

	Bitte hier unbedingt Matrikelnummer und Adresse eintragen, sonst keine Bearbeitung möglich
Postanschrift: FernUniversität D - 58084 Hagen	
_____ (Name, Vorname)	
_____ (Straße, Nr.)	
_____ (Auslandskennzeichen, PLZ, Wohnort)	

FERNUNIVERSITÄT -Gesamthochschule- EINGANG
INF
Bitte zurück an FERNUNIVERSITÄT in Hagen D - 58084 Hagen

Fachbereich Informatik

Kurs 1666 Datenbanken in Rechnernetzen

Kurseinheit 01 bis 07

Einsendeaufgaben

A. Hinweise zur Bearbeitung

1. Benutzen Sie bitte für Ihre Lösungen Papier im Format DIN A4.
2. Schreiben Sie bitte auf jedes Blatt Ihren Namen mit Matrikelnummer.
3. Schreiben Sie bitte deutlich, lassen Sie einen breiten Rand (ca. 4-5 cm) und numerieren Sie zum Schluß die Lösungsblätter durch.
4. Schicken Sie Ihre Lösungsblätter sowie das (grüne) Deckblatt komplett und zusammengeklammert zurück.
5. Kreuzen Sie bitte in der Zeile „bearbeitet“ die von Ihnen bearbeiteten Aufgaben an.

B. Hinweise zur Bewertung

1. Bei jeder Aufgabe bzw. Teilaufgabe ist die erreichbare Punktzahl vermerkt, sofern Punktwertung vorgenommen wird.
2. Insgesamt sind 200 Punkte erreichbar.
3. Die Einsendearbeit umfaßt 8 Aufgabe(n).

Letzer Einsendetag: 2.02.2007 (Datum des Poststempels)

Aufgabe	1	2	3	4	5	6	7	8	Summe
bearbeitet									
erreichte Punktzahl									

Datum: _____

Korrektur:

Aufgabe 1: Horizontale Partitionierung (25 Punkte)

Horizontale Partitionierungen sind durch $R_i = SL_{f_i} R (i = 1 \dots n)$ beschrieben.

a) Welche Eigenschaften (als prädikatenlogische Formeln) müssen die Selektionsformeln f_i aufweisen, um

- Diskunktheit
- Vollständigkeit

zu garantieren? Wir gehen davon aus, dass Sie wissen, dass die f_i als Prädikate zu TRUE oder FALSE evaluieren.

b) Verifizieren Sie mit Hilfe algebraischer Umformungen und den Eigenschaften der Selektionsformeln, dass Disjunktheit (der Schnitt zweier verschiedener R_i ist leer) und Vollständigkeit (d.h. Rekonstruierbarkeit der Ausgangsrelation) erreicht wird.

c) Finden Sie Beispiele für Selektionsformeln, die

- vollständig aber nicht disjunkt sind.
- disjunkt aber nicht vollständig sind.

Aufgabe 2: Partitionierung und Allokation (25 Punkte)

Ein Unternehmen ist auf drei Standorte in Deutschland verteilt. An jedem Standort gibt es eine Abteilung, die das rechtzeitige Eintreffen von Lieferungen von Zulieferern aus dem Umland überwacht. Die drei Abteilungen überwachen jeweils einen Postleitzahlbereich mit folgender Aufteilung:

Abteilung an Standort 1: PLZ 01000 bis 39999 (Der Norden, Osten und die Mitte)

Abteilung an Standort 2: PLZ 40000 bis 69999 (Der Westen)

Abteilung an Standort 3: PLZ 70000 bis 99999 (Der Süden)

Die Verwaltung der Lager des Unternehmens wird über die drei Relationen Lieferant, Teil, und Bestellung realisiert. Der Primärschlüssel ist jeweils unterstrichen:

Lieferant(<u>LiefNr</u> , LiefName, LiefPLZ, LiefStrasse)	Teil(<u>TeilNr</u> , TeilName, LagerNr, RegalNr, Bestand)	Bestellung(<u>LiefNr</u> , <u>TeilNr</u> , Anz, Termin, Status)
--	---	---

- a) Diese drei Relationen sollen sinnvoll partitioniert werden. Es wird dabei von folgenden Bedingungen ausgegangen:
- Jede Abteilung überwacht nur die Bestellungen der Lieferanten ihres PLZ-Bereiches
 - Es gibt drei Materiallager: Lager1, Lager2 und Lager3, die eigene Verbundrechner haben. Die Abteilungen sind von diesen Lagern weit entfernt und greifen über die TeilNr der Teil Relation meist nur auf den aktuellen Bestand zu.
 - Die Materiallager greifen hauptsächlich auf die Attribute TeilNr, TeilName, LagerNr und RegalNr der Teil Relation zu. Lager1 verwaltet alle Teile mit LagerNr = 1 und so weiter.
- b) Wie könnte eine geeignete zugehörige redundante/nicht-redundante Allokation aussehen, die folgenden Bedingungen entspricht:
- Globale Zugriffe finden eher selten statt.
 - Die Materiallager greifen häufig lesend auf Bestand in Teil zu.

Aufgabe 3: Algorithmus HORIZ_PART (25 Punkte)

Gegeben sei folgende Relation

```
Teil(  
    TeilNr,  
    TeilBezeichn  
    Preis)
```

Das Attribut `TeilNr` kann Werte im Bereich von 100 bis 399 annehmen. Fünf verschiedene Anwendungen greifen besonders häufig folgendermaßen auf diese Relation zu:

Anwendungen:

$A_1: \text{TeilNr} \in [100, 199]$

$A_2: \text{TeilNr} \in [200, 299]$

$A_3: \text{TeilNr} \in [300, 399]$

$A_4: \text{TeilNr} \in [100, 219]$

$A_5: \text{TeilNr} \in [200, 399]$

Ermitteln Sie mit Hilfe des Algorithmus HORIZ_PART eine geeignete horizontale Partitionierung der Relation `Teil`. Dokumentieren Sie dabei bitte alle Teilschritte.

Hinweis:

Der Ausdruck "*wesentliche Verbesserung*" im Algorithmus bedeutet, dass eine Anwendung nach der Änderung weniger Tupel lesen muss, um das Zugriffsprädikat auszuwerten.

Aufgabe 4: Nicht-redundante Allokation (25 Punkte)

Drei Zweigstellen einer DVD-Videothek verwalten ihre Datenbestände über ein vDBMS. Jeder Zweigstelle ist dabei ein Rechnerknoten mit unbegrenzter Speicherkapazität zugeordnet.

Folgende Bedingungen gelten:

Übertragungskosten zwischen zwei Knoten = 100, sonst 0.

Die Titel-Relation der DVD Datenbank wurde folgendermaßen logisch in zwei Partitionen zerlegt:

- Partition 1 enthält die Spielfilme mit 12000 Datensätzen
- Partition 2 enthält alle anderen DVD Titel (z.B. Dokumentationen) mit 9000 Datensätzen

Jeder Datensatz entspricht dabei genau einer Dateneinheit.

Die Größe des Anfragestrings bei Leseoperationen und die der Resultate bei Änderungsoperationen seien vernachlässigbar.

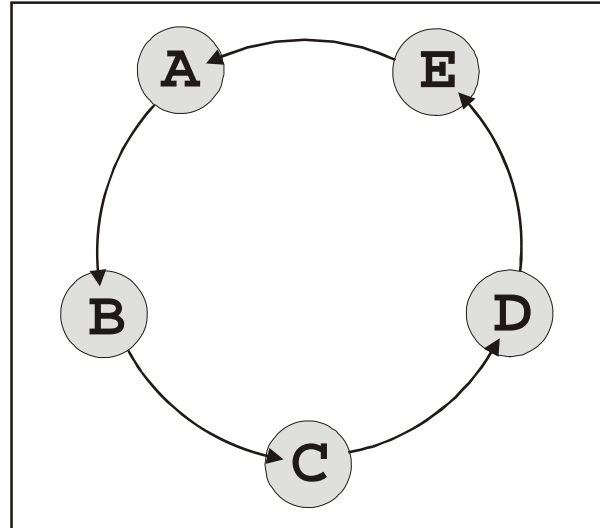
Die Speicherkosten und Zugriffe verteilen sich folgendermaßen auf die Knoten:

Knoten Nr.	Speicherkosten / Dateneinheit	Zugriffe / Tag
Knoten 1	120	200 lesend (Typ 1) auf Partition1, Selektivität 4%. 90 ändernd (Typ 2) auf Partition 1 und 2, 1.5% der Datensätze werden verändert.
Knoten 2	100	400 lesend (Typ 3) auf Partition1 und 2, Selektivität 2%
Knoten 3	80	700 lesend (Typ 4) auf Partition 1 und 2, Selektivität 1%

- a) Stellen Sie die Eingabegrößen K , P , T , S_i , G_p , U_{ij} , O_{tp} , R_{tp} und H_{it} für das Optimierungssystem zusammen.
- b) Ermitteln Sie damit, welche der drei folgenden Allokationsmöglichkeiten die beste Alternative darstellt.
 - I. Partition 1 allokiert auf Knoten 1, Partition 2 allokiert auf Knoten 2
 - II. Partition 1 und 2 allokiert auf Knoten 2
 - III. Partition 1 und 2 allokiert auf Knoten 3
- c) Es tritt eine asymmetrische Verringerung der Übertragungskosten ein. Die Kosten der Übertragung von Knoten 2 zu Knoten 1 halbiert sich auf 50. Die Kosten für die Gegenrichtung $1 \rightarrow 2$ bleiben bei 100. Werten Sie nochmals die unter b) genannten Allokationsmöglichkeiten unter den veränderten Bedingungen aus.

Aufgabe 6: Replikationsverfahren: Majority Consensus (25 Punkte)

In einer voll replizierten verteilten Datenbank, bestehend aus fünf Knoten sei die abgebildete logische Knotenreihenfolge für die Durchführung des Majority Consensus Verfahrens gegeben. Wir gehen zur Vereinfachung von einem globalen Zeittakt aus. Außerdem sollen an jedem Knoten jeweils innerhalb eines Taktschrittes eingehende Anforderungen geprüft, eine Entscheidung herbeigeführt und gegebenenfalls eine Nachricht an den nachfolgenden Knoten weitergegeben werden. Diese Nachricht verarbeitet dieser nachfolgende Knoten dann im nächsten Taktschritt. Commit und Abort Nachrichten



werden parallel an alle Knoten gleichzeitig verschickt. Diese werden dann ebenfalls im nächsten Taktschritt verarbeitet. Zu Beginn haben alle Objektzeitstempel den Wert 0. Bitte stellen Sie jeweils für beide Teilaufgaben die Zustände der Knoten zu den Zeittakten $t = 1, 2, 3, \dots$ dar bis keine Änderung mehr auftritt. Dabei müssen, soweit nötig, zu jedem Knoten der Objektzeitstempel, bisherige Entscheidungen, eintreffende Nachrichten, neue Entscheidungen und ausgehende Nachrichten unter Berücksichtigung der Transaktionszeitstempel angegeben werden.

- Zum Zeitpunkt $t = 1$ wird an Knoten A eine Update-Transaktion T_1 für das Objekt a gestartet. Einen Zeittakt später ($t = 2$) startet Knoten C ebenfalls eine Update-Transaktion T_2 für das Objekt a . Verdeutlichen Sie die einzelnen Schritte des Abstimmungsverfahrens.
- Ausgehend vom Anfangszustand wird in diesem Fall bei $t = 1$ an Knoten B die Update-Transaktion T_1 für das Objekt a gestartet. Zum Zeitpunkt $t = 2$ startet Knoten A ebenfalls eine Update-Transaktion T_2 für das Objekt a . Zeigen Sie auch hier den Verlauf des Abstimmungsverfahrens.

Wenn Sie möchten, können Sie das beigeheftete Blatt mit dem Schema der Knotenreihenfolge oder ggf. weitere Kopien davon für Ihre Antwort benutzen.

Aufgabe 7: Recovery (25 Punkte)

Betrachten Sie folgendes Szenario:

SP (Savepoint) steht für Sicherungspunkt.
TU (time unit/unit of time) steht für Zeiteinheit.
EOT (end of transaction) steht für Transaktionsende.

Zeitpkt.

0	SP C^1 an Knoten A, B und C abgeschlossen
1	Globale Transaktion T_{1A} startet an Knoten A Nach 2 TU initiiert T_{1A} eine Subtransaktion T_{1B} an Knoten B, Dauer: 5 TU Nach Ende von T_{1B} benötigt T_{1A} noch 5 TU bis EOT.
5	Knoten C initiiert die Erzeugung von SP C^2 . Eine lokale SP Erzeugung dauert 5 TU.
10	Knoten A initiiert Transaktion T_{2A} . Nach 1 TU initiiert T_{2A} eine Subtransaktion T_{2C} mit Dauer 5 TU an Knoten C. Nach dem Ende von T_{2C} benötigt T_{2A} noch 2 TU bis EOT.
11	Eine globale Transaktion T_C mit Dauer 7 TU wird an Knoten C gestartet. Eine globale Transaktion T_{3C} wird an Knoten C gestartet. Nach 2 TU initiiert T_{3C} eine Subtransaktion T_{3A} mit Dauer 3 TU an Knoten A. T_{3C} benötigt nach dem Ende von T_{3A} noch 3 TU bis EOT.

Hinweise:

- Die angegebenen Zeitdauern sind jeweils Nettozeiten ohne Berücksichtigung von synchronisationsbedingten Verzögerungen.
- Wenn die Erzeugung eines SP "angemeldet" wurde, werden lokal keine neuen Transaktionen gestartet.
- Zur Vereinfachung gilt: das Ende einer Subtransaktion ist gleichbedeutend mit ihrem Commit. Die Primärtransaktion läuft nach dem Ende der Subtransaktion unter Umständen noch ein wenig weiter.
- Die einzelnen Transaktionen stehen nicht in Konflikt miteinander.

Prüfen / zeigen Sie für die folgenden Varianten, wann die Erzeugung von C^2 und die einzelnen Transaktionen (incl. der zum Zeitpunkt 11) gestarteten abgeschlossen sind. Veranschaulichen Sie ihre Lösung bitte graphisch.

- a) Die Erzeugung von C^2 erfolgt strikt synchronisiert.
- b) Die Erzeugung von C^2 erfolgt lose synchronisiert.

Aufgabe 8: Client/Server-Systeme (25 Punkte)

a) Betrachten Sie die beiden folgenden Relationen:

Teil(<u>TeilNr</u> , Bestand, Mindestbestand, Bestellmenge)	Bestellung(TeilNr, Menge)
--	----------------------------------

Verringert sich der Bestand an Teilen, soll jeweils geprüft werden, ob der vorgegebene Mindestbestand unterschritten wurde. Wenn dem so ist, soll ein entsprechendes „(Nach-) Bestell“ Tupel in die Relation Bestellung eingefügt werden, falls dort nicht schon ein solches vorhanden ist. Falls schon eine Bestellung vorliegt, soll nichts weiter getan werden. Lösen Sie diese Aufgabe mit Hilfe von JDBC oder mittels Triggern, indem Sie einen entsprechenden Bedingungsteil benutzen. Bei der Lösung mit JDBC können sie die Details des Verbindungsaufbaus zur Datenbank abstrahieren. Liefern Sie beide Lösungen, so erhalten Sie Zusatzpunkte.

- b) Kennen Sie außer Triggern noch weitere Mechanismen zur Reduktion von (entfernten) Datenbankaufrufen? Diskutieren Sie Vor- und Nachteile
- c) Beschreiben Sie den Ablauf eines Remote Procedure Calls (RPC). Was versteht man unter RMI?
- d) Welche Mechanismen können zur Synchronisation von (Betriebssystem-) Prozessen eingesetzt werden?
- e) Was sind Threads?

