

Hauptdiplomklausur Datenbanksysteme I Sommersemester 1999

Name:
Vorname:
Matrikelnummer:
Studienfach:

Wichtige Hinweise:

1. Prüfen Sie Ihr Klausurexemplar auf Vollständigkeit (17 Seiten).
2. Es sind keine Hilfsmittel zugelassen.
3. Die Klausur dauert 100 Minuten.
4. Jede Aufgabe ist auf dem zugehörigen Aufgabenblatt (und ggf. auf separaten Lösungsblättern) zu bearbeiten.
5. Vermerken Sie Ihren Namen und Ihre Matrikelnummer auf jedem Aufgaben- (bzw. Lösungsblatt). Blätter ohne Namens- und Matrikelnummerangabe werden nicht bewertet.
6. Das Deckblatt sowie alle Aufgabenblätter (evtl. Lösungsblätter) sind abzugeben.

	maximale Anzahl Punkte	erreichte Anzahl Punkte
Aufgabe 1	15	
Aufgabe 2	29	
Aufgabe 3	3	
Aufgabe 4	6	
Aufgabe 5	10	
Aufgabe 6	12	
Aufgabe 7	8	
Aufgabe 8	17	
	100	

1. Nach dem Entwurf und dem Laden der Daten in ein relationales Datenbanksystem sind leider Informationen über die ursprünglichen funktionalen Abhängigkeiten verloren gegangen. (Abbildung 1 zeigt das Schema der Datenbank.) Rekonstruieren Sie in den folgenden Teilaufgaben die fehlenden Abhängigkeiten. Begründen Sie jeweils Ihre Antwort kurz.

```
create table Rel1 (  
  G integer,  
  H integer,  
  I integer,  
  J integer,  
  K integer,  
  primary key(I,K)  
);  
  
create table Rel2 (  
  A integer,  
  B integer,  
  C integer,  
  D integer,  
  E integer,  
  F integer,  
  primary key(B,D,F),  
  unique(A,B,F),  
  unique(C,F)  
);  
  
create table Rel3 (  
  M integer,  
  N integer,  
  O integer,  
  P integer,  
  Q integer,  
  primary key(M)  
);
```

Abbildung 1: Schema einer relationalen Datenbank

(a) (5 Punkte)

Von den Attributen in Relation *Rel1* ist bekannt, daß die funktionalen Abhängigkeiten $I \rightarrow G$ und $K \rightarrow J$ zwischen ihnen herrschen. Es existierte noch eine dritte funktionale Abhängigkeit von der lediglich bekannt ist, daß sie entweder $G \rightarrow H$ oder $G \rightarrow I$ oder $K \rightarrow G$ lautete. Welche dieser drei Abhängigkeiten ist die richtige?

(b) (5 Punkte)

Für *Rel2* gilt: $AB \rightarrow C \in \mathcal{F}_{\text{Rel2}}$ und $DF \rightarrow A \in \mathcal{F}_{\text{Rel2}}$. Welche der folgenden Abhängigkeiten ergänzt $\mathcal{F}_{\text{Rel2}}$ korrekt: $C \rightarrow E$, $C \rightarrow DE$, oder $C \rightarrow BDE$?

(c) (5 Punkte)

Für *Rel3* gilt: $M \rightarrow N \in \mathcal{F}_{\text{Rel3}}$, $N \rightarrow O \in \mathcal{F}_{\text{Rel3}}$ und $P \rightarrow Q \in \mathcal{F}_{\text{Rel3}}$. Von der vierten fehlenden Abhängigkeit ist bekannt, daß sowohl auf der linken wie auf der rechten Seite der Abhängigkeit nur ein Attribut stand. Geben Sie alle möglichen Abhängigkeiten an, die $\mathcal{F}_{\text{Rel3}}$ korrekt ergänzen.

2. Die Daten zur Tischreservierung einiger Restaurants werden zentral in einem relationalen Datenbanksystem gehalten. Es existieren vier verschiedene Relationen mit folgenden Schema:

- Restaurant(RID, Name, Straße, Stadt, geöffnetVon, geöffnetBis, Telefon)
- Tisch(TID, Art, RID)
- Platz(PID, TID)
- Belegung(PID, Datum, Ankunftszeit, Weggangszeit, Name)

In der Relation *Restaurant* steht die ID, der Name, die Adresse, die Öffnungszeiten und die Telefonnummer des jeweiligen Restaurants. Die Relation *Tisch* gibt sowohl die ID eines Tisches als auch den Standort (RID) und die Art des Tisches (Raucher/Nichtraucher) an. In der Relation *Platz* werden die einzelnen Plätze verwaltet. Die Relation *Belegung* hält für jeden reservierten Platz den Zeitraum der Belegung und den Namen des Reservierenden zu einem Tisch fest.

Formulieren Sie folgende Anfragen in SQL:

(Hinweis: Sie dürfen Datum und Uhrzeit direkt mit <, > und = miteinander vergleichen.)

(a) (5 Punkte)

Geben Sie die Namen und Telefonnummern aller Restaurants an, die Nichtraucherische anbieten. Achten Sie darauf, daß keine doppelten Einträge in der Liste auftauchen.

Restaurant(RID, Name, Straße, Stadt, geöffnetVon, geöffnetBis, Telefon)

Tisch(TID, Art, RID)

Platz(PID, TID)

Belegung(PID, Datum, Ankunftszeit, Weggangszeit, Name)

(b) (5 Punkte)

Geben Sie für alle Restaurants in Mannheim die maximale Kapazität an (Summe aller Sitzplätze).

Restaurant(RID, Name, Straße, Stadt, geöffnetVon, geöffnetBis, Telefon)

Tisch(TID, Art, RID)

Platz(PID, TID)

Belegung(PID, Datum, Ankunftszeit, Weggangszeit, Name)

(c) (9 Punkte)

Geben Sie für alle Restaurants in Mannheim die Anzahl der Nichtraucherplätze an, die am 29.08.1999 um 14:00 Uhr reserviert sind. Es sollen auch die Plätze mitgezählt werden, die zu einem früheren Zeitpunkt reserviert, aber um 14:00 Uhr noch nicht frei sind.

Restaurant(RID, Name, Straße, Stadt, geöffnetVon, geöffnetBis, Telefon)

Tisch(TID, Art, RID)

Platz(PID, TID)

Belegung(PID, Datum, Ankunftszeit, Weggangszeit, Name)

(d) (10 Punkte)

Geben Sie für alle Restaurants die TIDs der freien Raucherische zwischen 18:00 und 21:00 Uhr am 28.08.1999 aus. Ein Tisch ist frei, wenn alle Plätze über die ganzen drei Stunden an dem Tisch frei sind.

3. (3 Punkte)

Kreuzen Sie für jede der folgenden Aufgaben, die beim Entwurf eines Datenbanksystems anfallen, an, in welchem Schritt sie ausgeführt werden.

	Anforderungs- analyse	Konzeptueller Entwurf	Implementations- entwurf	Physischer Entwurf
Erstellung eines ER-Schemas				
Anlegen eines Index				
Erstellung eines relationalen Schemas				
Erstellung eines Pflichtenhefts				
Ballung logisch verwandter Datensätze				
Klassendefinitionen im objektorientierten Modell				

4. (6 Punkte)

Unter der Einbringstrategie versteht man die Methodik, nach der Änderungen in die materialisierte Datenbasis eingebracht werden. Dabei unterscheidet man zwischen zwei verschiedenen Techniken, den *direkten* und den *indirekten* Einbringstrategien. Erläutern Sie den Unterschied zwischen diesen zwei Strategien und geben Sie ein Beispiel für jede Strategie an.

5. In einem verteilten Datenbanksystem werden die Transaktionen T_1 und T_2 abgearbeitet. Die Transaktionen greifen auf die Datenelemente x , y und z zu, die jeweils auf die Stationen S_1 , S_2 und S_3 verteilt sind. Jede Teiltransaktion soll **lokal** (d.h. auf der jeweiligen Station) die 2PL-Regeln erfüllen. Geben Sie für jeden Zugriff explizit die Operationen zum Setzen und Freigeben der Sperren an: $rl_i[a]$: Transaktion i setzt Lesesperre auf Datenelement a , $wl_i[a]$: Transaktion i setzt Schreibsperre auf Datenelement a , $ul_i[a]$: Transaktion i gibt Sperre auf Datenelement a frei. Die Operationen der Transaktionen T_1 und T_2 lauten:

- $T_1 : r_1[x] w_1[z] r_1[y] commit$
- $T_2 : w_2[x] w_2[y] commit$

(a) (5 Punkte)

Geben Sie ein Beispiel einer verteilten Ausführung von T_1 und T_2 an, die nicht serialisierbar ist.

(b) (1 Punkt)

Gegen welche 2PL-Regel wird bei der Ausführung in Teil (a) **global** (d.h. aus Sicht einer gesamten Transaktion) verstoßen?

(c) (4 Punkte)

Angenommen in der verteilten Datenbank würde das strenge 2PL-Verfahren eingesetzt. Könnten jetzt immer noch nicht serialisierbare Pläne erzeugt werden? Begründen Sie Ihre Antwort kurz.

6. Die Änderbarkeit von Sichten wird in SQL sehr restriktiv gehandhabt, d.h. es werden auch Änderungen abgewiesen, die durchaus sinnvoll wären. Gegeben sei eine Datenbank zur Verwaltung des Universitätsbetriebs. Auf den Relationen *Professoren*(PersNr, Name, Rang, Raum) und *Vorlesungen*(VorlNr, Titel, SWS, gelesenVon) wurde folgende Sicht angelegt:

```
create view ProfVorlesung(PersNr, Name, VorlNr, Titel, SWS) as
  select p.PersNr, p.Name, v.VorlNr, v.Titel, v.SWS
  from Professoren p, Vorlesungen v
  where p.PersNr = v.gelesenVon;
```

Die Relationen *Professoren* und *Vorlesungen* enthalten folgende Tupel:

Professoren				Vorlesungen			
2125	Sokrates	C4	226	5041	Ethik	4	2125
2126	Russel	C4	232	4052	Logik	4	2125
2127	Kopernikus	C4	310	5043	Erkenntnistheorie	3	2126
				5123	Astronomie	4	2127

Entscheiden Sie, ob für die folgenden Änderungsoperationen auf der Sicht *ProfVorlesung* Änderungsoperationen auf den Basisrelationen existieren, die die gewünschte Änderung in der Sicht hervorrufen. Falls eine Umsetzung existiert, geben Sie die Änderungsoperationen auf den Basisrelationen an. Falls nicht, begründen Sie Ihre Antwort kurz.

(a) (2 Punkte)

```
update ProfVorlesung
set    Rang='C3'
where  PersNr=2125;
```

(b) (2 Punkte)

```
insert into ProfVorlesung
values(2125, Sokrates, 4920, Philosophie, 4);
```

(c) (2 Punkte)

```
insert into ProfVorlesung
values(2128, Curie, 5228, Radiochemie, 4);
```

(d) (2 Punkte)

```
insert into ProfVorlesung  
values(2127, Galilei, 5125, Kosmologie, 2);
```

(e) (2 Punkte)

```
delete from ProfVorlesung  
where Titel='Logik';
```

(f) (2 Punkte)

```
delete from ProfVorlesung  
where Name='Russel';
```

7. (a) (6 Punkte)

Geben Sie Definitionen für die folgenden Eigenschaften von Historien an:

i. rücksetzbar

ii. vermeidet kaskadierendes Rücksetzen

iii. strikt

(b) (2 Punkte)

Geben Sie für die folgenden Aussagen an, ob sie wahr oder falsch sind. (Bei einer falschen Antwort werden Punkte abgezogen, die Gesamtpunktzahl kann jedoch nicht unter 0 Punkte sinken.)

	wahr	falsch
Die Menge der rücksetzbaren Historien ist eine Untermenge der strikten Historien.	<input type="checkbox"/>	<input type="checkbox"/>
Alle seriellen Historien sind auch serialisierbar.	<input type="checkbox"/>	<input type="checkbox"/>
Historien, die kaskadierendes Rücksetzen vermeiden, sind auch rücksetzbar.	<input type="checkbox"/>	<input type="checkbox"/>
Es existieren Historien, die strikt sind, aber nicht serialisierbar.	<input type="checkbox"/>	<input type="checkbox"/>

8. Eine Versandfirma möchte ihre Lieferaufträge über eine verteilte relationale Datenbank abwickeln. Das Schema der Datenbank enthält unter anderem folgende Relation:

- Auftrag(AuftragsNr, Artikel, Menge, Preis, Steuer, Lieferadresse, Liefermodus)

Eine Abteilung der Firma kümmert sich um die Annahme von Aufträgen. Die häufigsten Anfragen aus dieser Abteilung beziehen sich auf die Attribute *Artikel*, *Menge*, *Preis* und *Steuer*. Eine andere Abteilung macht die Bestellungen versandfertig und interessiert sich hauptsächlich für die Attribute *Artikel*, *Menge*, *Lieferadresse* und *Liefermodus*.

(a) (1 Punkt)

Welche Fragmentierung ist hier sinnvoll (horizontal oder vertikal)?

(b) (6 Punkte)

Zerlegen Sie die Relation in geeignete Fragmente. Geben Sie das Schema jedes Fragments und die SQL-Befehle (insert-Befehle) zum Füllen der Fragmente an.

(c) (10 Punkte)

In der Lagerabteilung wird folgende Anfrage auf dem verteilten Datenbanksystem gestartet:

```
select Artikel, Menge, Preis
from   Auftrag
where  Lieferadresse = 'Meier, Mannheim';
```

i. Welche Transparenz liegt hier vor?

ii. Geben Sie einen kanonischen Operatorbaum für die Anfrage an.

- iii. Optimieren Sie diesen Baum nach den in der Vorlesung vorgestellten Regeln.