra und optimieren Sie den Ausdruck logisch. Verwenden Sie die Operatorbaum-Syntax.

Fortsetzung nächste Seite!

Aufgabe 2: ER Modellierung

Sie sind beauftragt, bei der Entwicklung des ersten Prototypen eines sozialen Netzwerkes die Datenmodellierung zu übernehmen. Der erste Prototyp soll folgende Anforderungen erfüllen:

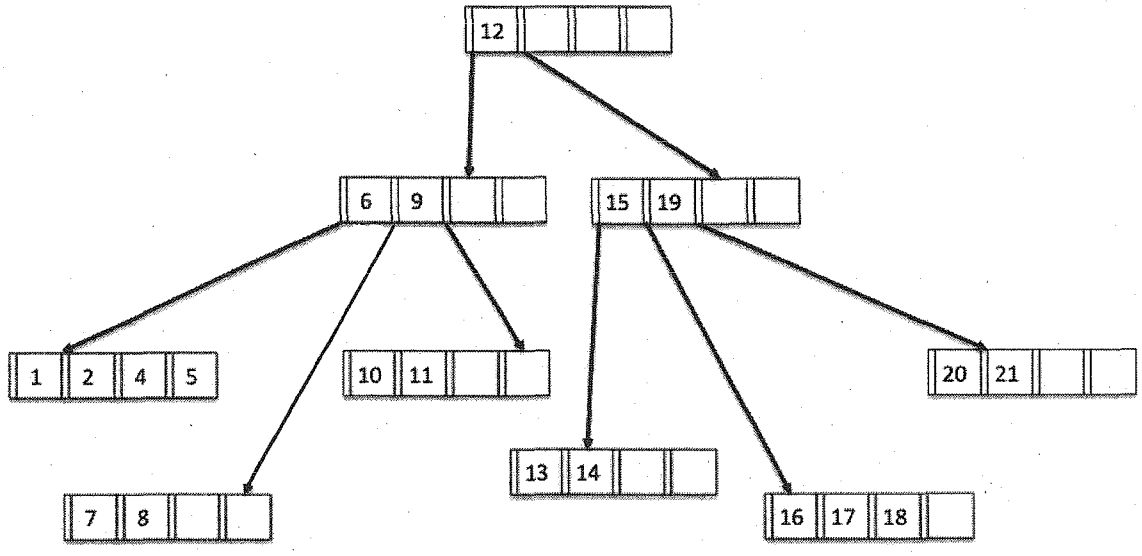
Benutzer können sich mit ihrem Nutzernamen und Kennwort anmelden. Ein Benutzer hat eine sogenannte Wall, auf die andere Benutzer Texte schreiben können. Benutzer können Freunde von anderen Benutzern sein. Außerdem können Benutzer sich in verschiedenen Gruppen anmelden. Gruppen haben ebenfalls eine sogenannte Wall, auf der Gruppenmitglieder Nachrichten hinterlassen können.

Erstellen Sie ein ER-Diagramm, welches die obengenannten Anforderungen erfüllt. Denken Sie daran Ihrem ER-Diagramm Funktionalitätsangaben hinzuzufügen.

Fortsetzung nächste Seite!

Aufgabe 3: B-Baum

Gegeben sei der folgende B-Baum



1. Was bedeutet *k* bei einem B-Baum mit Grad *k*? Geben Sie *k* für den obigen B-Baum an.
2. Was sind die Vorteile von B-Bäumen im Vergleich zu binären Baumen?
3. Wozu werden B-Bäume in der Regel verwendet und wieso?
4. Fügen Sie den Wert **3** in den B-Baum ein, und zeichnen Sie den vollständigen B-Baum nach dem Einfügen und möglichen darauf folgenden Operationen.
5. Entfernen Sie aus dem **ursprünglichen** B-Baum den Wert **19**. Zeichnen Sie das vollständige Ergebnis nach dem Löschen und möglichen darauf folgenden Operationen. Sollte es mehrere richtige Lösungen geben, reicht es eine Lösung zu zeichnen.

Themenschwerpunkt B
(Betriebssysteme)

Teilaufgabe 1:

Aufgabe 1:

Synchronisation mit Semaphoren

a) Gegeben sei folgende informelle Definition der Operation `V` auf Semaphore `s` durch Pseudocode. Die Semaphore sei symbolisch durch `int s` dargestellt. Es wird als gegeben angenommen, dass die Operation atomar ist und wechselseitig ausgeschlossen ausgeführt wird.

```
public void V (int s) {
   s = s + 1;
   if (s <= 0) { Führe genau einen der bezüglich s
     wartenden Prozesse in den Zustand rechenwillig über }
}
```

Geben Sie eine analoge Definition der Operation `P` an.

b) Gegeben sei ein System mit drei Prozessen V1, V2, E und vier boolschen Semaphoren R, S, T, U (jeweils mit 1 initialisiert). Die drei Prozesse laufen folgendermaßen ab:

```
Process V1
{
 while (TRUE)
 {
  P(S);
  P(T);
  P(U);
  <Aktion V1>
  V(T);
  V(S);
  V(U);
 }
}
```

```
Process V2
{
 while (TRUE)
 {
  P(R);
  P(T);
  P(U);
  <Aktion V2>
  V(R);
  V(U);
  V(T);
 }
}
```

```
Process E
{
 while (TRUE)
 {
  P(U);
  P(T);

  <Aktion E>
  V(T);
  V(U);

 }
}
```

Ist in dem gegebenen Prozesssystem eine Verklemmung möglich? Begründen Sie Ihre Antwort.

Fortsetzung nächste Seite!

c) Ein Obsthändler hat ein Geschäft für Obst mit nur einem Parkplatz. Beliefert wird er von einem Lieferanten L, der bei jeder Lieferung drei Kisten Obst bringt. Kunden K holen jeweils eine Kiste Obst ab. Zum Anliefern und Abholen muss der einzige Parkplatz benutzt werden. Für sich alleine betrachtet laufen die beteiligten Prozesse folgendermaßen ab:

```
Process K
{
 while (TRUE)
 {
  <zum Parkplatz fahren>;
  <1 Kiste Obst abholen>;
  <Parkplatz verlassen>;
 }
}
```

```
Process L
{
 while (TRUE)
 {
  <zum Parkplatz fahren>;
  <3 Kisten Obst anliefern>;
  <Parkplatz verlassen>;
 }
}
```

Synchronisieren Sie diese zwei Prozesse, indem Sie in Pseudocode-Notation in geeigneter Weise Semaphore deklarieren und Semaphor-Operationen einfügen.

Die Pseudocode-Operation zum Deklarieren eines Semaphors mit Namen `name` mit Startwert `n` laute:

```
name(n);
```

Die Pseudocode-Operationen zum Aufrufen von P und V auf der Semaphore mit Namen `name` lauten:

```
P(name);
V(name);
```

Sperrphasen sind möglichst kurz zu halten und folgende Bedingungen müssen erfüllt sein:

i) Der Parkplatz darf nur von einem Prozess gleichzeitig benutzt werden.

ii) Ein Kunde K darf nur zum Parkplatz fahren, wenn noch mindestens eine Kiste Obst im Geschäft ist.

iii) Das Geschäft hat beschränkte Kapazität; es kann höchstens 12 Kisten Obst aufnehmen.

iv) Der Lieferant L darf nur zum Parkplatz fahren, wenn er seine Lieferung vollständig abladen kann.

v) Zu Beginn sei das Geschäft leer und der Parkplatz frei.

d) Listen Sie alle benötigten Semaphoren auf und erklären Sie ihren Zweck bzw. ihre Pragmatik.

e) Fügen Sie in dem oben aufgelisteten Pseudocode des Ablaufs der beteiligten Prozesse an geeigneter Stelle entsprechende Aufrufe zum Deklarieren der Semaphoren und der Aufrufe von P und V ein.

Fortsetzung nächste Seite!

Aufgabe 2:

Scheduling

a) Erläutern Sie kurz allgemein die Aufgabe des *Schedulers* im Betriebssystem.

b) Gegeben seien fünf Prozesse mit ihren jeweiligen Ankunftszeiten, Bedienzeiten und Deadlines:

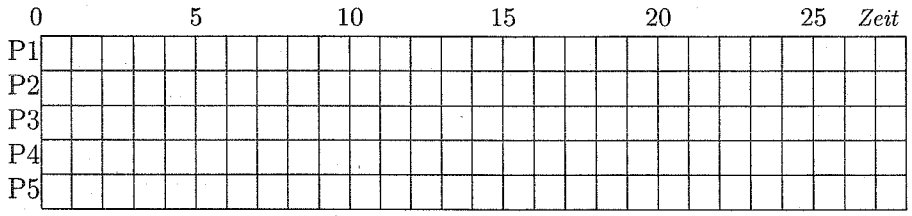
Prozess	*Ankunfszeit*	*Bedienzeit*	*Deadline*
P1	0	4	19
P2	7	5	15
P3	3	4	17
P4	10	2	18
P5	11	3	21

Die *Deadline* ist der Zeitpunkt, an dem ein Prozess spätestens beendet werden sollte. Alle Prozesse sollen aber in jedem Fall komplett bedient werden, auch wenn ihre Deadline bereits abgelaufen ist.

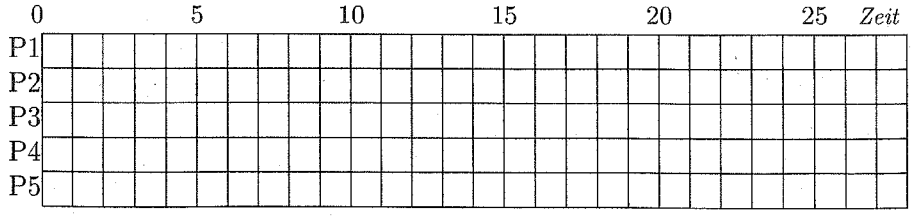
Ein *Kontextwechsel* benötige in diesem System *1 Zeiteinheit.* Der Kontextwechsel ist auch für den ersten rechnenden Prozess zu berücksichtigen.

Übertragen Sie die nachfolgenden Gantt-Diagramme auf Ihr Blatt und tragen Sie für die angegebenen Scheduling-Verfahren ein, welcher Prozess zu welcher Zeit die CPU erhält. Markieren Sie rechenwillige Prozesse mit - und rechnende Prozesse mit X in der jeweiligen Spalte.

i) *Shortest Job First (SJF)*: Prozess mit der kürzesten Bedienzeit wird ausgeführt, nicht unterbrechend.



ii) *Earliest Deadline First (EDF)*: Prozess mit der nächsten, in der Zukunft liegenden, Deadline wird ausgewählt, unterbrechend nur beim Eintreffen neuer Prozesse.



Fortsetzung nächste Seite!

c) Berechnen Sie die *mittlere Verweilzeit* $\overline{V}$ für die beiden Strategien der vorangehenden Teilaufgabe. Nutzen Sie zur Berechnung folgende Formel:

- c_i bezeichnet die **Abgangszeit** (der Zeitpunkt seiner fertigen Bearbeitung) des Prozesses Pi und a_i seine Ankunftszeit
- $v_i = c_i - a_i$ bezeichnet die **Verweilzeit** des Prozesses Pi;
- Gibt es $n \in N$ Prozesse, so gilt für die **mittlere Verweilzeit**:

$$\overline{V} = \frac{1}{n} \sum_{i=1}^{n} v_i$$

d) Für welche Art von Systemen erscheint Earliest Deadline First (EDF) prinzipiell sehr gut geeignet?

Fortsetzung nächste Seite!

Aufgabe 3:

Seitenersetzung

a) Gegeben sei ein System mit drei verfügbaren Frames ***F0, F1, F2***, welche zu Beginn leer sind. Tragen Sie für die Seitenanforderungsfolge **1, 2, 3, 4, 5, 1, 2, 5, 1, 2, 3, 4, 5** und die Verfahren **FIFO** (First-In-First-Out), **LRU** (Least-Recently-Used) und **OPT** (Belady's optimale Strategie) die jeweils entstehende Belegung der Speicherkacheln ein.

Übertragen Sie hierfür die unten gegebenen Schemata auf Ihr Blatt und vervollständigen Sie sie. Geben Sie für jedes Verfahren die Anzahl der resultierenden Seitenfehler (engl. page faults) an.

FIFO

	1	2	3	4	5	1	2	5	1	2	3	4	5
F0													
F1													
F2													

LRU

	1	2	3	4	5	1	2	5	1	2	3	4	5
F0													
F1													
F2													

OPT

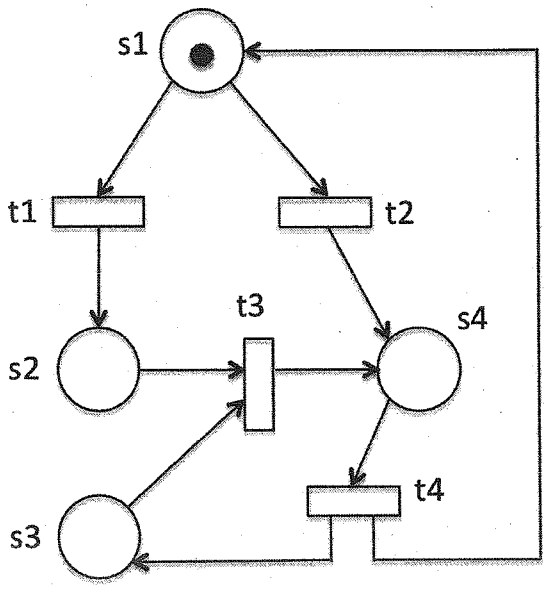
	1	2	3	4	5	1	2	5	1	2	3	4	5
F0													
F1													
F2													

b) Die optimale Strategie von Belady minimiert die Anzahl der Seitenfehler. Warum wird sie trotzdem in realen Paging-Systemen nicht eingesetzt?

c) Was versteht man unter Thrashing? Nennen Sie mindestens eine sinnvolle Gegenmaßnahme, die das Betriebssystem ergreifen kann.

Fortsetzung nächste Seite!

Aufgabe 4:
Petri-Netze
Gegeben sei folgendes boolsches Petri-Netz



a) Erläutern Sie zunächst allgemein, was man unter *Aushungern (engl. starvation)* bei Petri-Netzen versteht.

b) Geben Sie den *Erreichbarkeitsgraphen* des gegebenen Petri-Netzes an.

c) Existiert in dem gegebenen Petri-Netz ein *nicht-sequentieller Ablauf*? Falls ja, geben Sie ein Beispiel an. Falls nein, begründen Sie Ihre Antwort.

d) Ist in dem gegebenen Petri-Netz eine *Verklemmung (engl. deadlock)* erreichbar? Begründen Sie Ihre Antwort.

Fortsetzung nächste Seite!

Teilaufgabe 2:

Aufgabe 1:

Prozesse
Beantworten Sie folgende Fragen zum Thema Prozesse.

a) Welche drei Arten von Informationen enthält der Prozesskontrollblock (PCB)? Nennen Sie je ein Bespiel für jede der drei Arten.

b) Was ist der Unterschied zwischen PCB und Prozess-Image?

c) Nennen Sie einen Vorteil der sich ergibt, wenn ein Prozessmodell zwischen vielen verschiedenen Prozesszuständen differenziert. Geben Sie dafür ein Beispiel an!

d) Nennen Sie einen Nachteil der sich ergibt, wenn ein Prozessmodell zwischen vielen verschiedenen Prozesszuständen differenziert. Geben Sie dafür ein Beispiel an!

Fortsetzung nächste Seite!

Aufgabe 2:

Multilevel Feedback Queuing

In dieser Aufgabe sollen Sie ein modifiziertes Multilevel Feedback Queuing (MLFQ) Verfahren anwenden, welches auf unterschiedlichen Leveln verschieden lange Zeitscheiben verwendet. Die drei folgenden Prioritätsklassen stehen zur Verfügung:

Priorität	Verfahren
0	FIFO mit Quantum 1
1	FIFO mit Quantum 2
2	Round Robin mit Quantum 4

Das Quantum gibt die Dauer der Zeitscheibe auf dem entsprechenden Level an.

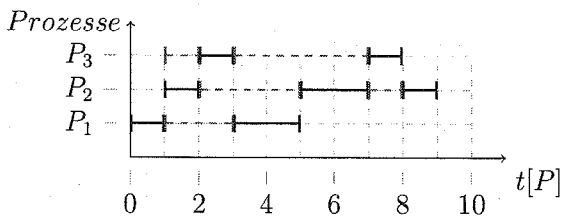
Gehen Sie davon aus, dass jeder Prozess sein Zeitquantum stets vollständig ausnutzt d.h. kein Prozess gibt den Prozessor freiwillig frei (Ausnahme: bei Prozessende).

Trifft ein Prozess zum Zeitpunkt t ein, so wird er direkt zum Zeitpunkt t beim Scheduling berücksichtigt. Wird ein Prozess zum Zeitpunkt t' unterbrochen, so reiht er sich auch zum Zeitpunkt t' wieder in die Warteschlange ein. Sind zwei Prozesse absolut identisch bezüglich ihrer relevanten Werte, so werden die Prozesse nach aufsteigender Prozess–ID in die Warteschlange eingereiht. Diese Annahme gilt sowohl für neu im System eintreffende Prozesse, als auch für den Prozess, dem der Prozessor u.U. gerade entzogen wird!

Beispiel: Es seien folgende Ankunfts– und Rechenzeiten für die drei Beispielprozesse P_1, P_2 und P_3 gegeben:

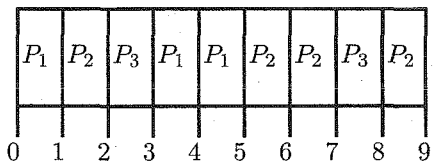
Prozess	Ankunftszeitpunkt	Rechenzeit
P_1	0	3
P_2	1	4
P_3	1	2

Das folgende Diagramm veranschaulicht das Scheduling der drei Prozesse P_1, P_2 und P_3 entsprechend dem oben beschriebenen Verfahren.



Fortsetzung nächste Seite!

Das Scheduling kann man auch mit Hilfe eines Gantt–Charts visualisieren:

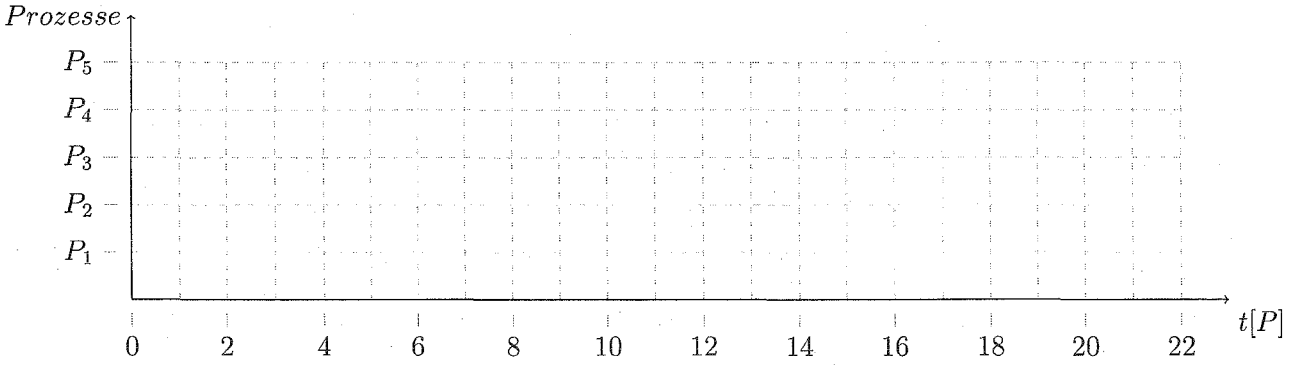


Bearbeiten Sie unter den gegebenen Voraussetzungen nun die folgenden Aufgaben:

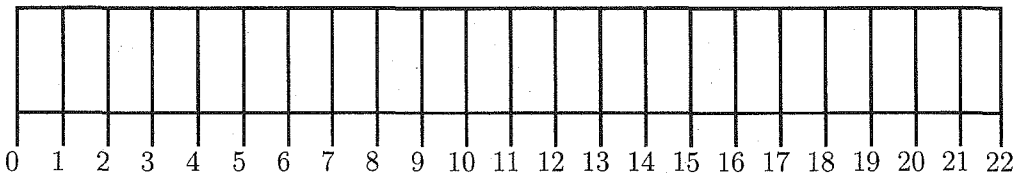
a) Gegeben seien die Prozesse $P_1, \ldots, P_5$ mit den folgenden Ankunfts– und Rechenzeiten:

Prozess	Ankunftszeitpunkt	Rechenzeit
P_1	1	4
P_2	3	4
P_3	3	2
P_4	5	6
P_5	7	5

Übertragen Sie das nachfolgende Diagramm auf Ihr Blatt und veranschaulichen Sie die Zuteilung der Rechenzeit an die Prozesse für die oben beschriebene Variante des MLFQ-Verfahrens im Diagramm. Kennzeichnen Sie zudem für jeden Prozess seine Ankunftszeit.



b) Übertragen Sie das nachfolgende Gantt-Diagramm auf Ihr Blatt. Veranschaulichen Sie die Zuteilung der Rechenzeiten an die Prozesse $P_1, \ldots, P_5$ aus Aufgabe a) des nachfolgenden Gantt-Diagramms.



c) Geben Sie für die Prozesse P_4 und P_5 jeweils die Verweilzeit (Antwortzeit) und die Wartezeit an.

d) Mit welcher Strategie ist Round Robin mit einem Quantum $Q = \infty$ gleichzusetzen?

Fortsetzung nächste Seite!

Aufgabe 3:

Buddy-Systeme

Ein Computer verfüge über einen 2 GByte großen Hauptspeicher, der nach dem Buddy-Verfahren verwaltet wird.

a) Wieviele Bits benötigt man mindestens, um diesen Speicher byteweise zu adressieren?

b) Nacheinander sollen nun die folgenden vier Programme in den Speicher geladen werden:

- P_1: 100 MByte
- P_2: 300 MByte
- P_3: 40 MByte
- P_4: 80 MByte

Zeichnen Sie den Buddy-Baum jedesmal, nachdem eines der Programme in den Speicher geladen wurde. Verwenden Sie dabei folgende Notation:

freies Segment

gesplittetes Segment

von Programm P_i belegtes Segment

Tragen Sie neben jedem freien Segment die Größe des freien Speicherbereichs an. Tragen Sie neben jedem belegten Segment die Größe des allokierten Speicherbereichs sowie die Speicheradressen des Segments an.

Hinweis: Es wird immer das am weitesten links stehende Segment gesplittet und der am weitesten links stehende Buddy belegt.

c) Die Programme aus Teilaufgabe b) benötigen insgesamt 520 MByte Speicherplatz. Damit müssten noch 1528 MByte Speicher nutzbar sein. Warum ist das im Beispiel nicht der Fall? Welcher Effekt kommt hier zum Tragen? Wieviel nutzbarer Speicherplatz steht für weitere Programme insgesamt noch zur Verfügung?

d) Gegeben ist eine weitere Anfrage:

- P_5: 600 MByte

Kann P_5 noch zusätzlich in den Speicher geladen werden? Falls ja, zeichnen Sie den Buddy-Baum nach der Belege-Operation. Falls nein, begründen Sie Ihre Antwort.

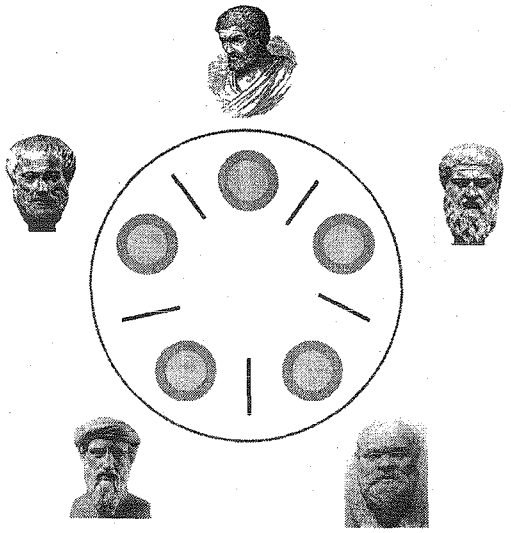
Fortsetzung nächste Seite!

Aufgabe 4:

Koordination von Threads

In dieser Aufgabe soll das bekannte Philosophenproblem (Dining Philosophers) mit Hilfe der Programmiersprache Java implementiert werden.

Beim Philosophenproblem sitzen fünf Philosophen an einem runden Tisch beim Essen (siehe Abbildung). Vor jedem Philosophen befindet sich ein Teller voller Reis und zwischen je zwei Tellern befindet sich ein Stäbchen. Ein Philosoph, der hungrig ist, benötigt sowohl das linke als auch das rechte Stäbchen, um essen zu können. Hat ein Philosoph zwei Stäbchen, so isst dieser, bis er satt ist, erst dann legt er die Stäbchen an die ursprünglichen Plätze zurück, so dass seine Nachbarn davon Gebrauch machen können. Ein Philosoph legt ein Stäbchen niemals zurück, bevor er nicht gegessen hat.



a) Beschreiben Sie, ab wann man sich beim Philosophenproblem im kritischen Bereich befindet und ab wann man diesen wieder verlässt.

b) Wenn man beim Philosophenproblem keine Einschränkung beim Zugriff auf die Stäbchen formuliert, kann es zu einem Deadlock kommen.

   i) Beschreiben Sie informell einen Ablauf für das Philosophenproblem, der zu einem Deadlock fährt.

   ii) Geben Sie zwei Möglichkeiten an, wie sich die Deadlock-Bedingung "Circular Wait" vermeiden lässt, so dass keine Deadlock-Situation im Fall des Fünf-Philosophenproblems auftritt.

Fortsetzung nächste Seite!

c) Im Folgenden sollen Sie das Philosophenproblem in Form der drei Java Klassen `Philosoph`, `Staebchen` und `Dinner` implementieren. Der Quelltext der Klasse `Dinner` ist bereits vorgegeben und sieht folgendermaßen aus:

```
1  public class Dinner {
2     public static void main(String[] args) {
3        Thread[] philosoph = new Thread[5];
4        Staebchen staebchen[] = new Staebchen[5];
5
6        for (int i = 0; i < 5; ++i) {
7           staebchen[i] = new Staebchen();
8        }
9        for (int i = 0; i < 5; ++i) {
10          philosoph[i] =
11             new Philosoph(i, staebchen[(i+4)%5], staebchen[i]);
12          philosoph[i].start();
13       }
14    }
15 }
```

i) Implementieren Sie zunächst die Klasse `Staebchen`. Diese soll die beiden Methoden `get()` (Stäbchen vom Tisch nehmen) und `put()` (Stäbchen an den ursprünglichen Platz zurücklegen) zur Verfügung stellen. Halten Sie sich dazu an den nachfolgenden Code-Rahmen. Referenzieren Sie jeden Teil Ihre Lösung entsprechend den Angaben im Code-Rahmen. Achten Sie darauf, dass der Zugriff auf ein Stäbchen im wechselseitigen Ausschluss erfolgt und keine Race Conditions auftreten können! Verwenden Sie für Ihre Implementierung das vorgegebene Attribut `wirdBenutzt`.

Kommentieren Sie Ihre Lösung ausführlich!

Code-Rahmen der Klasse `Staebchen`. Übertragen Sie den Code-Rahmen mit Ihren Ergänzungen auf Ihr Blatt.

```
1  public class Staebchen {
2
3     // Attribut
4     private boolean wirdBenutzt = false;
5
6     // ==================================================================
7     // Teil (1):
8     // Fügen Sie hier den Programmtext der get()-Methode ein.
9     // ==================================================================
10    // ...
11
12    // ==================================================================
13    // Teil (2):
14    // Fügen Sie hier den Programmtext der put()-Methode ein.
15    // ==================================================================
16    // ...
17 }
```

Fortsetzung nächste Seite!

ii) Implementieren Sie nun die Klasse `Philosoph`. Achten Sie darauf, dass Ihre Implementierung frei von Deadlocks ist. Verwenden Sie dazu den nachfolgenden Code-Rahmen (Fortsetzung auf der nächsten Seite!). Verwenden Sie ausschließlich die vorgegebenen Attribute. Achten Sie darauf, dass sich Ihre Implementierung zusammen mit der Klasse `Dinner` fehlerfrei übersetzen lässt.

Kommentieren Sie Ihre Lösung ausführlich!

Hinweis: Die Klasse `Random` dient dazu, Zufallszahlen zu erzeugen. Die Methode `nextInt(int positiveZahl)` liefert eine Zahl zwischen 0 (inklusive) und
`positiveZahl` (exklusive). Mit Hilfe dieser Methode wird die Zeit, die ein Philosoph isst bzw. denkt simuliert.

Code-Rahmen der Klasse `Philosoph`. Übertragen Sie den Code-Rahmen mit Ihren Ergänzungen auf Ihr Blatt.

```
1  import java.util.Random;
2
3  public class Philosoph extends Thread {
4     private static Random rand = new Random();
5
6     // Attribute
7     private Staebchen links;
8     private Staebchen rechts;
9     private int id;
10
11    // ==========================================================
12    // Teil (3):
13    // Fügen Sie hier den Programmtext des Konstruktors ein.
14    // ==========================================================
15    // ...
16
17
18    // run()-Methode
19    public void run() {
20       try {
21          while(true) {
22             // Philosoph denkt
23             System.out.println("Philosoph "+ id + " denkt");
24             Thread.sleep(rand.nextInt(500) + 500);
25             // ==========================================================
26             // Teil (4):
27             // Philosoph ist hungrig.
28             // Fügen Sie hier den Programmtext zum Erhalten der
29             // Stäbchen ein.
30             // ==========================================================
31             // ...
32
```

Fortsetzung nächste Seite!

```
33                  // Philosoph isst
34                  System.out.println("Philosoph "+ id + " isst");
35                  Thread.sleep(rand.nextInt(500) + 500);
36                  // ==========================================
37                  // Teil (5):
38                  // Philosoph ist satt.
39                  // Fügen Sie hier den Programmtext zum Zurücklegen der
40                  // Stäbchen ein.
41                  // ==========================================
42                  // ...
43
44              }
45          }
46          catch(InterruptedException ie) {
47          }
48      }
49  }
```

66114

Datenbank- und Betriebssysteme (vertieft)

Herbst 2013

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____ Kennwort: _____ Arbeitsplatz-Nr.: _____	**Herbst 2013**	**66114**

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

— Prüfungsaufgaben —

Fach:	**Informatik (vertieft studiert)**
Einzelprüfung:	**Datenbank- und Betriebssysteme**
Anzahl der gestellten Themen (Aufgaben):	**4 Aufgaben, von denen zwei gemäß untenstehender Auswahlregel zu bearbeiten sind!**
Anzahl der Druckseiten dieser Vorlage:	**18**

Zu den zwei Themenschwerpunkten A (Datenbanksysteme) und B (Betriebssysteme) ist jeweils entweder die Teilaufgabe 1 oder 2 zu wählen.
Auf der Vorderseite des Kopfbogens sind im Feld „gewähltes Thema: Nr." die Nummern der beiden gewählten Teilaufgaben anzugeben (z. B. A2, B1)!

Bitte wenden!

Themenschwerpunkt A
(Datenbanksysteme)

Teilaufgabe 1:

1. Grundwissen

1.1 Bewerten Sie die folgenden Aussagen.

Geben Sie für die folgenden Aussagen an, ob sie richtig oder falsch sind. Begründen Sie Ihre Antwort kurz.

a) Ein perfekter Datenbankpuffer müsste die zukünftige Seitenreferenzfolge kennen.

b) Ein Hash-Index eignet sich gut zur Beschleunigung von Bereichsanfragen.

c) Fremdschlüsselwerte müssen eindeutig sein.

d) Eine Relation kann mehrere Primärschlüsselattribute besitzen.

e) Die Begriffe Datenbankverwaltungssystem und Datenbank sind äquivalent.

1.2 Erläutern Sie kurz die folgenden Begriffe und Abkürzungen im Kontext von Datenbanksystemen.

a) Index

b) Normalisierung

c) Internes Schema

d) Stored Procedure

e) ORM

Fortsetzung nächste Seite!

2. Transaktionen

a) Erläutern Sie kurz die vier wesentlichen Eigenschaften einer Transaktion.

b) Können zwei parallel ablaufende Transaktionen, unter Garantie der Eigenschaften aus a), schreibend auf dasselbe Datenbankobjekt zugreifen? Begründen Sie Ihre Antwort.

c) Kann bei zwei parallel ablaufenden Transaktionen, unter Garantie der Eigenschaften aus a), eine Transaktion schreibend auf ein Datenbankobjekt zugreifen, das von der anderen Transaktion gelesen wurde? Begründen Sie ihre Antwort.

3. Relationale Algebra

Relationenalgebraische Ausdrücke werden genutzt, um die Bearbeitung von SQL-Anfragen zu optimieren. Die Literatur kennt dabei eine Menge von Transformationsregeln. Untersuchen Sie die folgenden drei Regeln auf Richtigkeit bzw. Nichtrichtigkeit und begründen Sie dies kurz, wobei R (A1, ..., An), S (B1, ..., Bm) und T (C1, ..., Ck) Relationen sind.

a) Der Verbund ist assoziativ, d. h. es gilt:
join(join (R, S),T) = join (R,(join (S, T))

b) Alle Mengenoperationen sind kommutativ.

c) Welche Operation der relationalen Algebra wird durch SQL zunächst nur unvollständig umgesetzt (Verletzung der Relationeneigenschaften bei Anwendung)? Welche Eigenschaft fehlt dieser vereinfachten Umsetzung?

d) Nennen Sie das SQL-Schlüsselwort, welches für die Operation aus c) ein Verhalten gemäß der relationalen Algebra erzwingt. Aus welchem Grund ist nicht Algebra konformes Verhalten in manchen Situationen besser?

Fortsetzung nächste Seite!

4 . Normalformen

a) Die Normalisierung von Relationenschemata dient der Vermeidung von Redundanzen und dadurch bedingter Anomalien. Geben Sie ein Beispiel für eine nicht-normalisierte Relation an und erläutern Sie zwei mögliche Anomalien an diesem Beispiel.

b) Normalisierung beruht auf dem Erkennen und Eliminieren von funktionalen Abhängigkeiten. Erläutern Sie in diesem Zusammenhang kurz die folgenden Begriffe

1. Funktionale Abhängigkeit
2. Voll-funktionale Abhängigkeit
3. Transitive funktionale Abhängigkeit
4. Superschlüssel
5. Determinante

c) In welcher Normalform ist das folgende Beispiel? Zeigen Sie, dass alle Bedingungen für diese Normalform erfüllt sind. Welche Bedingung der nächsthöheren Normalform ist verletzt?

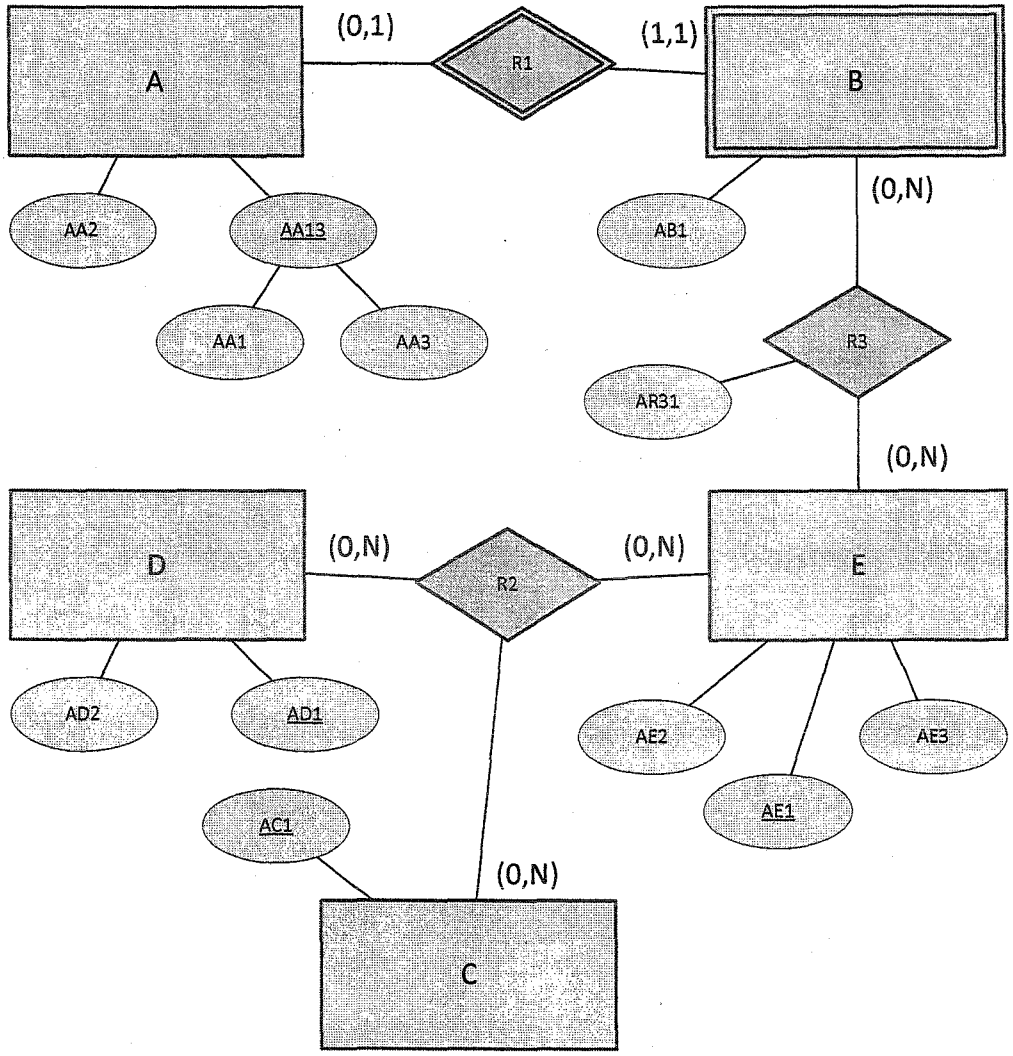
PersNr	Name	Land	Landesvorwahl	Vorwahl	Rufnummer
1	Maier	Frankreich	33	556	56881
2	Maier	Deutschland	49	821	4563
3	Pascal	Frankreich	33	556	56881
4	Konrad	Deutschland	49	841	923476
5	Schmidt	USA	1	0212	928357
6	Berner	Frankreich	33	556	183258

d) Erklären Sie die Begriffe Verlustlosigkeit und Abhängigkeitsbewahrung bei der Zerlegung von Relationen.

Fortsetzung nächste Seite!

5. E/R-Modellierung und Relationenmodell

Entwerfen Sie zum untenstehenden E/R-Diagramm ein Relationenschema in dritter Normalform (3. NF) mit möglichst wenigen Relationen!
Verwenden Sie dabei folgende Notation:
Primärschlüssel werden durch Unterstreichen gekennzeichnet und Fremdschlüssel durch die Nennung der Relation auf die sie verweisen in eckigen Klammern hinter dem Fremdschlüsselattribut. Attribute zusammengesetzter Fremdschlüssel werden durch runde Klammern als zusammengehörig markiert.
Beispiel: Relation1 (Primärschlüssel, Attribut1, Attribut2, Fremdschlüssel[Relation2])



Fortsetzung nächste Seite!

6. Relationen und SQL

Gegeben ist folgendes Relationenschema zur Verwaltung von Studenten und Kursen:
`STUDENT (Matrikelnummer, Name, Geburtsdatum)`
`KURS (Kursnummer, Kursname, Kursleiter)`
Die Primärschlüssel der Relationen sind wie üblich durch Unterstreichung gekennzeichnet.
Verwenden Sie für die zu formulierenden SQL-Anfragen nur standardkonformes SQL.

a) Erweitern Sie das obige Schema so, dass eine m:n-Beziehung zwischen STUDENT und KURS dargestellt werden kann: Ein Kurs kann aus mehreren Studenten bestehen, und ein Student kann an mehreren Kursen teilnehmen. Es ist darauf zu achten, dass alle Primärschlüssel nach wie vor genau ein Tupel identifizieren. Die Ausgangsrelationen STUDENT und KURS dürfen nicht verändert werden.

b) Formulieren Sie ein SQL-Statement, welches die Namen der Kursleiter alphabetisch sortiert ausgibt (keine Mehrfachnennung).

c) Formulieren Sie ausgehend von Ihrem in Teilaufgabe a) erweiterten Schema eine Anfrage in SQL, die die Namen und Leiter aller Kurse liefert, in denen Studenten mit Namen „Maier" teilnehmen.

d) Formulieren sie eine Anfrage in SQL, die die Namen und Geburtsdaten aller Kommilitonen liefert, die der Student mit der Matrikelnummer „12345678" in mindestens einem seiner Kurse außer sich selbst noch trifft? (keine Mehrfachnennungen)

e) Formulieren Sie eine SQL-Abfrage, die die Namen der 10 ältesten Studenten ausgibt, absteigend sortiert nach Alter. Gehen Sie davon aus, dass es keine zwei Studenten genau gleichen Alters gibt.

Teilaufgabe 2:

1. ER-Modellierung

Entwerfen Sie ein ER-Modell für die folgende Miniwelt, die sich mit Konzerten, Veranstaltern, Bands usw. befasst. Geben Sie Kardinalitäten in (*min, max*)-Notation an. Sie brauchen Schlüsselattribute nicht zu kennzeichnen.

- Es gibt *Bands*, die jeweils einen Namen tragen. Zu jeder Band soll ein Gründungs- und ein Auflösungsdatum vermerkt werden.
- *Musiker* haben einen Namen. Musiker spielen *Instrumente* in Bands. Dabei können sie in verschiedenen Zeitintervallen (von-bis) verschiedene Instrumente in wechselnden Bands spielen, auch mehrere zur gleichen Zeit.
- *Instrumente* haben einen Namen.
- Bands spielen *Konzerte*. Jedes Konzert wird von genau einer Band gespielt, zusätzlich kann es noch bis zu zwei Vorgruppen geben. Konzerte finden zu einem bestimmten Datum an einem bestimmten Ort statt. Weitere Musiker, die nicht zur Band gehören, können bei Konzerten als Gast auftreten.
- *Veranstalter* haben einen Namen und veranstalten Konzerte. Jedes Konzert hat genau einen Veranstalter.
- *Besucher* gehen zu Konzerten. Ein Konzert hat mindestens 100 und höchstens 100.000 Besucher. Zu einem Konzertbesuch soll notiert werden, wie hoch der Eintrittspreis war. Dieser kann für jeden Besucher verschieden sein.

Ihr Modell sollte folgendes ausdrücken können. Sie brauchen diese Fakten nicht zu modellieren, sie dienen nur zur Kontrolle, ob Ihr Modell die o. g. Anforderungen erfüllt.

- Der Musiker Eddie van Halen spielte vom 1.1.1977 bis zum 6.6.1994 Gitarre und Keyboard in der Band "Van Halen"; vom 3.4.1986 bis zum 12.8.1988 spielte er Bass in der "Sammy Hagar Band".
- Am 3.6.2007 spielte die Band "Linkin Park" ein Konzert in Hamburg, das 7.000 Besucher hatte. Vorgruppe war "P.O.D.", als Gast trat Fred Durst auf. Das Konzert wurde veranstaltet von Undercover Entertainment.
- Der Besucher Karl Napf war bei diesem Konzert und hat dafür 53 Euro Eintritt gezahlt.
- Die Besucherin Liese Laut war auch beim o. g. Konzert, hat aber nur 48 Euro Eintritt gezahlt.

Fortsetzung nächste Seite!

2. Relationale Algebra

Entsprechend dem Modell aus der vorigen Aufgabe sei eine Miniwelt über Konzerte, Veranstalter usw. entsprechend den folgenden Relationen modelliert:

- Besucher(persnr, name)
- Konzerte(konzertnr, bandnr, datum, ort, veranstalternr)
- Veranstalter(veranstalternr, name)
- KonzertBesuche(persnr, konzertnr, preis)
- Vorgruppen(bandnr, konzertnr)
- Bands(bandnr, name, gründungsdatum, auflösungsdatum)
- Musiker(musikernr, name)
- Instrumente(instrumentnr, name)
- Spielt(musikernr, bandnr, instrumentnr, von, bis)
- Gast(musikernr, konzertnr)

Geben Sie je einen Ausdruck in relationaler Algebra für folgende Anfragen an:

2.1 In welchen Bands (Nummern sind gesucht) hat Musiker Nr. 12 gespielt?

2.2 Wer spielte schon (Musikernummern sind gesucht) in der Band "The Hooters"?

2.3 Welche Musiker (Nummern sind gesucht) waren schon einmal Gast bei einem Konzert der "Dave Matthews Band"?

Fortsetzung nächste Seite!

3. SQL

Für diese Aufgabe betrachten wir eine relationale Datenbank in Anlehnung an das Schema aus Aufgabe 2.

3.1 Geben Sie je ein `CREATE-TABLE`-Statement zum Erzeugen der Relationen *Bands* und *Spielt* an. Achten Sie auf den Primärschlüssel und auf möglichst restriktive Fremdschlüssel und Integritätsbedingungen:
Ein Musiker soll ein bestimmtes Instrument in einer gegebenen Band nur einmal spielen können (nicht mehrere Zeitintervalle).
Wenn ein Instrument verschwindet, soll das entsprechende Attribut in *Spielt* auf `NULL` gesetzt werden. Ein Musiker soll nicht gelöscht werden können, solange er noch in einer Band spielt. Wenn Bands gelöscht werden, sollen die entsprechenden Tupel in *Spielt* auch verschwinden.

3.2 Geben Sie SQL-Statements für folgende Anfragen an:

3.2.1 In welchen Bands (Nummern sind gesucht) hat Musiker Nr. 28 gespielt?

3.3 In welchen Bands (Nummer gesucht) hat der Musiker namens "Phil Collins" gespielt?

3.4 Erzeugen Sie eine Sicht, die eine Übersicht über die Veranstalter und die Anzahlen ihrer Konzerte enthält.
Die Sicht soll also Tupel der Form (*Veranstalternr, Veranstaltername, Anzahl Konzerte*) enthalten.

3.5 Geben Sie eine Anfrage an, die für jede Band den Namen der Band und die Gesamtzahl der jemals dort spielenden Musiker ausgibt. (Vorsicht, Musiker mit mehreren Instrumenten nicht mehrfach zählen.)

3.6 Was leistet folgende SQL-Anfrage:

```
select v.name
from konzerte k, konzertbesuche b, veranstalter v
where k.konzertnr = b.konzertnr
and v.veranstalternr = k.veranstalternr
group by v.veranstalternr
having sum(b.preis) >= all (
    select sum(bb.preis)
    from konzerte kk, konzertbesuche bb
    where kk.konzertnr = bb.konzertnr
    group by kk.veranstalternr
)
```

3.7 Ist die Unteranfrage in 3.6 korreliert oder nicht? (Begründung.)

Fortsetzung nächste Seite!

4. Normalisierung

Gegeben seien die funktionalen Abhängigkeiten

$$
\begin{aligned}
F = \{ \quad & ABC \rightarrow E, && \text{(i)}\\
& B \rightarrow AC, && \text{(ii)}\\
& E \rightarrow BCD, && \text{(iii)}\\
& F \rightarrow ABE, && \text{(iv)}\\
& D \rightarrow EF \quad \} && \text{(v)}
\end{aligned}
$$

der Relation $R(A, B, C, D, E, F)$.
Berechnen Sie eine kanonische Überdeckung F_C von F. Geben Sie alle Zwischenschritte und die angewendeten Transformationen an.

5. Normalisierung

Gegeben sei folgendes Schema, das die Bücher eines Verlages repräsentiert:

Bücher (ISBN, AutorNr, Autorname, Autoradresse, LektorNr, Lektorname, Titel, Seiten, Preis, Auflage, Jahr)

mit den funktionalen Abhängigkeiten

$$
\begin{aligned}
F \quad = \{ \quad & \text{AutorNr} \rightarrow \text{Autorname, Autoradresse,} && \text{(i)}\\
& \text{LektorNr} \rightarrow \text{Lektorname,} && \text{(ii)}\\
& \text{ISBN} \rightarrow \text{Titel, AutorNr, LektorNr,} && \text{(iii)}\\
& \text{ISBN, Auflage} \rightarrow \text{Seiten, Preis, Jahr} \quad \} && \text{(iv)}
\end{aligned}
$$

Hinweis: Sie können folgende Abkürzungen verwenden:

I	ISBN	T	Titel
A#	AutorNr	S	Seiten
An	Autorname	P	Preis
L#	Lektornr	J	Jahr
Ln	Lektorname	Au	Auflage
		Ad	Autoradresse

5.1 In welcher Normalform ist dieses Schema? (Begründung.)

5.2 Nennen Sie je ein Beispiel für eine Einfüge-, Update- und Löschanomalie in diesem Schema.

Themenschwerpunkt B
(Betriebssysteme)

Teilaufgabe 1:

1. Schedulingstrategien

a) Definieren Sie den Begriff *Auslastung*

Definieren Sie den Begriff *Antwortzeit*

Inwieweit widersprechen sich die beiden Schedulingziele *maximale Auslastung* und *minimale Antwortzeit*?

b) In einem System wird das Scheduling gemäß *Round-Robin* mit einer Zeitscheibengröße von n durchgeführt. Ein Prozesswechsel dauert k Zeiteinheiten.

Was passiert wenn n unendlich gross wird?

Was ist das kleinstmögliche n? Wie verhält sich das System in diesem Falle und wann ist dies sinnvoll?

c) Gegeben sind folgende Prozesse mit ihren Bereitzeitpunkten und ihrem Rechenzeitbedarf. Führen Sie für eine Einprozessormaschine das nicht-preemptive Scheduling gemäß *Shortest Processing Time First* (SPF) und das preemptive Scheduling gemäß *Round-Robin* (RR) mit Zeitscheibenlänge 1 durch. Der Overhead für den Prozesswechsel ist hier vernachlässigbar.

Prozess	***Ankunftszeit***	***Benötigte Rechenzeit***
A	0	3
B	1	6
C	3	3
D	5	5
E	9	1
F	15	2

Ein Task kann zum Zeitpunkt seiner Ankunft bereits ausgeführt werden (Beispiel siehe Prozess A: Ankunftszeit = 0, Prozessor-Zuteilung = 0).

Fortsetzung nächste Seite!

Übertragen Sie die folgenden drei Tabellen auf Ihr Arbeitsblatt:

Tragen Sie in der Zeile Ausführung jenen Prozess ein, der für die entsprechende Zeitscheibe dem Prozessor zugeteilt wird. Die restlichen Zeilen der Tabelle stellen die Ready-Queue dar. Tragen Sie hier die Prozesse ein, die in der Warteschlange auf Zuteilung warten (beginnend in der obersten Zeile mit dem Prozess, der als nächster den Prozessor zugeteilt bekommt, usw.).

Shortest Processing Time First:

			0					5					10					15		
Ausf.																				
Ready-Queue																				

Round Robin:

			0					5					10					15		
Ausf.																				
Ready-Queue																				

Geben Sie für jeden Prozess die Verweildauer im System an:

Prozess	A	B	C	D	E	F
SPF						
RR						

Fortsetzung nächste Seite!

2. Synchronisation

a) Nennen Sie die vier hinreichenden und notwendigen Bedingungen für Verklemmungen.

b) Betrachten Sie die folgenden zwei Programme A und B, die konkurrierend auf den Puffer buf und auf die Logdatei lf zugreifen. Auf die Logdatei lf greifen auch noch andere Programme zu. Die beiden Semaphore SemBuf und SemLog sind jeweils mit 1 initialisiert!

`Program A:`	(1)	`Program B:`
`while (true) {`	(2)	`while (true) {`
`  ...`	(3)	`  P(SemBuf);`
`  P(SemLog);`	(4)	`  infoB=buf.removeNext();`
`  write(lf, "A: Got "+infoA);`	(5)	`  P(SemLog);`
`  P(SemBuf);`	(6)	`  write(lf, buf.length());`
`  buf.add(infoA);`	(7)	`  V(SemBuf);`
`  write(lf, buf.length());`	(8)	`  write(lf,"B: Got "+infoB);`
`  V(SemLog);`	(9)	`  V(SemLog);`
`  V(SemBuf);`	(10)	`  ...`
`}`		`}`

Können diese Programme verklemmungsfrei laufen? Begründen Sie die Verklemmungsfreiheit bzw. beschreiben und begründen Sie eine Änderung der Programme, so dass keine Verklemmungen auftreten können.

Fortsetzung nächste Seite!

3. Segmentierung

a) Erläutern Sie warum in heutigen Rechnern virtuelle Adressierung anstelle von physikalischer Adressierung verwendet wird. Gehen Sie dabei auf die vier Probleme physikalischer Adressierung ein.

b) Gegeben sei ein Rechner mit 5 Bit physikalischem Adressraum und 5 Bit segmentbasiertem, virtuellen Adressraum und 5 Bit breiten Worten. Der Inhalt des physikalischen Speichers ist unten angegeben.

- Es gibt 4 Segmente, d. h. die ersten (=höherwertigsten) 2 Bit einer Adresse bilden die Segmentnummer
- Die Basisadresse der Segmenttabelle ist 00000.
- Ein Eintrag in der Segmenttabelle benötigt jeweils zwei aufeinanderfolgende Speicherworte:
  - das erste Wort eines Eintrags gibt die Segmentbasisadresse an
  - vom zweiten Wort enthalten die ersten (=höherwertigsten) 3 Bit die Länge des Segments in Speicherworten, das 4. Bit sei hier nicht von Bedeutung und das 5. Bit sei das *present*- bzw. *mapped*-Bit.

Adresse	Inhalt	Adresse	Inhalt	Adresse	Inhalt	Adresse	Inhalt
00000	10101	01000	01110	10000	11111	11000	00110
00001	10011	01001	00011	10001	11010	11001	01010
00010	11011	01010	10010	10010	01011	11010	00010
00011	00100	01011	00000	10011	01101	11011	10001
00100	01000	01100	11000	10100	00101	11100	10100
00101	11101	01101	10000	10101	11100	11101	01001
00110	01111	01110	11110	10110	00111	11110	10110
00111	11001	01111	01100	10111	10101	11111	00001

1. Wieviele Segmente sind *present*?

2. Welche Daten stehen an den Adressen 01000, 00111 und 10001? Geben Sie jeweils auch die Adressumrechnungen mit an.

3. Geben Sie die virtuelle Adresse für die reale Adresse 10010 an.

Fortsetzung nächste Seite!

4. Virtueller Speicher und Seitenverdrängung

a) Gegeben sei ein Rechner mit 32 Bit virtuellem Adressraum und 1 GByte byteweise adressiertem, realem Speicher. Eine Seite sei 8 kBytes groß.

Hinweis:

- 1kByte = 1024 Byte = 2^{10} Byte
- 1GByte = 1024 MByte = 2^{30} Byte

Wie viel Bit einer Adresse werden für den Offset benötigt?

Wie viele virtuelle Seiten gibt es?

Wie viele Seitenrahmen gibt es?

b) Erklären Sie, nach wie vielen Schritten der Second Chance Algorithmus bei einer Speichergröße von N Seiten spätestens eine zu verdrängende Seite gefunden hat.

Teilaufgabe 2:

Aufgabe 1

a) Nennen Sie mindestens fünf wichtige Kategorien von Informationen, die in einem Prozesskontrollblock zu speichern sind.

b) Welche wesentlichen Eigenschaften unterscheiden Prozesse und Threads?

c) Schildern Sie knapp, welche Schritte bei der Initialisierung eines Prozesses üblicherweise durchlaufen werden.

d) Wofür werden die beiden Modi „User-Modus" und „Supervisor-Modus" benötigt? Nennen Sie hierzu jeweils mindestens ein Beispiel für einen Wechsel von User-Modus zu Supervisor-Modus und von Supervisor-Modus zu User-Modus.

Aufgabe 2

Gegeben sei ein Prozess-System mit vier Prozessen: $P_1, \ldots, P_4$ und die folgenden drei Ressourcen: R_1 (vier Einheiten) und R_2 (zwei Einheiten) und R_3 (drei Einheiten). Zu einem gegebenen Zeitpunkt besitzt

P_1 drei Einheiten von R_1
P_2 eine Einheit von R_1
P_3 eine Einheit von R_2
P_4 zwei Einheiten von R_3

a) Zusätzlich fordert nun
P_1 je eine Einheit von R_1 und R_3 an,
P_2 fordert zwei Einheiten von R_2,
P_3 und P_4 je eine Einheit von R_3.

Zeichnen Sie den zugehörigen Betriebsmittelgraphen (Ressource Allocation Graph) nach Holt.

b) Überprüfen Sie mit dem Banker's Algorithmus, ob bei Zuteilung der Ressourcen, wie in Teil a) von den Prozessen angefordert, ein sicherer Zustand vorliegen würde. Die Summe der belegten und der in Teil a) angeforderten Ressourcen stellen jeweils bereits die Maximalforderungen der Prozesse dar.

c) Konstruieren Sie für ein System von drei Prozessen und drei Ressourcentypen einen Systemzustand von Ressourcenbelegungen und –anforderungen, der einen geschlossenen Zyklus von Anforderungen und Belegungen enthält, der aber trotzdem kein Deadlock ist. Zeichnen Sie den zugehörigen Betriebsmittelgraphen.

Fortsetzung nächste Seite!

Aufgabe 3

Ein neu eröffneter Supermarkt hat zwei Selbstbedienungskassen eingerichtet, bei denen sich die Kunden per Chipkarte identifizieren, dann die Artikel unter den Kassenscanner legen und den Rechnungsbetrag direkt von ihrem Konto abbuchen lassen. Bei diesen beiden Kassen handelt es sich um eine „normale Automatik-Kasse" A und eine „Express-Kasse" E.

a) Anfangs ist die Nutzung der beiden Kassen folgendermaßen geregelt:

1. Jeder Kunde K (i), i > 0, kann sich an der Express-Kasse E anstellen, wenn die Anzahl der Artikel n(i) in seinem Warenkorb höchstens gleich drei ist.

2. Andernfalls muss er sich an der normalen Kasse A anstellen.

Implementieren Sie das Prozessverhalten des Kunden K(i) und der beiden Kassen A und E in Pseudo-Code. Die Synchronisation der Kunden K(i), der Kasse A und der Kasse E soll ausschließlich über allgemeine Semaphore, also Counting Semaphore mit wait() und signal(), geschehen.

b) Nach einem mehrmonatigen Betrieb der Selbstbedienungskassen werden anstatt des bisherigen Verfahrens die folgenden beiden Regeln für die Nutzung der Kassen A und E in Kraft gesetzt:

1. Warten bereits drei Kunden an der Express-Kasse E, so muss sich jeder weitere ankommende Kunde K(i) bei der normalen Kasse A anstellen, unabhängig von der Anzahl seiner Artikel im Warenkorb.

2. Kunden K(i) dürfen sich nun auch mit einer beliebigen Anzahl von Artikeln bei der Kasse E anstellen. Allerdings muss ein K(i) mit mehr als drei Artikeln direkt vor der Kasse E prüfen, ob mindestens ein weiterer Kunde hinter ihm in der Warteschlange steht. Ist dies der Fall, so muss K(i) zur Kasse A wechseln. Sonst wird er bei E bedient. Hat K(i) höchstens drei Artikel, so wird er ohne Prüfen der Länge der Warteschlange bei E bedient.

Implementieren Sie hier ebenfalls das Verhalten der Kunden K(i) in Pseudo-Code. Die Prozesse für die Kassen A und E sollen wie bei a) realisiert werden. Verwenden Sie wiederum counting Semaphore und zusätzlich zur Kontrolle der Warteschlangenlänge bei der Kasse E eine globale Zählvariable *anzahl_bei_E*.

Fortsetzung nächste Seite!

Aufgabe 4

In einem Arbeitsspeicher existieren die folgenden freien Blöcke in der angegebenen Reihenfolge mit den Größen

50KB, 48KB, 120KB, 66KB, 62KB

Geben Sie schrittweise die verbleibenden freien Speicherblöcke für die folgenden Anforderungen an (ebenfalls in dieser Reihenfolge) nach Blöcken der Größe:

48KB, 58KB, 64KB, 44KB

Verwenden Sie dabei die Vergabeverfahren:

a) First-Fit
b) Best-Fit
c) Worst-Fit
d) Rotating-First-Fit. Bei dieser Variante von First-Fit werden die freien Speicherblöcke in einer zyklisch verketteten Liste verwaltet. Nach jeder erfolgreichen Vergabe eines angeforderten Blocks startet die Suche für die nächste Anforderung jeweils beim Nachfolger des zuletzt betrachteten Speicherblocks.

66114

Datenbank- und Betriebssysteme (vertieft)

Frühjahr 2014

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____ Kennwort: _____ Arbeitsplatz-Nr.: _____	**Frühjahr 2014**	**66114**

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen
— Prüfungsaufgaben —

Fach: **Informatik (vertieft studiert)**

Einzelprüfung: **Datenbank- und Betriebssysteme**

Anzahl der gestellten Themen (Aufgaben): **4 Aufgaben, von denen zwei gemäß untenstehender Auswahlregel zu bearbeiten sind!**

Anzahl der Druckseiten dieser Vorlage: **20**

Zu den zwei Themenschwerpunkten A (Datenbanksysteme) und B (Betriebssysteme) ist jeweils entweder die Teilaufgabe 1 oder 2 zu wählen!
Auf der Vorderseite des Kopfbogens sind im Feld „gewähltes Thema: Nr." die Nummern der beiden gewählten Teilaufgaben anzugeben (z. B. A2, B1)!

Bitte wenden!

Themenschwerpunkt A
(Datenbanksysteme)

Teilaufgabe 1:

1. Modellierung

Der VDS (Verein deutscher Schwimmbäder) möchte Informationen über seine bestehenden Einrichtungen in Verbindung mit dessen Besucherzahlen festhalten und digitalisieren. Dazu soll eine Datenbank in zwei Schritten erstellt werden. Der erste soll folgende Informationen enthalten:

- Personen werden spezifiziert durch ihren Vor- sowie Nachnamen und einer Ausweisnummer.
- Ein Schwimmbad besitzt einen Namen und eine Stadt, in der es sich befindet.
- Personen besuchen Schwimmbäder.
- Für Schwimmbecken wird ein Name, die Länge, die Breite sowie die Tiefe gespeichert. Ein Schwimmbecken befindet sich in genau einem Schwimmbad und kann nur durch dieses eindeutig identifiziert werden.
- Personen können in einem Schwimmbecken schwimmen, wobei dazu ein Schwimmstil festgehalten werden soll.

Im zweiten Schritt der Datenbank wird spezieller auf die Personen im Schwimmbad eingegangen:

- Im Bezug auf die Domäne unterteilen sich Personen in Personal des Schwimmbads und Besucher. Angestellte des Schwimmbads können dieses aber auch außerhalb Ihrer Arbeitszeiten besuchen. Für das Personal wird eine Personalnummer, für die Besucher eine Besuchernummer gespeichert.
- Es werden Schwimmkurse angeboten. Zu diesen wird eine Kursnummer sowie dessen Gebiet festgehalten.
- Ein Angestellter kann eine beliebige Anzahl an Kursen leiten, wobei jeder Kurs eine beliebige Anzahl an Besuchern als Teilnehmer fassen kann.

Fortsetzung nächste Seite!

a) Entwerfen Sie für das beschriebene Szenario des **ersten** Schritts der Datenbank ein ER-Modell! Bestimmen Sie hierzu:

- die Entity-Typen, die Relationship-Typen und jeweils deren Attribute,
- die Primärschlüssel der Entity-Typen, welche Sie anschließend in das ER-Diagramm eintragen, und
- die Kardinalitäten der Relationship-Typen.

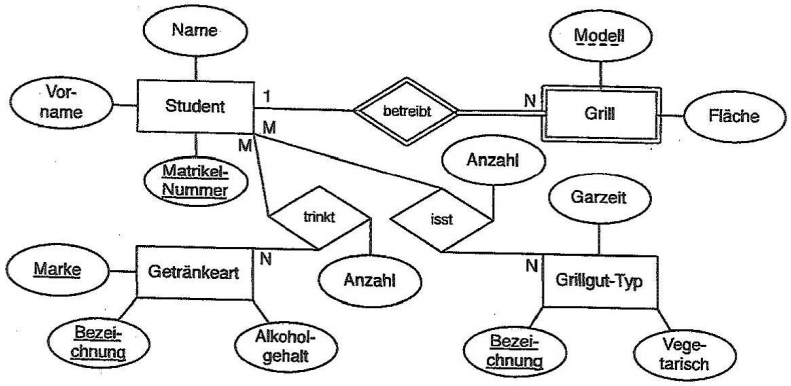
b) Übernehmen Sie den Entity-Typ für eine Person aus Aufgabenteil a) und entwerfen Sie ausgehend davon ein weiteres ER-Modell für den **zweiten** Teil der Datenbank! Bestimmten Sie dazu ebenfalls dieselben Kriterien wie in Aufgabenteil a)!

c) Überführen Sie das ER-Modell aus Aufgabe a) in ein verfeinertes relationales Modell! Geben Sie hierfür die verallgemeinerten Relationenschemata an! Achten Sie dabei insbesondere darauf, dass die Relationenschemata keine redundanten Attribute enthalten!

Fortsetzung nächste Seite!

2. SQL-Anfragen

Gegeben sei folgendes ER-Modell, welches eine Datenbank für Statistiken über Grillabende beschreibt. Darin sind Studenten, die Grills betreiben, enthalten. Zu jedem Studenten soll die Anzahl an Speisen und Getränken erfasst werden, welche sie zu sich genommen haben:



Gehen Sie von folgendem dazugehörigen relationalem Schema aus:

Student{[Matrikel_Nummer, Name, Vorname]}
Grill{[Modell, Matrikel_Nummer, Flaeche]}
Getraenkeart{[Marke, Bezeichnung, Alkoholgehalt]}
Grillgut-Typ{[Bezeichnung, Vegetarisch, Garzeit]}
Trinkt{[Getraenkeart_Marke, Getraenkeart-Bezeichnung, Matrikel-Nummer,Anzahl]}
Isst{[Grillgut-Typ-Bezeichnung, Matrikel-Nummer, Anzahl]}

Formulieren Sie folgende Anfragen in SQL. Achten Sie, falls nötig, auf duplikatfreie Ausgaben:

a) Geben Sie die Garzeit des Grillgut-Typs mit der Bezeichnung „Schweinenacken" aus!

b) Geben Sie die Namen und Vornamen aller Grillmeister aus! *Hinweis:* Als Grillmeister gelten Studierende, die mindestens einen Grill betreiben.

c) Geben Sie alle Attribute von Studierenden aus, die mindestens ein Getränk mit mehr als 5 % Alkohol getrunken haben!

d) Geben Sie alle Attribute von Studierenden aus, die mindestens zwei alkoholhaltige Getränke konsumiert haben!

Fortsetzung nächste Seite!

3. Normalformen

Gegeben sei folgendes relationales Schema R, dessen Attribute nur atomare Attributwerte besitzen.

$$R: \{[A, B, C, D, E, F, G, H]\}$$

Es gelte die folgende Menge FD von funkionalen Abhängigkeiten:
$FD = \{$

$$A \rightarrow B,$$
$$CD \rightarrow EB,$$
$$C \rightarrow DA,$$
$$CF \rightarrow FE,$$
$$F \rightarrow GHCA$$

$\}$

a) Nennen Sie die drei Änderungsanomalien, die auftreten können, wenn ein Datenbank-Schema nicht der Boyce-Codd-Normalform entspricht, und beschreiben Sie eine näher!

b) Definieren Sie die Begriffe Superschlüssel, Schlüsselkandidat und Primarschlüssel. Wie unterscheiden sich diese voneinander?

c) Finden Sie den/die Schlüsselkandidaten!

d) Prüfen Sie, in welcher Normalform R vorliegt!

e) Berechnen Sie die kanonische Überdeckung FD_c von FD!

Hinweis: Es genügt, wenn Sie für jeden der vier Einzelschritte die Menge der funktionalen Abhängigkeiten als Zwischenergebnis angeben.

f) Bestimmen Sie eine Zerlegung der Relation R in dritter Normalform. Verwenden Sie dafür den Synthesealgorithmus!

Hinweis: Sie können dazu die kanonische Überdeckung FD_c verwenden, die Sie in der vorherigen Teilaufgabe berechnet haben.

Fortsetzung nächste Seite!

Teilaufgabe 2:

Aufgabe 1: E/R-Modellierung, SQL-DDL

Für die Verwaltung von Museen soll folgendes System entworfen werden:

- Die zentrale Komponente ist das Museum mit eindeutigem Namen.
- Ein Museum besteht aus mindestens einem Themenbereich, welcher ebenfalls einen eindeutigen Namen besitzt.
- Ein Themenbereich erstreckt sich über mindestens einen Raum, der einen Namen hat und über eine eindeutige Raumnummer identifiziert werden kann. Ein Raum gehört zu genau einem Themenbereich.
- In einem Raum befindet sich mindestens ein Kunstwerk. Ein Kunstwerk gehört zum Eigentum genau eines Museums, welches aber (im Fall einer Leihgabe) nicht zwingend das Museum sein muss, in dem es ausgestellt ist. Ein Museum kann Eigentümer vieler Kunstwerke sein.
- Ein Kunstwerk wurde von genau einem Künstler erstellt und gehört einer speziellen Kunstepoche an. Jedes Kunstwerk hat einen eindeutigen Titel sowie einen Untertitel und ein Entstehungsjahr.
- Eine Kunstepoche wird durch einen eindeutigen Namen und einen Zeitraum (Start- und Endjahr) beschrieben.
- Ein Künstler wird durch seinen Namen und Vornamen identifiziert und hat außerdem ein Geburtsdatum und gegebenenfalls ein Sterbedatum. Von Künstlern kann man sagen, dass sie in ihrem Leben viele Kunstwerke erschaffen und in mindestens einer Kunstepoche gewirkt haben.
- Manche Künstler sind Mitglied in einer Künstlervereinigung mit eindeutigem Namen und einem Gründungsdatum.

1. Erstellen Sie ein Entity-Relationship-Diagramm für obige Datenbank!
2. Setzen Sie das gegebene E/R-Diagramm in ein entsprechendes relationales Datenbankschema um. Geben Sie die resultierenden Relationenschemata in folgender Schreibweise an:
   `Relation (Attribut1, Attribut2, ..., AttributN)`
   Identifizieren Sie für jede Relation einen Primärschlüssel und unterstreichen Sie diesen! Achten Sie auf eine möglichst kompakte Modellierung der Relationships! Geben Sie außerdem erforderliche Fremdschlüsselbeziehungen an!
3. Geben Sie die Anweisungen in SQL-DDL an, die notwendig sind, um die Relationen für ein **Kunstwerk** und eine **Kunstepoche** aus Teilaufgabe (2) in einer relationalen Datenbank zu erzeugen! Kennzeichnen Sie dabei die Primär- und Fremdschlüssel der Relationen. Geben Sie sinnvolle Integritätsbedingungen für Zahlen und Daten an und achten Sie auf die sinnvolle Behandlung möglicher leerer Ausprägungen!

Fortsetzung nächste Seite!

Aufgabe 2: Relationale Algebra und SQL

Gegeben sei das folgende relationale Schema mitsamt Beispieldaten für eine Datenbank für Bierbrauereien und ihre Produkte. Die Primärschlüssel-Attribute sind jeweils unterstrichen, Fremdschlüssel sind überstrichen.

"Brauerei":

ID	Name	Sitz	Gründungsjahr
B1	Augustiner	S4	1328
B2	Beck's	S3	1873
B3	Paulaner	S4	1634
B4	Krombacher	S5	1803
B5	Früh	S1	1904
B6	Gaffel	S1	1908
B7	Hoepfner	S2	1798
B8	Spaten	S4	1397

"Sitz":

ID	Stadt	Bundesland
S1	Köln	Nordrhein-Westfalen
S2	Karlsruhe	Baden-Württemberg
S3	Bremen	Bremen
S4	München	Bayern
S5	Krombach	Nordrhein-Westfalen

"Biersorte":

ID	Bezeichnung	Hefetyp
T1	Pils	untergärig
T2	Lager	untergärig
T3	Weizenbier	obergärig
T4	Helles	untergärig
T5	Starkbier	ober- und untergärig
T6	Export	untergärig
T7	Kölsch	obergärig
T8	Altbier	obergärig

"Bier":

ID	Brauerei	Biersorte	Alkoholgehalt
A01	B1	T4	5,2
A02	B1	T6	5,6
A03	B2	T1	4,9
A04	B3	T3	5,5
A05	B3	T4	4,9
A06	B4	T1	4,8
A07	B5	T7	4,8
A08	B6	T7	4,8
A09	B7	T1	4,7
A10	B8	T4	5,2

1. Formulieren Sie die folgenden Anfragen auf das gegebene Schema in relationaler Algebra:

   a) Finden Sie die Namen aller Brauereien, die vor dem Jahr 1800 gegründet wurden.

   b) Finden Sie die Bezeichnungen aller Biersorten mit einem Alkoholgehalt von 4,8% oder 4,9%.

   c) Finden Sie die Städte und zugehörigen Bundesländer, in denen Brauereien ihren Sitz haben, die im 19. Jahrhundert gegründet wurden.

   d) Finden Sie die Bezeichnungen aller Biersorten, die von der Brauerei 'Paulaner' hergestellt werden.

2. Geben Sie das Ergebnis (bezüglich der Beispieldaten) der folgenden Ausdrücke der relationalen Algebra als Tabellen an:

   a) $\pi_{Name,Bundesland}(Brauerei \bowtie_{Brauerei.Sitz=Sitz.ID} Sitz)$

   b) $\pi_{Gründungsjahr,\ Hefetyp}(Brauerei \bowtie_{Brauerei.ID=Bier.Brauerei} Bier \bowtie_{Bier.Biersorte=Biersorte.ID} \sigma_{Hefetyp\ =\ 'obergärig'}(Biersorte))$

Fortsetzung nächste Seite!

3. Formulieren Sie die folgenden Anfragen in SQL:

a) Finden Sie die Namen aller Brauereien, die mehr als eine Biersorte herstellen!

b) Geben Sie die Bezeichnung einer jeden Biersorte zusammen mit ihrem durchschnittlichen Alkoholgehalt an!

c) Geben Sie die Stadt aus, in der die älteste Brauerei ihren Sitz hat!

d) Finden Sie alle Bundesländer mit Brauereien, die nur untergärige Biersorten herstellen!

4. Wie sieht die Ergebnisrelation zu folgenden Anfragen auf den Beispieldaten aus?

a)
```
select Hefetyp, count(Bezeichnung) as Anzahl
from Biersorte
group by Hefetyp
having count(Bezeichnung) > 1;
```

b)
```
Select Bezeichnung
from Biersorte
where ID not in
 (select Biersorte
  from Bier join Brauerei on Bier.Brauerei = Brauerei.ID
            join Sitz on Brauerei.Sitz = Sitz.ID
  where Bundesland = 'Bayern');
```

Fortsetzung nächste Seite!

Aufgabe 3: Entwurfstheorie

Gegeben sei das folgende relationale Schema mitsamt Beispieldaten für eine Flugdatenbank. Die Primärschlüsselattribute sind unterstrichen.

Relation "Flüge":

FNr	Datum	VonS	VonL	NachS	NachL	Gesellschaft	Sitz	Flugzeug	Typ	Plätze
LH1432	03.04.11	München	GER	Frankfurt	GER	Lufthansa	Köln	Boeing	747	452
UA732	05.04.12	Chicago	USA	Brisbane	AUS	United	Chicago	Embraer	ERJ-195	108
UA735	05.04.12	Frankfurt	GER	Chicago	USA	United	Chicago	Airbus	320	152
EK935	01.02.11	Dubai	UAE	Montreal	CAN	Emirates	Dubai	Airbus	380	526
UA735	05.04.09	Frankfurt	GER	Chicago	USA	United	Chicago	Airbus	320	152

Gegeben seien ebenso die folgenden funktionalen Abhängigkeiten:

FD1: FNr → VonS, VonL, NachS, NachL, Gesellschaft, Sitz, Flugzeug, Typ, Plätze
FD2: VonS → VonL
FD3: NachS → NachL
FD4: Gesellschaft → Sitz
FD5: Flugzeug, Typ → Plätze

1. Beschreiben Sie kurz, welche Redundanzen in der Datenbank vorhanden sind und welche Anomalien auftreten können. Geben Sie ein Beispiel für jede Anomalie an.
2. Welchen Normalformen genügt das angegebene Relationenschema? Begründen Sie Ihre Entscheidung.
3. Überführen Sie das obige Relationenschema in die dritte Normalform und geben Sie die mit obigen Daten gefüllten Relationen an.
4. Erläutern Sie kurz, welchen Nachteil Normalisierung allgemein für die Anfragebearbeitung haben kann.

Themenschwerpunkt B
(Betriebssysteme)

Teilaufgabe 1:

Aufgabe 1: Petri-Netze

Gegeben sei das in folgender Abbildung dargestellte, boolesche Petri-Netz.

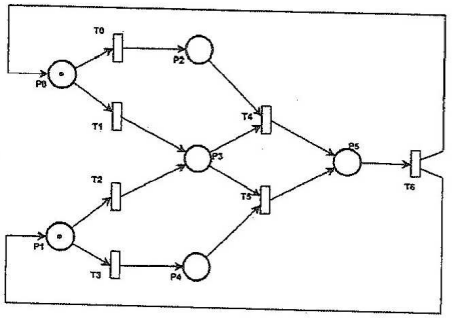


Abbildung 1: Boolesches Petri-Netz

1. Geben Sie den Erreichbarkeitsgraphen zu dem Petri-Netz aus Abbildung 1 an!
2. Begründen Sie, dass es in dem Petri-Netz aus Abbildung 1 zu Verklemmungen kommen kann! Sie können für Ihre Argumentation den Erreichbarkeitsgraphen aus der vorigen Teilaufgabe benutzen.

Fortsetzung nächste Seite!

Aufgabe 2: Seitenersetzung

2.1 Grundsätzliche Fragen

1. Erklären Sie die Seitenersetzungs-Strategien LIFO und LRU jeweils knapp!
2. Nennen und erklären Sie kurz zwei Elemente, die typischerweise in einem Seiten-Deskriptor enthalten sind!
3. Diskutieren Sie kurz Vor- und Nachteile großer bzw. kleiner Seitengrößen!

2.2 LFU

1. Sei eine Menge von Seiten $N = \{0, 1, 2, 3, 4, 5, 6\}$ und eine Menge von Kacheln (Frames) $F = \{f_1, f_2, f_3, f_4\}$ gegeben. Nun wird in folgender Reihenfolge auf die Seiten zugegriffen:

   $w = 3\ 4\ 6\ 0\ 4\ 5\ 1\ 2\ 3\ 4\ 5\ 2\ 5\ 6$

   Vollziehen Sie die Strategie LFU (Least Frequently Used) schrittweise nach, indem Sie die folgende Tabelle auf Ihr Blatt übertragen und vervollständigen! Wenn die Wahl der zu verdrängenden Seite nicht eindeutig ist, soll immer die Seite mit dem kleineren Seiten-Index verdrängt werden. Wir gehen davon aus, dass der Zugriffszähler einer Seite auch beim Verdrängen nicht gelöscht wird.

Anfrage	f_1	f_2	f_3	f_4	aufaddierte Zahl der Seitenfehler
3	3	-	-	-	1
4	3	4	-	-	2
6	3	4	6	-	3
0	3	4	6	0	4
4	3	4	6	0	4
5					
1					
2					
3					
4					
5					
2					
5					
6					

2. Welches Problem kann bei der LFU Strategie auftreten? Diskutieren Sie kurz!

Fortsetzung nächste Seite!

Aufgabe 3: Memory-Management

1. Welche Art von Daten verwaltet das Betriebssystem typischerweise in einer Halde (Heap) und welche Art von Daten in einem Keller (Stack)? Erläutern Sie an Beispielen!
2. Worin besteht der Unterschied zwischen externer und interner Fragmentierung von Speicherbereichen? Erläutern Sie!
3. Skizzieren Sie eine typische Speicherhierarchie! Was ist das fundamentale Konzept von Speicherhierarchien? Warum funktionieren sie bspw. in einem gewöhnlichen PC recht gut?
4. Sie haben ein Geschäft für Computer-Hardware. Ein guter Kunde möchte den Arbeitsspeicher seines 32 Bit-Systems von 4 GB auf 16 GB aufrüsten. Welche Speicherbausteine verkaufen Sie dem Kunden? Sein virtueller Speicher ist in 2^{16} Seiten eingeteilt. Wie groß ist eine solche Seite demnach, wenn man von byteweiser Adressierung ausgeht?

Fortsetzung nächste Seite!

Aufgabe 4: Wechselseitiger Ausschluss

Ein Gemüsehändler führt Gurken und Karotten. Er hat einen Lieferanten C, der ihm pro Lieferung eine Kiste Gurken und eine Kiste Karotten liefert. Er hat einen Kunden A, der bei ihm jeweils eine Kiste Karotten abholt, und einen Kunden B, der bei ihm jeweils eine Kiste Gurken abholt. Für sich alleine betrachtet laufen die Prozesse folgendermaßen ab:

```
Process A
{
 while (TRUE)
 {
  <an den Gemuesestand treten>;
  <Kiste Karotten abholen>;
  <Gemuesestand verlassen>;
 }
}
```

```
Process B
{
 while (TRUE)
 {
  <an den Gemuesestand treten>;
  <Kiste Gurken abholen>;
  <Gemuesestand verlassen>;
 }
}
```

```
Process C
{
 while (TRUE)
 {
  <an den Gemuesestand treten>;
  <Kiste Gurken abladen>;
  <Kiste Karotten abladen>;
  <Gemuesestand verlassen>;
 }
}
```

Synchronisieren Sie diese drei Prozesse, indem Sie in Pseudocode-Notation in geeigneter Weise Semaphore deklarieren und Semaphor-Operationen einfügen.

Die Pseudocode-Operation zum Deklarieren eines Semaphors mit Namen `name` mit Startwert `n` laute:

```
name(n);
```

Die Pseudocode-Operationen zum Aufrufen von P und V auf der Semaphore mit Namen `name` lauten:

```
P(name)
V(name)
```

Sperrphasen sind möglichst kurz zu halten und folgende Bedingungen müssen erfüllt sein:

- An den Gemüsestand können jeweils nur zwei Personen herantreten
- Kunde A darf nur an den Stand herantreten, wenn mindestens eine Kiste Karotten vorrätig ist.
- Kunde B darf nur an den Stand herantreten, wenn mindestens eine Kiste Gurken vorrätig ist.
- Der Gemüsestand hat beschränkte Kapazität, er kann höchstens 10 Gemüsekisten aufnehmen.
- Der Lieferant darf nur an den Gemüsestand herantreten, wenn er seine Lieferung vollständig abladen kann.
- Zu Beginn sei der Stand leer und niemand steht dort.

1. Listen Sie alle benötigten Semaphoren auf und erklären Sie ihren Zweck bzw. ihre Pragmatik.
2. Fügen Sie in den bzw. zu dem oben aufgelisteten Pseudocode des Ablaufs der beteiligten Prozesse an geeigneter Stelle entsprechende Aufrufe zum Deklarieren der Semaphoren und der Aufrufe von P und V ein!

Fortsetzung nächste Seite!

Aufgabe 5: Prozesse, Threads und Scheduling

5.1 Prozesswechsel

Unter *Kontextwechsel* versteht man das Wegnehmen des Rechenkerns von einem Prozess und die Zuweisung des Rechenkerns an einen anderen Prozess.

1. Beschreiben Sie stichpunktartig den Ablauf eines Kontextwechsels!
2. Welche Rolle spielt die *Größe* des Prozesskontrollblocks beim Kontextwechsel? Betrachten Sie diesen Aspekt im Hinblick auf *Häufigkeit* und *Effizienz* multipler Kontextwechsel.
3. Grenzen Sie die Aufgabe des Dispatchers von der des Schedulers ab! Wie gestaltet sich die Zusammenarbeit dieser beiden Komponenten?

5.2 Scheduling

Ein Scheduler verwende eine priorisierte Zeitscheibenstrategie:

- In jedem Zeitquantum wird der Prozess mit der höchsten Priorität ausgewählt.
- Jeder Prozess P_i besitzt eine Initialpriorität I_i.
- Das Quantum beträgt $q = 4$ Zeiteinheiten. Das zugeteilte Quantum wird bei Prioritätsänderungen nicht unterbrochen.
- Im rechnenden Zustand wird die Priorität des Prozesses nach je einer Zeiteinheit um 2 erniedrigt.
- Im rechenwilligen Zustand wird die Priorität des Prozesses nach je einer Zeiteinheit um 1 erhöht.
- Prioritäten reichen von 0 bis 25.

Nun seien drei Prozesse P_1, P_2 und P_3 gegeben. Der Vektor ihrer Initialprioritäten sei:

$$I = (8, 6, 12)$$

Der Vektor der Ankunftszeiten der Prozesse sei:

$$a = (0, 2, 0)$$

d.h. P_1 startet zum Zeitpunkt 0, P_2 zum Zeitpunkt 2 und P_3 zum Zeitpunkt 0. Ferner seien die Bedienzeiten der Prozesse bekannt:

$$b = (8, 4, 8)$$

1. Visualisieren Sie die Ausführung von P_1, P_2 und P_3 in einem geeigneten Diagramm, indem Sie die zeitliche Abfolge der Bedienung der Prozesse darstellen. Skizzieren Sie sowohl die Bedienung als auch die Wartezeit des jeweiligen Prozesses!
2. Berechnen Sie die mittlere Wartezeit $\overline{W}$ und die mittlere Verweilzeit $\overline{V}$ für dieses Szenario!

Fortsetzung nächste Seite!

Teilaufgabe 2:

Aufgabe 1

Die folgenden drei Jobs stehen zu den angegebenen Zeiten (in Minuten) zur Ausführung an:

Job	Ankunftszeit	Bearbeitungszeit
A	0	6
B	2	3
C	5	2

a) Untersuchen Sie hierfür die folgenden Scheduling-Strategien:

1. First-Come-First-Served (FCFS)
2. Shortest-Job-First (SJF)
3. Round-Robin (RR) mit Zeitquantum q = 2

Geben Sie dazu jeweils in einem Gantt-Diagramm, mit je einer Zeile pro Job, für alle Jobs die Bearbeitungs- und Wartezeiten an.

b) Zusätzlich zu den obigen Jobs A, B, C (der Kategorie Batch) sollen zu den Zeitpunkten 4 und 6 die Jobs Nummer D und E (der Kategorie Interactive) mit jeweils einer Laufzeit von 6 Zeiteinheiten zur Bearbeitung ankommen.

Job	Ankunftszeit	Bearbeitungszeit
A	0	6
B	2	3
C	5	2
D	4	6
E	6	6

Nun soll ein Multi-Level-Scheduling eingesetzt werden derart, dass jeweils eine Warteschlange für die Kategorien Interactive und Batch existiert, wobei bei den interaktiven Jobs Round-Robin (mit Zeitscheibe q=2) und bei den Batch-Jobs FCFS verwendet wird. Die interaktiven Jobs haben Priorität über die Batch-Jobs, d. h. falls gerade ein Batch-Job bearbeitet wird und es kommt ein neuer interaktiver Job an, so wird dieser Batch-Job genau solange angehalten, bis keine interaktiven Jobs mehr im System sind.

Geben Sie in einem Gantt-Diagramm, mit je einer Zeile pro Job, für alle Jobs die Bearbeitungs- und Wartezeiten an!

Fortsetzung nächste Seite!

Aufgabe 2

Ein System mit drei Betriebsmitteltypen R_1, R_2, R_3 und fünf Prozessen A, B, C, D, E befinde sich aktuell im unten gezeigten Zustand.

Verfügbarkeitsvektor

R_1	R_2	R_3
3	3	2

Zuteilungsmatrix

	R_1	R_2	R_3
A	0	1	0
B	2	0	0
C	3	0	2
D	2	1	1
E	0	0	2

Maximale Anforderungen inkl. bereits zugeteilter Betriebsmittel

	R_1	R_2	R_3
A	7	5	3
B	3	2	2
C	9	0	2
D	2	2	2
E	4	3	3

a) Erklären Sie informell die Vorgehensweise des Banker's-Algorithmus'!

b) Berechnen Sie die Matrix der Maximalen Restforderungen!

c) Zeigen Sie mittels der Einzelschritte der Safety-Prüfung aus dem Banker's-Algorithmus, dass der oben beschriebene Zustand sicher ist!

d) Kann eine neue Anforderung (1 0 2) von Prozess A unter Erhalt der Safety des Systems nun sofort erfüllt werden?

Fortsetzung nächste Seite!

Aufgabe 4

a) Gegeben sei ein System mit drei verfügbaren Frames ***F0, F1, F2***, welche zu Beginn leer sind. Tragen Sie für die Seitenanforderungsfolge **1, 2, 3, 4, 5, 1, 2, 5, 1, 2, 3, 4, 5** und die Verfahren **OPT** (Belady's optimale Strategie), **FIFO** (First-In-First-Out), **LRU** (Least-Recently-Used) die jeweils entstehende Belegung der Speicherkacheln an!

Übertragen Sie die unten gegebenen Schemata auf Ihr Blatt und ergänzen Sie diese! Geben Sie für jedes Verfahren die Anzahl der resultierenden Page-Faults an!

OPT

	1	2	3	4	5	1	2	5	1	2	3	4	5
F0													
F1													
F2													

FIFO

	1	2	3	4	5	1	2	5	1	2	3	4	5
F0													
F1													
F2													

LRU

	1	2	3	4	5	1	2	5	1	2	3	4	5
F0													
F1													
F2													

b) Der Algorithmus von Belady minimiert die Anzahl der Page-Faults. Warum wird er trotzdem in realen Paging-Systemen nicht eingesetzt?

66114

Datenbank- und Betriebssysteme (vertieft)

Herbst 2014

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____ Kennwort: _____ Arbeitsplatz-Nr.: _____	**Herbst 2014**	**66114**

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

— Prüfungsaufgaben —

Fach: **Informatik (vertieft studiert)**

Einzelprüfung: **Datenbank- und Betriebssysteme**

Anzahl der gestellten Themen (Aufgaben): **4 Aufgaben, von denen zwei gemäß untenstehender Auswahlregel zu bearbeiten sind!**

Anzahl der Druckseiten dieser Vorlage: **16**

Zu den zwei Themenschwerpunkten A (Datenbanksysteme) und B (Betriebssysteme) ist jeweils entweder die Teilaufgabe 1 oder 2 zu wählen!
Auf der Vorderseite des Kopfbogens sind im Feld „gewähltes Thema: Nr." die Nummern der beiden gewählten Teilaufgaben anzugeben (z. B. A2, B1)!

Bitte wenden!

Themenschwerpunkt A
(Datenbanksysteme)

Teilaufgabe 1:

Aufgabe 1:

Entwerfen Sie ein ER-Modell für die folgende Miniwelt, die sich mit Zügen, Verbindungen, Orten usw. befasst. Geben Sie Kardinalitäten in (*min, max*)-Notation an. Sie brauchen Schlüsselattribute nicht zu kennzeichnen.

- Ein *Zugtyp* hat eine Bezeichnung, eine Höchstgeschwindigkeit und eine Kapazität.
- Es gibt *Strecken*, die einen Start- und ein Zielort haben.
- Ein *Zug* hat einen Namen und ist von einem bestimmten Zugtyp. Ein Zug kann auf einer Strecke verkehren und jede Strecke wird von mindestens einem Zug bedient. Jeder Zug, der auf einer Strecke fährt, bekommt für diese Fahrt eine eigene Bezeichnung.
- *Bahnunternehmen* besitzen Züge und bieten Verbindungen auf bestimmten Strecken an.
- *Reisende* nutzen auf Strecken angebotene Züge. Der Preis für die Strecke mit dem Zug soll protokolliert werden. Reisende haben Namen. Ein Zug verkehrt nur auf einer Strecke, wenn mindestens 10 Reisende mitfahren.

Aufgabe 2:

Wir betrachten die Schüler-Datenbank mit den folgenden Tabellen:

- Schüler(<u>Snr</u>, Vorname, Nachname, Geburtsdatum, Klasse): *Persönliche Angaben zu den Schülern.*
- Abschlussnote(<u>Snr, Fach</u>, Note): *Notenspiegel der Schüler.*
- Raum(<u>Rnr</u>, Typ, Sitzplätze): *Typ und Anzahl der Sitzplätze der Räume.*
- Raumbelegung(<u>Klasse, Fach</u>, Rnr): *Raumbelegung nach Klassen und Fächern.*

Erstellen Sie in SQL folgende Anfragen:

a) Bestimmen Sie alle Räume und die Anzahl ihrer Sitzplätze, in denen die Klasse '7b' Unterricht hat.

b) Bestimmen Sie alle Paare von Schülern (jeweils gegeben durch die beiden Schülernummern), die in dieselbe Klasse gehen und am selben Tag Geburtstag haben.

c) Bestimmen Sie die Durchschnittsnote des Schülers mit der Nummer '1'.

d) Bestimmen Sie die Fächer, in denen der Schüler mit der Nummer '1' seine schlechtesten Noten hat.

Fortsetzung nächste Seite!

Augabe 3:

Wir erweitern die Datenbank aus Aufgabe 2 um die folgenden beiden Tabellen:

- Fach(Fach, Fachbetreuer).
- Klasse(Klasse, Klassleiter)

a) Geben Sie ein ER-Diagramm für die erweiterte Datenbank an, in dem möglichst viele Relationen als Relationships modelliert sind.

b) Geben Sie geeignete Create Table-Statements mit Primär- und Fremdschlüsselbedingungen zur Erzeugung der Tabellen Schüler, Abschlussnote und Fach an.

c) Fügen Sie mit Hilfe eines SQL-Statements ein Attribut Ausbildungsrichtung mit einem geeigneten Wertebereich zur Tabelle Schüler hinzu.

d) Löschen Sie alle Abschlussnoten von Schülern der Klasse '9a'.

Aufgabe 4:

Gegeben seien die funktionalen Abhängigkeiten

$$
\begin{aligned}
F = \{ \quad & B \rightarrow AC, && \text{(i)} \\
& BE \rightarrow C, && \text{(ii)} \\
& F \rightarrow AE, && \text{(iii)} \\
& D \rightarrow EF \quad \} && \text{(iv)}
\end{aligned}
$$

der Relation $R(A,B,C,D,E,F)$.

a) Berechnen Sie eine kanonische Überdeckung F_C von F. Geben Sie alle Zwischenschritte und die angewendeten Transformationen an.

b) Geben Sie alle Kandidatenschlüssel für das Relationenschema R (jeweils mit Begründung) sowie die Nichtschlüsselattribute an.

c) In welcher Normalform ist R? Geben Sie eine Begründung an.

Teilaufgabe 2:

Aufgabe 1:

1.1 Beantworten Sie folgende Fragen.

a) Welche Auswirkungen hat eine Transaktion, wenn sie mit Abort abgebrochen wird? Begründen Sie Ihre Antwort.
b) Grenzen Sie die Begriffe Datenbankverwaltungssystem und Datenbank voneinander ab.
c) Was ist das Ergebnis einer Projektion und was einer Selektion (= Restriktion) in der relationalen Algebra?
d) Was bedeutet die Abkürzung SQL ausgeschrieben? Worin unterscheidet sich diese Sprache von Programmiersprachen wie Java/C/Basic/... und welche Vorteile ergeben sich daraus?
e) Wozu dient der Primärschlüssel und welche Eigenschaften muss er erfüllen?

1.2 Geben Sie für jede der folgenden Aussagen an, ob sie richtig oder falsch ist und begründen Sie Ihre Antwort.

a) Das Anlegen von Views führt zu Redundanzen in der Datenbank.
b) Bei einer 1:1-Beziehung braucht nur eine der beiden Relationen einen Primärschlüssel.
c) In einer Relation sind die Tupel nicht zwingend in der Einfügereihenfolge abgelegt.
d) Eine Datenbankoperation muss die Datenbank in einem konsistenten Zustand hinterlassen.
e) Die SQL-Anweisung *select* gehört zur Data-Definition-Language (DDL).

Aufgabe 2:

a) E/R-Modell

Im Folgenden finden Sie die Beschreibung der Inventarverwaltungssoftware eines Unternehmens. Erstellen Sie zu dieser ein ER-Diagramm. Geben Sie dabei auch die Kardinalitäten an. Verwenden Sie nur die im Folgenden genannten Attribute als Schlüsselkandidaten:

Mitarbeiter, denen eine eindeutige Personalnummer zugeordnet ist, arbeiten in genau einer Abteilung, die firmenweit durch eine Abteilungsnummer identifiziert wird und an ein oder mehreren Standorten untergebracht ist. Jeder Standort hat eine Straße, eine Hausnummer, eine Postleitzahl sowie einen Ort und wird durch diese Adresse eindeutig gekennzeichnet. An den einzelnen Standorten kann es Räume geben, die durch eine Raumnummer bezeichnet werden, die am jeweiligen Standort eindeutig ist.

Jeder Mitarbeiter kann Gegenstände besitzen, für die er alleine verantwortlich ist. Alle Gegenstände haben eine eindeutige Inventarnummer und sind von einem bestimmten Inventartyp, der eine eindeutige Bezeichnung hat (z.B. Laptop Modell 500). Es gibt zwei Arten von Inventartypen: Bewegliches Inventar und unbewegliches Inventar. Zu beweglichen Inventartypen wird das Gewicht gespeichert. Unbewegliche Inventartypen haben eine Position, die aus der geographischen Breite und der geographischen Länge besteht.

Gegenstände können von einem Raum in einen anderen Raum umgezogen werden. Für Umzugswünsche kann eine Priorität vergeben werden, um festzulegen, wie dringend der Gegenstand am neuen Ort gebraucht wird. Es kann nur einen Umzugswunsch pro Gegenstand geben.

Fortsetzung nächste Seite!

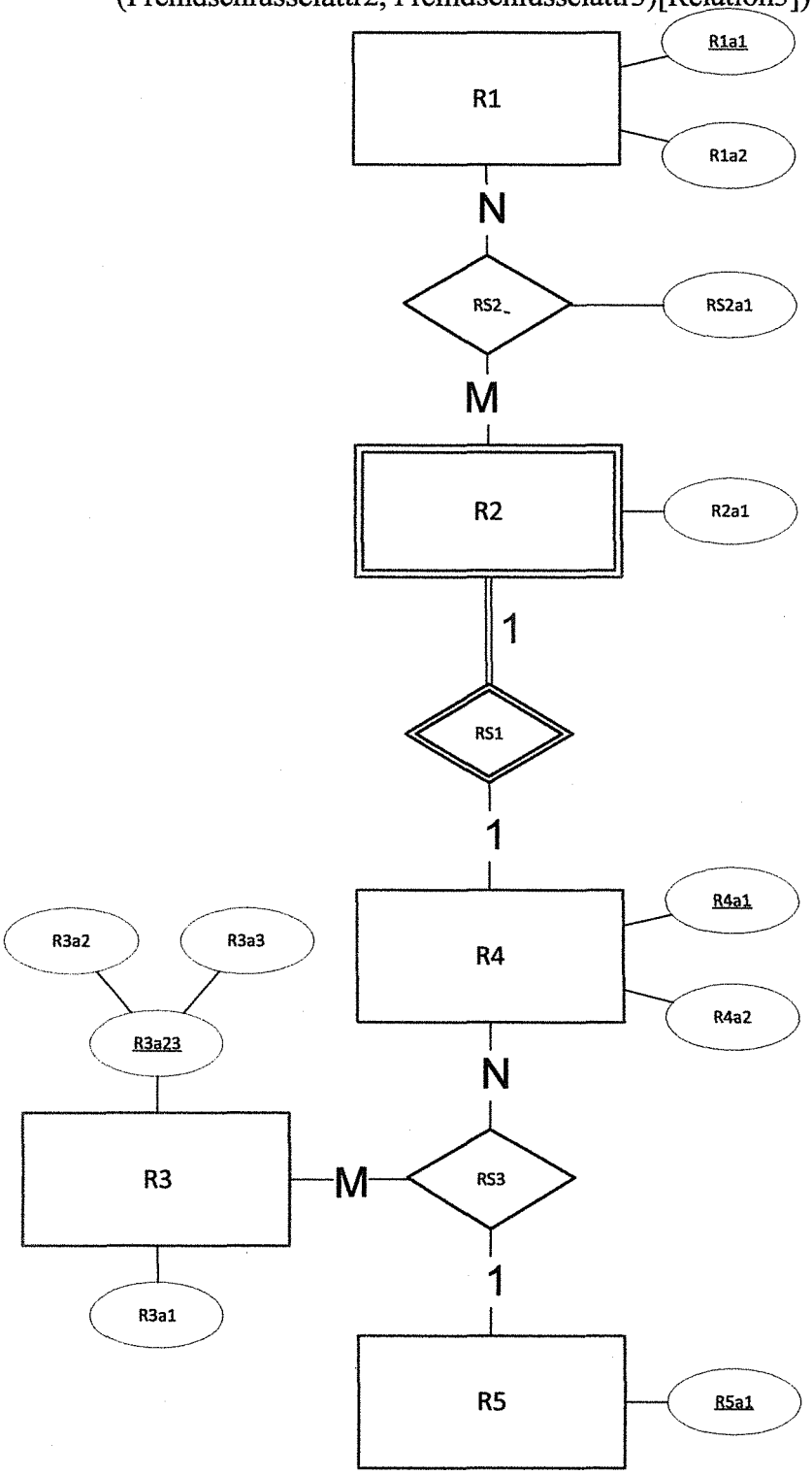
b) Abbildung auf Relationen

Entwerfen Sie zum untenstehenden E/R-Diagramm ein Relationenschema in dritter Normalform (3. NF) mit möglichst wenigen Relationen.

Verwenden Sie dabei folgende Notation:

Primärschlüssel werden durch Unterstreichen gekennzeichnet, Fremdschlüssel durch die Nennung der Relation, auf die sie verweisen, in eckigen Klammern hinter dem Fremdschlüsselattribut. Attribute zusammengesetzter Fremdschlüssel werden durch runde Klammern als zusammengehörig markiert.

Beispiel: Relation1 (Primärschlüssel, Attribut1, Attribut2, Fremdschlüsselattr1[Relation2], (Fremdschlüsselattr2, Fremdschlüsselattr3)[Relation3])



Fortsetzung nächste Seite!

Aufgabe 3:

Gegeben ist folgender Auszug aus einer Datenbank zur Verwaltung von Kochrezepten:

```
Autor(ID, Vorname, Nachname)
Rezept(Rezeptnummer, Name, Verfasser[Autor], Beschreibung)
Zutat(Bezeichnung, Einkaufspreis)
Rezept_Zutat(Rezept[Rezept], Zutat[Zutat], Menge)
```

Primärschlüssel sind unterstrichen, Fremdschlüssel dadurch gekennzeichnet, dass hinter ihnen in eckigen Klammern der Name der Relation angegeben ist, auf die sie sich beziehen.
Verwenden Sie bei der Formulierung der Anfragen nur standardkonformes SQL. Die Verwendung von Views ist erlaubt und empfohlen.

a) Schreiben Sie SQL-Statements, die die Tabellen zu den oben angegebenen Relationen anlegen. Sie können beliebige sinnvolle Datentypen wählen. Denken Sie auch an die Schlüssel und referentiellen Beziehungen.
b) Schreiben Sie ein SQL-Statement, das Vor- und Nachname des Autors mit der ID 42 ausgibt.
c) Schreiben Sie ein SQL-Statement, das alle Autoren, die weniger als 10 Rezepte verfasst haben, ausgibt, sortiert nach der Anzahl der von ihnen verfassten Rezepte. Ausgegeben werden soll die Anzahl der verfassten Rezepte, sowie Vor- und Nachname des Autors. Auch Autoren, die keine Rezepte verfasst haben, sollen in der Ausgabe enthalten sein.
d) Schreiben Sie eine SQL-Anfrage, die die Namen aller Rezepte zusammen mit der Anzahl verschiedener für sie benötigter Zutaten ausgibt, absteigend sortiert nach Anzahl der Zutaten.
e) Schreiben Sie eine SQL-Anfrage, die die Top-Ten der Rezepte mit den höchsten Zutatenkosten ausgibt. Die Kosten eines Rezepts berechnen sich als Summe über das Produkt aus Menge und Einkaufspreis aller Zutaten. Ausgegeben werden soll der Name des Rezepts, die Kosten und der Rang (1 = teuerstes Rezept bis 10), aufsteigend sortiert nach Rang. Gehen Sie davon aus, dass keine zwei Rezepte gleiche Kosten aufweisen.
   Verwenden Sie keine nicht-standardisierten Erweiterungen, wie `rownum` und ebenfalls nicht das Schlüsselwort `fetch`!

Aufgabe 4:

a) Von wann bis wann werden von einer Transaktion verwendete Sperren gehalten (dynamisches Sperren)? Begründen Sie, warum genau diese Zeitpunkte sinnvoll sind.
b) Bei der Transaktionsverarbeitung kann es zu Verklemmungen (Deadlocks) kommen. Erklären Sie anhand eines Beispiels, wie diese entstehen können. Erläutern Sie, wie ein Datenbankverwaltungssystem diese Situation erkennen und was es tun kann, um eine Verklemmung aufzulösen! Gehen Sie auch auf die entstehenden Konsequenzen ein.

Themenschwerpunkt B
(Betriebssysteme)

Teilaufgabe 1:

Aufgabe 1:

Beantworten Sie folgende Fragen zum Thema Prozesse.

1. Erläutern Sie kurz den Einfluss der Schedulingstrategie auf die Wahl eines geeigneten Prozesszustandsmodelles.
2. In den folgenden Fragen soll das 7–Zustands–Prozessmodell betrachtet werden.
   a) Skizzieren Sie das 7–Zustands–Prozessmodell. Kennzeichnen Sie alle Übergänge durch Pfeile. Beschriften Sie alle Pfeile mit der Aktion, die bei dem entsprechenden Übergang ausgeführt wird.
   b) Welche zusätzlichen Möglichkeiten ergeben sich im 7–Zustands-Prozessmodell, die im 5–Zustands-Prozessmodell nicht vorhanden sind? Geben Sie für jeden der zusätzlichen Zustandsübergänge ein Beispiel an.
3. Welche Problematik kann bei der Anwendung der nicht-präemptiven Schedulingstrategien First-Come First-Serve entstehen?
4. Erklären Sie kurz den Begriff Dispatching.
5. Erläutern und begründen Sie kurz. ob präemptive oder nicht-präemptive Schedulingstrategien weniger Verwaltungsoverhead erzeugen.

Fortsetzung nächste Seite!

Aufgabe 2:

In dieser Aufgabe sollen drei Scheduling-Strategien untersucht werden: die nicht-präemptive Strategie FCFS (First Come First Served), die nicht-präemptive Strategie SJF (Shortest Job First) und die präemptive Strategie RR (Round Robin). Dazu seien die folgenden Prozesse mit ihren Ankunftszeitpunkten und Rechenzeiten (in beliebigen Zeiteinheiten) gegeben.

Prozess	Ankunftszeitpunkt	Rechenzeit
P_1	0	3
P_2	1	5
P_3	2	4
P_4	3	2
P_5	5	3

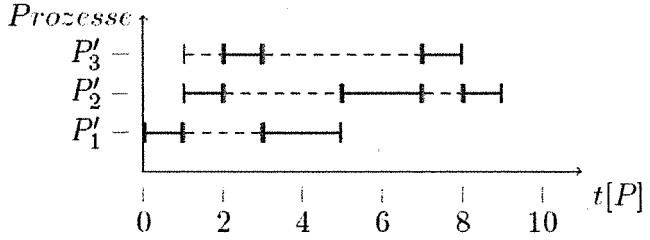
Gehen Sie davon aus, dass jeder Prozess sein Zeitquantum stets vollständig ausnutzt d.h. kein Prozess gibt den Prozessor freiwillig frei (Ausnahme: bei Prozessende).

Trifft ein Prozess zum Zeitpunkt t ein, so wird er direkt zum Zeitpunkt t beim Scheduling berücksichtigt. Wird ein Prozess zum Zeitpunkt t' unterbrochen, so reiht er sich auch zum Zeitpunkt t' wieder in die Warteschlange ein. Sind zwei Prozesse absolut identisch bezüglich ihrer relevanten Werte, so werden die Prozesse nach aufsteigender Prozess–ID in die Warteschlange eingereiht. Diese Annahme gilt sowohl für neu im System eintreffende Prozesse, als auch für den Prozess, dem der Prozessor u.U. gerade entzogen wird.

Beispiel: Es seien folgende Ankunfts– und Rechenzeiten für die drei Beispielprozesse P'_1, P'_2 und P'_3 gegeben:

Prozess	Ankunftszeitpunkt	Rechenzeit
P'_1	0	3
P'_2	1	4
P'_3	1	2

Das folgende Diagramm veranschaulicht ein beliebiges Scheduling der drei Prozesse P'_1, P'_2 und P'_3:



Bearbeiten Sie unter den gegebenen Voraussetzungen nun die folgenden Aufgaben:

1. Erstellen Sie entsprechend dem Beispiel ein Diagramm für die **nicht-präemptive Strategie FCFS**, das für die Prozesse P_1–P_5 angibt, wann welchem Prozess Rechenzeit zugeteilt wird und wann die Prozesse jeweils terminieren. Kennzeichnen Sie zudem für jeden Prozess seine Ankunftszeit.

2. Erstellen Sie entsprechend dem Beispiel ein Diagramm für die **nicht-präemptive Strategie SJF**, das für die Prozesse P_1–P_5 angibt, wann welchem Prozess Rechenzeit zugeteilt wird und wann die Prozesse jeweils terminieren. Kennzeichnen Sie zudem für jeden Prozess seine Ankunftszeit.

3. Erstellen Sie entsprechend dem Beispiel ein Diagramm für die **präemptive Strategie RR**, das für die Prozesse P_1–P_5 angibt, wann welchem Prozess Rechenzeit zugeteilt wird und wann die Prozesse jeweils terminieren. Die Dauer einer Zeitscheibe betrage 2 Zeiteinheiten. Gehen Sie davon aus, dass jeder Prozess die Dauer seiner Zeitscheibe stets vollständig ausnutzt, sofern er nicht terminiert. Terminiert ein Prozess aber vor Ablauf seiner Zeitscheibe, gibt er den Prozessor zum Zeitpunkt der Terminierung sofort frei. Trifft genau nach Ende einer Zeitscheibe ein neuer Prozess ein, so wird der neue Prozess **vor** dem gerade aktiven in die Warteschlange eingereiht.

Fortsetzung nächste Seite!

4. Berechnen Sie als Dezimalzahl mit einer Nachkommastelle die mittlere Verweil- und Wartezeit für die drei Verfahren FCFS, SJF und RR.
5. Warum ist der Einsatz von SJF in einem realen System in der Regel nicht realisierbar?
6. Welchen weiteren Nachteil hat SJF in Bezug auf die Verweildauer von Prozessen?

Fortsetzung nächste Seite!

Aufgabe 3:

Gegeben seien eine Menge an Seiten $N = \{ 0, 1, 2, 3, 4 \}$ und eine Menge der im Arbeitsspeicher zur Verfügung stehenden Seitenrahmen $F = \{ f_0, f_1, f_2 \}$. Auf die Seiten wird in der folgenden Reihenfolge zugegriffen:

`w = 0 3 4 2 2 0 3 1 2 3 4 0 4 1 3 4`

Ein Seitenfehler liegt immer dann vor, wenn sich eine referenzierte Seite nicht im Arbeitsspeicher befindet. Der Arbeitsspeicher ist zu Beginn leer.

1. Bei der Paging-Strategie FIFO (First In, First Out) wird bei einem Seitenfehler diejenige Seite im Arbeitsspeicher ersetzt, deren Zeitpunkt der Zuordnung zu einem Seitenrahmen am längsten zurück liegt.

   Ermitteln Sie die Anzahl der Seitenfehler für die Seitenersetzungsstrategie FIFO, indem Sie alle Veränderungen im Speicher dokumentieren. Erstellen Sie dazu eine Tabelle mit dem Schema:

Zeit	Referenzierte Seite	f_0, t	f_1, t	f_2, t	Summe Seitenfehler
⋮	⋮	⋮	⋮	⋮	⋮

   Ordnen Sie dabei jeder referenzierten Seite den entsprechenden Seitenrahmen f_i ($i \in \{0, \ldots, 2\}$) zu und dokumentieren Sie den Zeitpunkt t der Zuordnung. Geben Sie zudem nach jedem Seitenzugriff die aktuelle Summe an Seitenfehlern an.

   Achtung: Bereits in den Hauptspeicher geladene Seiten dürfen nicht von einem Seitenrahmen in einen anderen verschoben werden!

2. Ermitteln Sie die Anzahl der Seitenfehler für die Seitenersetzungsstrategie LRU (Least Recently Used), indem Sie eine Tabelle mit dem bekannten Schema aus Teilaufgabe 1 erstellen. Es sollen alle Seitenzugriffe seit dem Laden einer Seite berücksichtigt werden. Ordnen Sie in der Tabelle wieder jede referenzierte Seite dem entsprechenden Seitenrahmen f_i ($i \in \{0, \ldots, 2\}$) zu und dokumentieren Sie den Zeitpunkt t eines Seitenzugriffs. Geben Sie zudem nach jedem Seitenzugriff die aktuelle Summe an Seitenfehlern an.

   Achtung: Bereits in den Hauptspeicher geladene Seiten dürfen nicht von einem Seitenrahmen in einen anderen verschoben werden!

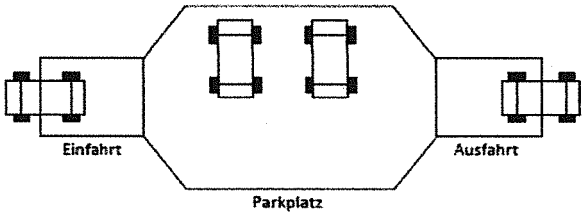
3. Nun soll theoretisch untersucht werden, wieviele Seitenfehler für das Beispiel aus Teilaufgabe 2 entstehen, wenn das System 1, 2, 3, 4 oder 5 Seitenrahmen besäße. Der zugehörige *Distance String* lautet ∞ ∞ ∞ ∞ 0 3 3 ∞ 3 2 4 4 1 4 3 2 . Der Rechenweg muss klar nachvollziehbar sein.

4. Welchen Effekt im Zusammenhang mit der Anzahl der Seitenrahmen eines Systems beschreibt die *Belady's Anomalie*? Kann diese Anomalie mit der Seitenersetzungsstrategie LRU (Least Recently Used) auftreten?

Fortsetzung nächste Seite!

Aufgabe 4:

In dieser Aufgabe soll ein Parkplatz simuliert werden, in dem aufgrund des Platzmangels die Anzahl der aktuell parkenden Autos zu jedem Zeitpunkt maximal `maxAutos` betragen darf. Die einzelnen parkenden Autos (die als Java Threads realisiert werden sollen) besitzen Tickets mit einer eindeutigen Ticketnummer. Sie können den Parkplatz durch eine Einfahrt befahren, indem sie ihr Ticket an einen dort angebrachten Ticketleser einer Schranke halten. Am anderen Ende des Parkplatzes können die parkenden Autos über die Ausfahrt wieder herausfahren, indem Sie ihr Ticket an den Ticketleser der dort installierten Schranke halten, welche anschließend das Passieren erlaubt. Der Parkplatz ist somit nur in eine Richtung passierbar. Die Schranken sind so konstruiert, dass immer nur genau ein Auto passieren kann, nachdem der Fahrer sein Ticket an den jeweiligen Ticketleser gehalten hat. Auf dem Parkplatz dürfen die Auto für eine beliebige Zeitspanne verweilen – somit können Autos den Parkplatz in einer anderen Reihenfolge verlassen, als sie ihn befahren haben. In folgender Abbildung ist ein solcher Parkplatz mit einer maximalen Anzahl an gleichzeitigen Besuchern von `maxAutos = 3` dargestellt, wobei sich gegenwärtig nur `anzahlBesucher = 2` Auto auf dem Parkplatz befinden. Ein Auto ist gerade dabei die Schranke an der Ausfahrt zu passieren und ein Auto wartet auf den Einlass an der Einfahrt.



Im Folgenden soll schrittweise eine Klasse `Parkplatz` implementiert werden. Die Beispielimplementierungen der Klassen `Auto` und `Simulation` sollen Ihnen verdeutlichen, wie die Klasse `Auto` verwendet werden kann:

Die Klasse `Simulation`:

```
1  public class Simulation {
2      private Parkplatz my_parkplatz;
3      private Auto[] autos;
4
5      public Simulation() {
6          my_parkplatz = new Parkplatz(3);
7          autos = new Auto[10];
8          for(int i = 0; i < autos.length; i++) {
9              autos[i] = new Auto(my_parkplatz, i);
10             autos[i].start();
11         }
12     }
13
14
15     public static void main(String[] args) {
16         new Simulation();
17     }
18 }
```

Die Klasse `Auto`:

```
1  import java.util.Random;
2
```

Fortsetzung nächste Seite!

```
3  public class Auto extends Thread {
4      private Parkplatz my_parkplatz;
5      private int ticket_nummer;
6      private Random generator;
7
8      public Auto(Parkplatz parkplatz, int ticket_nummer) {
9          this.my_parkplatz = parkplatz;
10         this.ticket_nummer = ticket_nummer;
11         generator = new Random();
12     }
13
14     public void run() {
15             try {
16                 Thread.sleep(generator.nextInt(2000) + 500);
17                 my_parkplatz.einfahren(ticket_nummer);
18                 Thread.sleep(generator.nextInt(2000) + 500);
19                 my_parkplatz.ausfahren(ticket_nummer);
20             }
21             catch(InterruptedException ie) {}
22     }
23 }
```

Der Klasse Parkplatz soll folgendes Interface zugrunde liegen:

```
1  interface Parkplatz {
2     // Konstruktor
3     public Parkplatz(int maxBesucher);
4
5     // Methoden
6     public synchronized void einfahren(int ticket_nummer);
7     public synchronized void ausfahren(int ticket_nummer);
8  }
```

1. Was sind die kritischen Bereiche bei diesem Problem?
2. Überlegen Sie sich die notwendigen Attribute der Klasse `Parkplatz` und implementieren Sie deren Konstruktor. *Kommentieren Sie Ihre Lösung ausführlich!*
3. Implementieren Sie die Methode `einfahren(int ticket_nummer)`, welche das Passieren der Schranke an der Einfahrt des Parkplatzes modelliert, sowie die Methode `ausfahren(int ticket_nummer)`, welche im Gegenzug das Verlassen des Parkplatzes über die Schranke an der Ausfahrt modelliert. Beachten Sie bei der Implementierung, dass alle oben genannten Bedingungen eingehalten werden. Schreiben Sie die Methode `einfahren(int ticket_nummer)` so, dass sie die eindeutige Ticketnummer des einfahrenden Autos sowie die Anzahl der sich gerade auf dem Parkplatz befindlichen Autos (inklusive des einfahrenden Autos) auf der Konsole ausgibt. Schreiben sie analog die Methode `ausfahren(int ticket_nummer)` so, dass sie die Ticketnummer des Autos, das gerade die Schranke am Ausgang passiert sowie die Anzahl der sich gerade auf dem Parkplatz befindlichen Autos (exklusive des Autos das den Parkplatz gerade verlässt) auf der Konsole ausgibt. Sie können davon ausgehen, dass die Methoden `einfahren(int ticket_nummer)` bzw. `ausfahren(int ticket_nummer)` immer in einer sinnvollen Reihenfolge aufgerufen werden (siehe Beispielimplementierung der Klasse Auto). *Kommentieren Sie Ihre Lösung ausführlich!*
4. Erläutern Sie kurz, warum Ihre Lösung des wechselseitigen Ausschlusses die Bedingung der *Mutual Exclusion* erfüllt.

Teilaufgabe 2:

Aufgabe 1:

Eine der wesentlichen Aufgaben eines Betriebssystems ist es, die vorhandenen Hardware- und Software-Betriebsmittel zu verwalten und für einen verklemmungsfreien Ablauf der einzelnen Prozesse zu sorgen.

(a) In welche Klassen können Betriebsmittel eingeteilt werden? Geben Sie für jede der genannten Klassen ein Beispiel an.

(b) Bei der Zuteilung von Betriebsmitteln an Prozesse sollte das Auftreten von Deadlocks ausgeschlossen werden. Welche vier Bedingungen sind Voraussetzung für einen Deadlock? Bei welchen der oben genannten Betriebsmittelklassen können diese Bedingungen eintreten?

(c) Welche drei grundsätzlichen Verfahren gibt es, um mit der Deadlock-Problematik umzugehen? Beschreiben Sie jedes Verfahren. Wie ist jeweils die grundsätzliche Vorgehensweise? Nennen Sie den Namen eines Algorithmus, der bei einem dieser Verfahren zum Einsatz kommt.

(d) Fünf Philosophen sitzen an einem runden Tisch, in dessen Mitte ein immer voller Teller mit Nudeln steht. Rechts und links von jedem Philosophen liegt jeweils eine Gabel. Zum Essen benötigt ein Philosoph zwei Gabeln. Die Philosophen sind nur mit zwei Dingen beschäftigt: Denken und Essen. Ist einer hungrig, so greift er zuerst nach der linken und dann nach der rechten Gabel und fängt an zu essen. Ist eine Gabel belegt, so wartet er, bis sie wieder frei ist. Hat der Philosoph sich satt gegessen, so legt er die Gabeln wieder an ihren Platz und fährt fort mit Denken.

Betrachten Sie unten stehenden Pseudocode, der einen einzelnen Philosophen simuliert. Erläutern Sie, wie es in dieser Lösung zu einem Deadlock kommen könnte, indem Sie die Bedingungen aus Aufgabenteil (b) nachweisen. Zu welcher Klasse nach Aufgabenteil (a) gehören die vorhandenen Betriebsmittel? Neben einem Deadlock könnte noch ein weiteres Problem auftauchen. Erläutern Sie kurz welches.

```
N=5; Procedure phil(i:INTEGER)
     WHILE (TRUE) DO
          think;                      //Philosoph denkt nach
          take_fork(i);               //nimmt die linke Gabel
          take_fork((i+1) mod N);     //nimmt die rechte Gabel
          eat;                        //isst
          put_fork(i);                //legt die linke Gabel zurück
          put_fork((i+1) mod N);      //legt die rechte Gabel zurück
     END;
END phil;
```

Fortsetzung nächste Seite!

Aufgabe 2:

Zu einem bestimmten Zeitpunkt T kommen folgende Jobs in der gegebenen Reihenfolge in einer Rechenanlage mit einer CPU zur Bearbeitung an. Der zusätzliche Aufwand beim Umschalten zwischen den Jobs soll vernachlässigt werden.

Job	Bearbeitungszeit
1	7
2	3
3	6
4	10
5	1
6	2

(a) Beschreiben Sie unter Verwendung von Gantt-Diagrammen (s. u.) die Bearbeitung der Jobs für die Schedulingverfahren *First-come-first-served* (FCFS), *Shortest-Job-First* (SJF) und *Round-Robin* mit Zeitquantum=3 (RR).

Beispiel für ein Gantt-Diagramm:

Zeit	1	2	3	4	5	6	8	9
Prozessor	Job#1			Job#2	Job#3			leer

(b) Geben Sie für alle drei Schedulingverfahren den „Turnaround" (Gesamtzeit im System) eines jeden Jobs an.

(c) Geben Sie für alle drei Schedulingverfahren die Anfangswartezeiten, d.h. die Zeit eines jeden Jobs an, die er warten muss, bis er zum ersten Mal die CPU zugeteilt bekommt.

(d) Welches Schedulingverfahren besitzt (angewandt auf das obige Beispiel und über alle Jobs gerechnet) die geringste durchschnittliche Gesamtwartezeit, d.h. bei welchem Verfahren ist die Zeit der Inaktivität der Prozesse durchschnittlich am geringsten?

Fortsetzung nächste Seite!

Aufgabe 3:

(a) Beschreiben Sie knapp die Speicherverwaltungsstrategien First-Fit, Best-Fit und Worst-Fit. Nennen Sie zu jeder Strategie mindestens einen Vorteil und einen Nachteil.

(b) Gegeben sei zusätzlich noch Fragment-Fit: die zu speichernden Daten können in Fragmente aufgeteilt und auf verschiedene Speicherbereiche verteilt werden. Dabei wird der zuerst gefundene Speicherblock gefüllt und ggf. überschüssige Daten in die nächsten freien Speicherbereiche geschrieben. Nennen die Vor- und Nachteile dieses Verfahrens gegenüber den oben genannten First-Fit, Best-Fit und Worst-Fit.

(c) Gegeben sei die Liste mit den folgenden, freien Speicherblöcken (in genau dieser Reihenfolge, wobei 1MB = 1024KB). Diese Blöcke liegen nicht nebeneinander im RAM.

800KB 1MB 260KB 120KB 5MB

Es werden nun an das Betriebssystem in folgender Reihenfolge Speicheranforderungen gestellt:

1000KB 550KB 90KB 160KB 4MB 900KB

Wie teilt jede der vier Strategien First-Fit, Best-Fit, Worst-Fit und Fragment-Fit den Speicher zu? Zur vollständigen Lösung gehört die Liste aller freien Blöcke nach jeder erfüllten Anforderung.

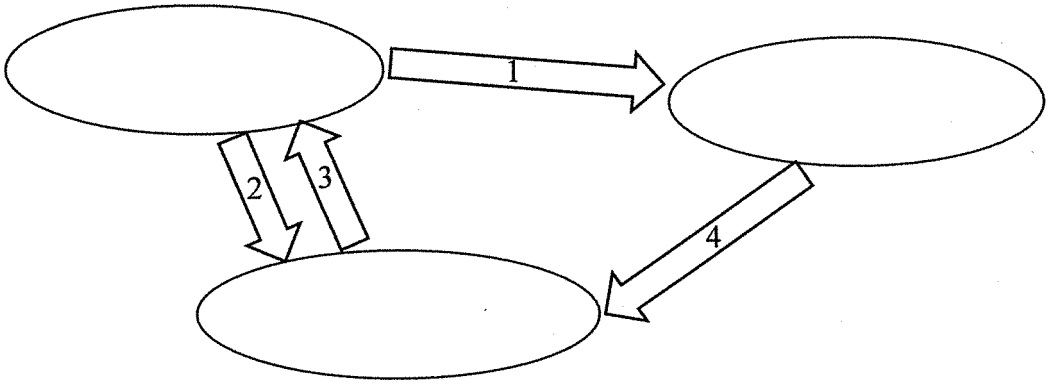
(d) Beantworten Sie bezogen auf das obige Beispiel die folgenden Fragen mit knapper Begründung: Welches Verfahren schneidet bzgl. Erfüllung der Anfragen am schlechtesten ab? Welches Verfahren arbeitet bzgl. der Speicherfragmentierung am effizientesten?

Fortsetzung nächste Seite!

Aufgabe 4:

(a) Definieren Sie den Begriff „Prozess“ in Betriebssystemen.

(b) Übertragen Sie die gegebene Grafik auf Ihr Blatt, fügen Sie die Prozesszustände „rechnend“, „blockiert“ und „rechenbereit“ ein und beschriften Sie die Übergänge entsprechend der Ereignisse, durch die sie ausgelöst werden.



66114

Datenbank- und Betriebssysteme (vertieft)

Herbst 2015

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____ Kennwort: _____ Arbeitsplatz-Nr.: _____	**Herbst 2015**	**66114**

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

— Prüfungsaufgaben —

Fach: **Informatik (vertieft studiert)**

Einzelprüfung: **Datenbank- und Betriebssysteme**

Anzahl der gestellten Themen (Aufgaben): **2**

Anzahl der Druckseiten dieser Vorlage: **12**

Zu den zwei Themenschwerpunkten A (Datenbanksysteme) und B (Betriebssysteme) ist jeweils entweder die Teilaufgabe 1 oder 2 zu wählen!
Auf der Vorderseite des Kopfbogens sind im Feld „Gewähltes Thema: Nr." die Nummern der beiden gewählten Teilaufgaben anzugeben (z. B. A2, B1)!

Bitte wenden!

Themenschwerpunkt A
(Datenbanksysteme)

Teilaufgabe 1

1. Modellierung

Wir wollen eine relationale Datenbankstruktur für ein Online-Auktionshaus modellieren.

Das Auktionshaus hat Mitglieder. Diese Mitglieder haben Kundennummern, Namen und Adressen. Sie können Verkäufer und/oder Käufer sein. Verkäufer können eine Laden-URL (eine Subdomain) erhalten, die zu einer Seite mit ihren aktuellen Auktionen führt. Verkäufer können neue Auktionen starten. Die Auktionen eines Verkäufers werden durchnummeriert. Diese Auktionsnummer ist nur je Verkäufer eindeutig. Jede Auktion hat ein Mindestgebot und eine Ablaufzeit.

Auf Auktionen können Gebote abgegeben werden. Die Gebote auf eine Auktion werden nach ihrem Eintreffen nummeriert. Diese Nummer ist nur innerhalb einer Auktion eindeutig. Zu einem Gebot werden noch die Zeit des Gebots sowie der gebotene Geldbetrag angegeben. Die Gebote werden von Käufern abgegeben. Jedes Gebot muss einem Käufer zugeordnet sein.

Jede Auktion besteht aus einer Menge von Artikeln, aber aus mindestens einem. Artikel haben eine Beschreibung und können und über ihre Artikel-ID identifiziert werden. Zur Katalogisierung der Artikel gibt es Kategorien. Kategorien haben eine eindeutige ID und einen Namen. Jede Kategorie kann Subkategorien besitzen. Auch Subkategorien sind Kategorien (und können damit weitere Subkategorien besitzen). Jeder Artikel kann beliebig vielen Kategorien zugeordnet werden.

Entwerfen Sie für das beschriebene Szenario ein ER-Diagramm. Bestimmen Sie hierzu:
- die Entity-Typen, die Relationship-Typen und jeweils deren Attribute,
- ein passendes ER-Diagramm,
- die Primärschlüssel der Entity-Typen, welche Sie anschließend in das ER-Diagramm eintragen,
- die Funktionalitäten der Relationship-Typen, welche Sie ebenfalls in das ER-Diagramm eintragen.

Fortsetzung nächste Seite!

2. Normalformen

Gegeben sei folgendes verallgemeinerte Relationenschema in 1. Normalform:

$$R:\{[A,B,C,D,E,F,G,H]\}$$

Für R soll die folgende Menge FD von funktionalen Abhängigkeiten gelten:

- $F \rightarrow E$
- $A \rightarrow BD$
- $AE \rightarrow D$
- $A \rightarrow EF$
- $AG \rightarrow H$

Bearbeiten Sie mit diesen Informationen folgende Teilaufgaben. Vergessen Sie dabei nicht Ihr Vorgehen stichpunktartig zu dokumentieren und zu begründen.

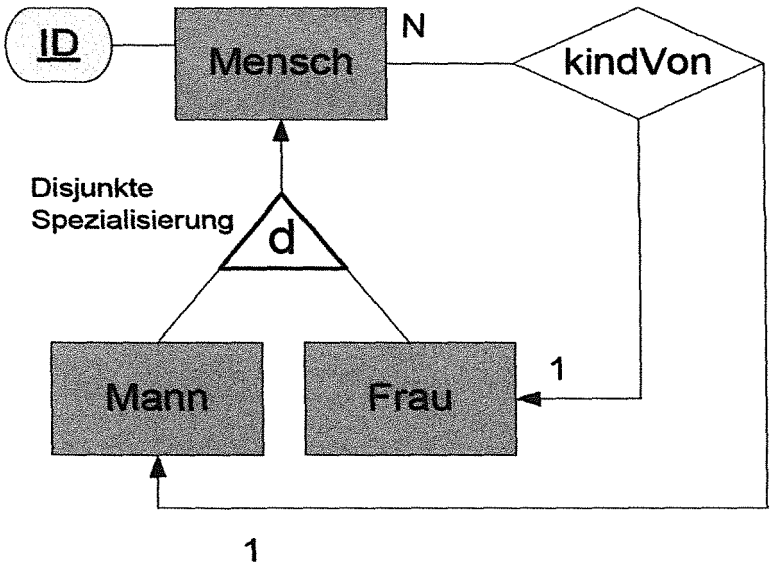
a) Bestimmen Sie alle Schlüsselkandidaten von R. Begründen Sie stichpunktartig, warum es außer den von Ihnen gefundenen Schlüsselkandidaten keine weiteren geben kann.
b) Ist R in 2NF, 3NF?
c) Berechnen Sie eine kanonische Überdeckung von FD. Es genügt, wenn Sie für jeden der vier Einzelschritte die Menge der funktionalen Abhängigkeiten als Zwischenergebnis angeben.
d) Bestimmen Sie eine Zerlegung von R in 3NF. Wenden Sie hierfür den Synthesealgorithmus an.

3. SQL

Gegeben seien folgende Relationen:

Mensch(ID, MutterID, VaterID)
Mann(ID)
Frau(ID)

Das zugehörige ER-Modell für dieses relationale Datenbankschema sieht folgendermaßen aus:



Fortsetzung nächste Seite!

Bearbeiten Sie folgende Teilaufgaben:

a) Finden Sie die Töchter der Frau mit *ID* 42.
b) Gibt es Männer, die ihre eigenen Großväter sind? Formulieren Sie eine geeignete SQL-Anfrage.
c) Definieren Sie eine View *VaterKind (VaterID; KindID)*, die allen Vätern *(VaterID)* ihre Kinder *(KinderID)* zuordnet. Diese View darf keine NULL-Werte enthalten.
d) Verwenden Sie die View aus c), um alle Väter zurückzugeben, absteigend geordnet nach der Anzahl ihrer Kinder.
e) Hugo möchte mit folgender Anfrage auf Basis der View aus c) alle kinderlosen Männer erhalten:

```
SELECT VaterID
FROM VaterKind
GROUP BY VaterID
HAVING COUNT(KindID) = 0
```

i. Was ist das Ergebnis von Hugos Anfrage und warum?
ii. Formulieren Sie eine Anfrage, die tatsächlich alle kinderlosen Männer zurückliefert.

Hinweis: Denken Sie daran, dass SQL auch Mengenoperationen kennt.

Teilaufgabe 2

1. Modellierung

Das Fitnessstudio „Bavariogym" plant zur Unterstützung des Trainings eine App für ihre Mitglieder. Betrachten Sie dazu folgendes Szenario:

- Zu einem Mitglied werden Vorname, Nachname, Eintrittsdatum und der Tarif gespeichert.
- Im Fitnessstudio arbeiten speziell ausgebildete Fitnesstrainer. Deren Aufgabe ist es, Trainingspläne zu erstellen. Ein Trainer besitzt immer eine eindeutige Personal-Nummer. Gehen Sie davon aus, dass ein Trainer kein Mitglied ist.
- Ein Trainingsplan ist nur in einem bestimmten Zeitraum gültig. Sehen Sie hierzu zwei Daten für die zeitlichen Grenzen vor. Um einen Trainingsplan zu identifizieren soll eine fortlaufende Nummer verwendet werden.
- Jeder Trainingsplan schreibt das Training an unterschiedlichen Geräten vor. Dabei sollen jeweils die Anzahl der Wiederholungen pro Satz, die Anzahl der Sätze und die Gewichtseinstellung berücksichtigt werden.
- Ein Mitglied kann nach mehreren Trainingsplänen gleichzeitig trainieren. Ebenso findet ein Trainingsplan bei mehreren verschiedenen Mitgliedern Anwendung.
- Ein Gerät besitzt einen eindeutigen Namen, sowie eine Information über die Muskelgruppe, die trainiert wird.

Fortsetzung nächste Seite!

a) Entwerfen Sie für das beschriebene Szenario ein ER-Modell in Chen-Notation. Bestimmen Sie hierzu:
   - Die Entity-Typen, die Relationship-Typen und jeweils deren Attribute.
   - Die Funktionalitäten der Relationship-Typen.

b) Wie würde sich ihr Modell aus Teilaufgabe a) ändern, wenn die Trainer den Mitgliedern individuelle, jedoch nicht exklusive Trainingspläne anfertigen? Zeichnen Sie ein aktualisiertes ER-Modell in Chen-Notation. Bestimmen Sie hierzu:
   - Die Entity-Typen, die Relationship-Typen, jeweils **ohne** deren Attribute.
   - Die Funktionalitäten der Relationship-Typen.

c) Wie würden Sie vorgehen, um ihr Modell aus Teilaufgabe a) so zu ändern, dass Sie für jede Person im Fitnessstudio den Vornamen, den Nachnamen, die Straße, die Hausnummer, die PLZ und den Ort speichern können. Beschränken Sie ihre Antwort auf maximal drei Sätze.

2. SQL-Anfragen

Gegeben ist das folgende relationale Schema:

$EQUIPE\ \{[\underline{id}, name]\}$
$FAHRER\ \{[\underline{id}, name, equipe_id]\}$
$ETAPE\ \{[\underline{id}, date, NbKm]\}$
$TEILNAHME\ \{[\underline{fahrer_id, etape_id}, zeit]\}$

Zur Modellierung der Tour de France im letzten Jahr (2014) wird folgendes relationales Schema verwendet. EQUIPE enthält Informationen zur Mannschaft (der EQUIPE), zu denen jeder Fahrer eindeutig zugeordnet ist. FAHRER enthält die Informationen zum Fahrer mit dem Fremdschlüssel equipe_id. ETAPE enthält die Information zu den Etapen, die gefahren werden, das Attribut NbKm ist die Länge der Etape in km. TEILNAHME stellt die Verbindung von Etapen und Fahrern her, das Attribut zeit ist die individuelle Zeit eines Fahrers fahrer_id in der Etape etape_id. Fällt der Fahrer für die Etape id aus, so erscheint er in der Relation TEILNAHME für dieses etape_id nicht.

Formulieren Sie folgende Anfragen in SQL. Achten Sie falls nötig auf duplikatfreie Ausgaben:

a) Wie lang war die Tour de France letztes Jahr?

b) Geben Sie die Namen der Fahrer der Equipe 'RENNSTALL' aus.

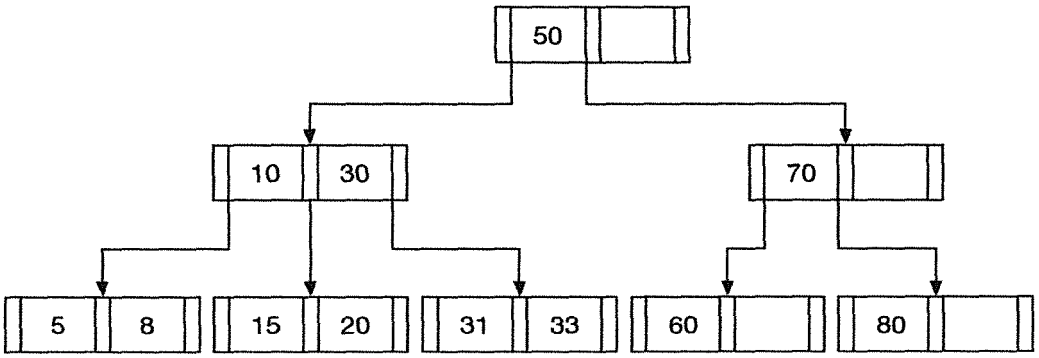
c) Geben Sie die Namen der Fahrer an, die an allen Etapen teilgenommen haben.

d) Geben Sie die Gesamtlaufzeit aller Läufer der Equipe 'RENNSTALL' in der 1. Etape (ETAPE.id=1) aus.

Fortsetzung nächste Seite!

3. Indexstrukturen

Gegeben ist der folgende B-Baum der Ordnung 3 (max. drei Kindknoten, max. zwei Schlüssel pro Knoten):



a) Fügen Sie die Werte 9 und 45 ein. Löschen Sie anschließend die Werte 30 und 70. Zeichnen Sie den Baum nach jeder Einfüge- bzw. Lösch-Operation.

b) Geben Sie das Vorgehen für exakte Suche und Bereichssuche im B+-Baum an.

c) Geben Sie einen Vorteil und einen Nachteil von B+-Bäume im Vergleich zu Hash-Tabellen an.

Themenschwerpunkt B
(Betriebssysteme)

Teilaufgabe 1

In allen gängigen Betriebssystemen wird heute die Parallelität durch Prozesse, teils auch zusätzlich durch Threads, umgesetzt. Zur Verwaltung der Prozesse sind die Zustände wichtig, die Prozesse während ihrer Lebenszeit annehmen können; zudem sind die Wechselwirkungen wichtig, die sich in der Synchronisation und Verklemmungen niederschlagen, sowie für die Zuordnung zu den Prozessoren das Scheduling und das Dispatching. Der Prozessadressraum basiert in der Regel auf einem virtuellen Speicher, so dass das Paging eine zentrale Rolle spielt. Für die langfristige, persistente Speicherung von Daten werden Dateisysteme herangezogen.

1. Prozesse und ihre Zustände

   Prozesse durchlaufen während ihrer Lebenszeit einige Zustände. Benennen Sie die wichtigsten dieser Zustände, erläutern Sie diese kurz und skizzieren Sie ein Diagramm, das die Zustandsübergänge aufzeigt. Beschreiben Sie kurz die Bedeutung der jeweiligen Zustandsübergänge.

2. Synchronisation zwischen Prozessen

   Gegeben sei die folgende Festlegung mit einer Klasse für Semaphore. Nutzen Sie diese für die Lösung der nachfolgenden Teilaufgaben a) und b).

```
public class Semaphor
{
   // Konstruktor
   public Semaphor (int init) {...}

   // Methoden
   public void prolog () {...}
   public void epilog () {...}
}
```

Fortsetzung nächste Seite!

a) Gegeben seien zwei Prozesse P1 und P2. Der Einfachheit halber sollen beide Prozesse jeweils nur aus zwei Berechnungsschritten bestehen: P1 aus S11 gefolgt von S12 und P2 aus S21 gefolgt von S22. Ein einfaches Beispiel für die Synchronisation zwischen Prozessen besteht nun darin, dass der zweite Schritt des zweiten Prozesses (S22) erst dann ausgeführt werden darf, wenn der erste Schritt des ersten Prozesses erfolgreich abgeschlossen wurde. Geben Sie eine Lösung für diese Synchronisationsaufgabe unter Verwendung von Semaphoren an. Beschreiben Sie Ihre Lösung programmiersprachlich oder in Pseudocode und treffen Sie dazu die nötigen Annahmen.

b) Das bei den Semaphoren benötigte Warten kann entweder mit aktivem Warten oder mit passivem Warten realisiert werden. Beschreiben Sie kurz die Unterschiede zwischen diesen Varianten.

3. Scheduling von Prozessen

Gegeben sei ein Ein-Prozessor-System mit sechs Prozessen bzw. Aufträgen $A_1,\ldots,A_6$ und den dazugehörenden Bedienzeiten $b=(4,2,5,3,1,1)$, also den Zeiten, die die jeweiligen Prozesse zur Bearbeitung benötigen. Die Ankunftszeiten der Aufträge werden in den nachfolgenden Unteraufgaben angegeben. Vor dem Start des Schedulings liege kein Prozess bzw. Auftrag vor.

a) Der Vektor der Ankunftszeiten für die sechs Prozesse sei durch $a=(0,1,2,3,4,5)$ gegeben. Berechnen Sie die mittlere Verweilzeit und die mittlere Wartezeit für die nichtunterbrechende Strategie Shortest Processing Time (SPT), also die Strategie, bei der in jedem Zeitintervall einer der Aufträge mit kürzester Bedienzeit ausgeführt wird. Notieren Sie den Ablauf entsprechend.

b) Nun kommen die Prozesse (Aufträge) mit folgendem Vektor der Ankunftszeiten $a'=(5,4,3,2,1,0)$ an. Die Aufträge werden nun nach einer Zeitscheibenstrategie mit dem Quantum q bedient (Round-Robin Strategie). Die Reihenfolge wird durch die Zeitpunkte der Ankünfte festgelegt. Falls erforderlich werden Neuankünfte immer vor einem aktuellen Wiederholer eingereiht. Berechnen Sie die mittlere Verweilzeit und die mittlere Wartezeit für das Quantum $q=1/2$.

4. Virtueller Speicher, Seiteneinteilung und Paging

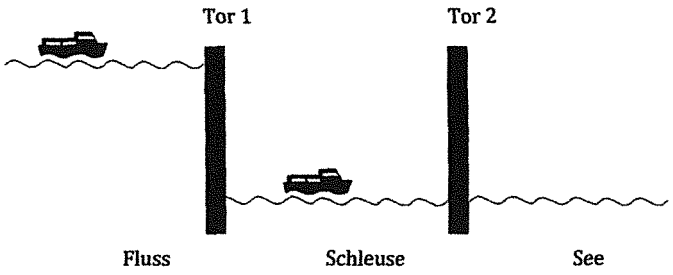
a) Gegeben sei ein Prozess mit einem virtuellen Speicher von sechs Seiten, für dessen Realisierung drei Seitenrahmen (Kachelmenge = $\{K_1,K_2,K_3\}$) zur Verfügung stehen. Geben Sie für die Strategien LRU (Least Recently Used), LFU (Least Frequently Used) und FIFO (First in First out) die Entwicklung der Seitenrahmenbelegung bei der Zugriffsreferenzkette a'= 1 3 2 6 4 2 1 6 5 1 2 1 1 3 4 3 5 1 an. Ist die Auswahlentscheidung nicht eindeutig, verwenden Sie zusätzlich FIFO. Die Seitenrahmen seien zu Beginn leer. Markieren Sie jedes Auftreten von Seitenfehlern und notieren Sie für die jeweilige Strategie die Gesamtzahl der Seitenfehler.

b) Die Menge der Seitenrahmen (Kacheln), in die die Seiten des virtuellen Speichers eingelagert werden, muss durch das Betriebssystem verwaltet werden. Geben Sie eine Datenstruktur zur Verwaltung der Kacheln an und listen Sie die wichtigsten Elemente der Seitenrahmendeskriptoren auf.

Fortsetzung nächste Seite!

Teilaufgabe 2

1. Modellierung paralleler Systeme

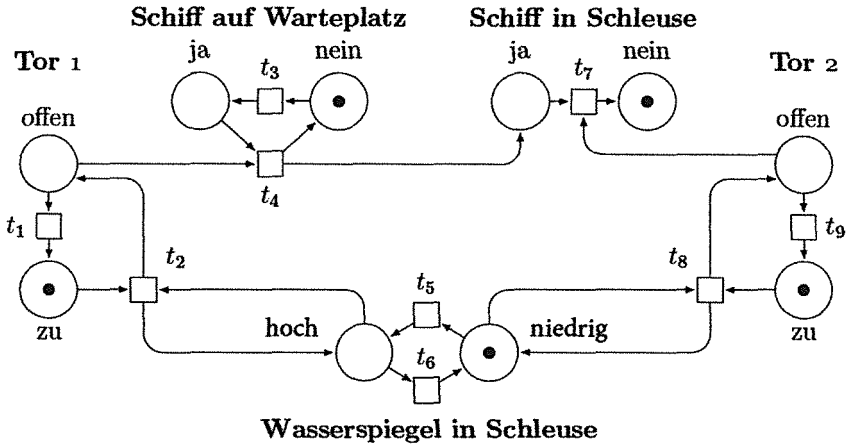


Wir betrachten eine Schleuse, die es Schiffen ermöglicht, von einem Fluss mit hohem Wasserspiegel in einen See mit niedrigem Wasserspiegel zu fahren (nur in diese Richtung). Die Schleuse hat zwei Tore. Der Wasserspiegel in der Schleuse kann entweder hoch oder niedrig sein. Im Fluss gibt es einen Warteplatz für ein Schiff, das überfahren will, und natürlich kann sich auch in der Schleuse selbst ein Schiff befinden. Es gelten folgende zusätzliche Bedingungen:

- Tor 1 darf nur geöffnet sein, wenn der Wasserspiegel in der Schleuse hoch ist, und Tor 2 darf nur geöffnet werden, wenn der Wasserspiegel in der Schleuse niedrig ist.
- Die Schleuse muss nach Durchfahrt eines Schiffes immer in der Lage sein, ein weiteres Schiff durchfahren zu lassen.
- Zustände sind dichotom: Die Tore sind entweder offen oder zu, der Wasserspiegel entweder hoch oder niedrig. Auf dem Warteplatz und in der Schleuse gibt es (unabhängig voneinander) ein Schiff oder nicht.

Fortsetzung nächste Seite!

Gegeben sei nun ein fehlerhafter Entwurf eines natürlichzahligen Petrinetzes zur Schaltung der Schleuse:



t_1: Tor 1 wird geschlossen
t_2: Tor 1 wird geöffnet
t_3: Schiff kommt an
t_4: Schiff fährt in die Schleuse
t_5: Wasser wird hoch
t_6: Wasser wird niedrig
t_7: Schiff fährt weg
t_8: Tor 2 wird geöffnet
t_9: Tor 2 wird geschlossen

Korrigieren Sie das obige Petrinetz durch Einzeichnen der sieben fehlenden Pfeile im gegebenen Diagramm.

Hinweis: Gehen Sie das Netz anhand eines naheliegenden Szenarios durch. Gibt es Schaltfolgen, nach denen eine der genannten Bedingungen nicht mehr erfüllt ist?

2. Prozess- und Threadsynchronisation

Das erste Readers-Writers-Problem beschreibt ein Szenario, bei dem mehrere Prozesse – manche lesend, manche schreibend – um den Zugriff auf eine Ressource konkurrieren. Mehrere lesende Prozesse können zeitgleich auf die Ressource zugreifen, solange kein schreibender Zugriff stattfindet. Will ein Prozess schreiben, wartet er, bis andere Zugriffe abgeschlossen sind. Greift ein Prozess schreibend auf die Ressource zu, ist anderen Prozessen der Zugriff so lange verwehrt. Der nachfolgende Pseudocode modelliert beide Prozesstypen und ist so zu vervollständigen, dass Prozesse schreiben → doActualWriting() respektive lesen → doActualReading() können, ohne die genannten Voraussetzungen zu verletzen. Verwenden Sie hierfür ausschließlich die bereits vorgegebenen Konstrukte. Gehen Sie davon aus, dass die untenstehenden Funktionen wiederholt aufgerufen werden. Ein Aufruf löst einen Schreib- bzw. Lesevorgang aus. Verzichten Sie daher auf Schleifen oder rekursive Funktionsaufrufe.

Fortsetzung nächste Seite!

```
Semaphore write=1;
Semaphore lock=1;

int readcount=0; //zähle Lesezugriffe

writer()
{



    //schreibe
    doActualWriting();



}

reader()
{


    //lese
    doActualReading();





}
```

3. Scheduling

a) Nennen Sie außer dem Prozessnamen vier Inhalte des Prozesskontrollblocks eines Prozesses.

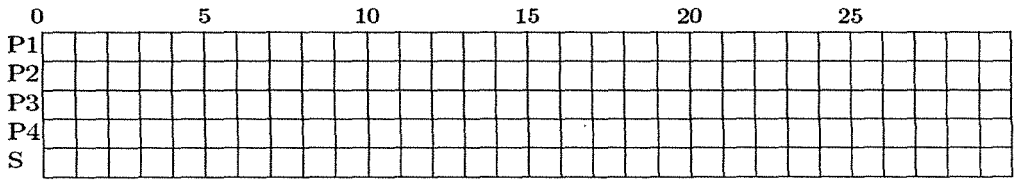
b) Welche Abwägung gilt es bei der Wahl des (fixen) Zeitquantums bei der Implementierung eines Round-Robin-Schedulers zu treffen?

Fortsetzung nächste Seite!

c) Gegeben seien vier Prozesse mit ihrer jeweiligen Ankunftszeit und Bedienzeit:

Prozess	*Ankunfszeit*	*Bedienzeit in ms*
P1	0	13
P2	0	4
P3	16	3
P4	17	4

Nehmen Sie an, dass jeder Kontextwechsel 1ms kostet. Bestimmen Sie die durchschnittliche Prozesswartezeit inklusive Kontextwechsel für einen Round-Robin-Scheduler mit einem Quantum von 4ms. Ist nach einem vollen Quantum ein Prozess nicht **sofort** verfügbar, so wird er nicht in Betracht gezogen – selbst dann nicht, wenn er nach dem Kontextwechsel verfügbar wäre. Bei nicht eindeutiger Entscheidbarkeit haben kleinere Prozess-IDs Vorrang. Das nachfolgende leere Gantt-Diagramm kann Sie bei der Berechnung unterstützen. Dokumentieren Sie Ihr Vorgehen.



4. Speicherverwaltung

a) Erläutern Sie worin der Unterschied zwischen externer und interner Fragmentierung von Speicherbereichen besteht!

b) Gegeben seien eine Menge von Seiten (pages) N = {0, 1, 2, 3, 4, 5, 6, 7} und eine Menge von Seitenrahmen (frames) F = {f1, f2, f3}. Nun wird in folgender Reihenfolge auf die Seiten zugegriffen:

w = 7 2 5 0 4 0 7 5 1 3

Vollziehen Sie schrittweise die Seitenersetzungsstrategie LRU (Least Recently Used), indem Sie die leere Tabelle befüllen.

An-forderung	f_1	f_2	f_3	Page-Fault j/n
7				
2				
5				
0				
4				
0				
7				
5				
1				
3				

66114

Datenbank- und Betriebssysteme (vertieft)

Frühjahr 2016

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____ Kennwort: _____ Arbeitsplatz-Nr.: _____	**Frühjahr 2016**	**66114**

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

— Prüfungsaufgaben —

Fach:	**Informatik (vertieft studiert)**
Einzelprüfung:	**Datenbank- und Betriebssysteme**
Anzahl der gestellten Themen (Aufgaben):	**4 Aufgaben, von denen zwei gemäß untenstehender Auswahlregel zu bearbeiten sind!**
Anzahl der Druckseiten dieser Vorlage:	**20**

Zu den zwei Themenschwerpunkten A (Datenbanksysteme) und B (Betriebssysteme) ist jeweils entweder die Teilaufgabe 1 oder 2 zu wählen!
Auf der Vorderseite des Kopfbogens sind im Feld „Gewähltes Thema: Nr." die Nummern der beiden ausgewählten Teilaufgaben anzugeben (z. B. A2, B1)!

Bitte wenden!

Themenschwerpunkt A
(Datenbanksysteme)

Teilaufgabe 1:

1. Modellierung

Für die bayerische Forstverwaltung wird eine Datenbank zur Erschließung einer Jagd-Statistik benötigt. Gehen Sie dabei von folgendem Szenario aus:

- Die Administration von Jagdgebieten obliegt den Landkreisen. Jeder Landkreis besitzt, neben seinem Namen und der Einwohnerzahl, ein eindeutiges KFZ-Kennzeichen.
- Die Jagd findet in Jagdgebieten statt. Ein Jagdgebiet soll dem Landkreis zugeteilt werden, indem es liegt. Gehen Sie davon aus, dass Jagdgebiete nicht in mehreren Landkreisen liegen können. Zusätzlich ist für jedes Jagdgebiet der Name und die Gesamtfläche zu speichern. Dabei ist zu beachten, dass die Namen nur innerhalb eines einzelnen Landkreises eindeutig sind.
- Die Erlaubnis zum Jagen wird durch einen Jagdschein erteilt. Dieser kann nur von einem Landkreis ausgestellt werden und beschränkt sich auf ein oder mehrere Jagdgebiete. Er wird durch eine Jagdschein-Nummer (JSNR) identifiziert und ist in einem bestimmtem Zeitintervall gültig. Dieses soll über zwei Zeitpunkte festgelegt werden (gültig von, gültig bis).
- Ein Jäger besitzt genau einen Jagdschein. Zu einem Jäger sollen Name, Stadt, Straße und Hausnummer, gespeichert werden. Da die Jagdtradition innerhalb einer Familie häufig von einer zur nächsten Generation weitergegeben wird, kann es vorkommen, dass Name und Adresse von zwei unterschiedlichen Jägern gleich ist (z.B. Vater und Sohn). Aus diesem Grund ist eine eindeutige Identifikationsnummer notwendig.
- Um Statistiken erheben zu können, muss berücksichtigt werden, welches Wild von welchen Jägern zu welchem Zeitpunkt in welchem Jagdgebiet erlegt worden ist. Gehen Sie davon aus, dass es mehrere Jäger geben kann, die gemeinsam ein Wild erlegen (z.B. in einer Jagdgesellschaft). Zu einem Wild gehört die Art (z.B. Reh), die Größe, das Gewicht, sowie eine eindeutige Identifikationsnummer. Zusätzlich unterscheidet man zwischen Haarwild und Federwild, wobei beim Haarwild der Typ des Gehörns (z.B. Hirschgeweih) und beim Federwild die Flügelspannweite betrachtet werden soll.

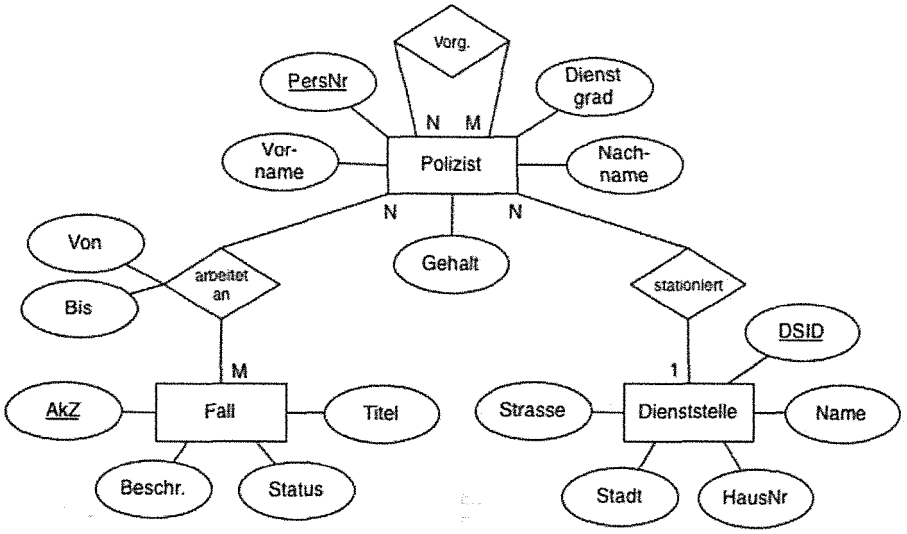
a) Entwerfen Sie für das beschriebene Szenario ein ER-Modell in Chen-Notation. Bestimmen Sie hierzu:
   - die Entity-Typen, die Relationship-Typen und jeweils deren Attribute,
   - die Primärschlüssel der Entity-Typen, welche Sie anschließend in das ER-Diagramm eintragen, und
   - die Funktionalitäten der Relationship-Typen.

b) Überführen Sie das ER-Modell aus Aufgabe a) in ein verfeinertes relationales Modell. Geben Sie hierfür die verallgemeinerten Relationenschemata an. Achten Sie dabei insbesondere darauf, dass die Relationenschemata keine redundanten Attribute enthalten.

Fortsetzung nächste Seite!

2. Relationale Algebra, SQL

Gegeben sei folgendes ER-Modell, welches Polizisten, deren Dienststelle und Fälle, an denen sie arbeiten, speichert:



Gehen Sie dabei von dem dazugehörigen relationalen Schema aus:

Polizist: {[*PersNr, DSID, Vorname, Nachname, Dienstgrad, Gehalt*]}
Dienststelle: {[*DSID, Name, Strasse, HausNr, Stadt*]}
Fall: {[{*AkZ, Titel, Beschreibung, Status*]}
Arbeitet_An: {[*PersNr, AkZ, Von, Bis*]}
Vorgesetzte: {[*PersNr, PersNr_Vorgesetzter*]}

a) Formulieren Sie eine Anfrage in relationaler Algebra, welche den Vornamen und Nachnamen von Polizisten zurückgibt, deren Dienstgrad 'Polizeikommissar' ist und mehr als 1500 Euro verdienen.

b) Formulieren Sie eine Anfrage in relationaler Algebra, welche die Titel der Fälle ausgibt, die von Polizisten mit dem Nachnamen 'Mayer' bearbeitet wurden.

c) Formulieren Sie eine SQL-Anfrage, welche die Anzahl der Polizisten ausgibt, die in der Stadt 'München' arbeiten und mit Nachnamen 'Schmidt' heißen.

d) Formulieren Sie eine SQL-Anfrage, welche die Namen der Dienststellen ausgibt, die am 14.02.2012 an dem Fall mit dem Aktenzeichen XZ1508 beteiligt waren. Ordnen Sie die Ergebnismenge alphabetisch (aufsteigend) und achten Sie darauf, dass keine Duplikate enthalten sind.

e) Definieren Sie die View 'Erstrebenswerte_Dienstgrade', welche Dienstgrade enthalten soll, die in München mit durchschnittlich mehr als 2500 Euro besoldet werden.

Fortsetzung nächste Seite!

f) Formulieren Sie eine SQL-Anfrage, welche Vorname, Nachname und Dienstgrad von Polizisten mit Vorname, Nachname und Dienstgrad ihrer Vorgesetzten als ein Ergebnis-Tupel ausgibt (siehe Beispiel-Tabelle). Dabei sind nur Polizisten zu selektieren, die an Fällen gearbeitet haben, deren Titel den Ausdruck 'Fussball' beinhalten. An Vorgesetzte sind keine Bedingungen gebunden. Achten Sie darauf, dass Sie nicht nur direkte Vorgesetzte, sondern alle Vorgesetzte innerhalb der Vorgesetzten-Hierarchie betrachten. Ordnen Sie ihre Ergebnismenge alphabetisch (absteigend) nach Nachnamen des Polizisten.

VN	NN	DG	VN_VG	NN_VG	DG_VG
Hans	Müller	Polizeimeister	Andreas	Schmidt	Polizeikommissar
Hans	Müller	Polizeimeister	Stefan	Hoffmann	Polizeidirektor
Josef	Fischer	Polizeihauptmeister	Sebastian	Wagner	Polizeioberkommissar
Josef	Fischer	Polizeihauptmeister	Stefan	Hoffmann	Polizeidirektor

Abbildung 1: Ergebnistabelle

Hinweis: Sie dürfen Views verwenden, um Teilergebnisse auszudrücken.

Fortsetzung nächste Seite!

3. Normalformen

Gegeben sei ein Auszug der Relation S einer relationalen Datenbank in Abbildung 2. Darin sind Informationen zu Schrauben eines Baumarkts enthalten. Jeder Schraubentyp besitzt eine eindeutige Artikelnummer, sowie eine Kopfform, ein Formelement, einen Gewindetyp, das Material und die jeweilige Länge. Zusätzlich ist zu jeder Schraube gespeichert, welche DIN Norm sie erfüllt. Diese Normen standardisieren Kopfform, Formelement, Material und Gewinde. Zum Beispiel erfüllt eine Senk-Holzschraube mit Schlitz die DIN Norm 97.

ArtNr	DIN	Kopfform	Formelement	Gewinde	Material	Länge
SR2141	DIN571	Sechskant	Kreuz	Holzschraube	Holz	34mm
SR8923	DIN478	Vierkant	Trox-TR	Rechts	Stahl	41mm
SR0984	DIN7996	Rund	Schlitz	Links	Messing	80mm
SR7688	DIN571	Sechskant	Kreuz	Holzschraube	Holz	52mm
SR3579	DIN965	Senk	Kreuz	Rohr	Keramik	29mm
SR4139	DIN571	Sechskant	Kreuz	Holzschraube	Holz	78mm
SR6760	DIN97	Senk	Schlitz	Holzschraube	Holz	29mm

Abbildung 2: Relation S

Bearbeiten Sie mit diesen Informationen folgende Teilaufgaben. Vergessen Sie dabei nicht, Ihr Vorgehen stichpunktartig zu dokumentieren und zu begründen.

a) Bestimmen Sie alle funktionalen Abhängigkeiten von S.

b) Bestimmen Sie alle Kandidatenschlüssel von S.

c) Zeigen Sie an einer geeigneten Instanz, dass in der Relation S eine Update-Anomalie auftreten kann.
Hinweis: Beachten Sie, dass Ihre Instanz alle Funktionalen Abhängigkeiten aus der Aufgabenstellung erfüllt.

d) Welcher Normalform genügt die Relation S (1NF, 2NF, 3NF, BCNF)? Begründen Sie Ihre Aussagen ausführlich.

Fortsetzung nächste Seite!

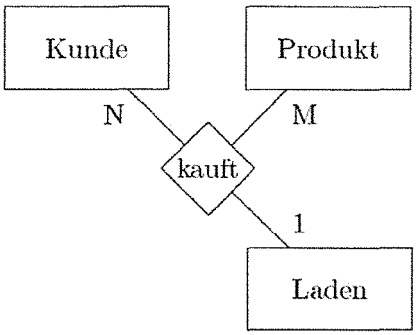
Teilaufgabe 2

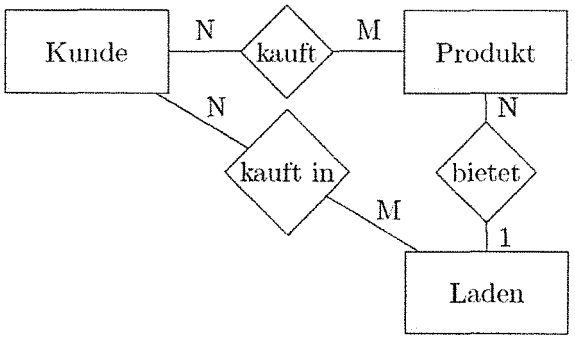
1. ER-Diagramme

1.1 Modellieren Sie den Aufbau eines Sportklubs mit einem ER-Diagramm. Kardinalitäten sind hierbei in der (min, max)-Notation anzugeben.

Personen unterteilen sich in Sportler und Betreuer. Personen haben einen Namen, eine Adresse, eine Telefonnummer sowie ein Geburtsdatum. Identifiziert werden sie über eine eindeutige Kennung. Zu den Sportlern ist ihr Gewicht sowie ihr Geburtsland bekannt, zu den Betreuern ihr Gehalt sowie eine Liste ihrer Führerscheinklassen.
Sportler können Mitglied eines oder mehrerer Teams sein, die über ihren Namen identifiziert werden. Jedes Team wird von mindestens einem Betreuer in mindestens einer Sportart betreut. Sportarten haben eine Kennung, einen Namen und eine Angabe zu den durchschnittlichen monatlichen Kosten. Sportler können mehrere Sportarten betreiben, wobei sie maximal eine Lieblingssportart haben.

1.2 Welche Aussagen lassen sich für folgende Modelle unter einer üblichen Interpretation treffen? Inwieweit unterscheiden bzw. gleichen sie sich? Es kommt hier vor allem auf die Gesamtaussage an. Formulieren Sie die Aussagen auch mit Hilfe partieller Funktionen.



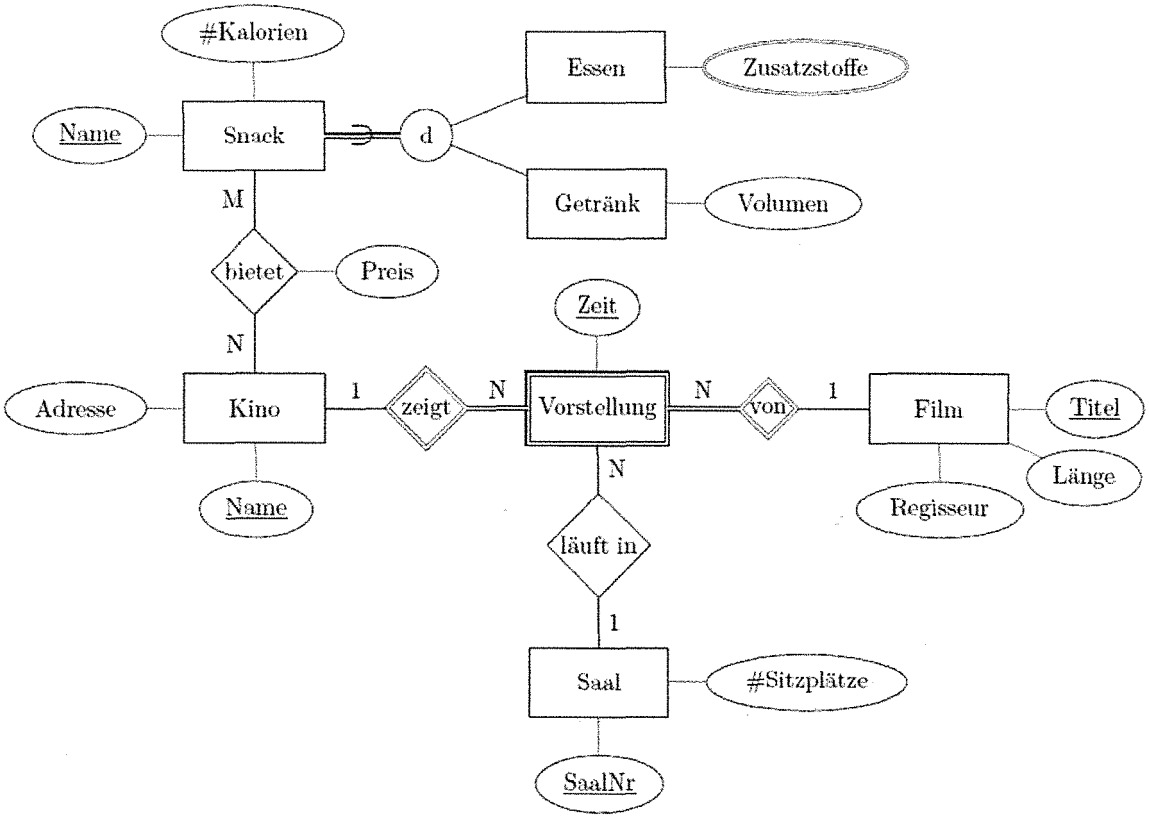


Bemerkung: Die Attribute wurden der Übersichtlichkeit wegen weggelassen.

Fortsetzung nächste Seite!

2. Relationales Modell

2.1 Überführen Sie das folgende ER-Modell in ein relationales Modell.



2.2 Macht es im ER-Modell von Aufgabe 2.1 einen Unterschied, zwischen Entitätstyp „Kino“ und Entitätstyp „Saal“ noch eine zusätzliche Beziehung einzuführen? Wenn ja, wo liegt der Unterschied? Wenn nein, warum nicht?

Fortsetzung nächste Seite!

3. Normalisierung

Gegeben ist eine Relation „Mitarbeiter" sowie alle für die Attribute der Relation geltenden funktionalen Abhängigkeiten.

3.1 Ist die Relation in erster Normalform? Ist sie in zweiter Normalform? Ist sie in dritter Normalform? Geben Sie jeweils eine Begründung für Ihre Antwort mit an.

Mitarbeiter(MID, Vorname, Nachname, Geburtsdatum, Straße, Hausnummer, Postleitzahl, Ort, PrivatVorwahl, PrivatTelefonnummer, Abteilungsnummer, Abteilungsname, Bereichsnummer, Bereichsname, Eintrittsdatum, Dienstalter, Gehaltsklasse, Gehalt)

Zusätzlich zur Schlüsseleigenschaft des Attributs „MID" (MID → Mitarbeiter) gelten noch weitere funktionalen Abhängigkeiten. Alle Abhängigkeiten sind in der Menge FD zusammengefasst:

FD={
MID → Vorname Nachname Geburtsdatum Straße Hausnummer Postleitzahl Ort PrivatVorwahl PrivatTelefonnummer Abteilungsnummer Abteilungsname Bereichsnummer Bereichsname Eintrittsdatum Dienstalter Gehaltsklasse Gehalt,
Postleitzahl → Ort,
Postleitzahl → PrivatVorwahl,
PrivatVorwahl → Ort,
Ort → Privatvorwahl,
Abteilungsnummer → Abteilungsname,
Abteilungsname → Abteilungsnummer,
Bereichsnummer → Bereichsname,
Bereichsname → Bereichsnummer,
Bereichsnummer → Abteilungsnummer,
Eintrittsdatum → Dienstalter,
Dienstalter → Eintrittsdatum,
Eintrittsdatum Gehaltsklasse → Gehalt
}

Bei der Bestimmung der kanonischen Überdeckung für die Menge FD muss eine Links- und eine Rechtsreduktion durchgeführt werden.

3.2 Warum führt eine Linksreduktion von FD zu keiner Veränderung?

Fortsetzung nächste Seite!

3.3 Welche Attribute auf der rechten Seite der funktionalen Abhängigkeit

`MID → Vorname Nachname Geburtsdatum Straße Hausnummer Postleitzahl Ort PrivatVorwahl PrivatTelefonnummer Abteilungsnummer Abteilungsname Bereichsnummer Bereichsname Eintrittsdatum Dienstalter Gehaltsklasse Gehalt`

sind redundant, wenn noch keine der anderen Abhängigkeiten entfernt wurden?

4. Relationale Algebra und SQL

Nachfolgend ist das Datenbankschema einer Schule gegeben. Verschiedene Anfragen in SQL sollen auf diesem Schema durchgeführt werden.

```
Lehrkraft (Personalnummer, Name, Geschlecht, Wohnort,  Geburtsjahr)
Schüler (Schülernummer, Name, Konfession, Klassennummer)
Fach (Fachname, Beschreibung)
unterrichtet (Personalnummer, Schülernummer, Fachname, Stundenzahl)
```

Bemerkung: Das Geschlecht einer Lehrkraft ist entweder 'm' oder 'w'.
In unterrichtet: Personalnummer ist Fremdschlüssel und referenziert den Primärschlüssel von Lehrkraft; Schülernummer ist Fremdschlüssel und referenziert den Primärschlüssel von Schüler; Fachname ist Fremdschlüssel und referenziert den Primärschlüssel von Fach.

4.1 Es soll eine Liste mit Schülernamen und den zugehörigen Klassen ausgegeben werden, wobei die Ausgabe absteigend nach Klassennummer und innerhalb der Klassen aufsteigend nach Name erfolgen soll.

4.2 Aus welchen Wohnorten stammen die Lehrerinnen und wie viele kommen jeweils aus einem Ort?

4.3 Wie lautet das durchschnittliche Geburtsjahr aller Lehrkräfte?

4.4 Zu welchem Fach gibt es momentan nur eine unterrichtende Lehrkraft?

4.5 Welche Lehrkräfte unterrichten mindestens zwei verschiedene Klassen im Fach Mathematik? Geben Sie Personalnummer und Name der entsprechenden Lehrkräfte aus.

Nachfolgende Anfrage soll in relationaler Algebra formuliert werden.

4.6 Geben Sie alle Schüler mit Nummer aus, die von Lehrkraft Maier unterrichtet werden.

5. Transaktionen

5.1 Welche vier grundlegenden Eigenschaften soll eine Datenbanktransaktion erfüllen? Nennen Sie diese vier Eigen-schaften und beschreiben Sie sie kurz.

5.2 Geben Sie zwei Fehler an, die im unkontrollierten (nicht synchronisierten) Mehrbenutzerbetrieb auftreten können und beschreiben Sie diese jeweils anhand eines Beispiels.

Themenschwerpunkt B
(Betriebssysteme)

Teilaufgabe 1

Aufgabe 1:
Speicherverwaltungstechniken

a) Bei der dynamischen Speicherzuteilung muss der freie Speicher verwaltet werden. Im Folgenden sind vier verschiedene Techniken aufgelistet, die in unterschiedlichen Speicherverwaltungsverfahren eingesetzt werden. Beschreiben Sie jeweils kurz diese Techniken, ihre Eigenschaften sowie ihre Vor- und Nachteile.

- Freispeicherverwaltung mit einer Bitliste
- Freispeicherverwaltung mit einer separat liegenden verketteten Liste
- Freispeicherverwaltung mit der Verkettung von freien Blöcken
- Freispeicherverwaltung mit dem Buddy-Verfahren

b) Bei der Suche nach freien Speicherbereichen innerhalb einer Freispeicherverwaltung, die mit Verkettung arbeitet, gibt es vier unterschiedliche Vergabestrategien. Benennen Sie diese Verfahren und beschreiben Sie ihre Arbeitsweise. Geben Sie in O-Notation an, wie aufwändig jeweils die Suche nach einem freien Block ausreichender Größe, das Einsortieren eines Blocks in die Freispeicherliste und das Verschmelzen benachbarter Freispeicherblöcke sind. Wie verhalten sich die unterschiedlichen Verfahren in Bezug auf die Fragmentierung des Speichers?

Aufgabe 2:
Scheduling

a) Geben Sie einen Überblick über drei Ihnen bekannte Schedulingstrategien.
Beschreiben Sie jeweils die Funktionsweise sowie die Eigenschaften (Vor-/Nachteile und sonstige Besonderheiten).

b) Welche Strategie eignet sich besonders gut für ein Mehrbenutzer-Betriebssystem, in dem gleichzeitig mehrere interaktive Anwendungen und sehr rechenzeitintensive Hintergrundaufgaben zu bearbeiten sind?
Die Gesamteffizienz der Strategie hängt sicherlich von der Wahl geeigneter Parameter ab. Welche Parameter sind bei der von Ihnen gewählten Strategie einstellbar und wie wirkt sich welche Art der Einstellung auf das Systemverhalten aus?

Fortsetzung nächste Seite!

Aufgabe 3:
Virtueller Speicher

a) Bei Virtuellem Speicher kann es zu Seitenfehlern (Page-Faults) kommen. In welchen Situationen tritt ein Seitenfehler auf und was muss das Betriebssystem jeweils zur Behebung tun (beschreiben Sie drei solche Situationen)?

b) Falls eine neue Seite allokiert werden soll, aber keine freie Seitenkachel im Hauptspeicher mehr verfügbar ist, muss eine Seite ausgelagert werden. Wie würde die optimale Seitenersetzungsstrategie funktionieren und warum ist sie in der Praxis nicht realisierbar?

c) Eine mögliche Approximation der optimalen Strategie wäre LRU. Beschreiben Sie die Funktionsweise von LRU. Wie könnte man LRU implementieren? Ist LRU eine in der Praxis gut oder problematisch implementierbare Strategie? Warum?

d) Clock (oder Second Chance) ist eine weitere Strategie, die eine möglichst optimale Seitenersetzung anstrebt. Beschreiben Sie die grundsätzliche Funktionsweise dieser Strategie. Hat sie gegenüber LRU Vor- bzw. Nachteile? Warum?

Fortsetzung nächste Seite!

Aufgabe 4:
Dateisysteme

a) Was versteht man unter einem Inode in einem Unix/Linux-Dateisystem? Geben Sie vier Beispiele für Informationen, die darin gespeichert sind.

b) Wie werden Dateiattribute und Dateiinhalt im Gegensatz zu Linux in einem Windows-NT-Dateisystem (NTFS) verwaltet?

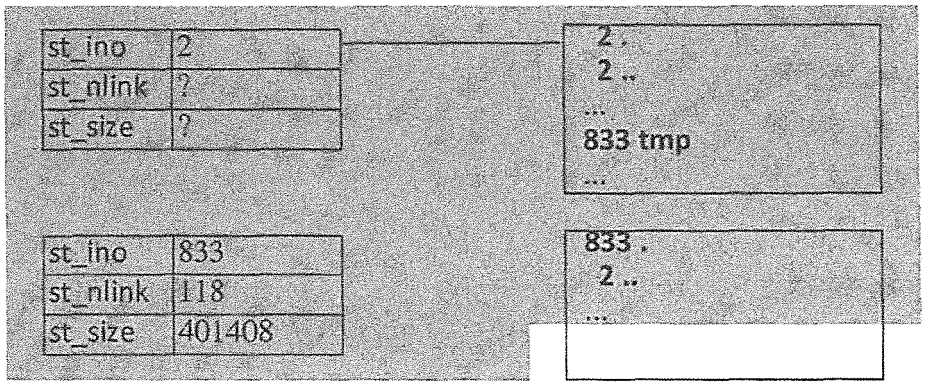
c) Gegeben ist die folgende Ausgabe des Kommandos `ls -alRi /tmp/sp` (rekursiv absteigende Ausgabe aller Dateien und Verzeichnisse unter `/tmp/sp` mit Angabe der Inode-Nummer) auf einem Linux-System:

```
fauixx> ls -alRi /tmp/sp
/tmp/sp:
total 408
 91 drwxrwxr-x  4 jklein i4staff  4096 Jul 19 15:17 .
833 drwxrwxrwt 118 root   root   401408 Jul 19 15:30 ..
 96 drwxrwxr-x  2 jklein i4staff  4096 Jul 19 15:21 dir1
 98 drwxrwxr-x  2 jklein i4staff  4096 Jul 19 15:22 dir2

/tmp/sp/dir1:
total 68
 96 drwxrwxr-x 2 jklein i4staff  4096 Jul 19 15:21 .
 91 drwxrwxr-x 4 jklein i4staff  4096 Jul 19 15:17 ..
536 -rwxr--r-- 2 jklein i4staff 52224 Jul 19 15:20 datei1
258 -rw-rw-r-- 1 jklein i4staff    30 Jul 19 15:21 datei2

/tmp/sp/dir2:
total 68
 98 drwxrwxr-x 2 jklein i4staff  4096 Jul 19 15:22 .
 91 drwxrwxr-x 4 jklein i4staff  4096 Jul 19 15:17 ..
896 lrwxrwxrwx 1 jklein i4staff    14 Jul 19 15:22 dat1 -> ../dir1/datei2
536 -rwxr--r-- 2 jklein i4staff 52224 Jul 19 15:20 xxx
900 -rw-rw-r-- 1 jens   i4staff    30 Jul 19 15:22 yyy
```

Ergänzen Sie im weißen Bereich die auf der folgenden Seite im grauen Bereich bereits angefangene Skizze der Inodes und Datenblöcke des Linux-Dateisystems um alle entsprechenden Informationen, die Sie aus obiger Ausgabe entnehmen können.

Fortsetzung nächste Seite!

Inodes **Inhalt der Datenblöcke**



st_ino	
st_nlink	
st_size	

st_ino	
st_nlink	
st_size	

st_ino	
st_nlink	
st_size	

st_ino	
st_nlink	
st_size	

st_ino	
st_nlink	
st_size	

st_ino	
st_nlink	
st_size	

st_ino	
st_nlink	
st_size	

Fortsetzung nächste Seite!

d) Beschreiben Sie kurz (in Stichworten) die grundsätzliche Funktionsweise eines *Journaling-File-Systems*.

Teilaufgabe 2

Aufgabe 1: Grundlagen zu Betriebssystemen

Beantworten Sie folgende allgemeine Fragen aus dem Bereich der Threads:

a) Erläutern Sie 2 wesentliche Unterschiede zwischen Threads und Prozessen.
b) In welchen 3 grundsätzlichen Zuständen kann sich ein Thread befinden?
c) Es gibt 2 wesentliche Arten von Threads. Nennen Sie beide Arten und geben Sie jeweils einen Vorteil an.
d) Was versteht man unter Multithreading?
e) Nennen Sie 3 Bestandteile, die jedem Thread innerhalb eines Prozesses zugeordnet werden.

Beantworten Sie folgende allgemeine Fragen aus dem Bereich der Zustands-Prozessmodelle:

a) Benennen Sie die 5 grundsätzlichen Zustände des 5-Zustands-Prozessmodells.
b) Das 7-Zustandsmodell ist eine Erweiterung des 5-Zustandsmodells. Nennen Sie nur die zusätzlichen Zustände im Vergleich zum 5-Zustandsmodell und geben Sie jeweils den Grund dafür an, warum ein solcher zusätzlicher Zustand in einem Betriebssystem gebraucht wird.
c) Welcher Zustandsübergang wird durch den Dispatcher realisiert?
d) Müsste man das Zustands-Prozessmodell anpassen, wenn man es auf eine Multiprozessorumgebung anwenden will? Begründen Sie kurz Ihre Antwort.

Fortsetzung nächste Seite!

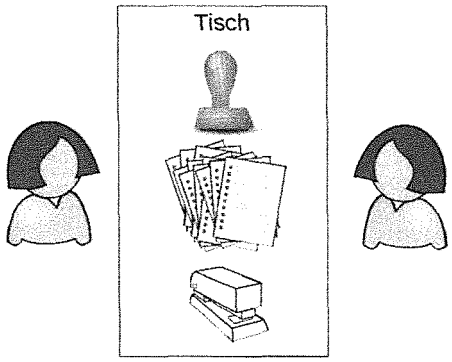
Aufgabe 2: Petrinetz und Erreichbarkeitsgraphen

Gegeben Sei folgendes Szenario, welches als Petrinetz modelliert werden soll:

2 Sekretärinnen eines Instituts müssen einen riesigen Stapel an Unterlagen bearbeiten. Dazu benötigen sie gleichzeitig 2 Gegenstände, um zusammengehörende Unterlagen aus dem Gesamtstapel zu verarbeiten. Die beiden Gegenstände sind:

- Der offizielle Institutsstempel
- Der Tacker

Beide Sekretärinnen sitzen zusammen an einem Tisch gegenüber. Auf dem Tisch liegt in der Mitte der Stapel an Unterlagen, sowie der Stempel und der Tacker, wie in folgender Abbildung gezeigt:



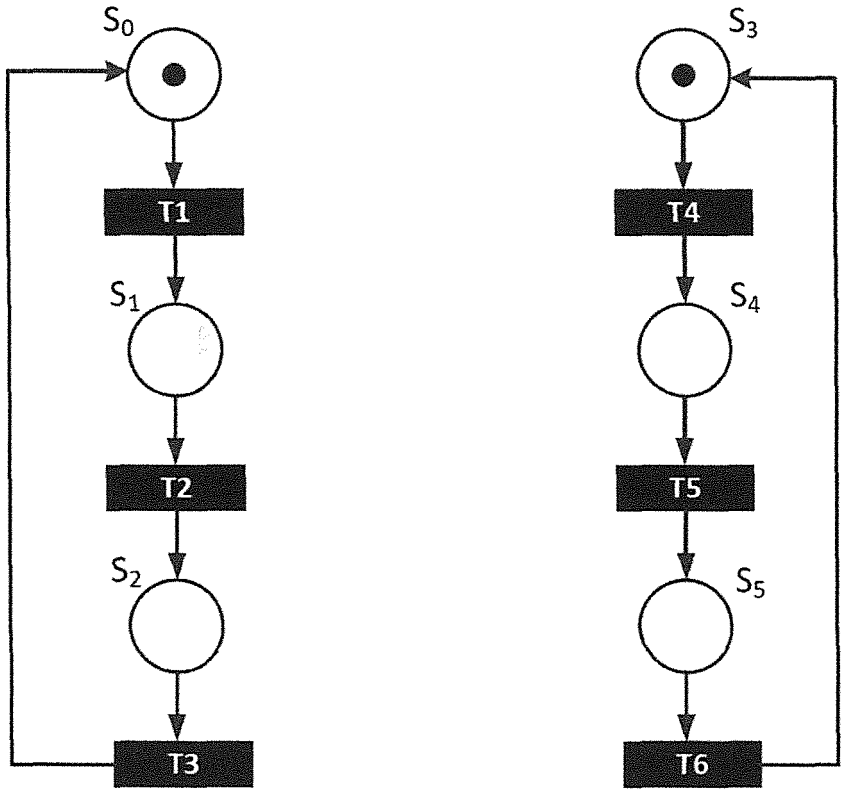
In diesem Szenario sind beide Sekretärinnen Rechtshänder. So greift jede zuerst nach dem Gegenstand, der rechts von ihr liegt, und dann zu dem anderen Gegenstand. Die eine greift somit immer zuerst nach dem Tacker und dann zum Stempel und die andere genau anders herum. Beachten Sie bitte, dass bei diesem Szenario eine Sekretärin immer beide Gegenstände gleichzeitig benutzen muss, damit sie einen Teil der Unterlagen verarbeiten kann. Nachdem eine Sekretärin die zusammengehörenden Unterlagen verarbeitet hat (getackert und gestempelt), legt sie beide Gegenstände wieder genau so auf den Tisch wie zuvor und ruht einen kurzen Moment. Eine Sekretärin hat somit drei Zustände:

- Ruhen
- Hat einen Gegenstand
- Verarbeitet Unterlagen.

Fortsetzung nächste Seite!

Bearbeiten Sie zu diesem Szenario die folgenden Teilaufgaben:

a) Übertragen Sie das folgende Petrinetz auf Ihr Blatt und ergänzen Sie es durch Einfügen einer minimalen Anzahl von Stellen, Transitionen, Kanten oder Marken, so dass das oben beschriebene Szenario korrekt modelliert wird. Beschriften Sie außerdem alle neu eingezeichneten Stellen bzw. Transitionen. Geben Sie zusätzlich für alle Stellen und Transitionen in dem kompletten Petrinetz eine kurze Beschreibung an:



b) Erstellen Sie den Erreichbarkeitsgraphen zu dem modellierten Petrinetz. Geben Sie hierbei auch an, wie in den Markierungen des Erreichbarkeitsgraphen die Stellen angeordnet sind. Beschriften Sie alle Übergänge zwischen Markierungen mit der Bezeichnung der Transition, die hierfür feuern muss.

c) Begründen Sie mittels des Erreichbarkeitsgraphen jeweils separat, ob in dem angegebenen Szenario ein Deadlock bzw. ein partieller Deadlock auftreten kann.

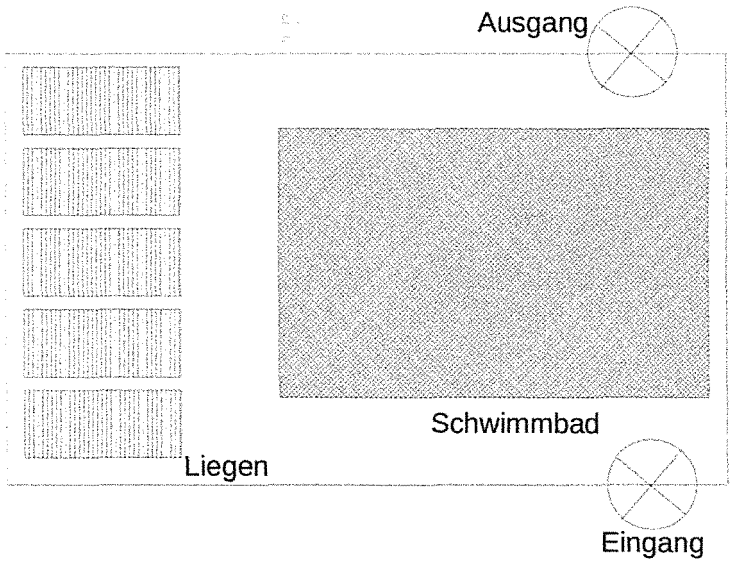
Fortsetzung nächste Seite!

Aufgabe 3: Koordination von Threads

In dieser Aufgabe sollen Sie Badegäste eines kleinen Schwimmbads synchronisieren. Die Badegäste sollen dabei als Java-Threads realisiert werden. Betrachten Sie dazu folgendes Szenario:

- Das kleine Schwimmbad besitzt maximal *maxLiegen*, die von den Badegästen genutzt werden können.
- Die Badegäste können das Schwimmbad über ein Drehkreuz betreten und über ein anderes Drehkreuz verlassen.
- Zu jedem Zeitpunkt kann immer nur eine Person durch das jeweilige Drehkreuz gehen.
- Es dürfen sich stets auch nur maximal so viele Personen im Schwimmbad befinden (inklusive der beiden Drehtüren), wie Liegen vorhanden sind.

Die nachfolgende Abbildung zeigt den schematischen Aufbau des beschriebenen Szenarios für *maxLiegen=5* und *anzahlGaeste=0:*



Übertragen Sie den folgenden Coderahmen der Klasse *Schwimmbad* und vervollständigen Sie ihn. Die Beispielimplementierungen der Klassen *Badegast* und *Simulation* sollen Ihnen verdeutlichen, wie die Klasse *Schwimmbad* verwendet werden kann.

a) Überlegen Sie sich die notwendigen Attribute der Klasse *Schwimmbad* und implementieren Sie den Konstruktor der Klasse an der entsprechenden Stelle, die im Coderahmen dafür vorgesehen ist. Kommentieren Sie Ihre Lösung ausführlich.

Fortsetzung nächste Seite!

b) Implementieren Sie die Methode *betreten()*, welche das Passieren des Eingang-Drehkreuzes modelliert, sowie die Methode *verlassen()*, welche im Gegenzug das Verlassen des Schwimmbads über das Ausgangs-Drehkreuzes modelliert. Kommentieren Sie Ihre Lösung wieder ausführlich. Beachten Sie bei der Implementierung, dass alle oben genannten Bedingungen eingehalten werden.

c) Was sind die kritischen Bereiche bei diesem Problem?

d) Erläutern Sie kurz (2 Sätze), warum Ihre Lösung des wechselseitigen Ausschlusses die Bedingung der Mutual Exclusion erfüllt.

e) Beschreiben Sie ausgehend von Java die Einschränkungen bei der Verwendung von *wait()*, *notify()* und *notifyAll()* gegenüber dem Monitorkonzept von C. Hoare (1974). **Hinweis**: Überlegen Sie sich dazu, wie im Falle des Monitorkonzepts von Hoare bzw. im Falle des Java Synchronisierungsmechanismus der nächste aktive Thread selektiert wird.

Fortsetzung nächste Seite!

```
public class Schwimmbad {
   private static int maxLiegen;
   private int anzahlGaeste;

   //Schreiben Sie hier den Konstruktor:

   // Implementierung der Methode "betreten":

   // Implementierung der Methode "verlassen":

}
```

Fortsetzung nächste Seite!

Die Klasse `Simulation`:

```
1  public class Simulation {
2      private Schwimmbad my_schwimmbad;
3
4      public Simulation() {
5          my_schwimmbad = new Schwimmbad(5);
6
7          for(int i=0; i< 50; i++)
8          {
9                  Badegast gast = new Badegast(my_schwimmbad);
10                 gast.start();
11         }
12     }
13     public static void main(String[] args) {
14         new Simulation();
15     }
16 }
```

Die Klasse `Badegast`:

```
1  import java.util.Random;
2
3  public class Badegast extends Thread {
4      private Schwimmbad my_schwimmbad;
5      private Random generator;
6
7      public Badegast(Schwimmbad my_schwimmbad) {
8          this.my_schwimmbad = my_schwimmbad;
9          generator = new Random();
10     }
11
12     public void run() {
13             try {
14                 Thread.sleep(generator.nextInt(2000) + 500);
15                 my_schwimmbad.betreten();
16                 //Liege belegen, schwimmen usw.
17                 Thread.sleep(generator.nextInt(2000) + 500);
18                 my_schwimmbad.verlassen();
19             }
20             catch(InterruptedException ie) {}
21     }
22 }
```

66114

Datenbank- und Betriebssysteme (vertieft)

Herbst 2016

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____	**Herbst 2016**	**66114**
Kennwort: _____		
Arbeitsplatz-Nr.: _____		

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

— Prüfungsaufgaben —

Fach: **Informatik (vertieft studiert)**

Einzelprüfung: **Datenbank- und Betriebssysteme**

Anzahl der gestellten Themen (Aufgaben): **4 Aufgaben, von denen zwei gemäß untenstehender Auswahlregel zu bearbeiten sind!**

Anzahl der Druckseiten dieser Vorlage: **26**

Zu den zwei Themenschwerpunkten A (Datenbanksysteme) und B (Betriebssysteme) ist jeweils entweder die Teilaufgabe 1 oder 2 zu wählen!
Auf der Vorderseite des Kopfbogens sind im Feld „Gewähltes Thema: Nr." die Nummern der beiden ausgewählten Teilaufgaben anzugeben (z. B. A2, B1)!

Bitte wenden!

Themenschwerpunkt A
(Datenbanksysteme)

Teilaufgabe 1

1. ER-Modellierung

Gegeben seien folgende Informationen:

- Krankenhäuser bestehen aus Stationen. Zu jedem Krankenhaus ist dessen Adresse (wodurch es identifiziert werden kann) und die Anzahl an Betten bekannt. Stationen besitzen einen Namen, der nur innerhalb eines Krankenhauses eindeutig ist.
- Stationen bestehen wiederum aus Zimmern, welche nummeriert sind. Eine solche Nummer ist innerhalb einer Station eindeutig.
- Jedes Krankenhaus beschäftigt Personal. Dabei wird festgehalten, welches Gehalt bezahlt wird. Personal kann in verschiedenen Krankenhäusern beschäftigt sein.
- Personal ist durch eine Personal-Nummer gekennzeichnet und kann unter anderem in Ärzte und Krankenpfleger unterteilt werden. Zu einem Arzt ist sein Fachbereich bekannt und von welchen Krankenpflegern er Vorgesetzter ist. Kein Krankenpfleger kann zugleich Arzt sein, aber durchaus mehrere vorgesetzte Ärzte haben.
- Eine Station kann von mehreren Ärzten gleitet werden. Ein Arzt kann ebenso mehrere Stationen leiten. Außerdem ist bekannt, ob und in welchem Zimmer ein Arzt sein Büro hat. Kein Arzt muss sich sein Büro mit einem anderen Arzt teilen.
- Personal arbeitet auf Stationen in Schichten. Eine Schicht kann über Datum und Zeitraum eindeutig identifiziert werden. Eine Person kann in einer Schicht auf nur einer Station arbeiten.

a) Erstellen Sie für das oben gegebene Szenario ein geeignetes ER-Diagramm.
Verwenden Sie dabei – wenn angebracht – das Prinzip der Spezialisierung. Kennzeichnen Sie die Primärschlüssel der Entity-Typen, totale Teilnahmen und schwache Entity-Typen.
Zeichnen Sie die Funktionalitäten der Relationship-Typen in das Diagramm ein.

b) Überführen Sie Ihr in Aufgabe a) erstelltes Modell in ein verfeinertes relationales Schema.
Kennzeichnen Sie die Schlüssel durch Unterstreichen. Datentypen müssen nicht angegeben werden.

Fortsetzung nächste Seite!

2. SQL und relationale Algebra

Gegeben sei der folgende Ausschnitt aus dem Schema einer Schulverwaltung:

Person : {[
  ID : INTEGER,
  Name : VARCHAR(255),
  Wohnort : VARCHAR(255),
  Typ : CHAR(1)
]}

Unterricht : {[
  Klassenbezeichnung : VARCHAR(20),
  Schuljahr : INTEGER,
  Lehrer : INTEGER,
  Fach : VARCHAR(100)
]}

Klasse : {[
  Klassenbezeichnung : VARCHAR(20),
  Schuljahr : INTEGER,
  Klassenlehrer : INTEGER
]}

Klassenverband : {[
  Schüler : INTEGER,
  Klassenbezeichnung : VARCHAR(20),
  Schuljahr : INTEGER
]}

Hierbei enthält die Tabelle *Person* Informationen über Lehrer (Typ 'L') und Schüler (Typ 'S'); andere Werte für Typ sind nicht zulässig. *Klasse* beschreibt die Klassen, die in jedem Schuljahr gebildet wurden, zusammen mit ihrem Klassenlehrer. In *Unterricht* wird abgelegt, welcher Lehrer welches Fach in welcher Klasse unterrichtet; es ist möglich, dass derselbe Lehrer mehr als ein Fach in einer Klasse unterrichtet. *Klassenverband* beschreibt die Zuordnung der Schüler zu den Klassen.

a) Schreiben Sie eine SQL-Anweisung, die die Tabelle *Unterricht* mit allen ihren Constraints (einschließlich Fremdschlüsselconstraints) anlegt.

b) Definieren Sie ein geeignetes Constraint, das sicherstellt, dass nur zulässige Werte im Attribut Typ der (bereits angelegten) Tabelle *Person* eingefügt werden können.

c) Schreiben Sie eine SQL-Anweisung, die die Bezeichnung der Klassen bestimmt, die im Schuljahr 2015 die meisten Schüler haben.

d) Schreiben Sie eine SQL-Anweisung, die die Namen aller Lehrer bestimmt, die nur Schüler aus ihrem Wohnort unterrichtet haben.

e) Schreiben Sie eine SQL-Anweisung, die die Namen aller Schüler bestimmt, die immer den gleichen Klassenlehrer hatten.

f) Schreiben Sie eine SQL-Anweisung, die alle Paare von Schülern bestimmt, die mindestens einmal in der gleichen Klasse waren. Es genügt dabei, wenn Sie die ID der Schüler bestimmen.

g) Formulieren Sie eine Anfrage in der relationalen Algebra, die die ID aller Schüler bestimmt, die mindestens einmal von 'Ludwig Lehrer' unterrichtet wurden.

h) Formulieren Sie eine Anfrage in der relationalen Algebra, die Namen und ID der Schüler bestimmt, die von allen Lehrern unterrichtet wurden.

Beachten Sie bei der Formulierung der SQL-Anfragen, dass die Ergebnisrelationen keine Duplikate enthalten dürfen. Sie dürfen geeignete Views definieren.

Fortsetzung nächste Seite!

3. Entwurfstheorie

Gegeben sei folgendes relationale Schema R in erster Normalform:

$$R : \{[A,B,C,D,E,F]\}$$

Für R gelte folgende Menge FD funktionaler Abhängigkeiten:
$FD = \{$

$$\begin{aligned} AC &\rightarrow B, \\ DEF &\rightarrow BC, \\ F &\rightarrow AB, \\ D &\rightarrow F, \\ BC &\rightarrow E \end{aligned}$$

$\}$

a) Bestimmen Sie alle Kandidatenschlüssel von R mit FD.
Hinweis: Die Angabe von Attributmengen, die keine Kandidatenschlüssel sind, führt zu Abzügen.

b) Prüfen Sie, ob R mit FD in 2NF bzw. 3NF ist.

c) Bestimmen Sie mit folgenden Schritten eine kanonische Überdeckung FD_C von FD:
   i. Führen Sie eine Linksreduktion von FD durch. Geben Sie die Menge funktionaler Abhängigkeiten nach der Linksreduktion an (FD_L).

   ii. Führen Sie eine Rechtsreduktion des Ergebnisses der Linksreduktion (FD_L) durch. Geben Sie die Menge funktionaler Abhängigkeiten nach der Rechtsreduktion an (FD_R).

   iii. Bestimmen Sie eine kanonische Überdeckung FD_C von FD auf Basis des Ergebnisses der Rechtsreduktion (FD_R).

d) Zerlegen Sie R mit FD_C mithilfe des *Synthesealgorithmus* in 3NF.

e) Prüfen Sie *für alle Relationen* der Zerlegung aus d), ob sie jeweils in BCNF sind.

Fortsetzung nächste Seite!

4. Transaktionen

a) Erläutern Sie kurz die Konzepte *Atomarität* und *Isolation.*
b) Betrachten Sie den folgenden Schedule S:

T_1	T_2	T_3
$r_1(x)$		
	$r_2(x)$	
	$r_2(y)$	
	$w_2(x)$	
	c_2	
$r_1(y)$		
		$r_3(y)$
		$w_3(y)$
		c_3
$w_1(y)$		
c_1		

Geben Sie den Ausgabeschedule (einschließlich der Operationen zur Sperranforderung und -freigabe) im strikten Zweiphasen-Sperrprotokoll für den obigen Eingabeschedule S an.
Hinweis: Sollte während der Ausführung ein Deadlock auftreten, führen Sie ein Rollback einer der beteiligten Transaktionen durch.

Fortsetzung nächste Seite!

Teilaufgabe 2

1. Konzeptioneller Entwurf

Im Folgenden ist die Beschreibung einer Fotoverwaltung gegeben. Erstellen Sie zu dieser ein ER-Diagramm. Kennzeichnen Sie die Primärschlüssel durch passendes Unterstreichen und geben Sie die Kardinalitäten in Min-Max-Notation an. Modellieren Sie keine Attribute oder Entitytypen, die nicht aus dem Text hervorgehen.
Es werden Fotos verwaltet, die über eine eindeutige ID identifiziert werden und einen Aufnahmezeitpunkt sowie eine Beschreibung besitzen. Den Fotos können Stichworte zugeordnet werden, die einen eindeutigen Namen haben. Stichworte sind hierarchisch organisiert, d.h. ein Stichwort kann maximal einem anderen Stichwort untergeordnet sein. Jedes Foto wurde von genau einem Fotografen aufgenommen und es können Models auf ihm abgebildet sein. Zu Fotografen und Models werden Name, Vorname, Postleitzahl und Wohnort gespeichert. Zu Fotografen wird zusätzlich ihre Mitgliedsnummer im Verband deutscher Fotografen abgelegt, durch die sie eindeutig identifiziert werden können. Models gehören genau einer Agentur an und haben eine Mitgliedsnummer, die innerhalb ihrer Agentur eindeutig ist. Agenturen haben einen eindeutigen Namen sowie eine E-Mail-Adresse. Ein Fotograf kann auch gleichzeitig Model sein. Models können der Veröffentlichung von bestimmten Fotos in bestimmten Medien zustimmen. Zu jedem Medium werden ein eindeutiger Name sowie eine Beschreibung abgelegt. Zu einer Zustimmung wird das Datum gespeichert, an dem sie erfolgt ist.

Fortsetzung nächste Seite!

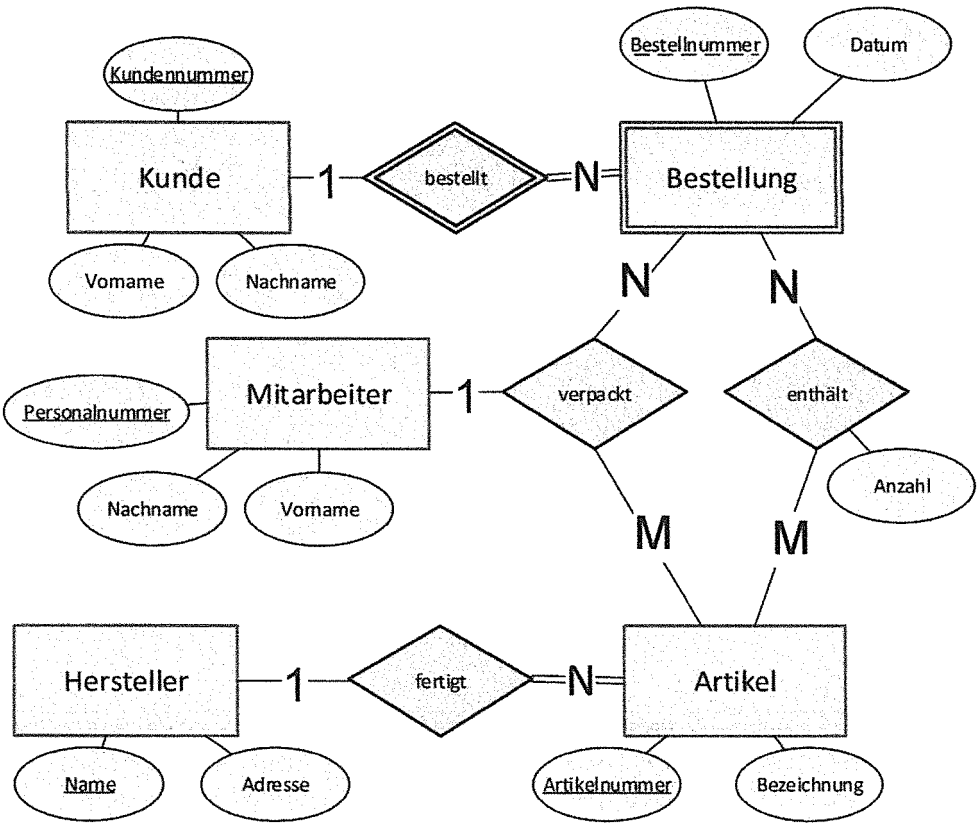
2. Relationenschema

Entwerfen Sie zum untenstehenden ER-Diagramm ein Relationenschema in dritter Normalform (3 NF) mit möglichst wenigen Relationen.

Verwenden Sie dabei folgende Notation:

Primärschlüssel werden durch Unterstreichen gekennzeichnet, Fremdschlüssel durch die Nennung der Relation, auf die sie verweisen, in eckigen Klammern hinter dem Fremdschlüsselattribut. Attribute zusammengesetzter Fremdschlüssel werden durch runde Klammern als zusammengehörig markiert.

Beispiel: Relation1 (Primärschlüssel, Attribut1, Attribut2, Fremdschlüssel[Relation2])



Fortsetzung nächste Seite!

3. Normalformen

a) Gegeben ist eine vollständige Extension der Relation Bestellungen mit dem zusammengesetzten Primärschlüssel {Kundennummer, Bestellnummer}:

Kundennummer	**Bestellnummer**	Kundennachname	Bestelldatum	Lieferdatum
1	1657	Frank	13.02.2015	15.02.2015
2	1306	Weizenbaum	14.01.2015	18.01.2015
3	2369	Weizenbaum	09.08.2014	10.01.2015
2	9847	Weizenbaum	21.11.2013	25.11.2013
5	2569	Schmitt	05.09.2015	05.09.2015
7	3457	Nürnberger	07.10.2012	12.12.2012
8	4468	Haberkorn	31.08.2015	01.10.2015

Geben Sie an, in welcher Normalform sich die Relation Bestellungen befindet. Begründen Sie weiterhin in zwei bis drei Sätzen, dass alle Bedingungen dieser Normalform erfüllt sind und dass nicht alle Bedingungen der nächsthöheren Normalform erfüllt sind.

b) Nennen Sie die Bezeichnungen der drei Anomalien, zu denen nicht normalisierte Relationenschemata führen können, und erläutern Sie die Anomalien in jeweils ein bis zwei Sätzen. Sie können dazu Beispiele aus der in der vorhergehenden Teilaufgabe gegebenen Relation nutzen.

Fortsetzung nächste Seite!

4. SQL

Gegeben sind folgende Relationen aus einer Personalverwaltung:

```
Mitarbeiter (MitarbeiterID, Vorname, Nachname, Vorgesetzter[Mitarbeiter],
  AbteilungsID[Abteilung], Telefonnummer, Gehalt)
Abteilung (AbteilungsID, Bezeichnung)
```

a) Schreiben Sie eine SQL-Anfrage, die Vor- und Nachnamen der Mitarbeiter aller Abteilungen mit Bezeichnung „Buchhaltung" ausgibt, absteigend sortiert nach MitarbeiterID.

b) Schreiben Sie eine SQL-Anfrage, die die Nachnamen aller Mitarbeiter zusammen mit dem Nachnamen ihres jeweiligen direkten Vorgesetzten ausgibt. Mitarbeiter ohne Vorgesetzten sollen in der Ausgabe ebenfalls enthalten sein. In diesem Fall soll der Nachname des Vorgesetzten NULL sein.

c) Schreiben Sie eine SQL-Anfrage, die die 10 Abteilungen ausgibt, deren Mitarbeiter das höchste Durchschnittsgehalt haben. Ausgegeben werden sollen der Rang (1 = höchstes Durchschnittsgehalt bis 10 = niedrigstes Durchschnittsgehalt), die Bezeichnung sowie das Durchschnittsgehalt der Abteilung.
Gehen Sie davon aus, dass es keine zwei Abteilungen mit gleichem Durchschnittsgehalt gibt. Sie können der Übersichtlichkeit halber Views oder With-Anweisungen verwenden. Verwenden Sie jedoch **keine** datenbanksystemspezifischen Erweiterungen wie limit oder rownum.

d) Schreiben Sie eine SQL-Anfrage, die das Gehalt aller Mitarbeiter aus der Abteilung mit AbteilungsID 42 um 5% erhöht.

e) Alle Abteilungen mit der Bezeichnung „Qualitätskontrolle" sollen zusammen mit den Datensätzen ihrer Mitarbeiter gelöscht werden. ON DELETE CASCADE ist für keine der Tabellen gesetzt. Schreiben Sie die zum Löschen notwendigen SQL-Anfragen.

f) Alle Mitarbeiter sollen mit SQL-Anfragen nach den Telefonnummern anderer Mitarbeiter suchen können. Sie dürfen jedoch das Gehalt der Mitarbeiter nicht sehen können. Erläutern Sie in zwei bis drei Sätzen eine Möglichkeit, wie dies in einem Datenbanksystem realisiert werden kann, ohne die gegebenen Relationen, die als Tabellen abgelegt sind, zu verändern. Sie brauchen hierzu keinen SQL-Code schreiben.

Fortsetzung nächste Seite!

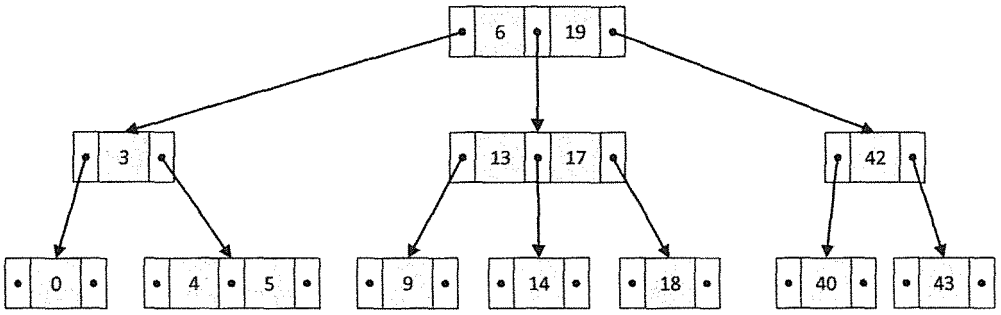
5. Physische Datenorganisation

a) Erläutern Sie die wesentliche Eigenschaft eines Tupel-Identifikators (TID) in ein bis zwei Sätzen.

b) Fügen Sie in einen anfangs leeren B-Baum mit k = 1 (maximal 2 Schlüsselwerte pro Knoten) die im Folgenden gegebenen Schlüsselwerte der Reihe nach ein. Zeichnen Sie den Endzustand des Baums nach jedem Einfügevorgang. Falls Sie Zwischenschritte zeichnen, kennzeichnen Sie die sieben Endzustände deutlich.

3, 7, 13, 11, 9, 10, 8

c) Gegeben ist der folgende B-Baum:



Die folgenden Teilaufgaben sind voneinander unabhängig.

i. Löschen Sie aus dem gegebenen B-Baum den Schlüssel 3 und zeichnen Sie den Endzustand des Baums nach dem Löschvorgang. Falls Sie Zwischenschritte zeichnen, kennzeichnen Sie den Endzustand deutlich.

ii. Löschen Sie aus dem (originalen) gegebenen B-Baum den Schlüssel 17 und zeichnen Sie den Endzustand des Baums nach dem Löschvorgang. Falls Sie Zwischenschritte zeichnen, kennzeichnen Sie den Endzustand deutlich.

iii. Löschen Sie aus dem (originalen) gegebenen B-Baum den Schlüssel 43 und zeichnen Sie den Endzustand des Baums nach dem Löschvorgang. Falls Sie Zwischenschritte zeichnen, kennzeichnen Sie den Endzustand deutlich.

6. Transaktionen

a) Nennen Sie in einem Satz den Unterschied zwischen dem strengen Zwei-Phasen-Sperrprotokoll und dem Zwei-Phasen-Sperrprotokoll.

b) Beim unkontrollierten Mehrbenutzerbetrieb kann es zum Problem der verlorengegangenen Änderungen (lost update) kommen. Erläutern Sie anhand eines Beispiels in drei bis vier Sätzen, wie dieses Problem auftreten kann und welche Folgen sich daraus ergeben.

Themenschwerpunkt B
(Betriebssysteme)

Teilaufgabe 1

Aufgabe 1:

a) Was versteht man unter einem Prozess und was ist der Unterschied zu einem Programm?

b) Das Betriebssystem muss in Zusammenhang mit einem Prozess Ressourcen verwalten. Welche Ressourcen sind minimal erforderlich, um einen Schutz gegenüber Operationen von anderen Prozessen (hierbei geht es nicht um den Schutz gegenüber anderen Benutzern) zu gewährleisten und Operationen auf Dateien zu ermöglichen?

c) Ressourcen und weitere Daten zu einem Prozess werden vom Betriebssystem in der Prozesstabelle verwaltet. Nennen Sie 5 weitere typische Einträge und erläutern Sie, zu welchem Zweck diese Daten vom Betriebssystem benötigt werden.

d) Erläutern Sie den Unterschied zwischen den Konzepten *Prozess* und *Thread*. Nennen Sie die unterschiedlichen Arten von Threads und erläutern Sie die Unterschiede und Vor- und Nachteile der Thread-Arten.

e) Welche Operationen bietet die Systemschnittstelle von UNIX-/Linux-Systemen zur Erzeugung von Prozessen und zum Laden von Programmen? Beschreiben Sie die Funktionsweise der einzelnen Operationen und was im Betriebssystem dabei jeweils abläuft.

f) Skizzieren Sie, wie eine Shell in einem UNIX- oder Linux-System funktioniert (vom Einlesen eines Kommandos über das Ausführen bis zur Ausgabe des nächsten Prompt-Symbols). Wie geht die Shell in diesem Zusammenhang vor, um Kommandos im Vordergrund bzw. Hintergrund auszuführen?

   Was tut die Shell dabei, um z.B. die Standardeingabe des auszuführenden Kommandos auf eine Datei umzuleiten (wie bei `Kommando <Datei1`)?

Fortsetzung nächste Seite!

Aufgabe 2:

a) Was versteht man in Betriebssystemen unter einem Semaphor? Beschreiben Sie die elementaren Operationen auf Semaphoren im Detail.

b) Man kann Semaphore sowohl für den gegenseitigen Ausschluss als auch zur Synchronisierung von mehreren Abläufen (einen Ablauf vor Ausführung der Operation B verzögern, bis ein anderer Ablauf einen bestimmten Punkt - d. h. Operation A wurde ausgeführt - erreicht hat) einsetzen. Beschreiben Sie jeweils eine typische Situation von solchen Anwendungsfällen und skizzieren Sie den Einsatz von Semaphoren (unter Angabe der Semaphor-Werte und -Operationen) in diesen Fällen in einer programmiersprachenähnlichen Form.

c) Was versteht man unter einem verdrängenden Scheduling-Verfahren mit dynamischen Prioritäten?

d) Beim Einsatz von Semaphoren bei mehreren Prozessen und verdrängendem Scheduling mit dynamischen Prioritäten kann es zu dem Phänomen der Prioritätsinversion kommen. Was versteht man darunter? Skizzieren Sie in einem Diagramm mit drei Prozessen eine Situation, in der das Phänomen auftritt. Was könnte man zur Vermeidung des Phänomens tun?

Fortsetzung nächste Seite!

Aufgabe 3:

a) Die Dateisysteme von UNIX-/Linux-Systemen (z. B. EXT2 oder EXT3) und Windows (z. B. NTFS) weisen eine Reihe von gemeinsamen Konzepten auf, unterscheiden sich aber auch in vielen Details. Ein gemeinsames Konzept sind die hierarchisch aufbaubaren Dateikataloge. Beschreiben Sie die Funktion eines Dateikatalogs (*Directory*), wie eine baumförmige Hierarchie aufgebaut werden kann und wie schließlich in solch einem System Dateien abgelegt, benannt und anhand des Namens gefunden werden können.

b) Die Verwaltungsdatenstruktur zu einer Datei in UNIX/Linux ist der sog. *Inode*, das Pendant in Windows-NTFS ist ein Eintrag in der Master-File-Table (MFT).
Vergleichen Sie die Konzepte *Inode* und *MFT-Eintrag*. Welche Daten werden darin typischerweise gespeichert? (Nennen Sie 5 Beispiele und erläutern Sie, wozu diese Daten jeweils benötigt werden). Beschreiben Sie zwei Vorteile des NTFS-Konzepts. Was versteht man in diesem Zusammenhang unter dem Konzept eines Datenstroms (*Streams*)?

c) Wie funktioniert die Verwaltung der Datenblöcke einer Datei in einem FAT-Dateisystem? Geben Sie die Belegung der FAT für drei Dateien an, wobei
Datei 1 in den Blöcken 8, 1, 5, 9;
Datei 2 in den Blöcken 12, 2, 11, 4, 7;
Datei 3 in den Blöcken 3, 6, 10;
gespeichert ist.

d) Moderne Dateisysteme werden typischerweise als Log-basierte (oder *Journaling*) Dateisysteme realisiert. Beschreiben Sie die grundlegende Funktionsweise eines solchen Dateisystems und welche Vorteile es mit sich bringt.

Fortsetzung nächste Seite!

Teilaufgabe 2

Aufgabe 1: Prozesse

1. Erläutern Sie kurz den Unterschied zwischen Multiprogramming und Multiprocessing.
2. Worin unterscheiden sich die Aufgaben des Schedulers und des Dispatchers?
3. Erläutern Sie kurz, ob der Einsatz des 5-Zustands-Prozess-Modells mit einem nicht-präemptiven Scheduling-Verfahren empfehlenswert ist.
4. In den folgenden Fragen soll das 7-Zustands-Prozessmodell betrachtet werden.
   a) Skizzieren Sie das 7-Zustands-Prozessmodell. Kennzeichnen Sie alle Übergänge durch Pfeile.
   Beschriften Sie alle Pfeile mit der Aktion, die bei dem entsprechenden Übergang ausgeführt wird.
   b) Welche zusätzlichen Möglichkeiten ergeben sich im 7-Zustands-Prozessmodell, die im 5-Zustands-Prozessmodell nicht vorhanden sind? Geben Sie für jeden der zusätzlichen Zustandsübergänge ein Beispiel an.

Fortsetzung nächste Seite!

Aufgabe 2: Scheduling-Strategien

In dieser Aufgabe sollen drei Scheduling-Strategien untersucht werden: die nicht-präemptive Strategie SJF (Shortest Job First), die präemptive Strategie SRPT (Shortest Remaining Processing Time) und die präemptive Strategie RR (Round Robin). Dazu seien die folgenden Prozesse mit ihren Ankunftszeitpunkten und Rechenzeiten (in beliebigen Zeiteinheiten) gegeben.

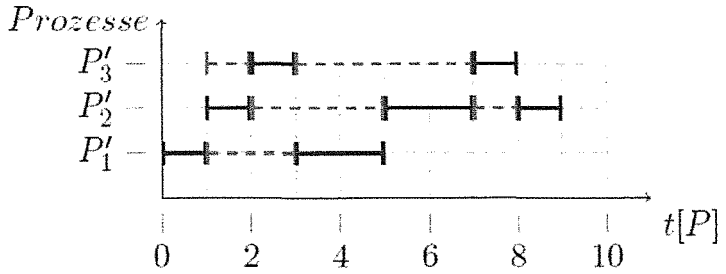
Prozess	Ankunftszeitpunkt	Rechenzeit
P_1	0	2
P_2	0	2
P_3	2	2
P_4	4	3
P_5	5	4
P_6	6	3
P_7	8	1

Gehen Sie davon aus, dass jeder Prozess sein Zeitquantum stets vollständig ausnutzt d.h. kein Prozess gibt den Prozessor freiwillig frei (Ausnahme: bei Prozessende).
Trifft ein Prozess zum Zeitpunkt t ein, so wird er direkt zum Zeitpunkt t beim Scheduling berücksichtigt.
Wird ein Prozess zum Zeitpunkt t' unterbrochen, so reiht er sich auch zum Zeitpunkt t' wieder in die Warteschlange ein. Sind zwei Prozesse absolut identisch bezüglich ihrer relevanten Werte, so werden die Prozesse nach aufsteigender Prozess-ID in die Warteschlange eingereiht. Diese Annahme gilt sowohl für neu im System eintreffende Prozesse, als auch für den Prozess, dem der Prozessor u.U. gerade entzogen wird.

Beispiel: Es seien folgende Ankunfts- und Rechenzeiten für die drei Beispielprozesse P'_1, P'_2 und P'_3 gegeben:

Prozess	Ankunftszeitpunkt	Rechenzeit
P'_1	0	3
P'_2	1	4
P'_3	1	2

Das folgende Diagramm veranschaulicht ein beliebiges Scheduling der drei Prozesse P'_1, P'_2 und P'_3:



Fortsetzung nächste Seite!

Bearbeiten Sie unter den gegebenen Voraussetzungen nun die folgenden Aufgaben:

1. Erstellen Sie entsprechend dem Beispiel ein Diagramm für die **nicht-präemptive Strategie SJF** das für die Prozesse P_1-P_7 angibt, wann welchem Prozess Rechenzeit zugeteilt wird und wann die Prozesse jeweils terminieren. Kennzeichnen Sie zudem für jeden Prozess seine Ankunftszeit.

2. Erstellen Sie entsprechend dem Beispiel ein Diagramm für die **präemptive Strategie SRPT** das für die Prozesse P_1-P_7 angibt, wann welchem Prozess Rechenzeit zugeteilt wird und wann die Prozesse jeweils terminieren. Kennzeichnen Sie zudem für jeden Prozess seine Ankunftszeit.

3. Erstellen Sie entsprechend dem Beispiel ein Diagramm für die **präemptive Strategie RR** das für die Prozesse P_1-P_7 angibt, wann welchem Prozess Rechenzeit zugeteilt wird und wann die Prozesse jeweils terminieren. Die Dauer einer Zeitscheibe betrage 2 Zeiteinheiten. Gehen Sie davon aus, dass jeder Prozess die Dauer seiner Zeitscheibe stets vollständig ausnutzt, sofern er nicht terminiert. Terminiert ein Prozess aber vor Ablauf seiner Zeitscheibe, gibt er den Prozessor zum Zeitpunkt der Terminierung sofort frei. Trifft genau nach Ende einer Zeitscheibe ein neuer Prozess ein, so wird der neue Prozess **vor** dem gerade aktiven in die Warteschlange eingereiht.

4. Berechnen Sie als Dezimalzahl mit einer Nachkommastelle die mittlere Verweil- und Wartezeit für die drei Verfahren SJF, SRPT und RR.

5. Warum ist der Einsatz von SJF in einem realen System in der Regel nicht realisierbar?

6. Welchen weiteren Nachteil hat SRPT in Bezug auf die Verweildauer von Prozessen?

Fortsetzung nächste Seite!

Aufgabe 3: Petrinetze

In dieser Aufgabe soll eine Modifikation des Erzeuger-/Verbraucherproblems als Petrinetz modelliert werden. Beim klassischen Erzeuger-/Verbraucherproblem liegt folgende Situation vor:

- Es gibt zwei Prozesse. Einen Erzeuger (E) und einen Verbraucher (V).
- E produziert Resultate (R), die zunächst in einem gemeinsamen Speicher (S) der Kapazität max abgelegt werden.
- V entnimmt die Resultate dem gemeinsamen Speicher S und verbraucht diese.
- V darf nur mittels des gemeinsamen Speichers S auf ein Resultat R zugreifen.
- E darf nur bei freien Plätzen ein Resultat R im gemeinsamen Speicher S ablegen.
- Der Speicher darf nicht gleichzeitig von E und V verändert werden.

Dieses klassische Problem wird nun durch Einführen der folgenden zusätzlichen Bedingung modifiziert:

- V darf ein Resultat nur aus dem Speicher S entfernen, wenn danach noch **mindestens zwei Resultate** darin enthalten sind.

Bearbeiten Sie unter Berücksichtigung der gegebenen Anforderungen nun die folgenden Teilaufgaben.

1. Modellieren Sie das beschriebene modifizierte Erzeuger-/Verbraucherproblem durch ein Petrinetz. Erweitern Sie dazu den nachfolgend angegebenen Rahmen um die minimal notwendigen Stellen, Marken, Kanten und Transitionen. Versehen Sie alle Stellen und Transitionen mit aussagekräftigen Bezeichnern. Gehen Sie von folgenden Bedingungen aus:

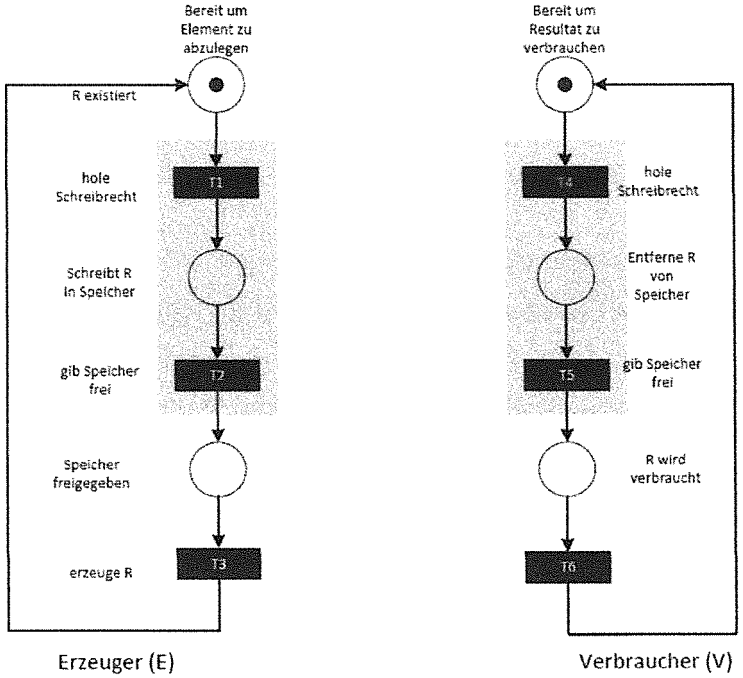
   - Die Kapazität max des gemeinsamen Speichers S beträgt 3.
   - Zu Beginn befindet sich bereits ein fertiges Resultat im gemeinsamen Speicher S.

Hinweis: Überlegen Sie sich zunächst die Funktionsweise des klassischen Erzeuger-/Verbraucherproblems und leiten Sie darauf basierend die nötigen Anpassungen her.

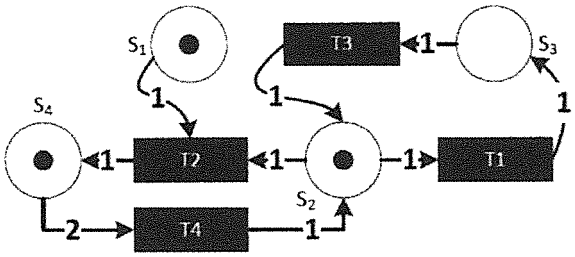
(Beachten Sie die Abbildung auf der nächsten Seite!)

Fortsetzung nächste Seite!

Der vorgegebene Rahmen ist:



2. Gegeben sei nun folgendes Petrinetz:



Leiten Sie den Erreichbarkeitsgraphen für dieses Petrinetz her.

3. Begründen Sie jeweils separat, ob im Petrinetz aus Teilaufgabe 2 ein Deadlock bzw. ein partieller Deadlock entstehen kann.

Fortsetzung nächste Seite!

Aufgabe 4: Seitenersetzungsstrategien

Gegeben seien eine Menge an Seiten $N = \{0, 1, 2, 3, 4\}$ und eine Menge der im Arbeitsspeicher zur Verfügung stehenden Seitenrahmen $F = \{f_0, f_1, f_2\}$ Auf die Seiten wird in der folgenden Reihenfolge zugegriffen:

```
W = 2, 4, 3, 0, 2, 4, 1, 3, 3, 4, 1, 0, 1, 4, 1, 2
```

Ein Seitenfehler liegt immer dann vor, wenn sich eine referenzierte Seite nicht im Arbeitsspeicher befindet. Der Arbeitsspeicher ist zu Beginn leer.

1. Bei der Paging-Strategie FIFO (First In, First Out) wird bei einem Seitenfehler diejenige Seite im Arbeitsspeicher ersetzt, deren Zeitpunkt der Zuordnung zu einem Seitenrahmen am längsten zurück liegt.
   Ermitteln Sie die Anzahl der Seitenfehler für die Seitenersetzungsstrategie FIFO, indem Sie alle Veränderungen im Speicher dokumentieren. Erstellen Sie dazu eine Tabelle mit dem Schema:

Zeit	Referenzierte Seite	f_0, t	f_1, t	f_2, t	Summe Seitenfehler
⋮	⋮	⋮	⋮	⋮	⋮

   Ordnen Sie dabei jeder referenzierten Seite den entsprechenden Seitenrahmen f_i ($i \in \{0,1,2\}$) zu und dokumentieren Sie den Zeitpunkt t der Zuordnung. Geben Sie zudem nach jedem Seitenzugriff die aktuelle Summe an Seitenfehlern an.

Achtung: Bereits in den Hauptspeicher geladene Seiten dürfen nicht von einem Seitenrahmen in einen anderen verschoben werden.

2. Ermitteln Sie die Anzahl der Seitenfehler für die Seitenersetzungsstrategie LFU (Least Frequently Used), indem Sie eine Tabelle mit dem bekannten Schema aus Teilaufgabe 1 erstellen. Statt der Zeit t soll allerdings nun für jeden Seitenrahmen die Anzahl der Seitenzugriffe seit dem Laden einer Seite festgehalten werden. Ordnen Sie in der Tabelle wieder jede referenzierte Seite dem entsprechenden Seitenrahmen f_i ($i \in \{0,1,2\}$) zu und dokumentieren Sie die Anzahl der Seitenzugriffe. Geben Sie zudem nach jedem Seitenzugriff die aktuelle Summe an Seitenfehlern an. Falls mehrere Seitenrahmen als Kandidaten für die neue Seite infrage kommen, so soll diejenige mit der kleinsten Indexnummer gewählt werden.

   Achtung: Bereits in den Hauptspeicher geladene Seiten dürfen nicht von einem Seitenrahmen in einen anderen verschoben werden.

3. Nun sollen die theoretischen Eigenschaften der Seitenersetzungsstrategie Least Frequently Used (LFU) untersucht werden. Dazu soll untersucht werden, wie viele Seitenfehler für das Beispiel aus Teilaufgabe 2 unter Verwendung von LFU entstehen, wenn das System 1, 2, 3, 4 oder 5 Seitenrahmen besäße. Der zugehörige *Distance String* lautet $\infty\ \infty\ \infty\ \infty\ 3\ 3\ \infty\ 4\ 0\ 2\ 2\ 4\ 1\ 2\ 1\ 4$. Der Rechenweg muss klar nachvollziehbar sein.

4. Welchen Effekt im Zusammenhang mit der Anzahl der Seitenrahmen eines Systems beschreibt die *Belady's Anomalie*? Kann diese Anomalie mit der Seitenersetzungsstrategie LFU (Least Frequently Used) auftreten?

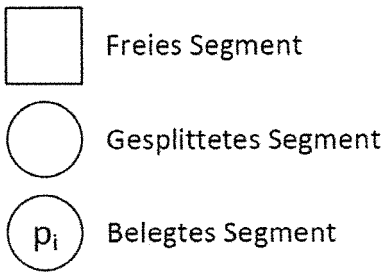
Fortsetzung nächste Seite!

Aufgabe 5: Buddy-Systeme

Gegeben sei ein mobiles System mit einem Hauptspeicher von 2 GB = 2048 MB, der byteweise adressiert wird. Zur Speicherverwaltung werden Buddy-Systeme eingesetzt. Dabei wird immer die am weitesten links stehende Speicherzelle geteilt, wenn ein neuer Prozess eingefügt wird. Die minimale Buddy-Größe soll 128 MB betragen. Bearbeiten Sie nun folgende Aufgaben:

1. Es werden 5 Prozesse der Reihe nach in das Buddy-System eingefügt:
   a) p_1: 510 MB
   b) p_2: 382 MB
   c) p_3: 76 MB
   d) p_4: 248 MB
   e) p_5: 426 MB

Zeichnen Sie insgesamt 5 Buddy-Bäume, die jeweils den Zustand der Speicherbelegung darstellen, nachdem ein weiterer der 5 Prozesse eingefügt wurde. Es muss genau ersichtlich sein, ob es sich um ein freies, ein gesplittetes bzw. um ein belegtes Segmente des Hauptspeichers handelt. Kennzeichnen Sie die entsprechenden Segmente eines Buddy-Baums mit den folgenden Symbolen:

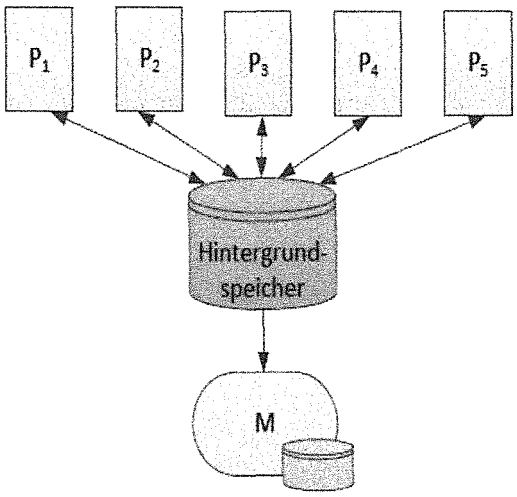


2. Wieviele Bits werden benötigt, um ein Byte im gegebenen Speicher adressieren zu können? Ergänzen Sie im letzten Buddy-Baum der Teilaufgabe 1 (also dem Buddy-Baum nach dem Einfügen von p_5) für alle freien, belegten und gesplitteten Segmente die ersten 6 Bits ihrer Speicheradresse.
3. Wieviel Speicher würde jedem der 5 Prozesse nach dem Buddy-Verfahren tatsächlich zur Verfügung stehen? Geben Sie den maximal zur Verfügung stehenden Speicherplatz für jeden der eingefügten Prozesse an.
4. Die eingefügten Prozesse p_1 bis p_5 benötigen insgesamt 1642 MB. Durch die Verwendung von Buddy-Systemen weicht der tatsächlich zugeordnete Hauptspeicher von diesem Wert ab. Wie viele von den ursprünglichen 2048 MB stehen für weitere Prozesse somit nur noch zur Verfügung und wieviel Speicher wird letztendlich verschwendet? Wie nennt man den für diese Verschwendung verantwortlichen Effekt?
5. Ein neuer Prozess p_6 mit einem Speicherbedarf von 240 MB soll gestartet werden. Ist es möglich diesen Prozess einem Buddy zuzuweisen? Falls ja, zeichnen sie den aktualisierten Buddy-Baum. Falls nein, erläutern Sie den Grund.
6. Zunächst terminiert der Prozess p_3 und danach der Prozess p_4. Zeichen Sie den aktualisierten Buddy-Baum nach jeder der beiden Prozessterminierungen. Kennzeichnen Sie dabei freie, gesplitteten bzw. belegten Segmente wieder mit den zuvor verwendeten Symbolen.

Fortsetzung nächste Seite!

Aufgabe 6: Koordination von Threads

In dieser Aufgabe soll der Betrieb eines Terminals in Java simuliert werden, auf dem insgesamt 5 Prozesse aktiv sind. Es wird angenommen, dass die Prozesse P_1 bis P_5 spezielle Rechenprozesse sind, die regelmäßig große Datenmengen in speziellen *Zugriffsphasen* auf den gemeinsamen Hintergrundspeicher schreiben. Aus Performanzgründen dürfen zu einem Zeitpunkt jedoch maximal 3 Rechenprozesse gleichzeitig den Hintergrundspeicher beschreiben. In den *Zwischenphasen* greifen die Rechenprozesse nicht auf den Hintergrundspeicher zu und führen spezielle Berechnungen durch. In regelmäßigen Abständen soll der aktuelle Stand des Hintergrundspeichers über ein Bandlaufwerk gesichert werden, wobei exakt der Zustand zu einem dedizierten Zeitpunkt gespeichert werden soll. Diese Datensicherung führt ein weiterer Datensicherungsprozess M durch.
Der Datensicherungsprozess beendet sich im Ruhezustand, wenn aktuell keine Sicherung nötig ist. Er geht in den Wartezustand über, wenn eine Sicherung nötig ist und meldet einen Sicherungswunsch an. Kein Rechenprozess darf dann mehr einen Schreibzugriff beginnen. Sobald alle noch laufenden Schreibzugriffe beendet wurden, beginnt der Datensicherungsprozess seine Arbeit und sichert die Daten auf ein Bandlaufwerk. Während der Datensicherung dürfen die Prozesse P_1 bis P_5 nicht auf die Festplatte zugreifen und müssen warten. Gleichzeitig gilt, dass M keine Festplattenzugriffe machen darf, solange einer der Prozesse P_1 bis P_5 aktuell noch auf die Festplatte zugreift. Das beschriebene Szenario ist in folgender Abbildung nochmals dargestellt:



Im Folgenden soll eine Klasse Speicher zur Simulation des Hintergrundspeichers implementiert werden.
Die Beispielimplementierungen der Klassen Rechenprozess, Datensicherungsprozess und Terminal soll Ihnen verdeutlichen, wie die Klasse Speicher verwendet werden kann:

Fortsetzung nächste Seite!

Die Klasse Terminal:

```
1  public class Terminal {
2      private Speicher my_speicher;
3      private Rechenprozess[] rechenprozesse;
4      private Datensicherungsprozess my_datensicherungsprozess;
5
6      public Terminal() {
7          my_speicher = new Speicher(3);
8          rechenprozesse = new Rechenprozess[5];
9          for(int i = 0; i < rechenprozesse.length; i++) {
10             rechenprozesse[i] = new Rechenprozess(my_speicher, i);
11             rechenprozesse[i].start();
12         }
13         my_datensicherungsprozess = new Datensicherungsprozess(my_speicher);
14         my_datensicherungsprozess.start();
15     }
16
17
18     public static void main(String[] args) {
19         new Terminal();
20     }
21 }
```

Die Klasse Rechenprozess:

```
1  import java.util.Random;
2
3  public class Rechenprozess extends Thread {
4      private Speicher my_speicher;
5      private int id;
6      private Random generator;
7
8      public Rechenprozess(Speicher speicher, int id) {
9          this.my_speicher = speicher;
10         this.id = id;
11         generator = new Random();
12     }
13
14     public void run() {
15             try {
16                 Thread.sleep(generator.nextInt(2000) + 500);
17                 my_speicher.schreibzugriffBeginnen(id);
18                 Thread.sleep(generator.nextInt(2000) + 500);
19                 my_speicher.schreibzugriffBeenden(id);
20             }
21             catch(InterruptedException ie) {}
22     }
23 }
```

Fortsetzung nächste Seite!

Die Klasse Datensicherungsprozess:

```
1  import java.util.Random;
2
3  public class Datensicherungsprozess extends Thread {
4          private Speicher my_speicher;
5          private Random generator;
6
7          public Datensicherungsprozess(Speicher speicher) {
8                  this.my_speicher = speicher;
9                  generator = new Random();
10         }
11
12         public void run() {
13                 while (true) {
14                         try {
15                                 Thread.sleep(generator.nextInt(2000) + 500);
16                                 my_speicher.sicherungBeginnen();
17                                 System.out.println("Daten werden gesichert");
18                                 Thread.sleep(generator.nextInt(2000) + 500);
19                                 my_speicher.sicherungBeenden();
20                                 Thread.sleep(1000);
21                         } catch (InterruptedException ie) {
22                         }
23                 }
24         }
25 }
```

Bearbeiten Sie nun die folgenden Aufgaben:

1. Was versteht man allgemein unter einem kritischen Bereich?
2. Implementieren Sie den Konstruktor der Klasse Speicher. Verwenden Sie dabei den Coderahmen am Ende der Aufgabe und *kommentieren Sie Ihre Lösung ausführlich.* Die Klassenattribute sind dorts bereits deklariert und müssen durch den Konstruktur initialisiert werden.

Fortsetzung nächste Seite!

3. Implementieren Sie die Methode `schreibzugriffBeginnen(int id)`, welche den Beginn des Zugriff eines Rechenprozesses auf den Hintergrundspeicher modelliert, sowie die Methode `schreibzugriffBeenden(int id)`, welche im Gegenzug das Beenden des Zugriffs eines Rechenprozesses auf den Hintergrundspeicher modelliert. Beachten Sie dazu, dass alle oben genannten Anforderungen beachtet werden müssen. Insbesondere:

   - Maximal 3 Rechenprozesse dürfen den Hintergrundspeicher gleichzeitig beschreiben.
   - Sobald der Datensicherungsprozess einen Sicherungswunsch hat, darf kein Rechenprozess den Hintergrundspeicher mehr beschreiben, bis der Datensicherungsprozess fertig ist und der Hintergrundspeicher wieder freigegeben ist.

   Ergänzen Sie dazu den Coderahmen am Ende der Aufgabe und *kommentieren Sie Ihre Lösung ausführlich.*
   Hinweis: Sie können davon ausgehen, dass die Methoden `schreibzugriffBeginnen(int id)` bzw. `schreibzugriffBeenden(int id)` immer in einer sinnvollen Reihenfolge aufgerufen werden (siehe Beispielimplementierung der Klasse Rechenprozess).

4. Implementieren Sie nun die Methoden für den Datensicherungsprozess. Vervollständigen Sie dazu den Coderahmen für die Methoden `sicherungBeginnen()` und `sicherungBeenden()` in dem Coderahmen am Ende der Aufgabe.
   Hinweis: Sie können davon ausgehen, dass die Methoden `sicherungBeginnen()` und `sicherungBeenden()` immer in einer sinnvollen Reihenfolge aufgerufen werden (siehe Beispielimplementierung der Klasse Datensicherungsprozess).

5. Zeigen Sie zwei kritische Bereiche in Ihrem Programm auf. Wie wird hier sichergestellt, dass die Bedingung der Mutual Exclusion erfüllt ist?

Fortsetzung nächste Seite!

Folgender Coderahmen steht Ihnen zur Bearbeitung der Teilaufgaben 2), 3) und 4) zur Verfügung:

```
1  public class Speicher {
2          private static int maxRechenprozesse;
3          private int anzahlRechenprozesse;
4          private boolean hat_sicherungswunsch;
5
6
7
8          public Speicher(                                                    ) {
9
10
11
12
13
14
15
16
17
18         }
19
20         public synchronized void schreibzugriffBeginnen(int id) {
21                 // Wird durch einen Rechenprozess aufgerufen.
22                 try {
23                    while (                                                  ) {
24
25
26
27
28                        }
29                 } catch (InterruptedException ie) {
30                 }
31
32
33
34
35
36
37
38                 System.out.println("Rechenprozess " + id + " ==> Speicher ("
39                                 + anzahlRechenprozesse + " schreibend)");
40
41
42
43
44         }
45
46         public synchronized void schreibzugriffBeenden(int id) {
47                 // Wird durch einen Rechenprozess aufgerufen.
48
49
50
51
```

Fortsetzung nächste Seite!

```
52
53
54                  System.out.println("Speicher ==> Rechenprozess " + id + " ("
55                                  + anzahlRechenprozesse + " schreibend)");
56
57
58
59
60
61
62          }
63
64          public synchronized void sicherungBeginnen() {
65                  // Wird durch den Datensicherungsprozess aufgerufen.
66                  this.hat_sicherungswunsch = true;
67                  try {
68                     while (                                          ) {

69
70
71
72
73                          }
74                  } catch (InterruptedException ie) {
75                  }
76                  System.out.println("Datensicherungsprozess ==> Speicher ("
77                                  + anzahlRechenprozesse
78                                  + " Rechenprozesse schreibend)");
79
80
81
82
83
84          }
85
86          public synchronized void sicherungBeenden() {
87                  // Wird durch den Datensicherungsprozess aufgerufen.
88                  this.hat_sicherungswunsch = false;
89
90
91
92
93
94
95
96                  System.out.println("Speicher ==> Datensicherungsprozess ("
97                                  + anzahlRechenprozesse
98                                  + " Rechenprozesse schreibend)");
99
100
101
102
103

105
106
107          }
108
109 }
```