
Prüfungsteilnehmer

Prüfungstermin

Einzelprüfungsnummer

Kennzahl: _____

Frühjahr

Kennwort: _____

2006

66113

Arbeitsplatz-Nr.: _____

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

- Prüfungsaufgaben -

Fach: Informatik (vertieft studiert)

Einzelprüfung: Rechnerarchitektur, Datenbank, Betriebssysteme

Anzahl der gestellten Themen (Aufgaben): 2

Anzahl der Druckseiten dieser Vorlage: 14

Bitte wenden!

Thema Nr. 1

Sämtliche Teilaufgaben sind zu bearbeiten!

A. Themengebiet Datenbanksysteme**Aufgabe 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

Aufgabe 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.

Aufgabe 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.

Aufgabe 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.

Aufgabe 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.

- Geben Sie den Namen und Vornamen aller gespeicherten Weinfreunde aus, die jünger als 30 Jahre sind.
- Geben Sie eine Liste aller gespeicherten Weine des Weinbergs „Katzkopf“ aufsteigend geordnet nach dem Jahrgang aus.
- Geben Sie die durchschnittlich bestellte Menge Wein aller erfassten Bestellungen an.
- 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.

- 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 letzten 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.

B. Themengebiet Betriebssysteme

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.

Aufgabe 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.

Aufgabe 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.

Aufgabe 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 Prozesskoordination ist hier erforderlich?
Skizzieren Sie die Koordinierung für die beiden Prozesse in einer programmiersprachlichen Form.

Thema Nr. 2

Sämtliche Teilaufgaben sind zu bearbeiten!

A. Themengebiet Datenbankensysteme**Teilaufgabe 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.

Teilaufgabe 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 in 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.

Teilaufgabe 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).

B. Themengebiet Betriebssysteme**Teilaufgabe 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();
        }
    }
}
```

Teilaufgabe 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?

Teilaufgabe 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?