

# Hauptdiplomklausur Datenbanksysteme I Sommersemester 1998

Name: .....  
Vorname: .....  
Matrikelnummer: .....  
Studienfach: .....

## Wichtige Hinweise:

1. Prüfen Sie Ihr Klausurexemplar auf Vollständigkeit (19 Seiten).
2. Aufgabe 1 bis 6 sind von allen zu bearbeiten. Aufgabe 7 bis 10 von den Hörern von Prof.Dr. Majster-Cederbaum, Aufgabe 11 bis 13 von den Hörern von Prof.Dr. Moerkotte.
3. Es sind keine Hilfsmittel zugelassen.
4. Die Klausur dauert 100 Minuten.
5. Jede Aufgabe ist auf dem zugehörigen Aufgabenblatt (und ggf. auf separaten Lösungsblättern) zu bearbeiten.
6. Vermerken Sie Ihren Namen und Ihre Matrikelnummer auf jedem Aufgaben- (bzw. Lösungsblatt). Blätter ohne Namens- und Matrikelnummerangabe werden nicht bewertet.
7. Das Deckblatt sowie alle Aufgabenblätter (evtl. Lösungsblätter) sind abzugeben.

	maximale Anzahl Punkte	erreichte Anzahl Punkte
Aufgabe 1	7	
Aufgabe 2	9	
Aufgabe 3	4	
Aufgabe 4	16	
Aufgabe 5	14	
Aufgabe 6	29	
Aufgabe 7 / 11	5 / 9	
Aufgabe 8 / 12	7 / 10	
Aufgabe 9 / 13	4 / 2	
Aufgabe 10 / -	5 / -	
	100	

1. In Abbildung 1 sehen Sie ein ER-Diagramm für einen WWW-Provider.

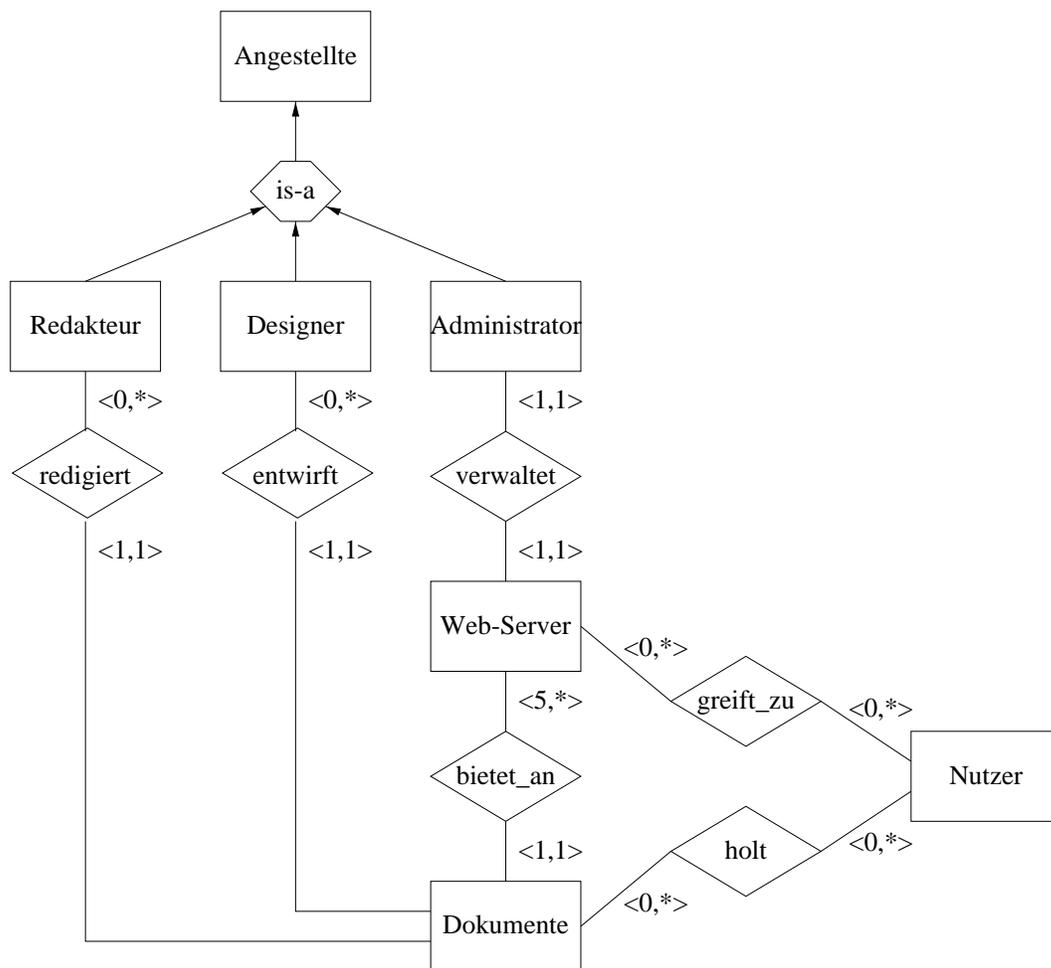


Abbildung 1: ER-Diagramm für WWW-Provider

Beantworten Sie folgenden Fragen zum ER-Diagramm (begründen Sie jeweils Ihre Antwort):

(a) (1 Punkt)

Kann es Web-Server ohne Administrator geben?

(b) (1 Punkt)

Verwaltet ein Administrator mehrere Web-Server?

- (c) (1 Punkt)  
Werden alle Dokumente (von Nutzern) geholt?
- (d) (1 Punkt)  
Wieviel Administratoren sind für ein Dokument verantwortlich?
- (e) (1 Punkt)  
Können Dokumente von einem Designer redigiert werden (disjunkte Spezialisierung vorausgesetzt)?
- (f) (1 Punkt)  
Enthält ein Web-Server immer mehrere Dokumente?
- (g) (1 Punkt)  
Gibt es überflüssige Beziehungen in diesem Diagramm? (D.h. existieren Beziehungen, die aus anderen Beziehungen hergeleitet werden können?)

2. Finden Sie für die folgenden Relationenschemata alle Schlüssel und bestimmen Sie in welcher Normalform sich die Schemata befinden.

(a) (2 Punkte)

$\mathcal{R}(A, B, C, D, E, F)$

mit  $\mathcal{F} = \{\}$

(b) (3 Punkte)

$\mathcal{S}(A, B, C, D, E, F)$

mit  $\mathcal{F} = \{A \rightarrow D, B \rightarrow F, C \rightarrow E, D \rightarrow B, E \rightarrow A, F \rightarrow C\}$

(c) (4 Punkte)

$\mathcal{T}(A, B, C, D, E, F)$

mit  $\mathcal{F} = \{AC \rightarrow D, D \rightarrow EF, F \rightarrow ABC, B \twoheadrightarrow E\}$

3. (4 Punkte)

Geben Sie ein Beispiel für ein Relationenschema (mit einer Menge von funktionalen Abhängigkeiten) an, das in 3NF **nicht** aber in BCNF ist.

4. Gegeben seien die Relationenschemata  $\mathcal{R}(\underline{A}, B, C, D)$  und  $\mathcal{S}(\underline{E}, F, G)$ .  $F$  sei Fremdschlüssel, d.h. die Werte von  $F$  verweisen auf Tupel aus  $R$ .

(a) (4 Punkte)

Welche zwei Eigenschaften müssen die Relationen  $R$  und  $S$  erfüllen, um die referentielle Integrität zu gewährleisten?

Betrachten Sie folgende Ausprägungen der Relationen  $R$  und  $S$  in einer Datenbank:

TupelNr	$R$				TupelNr	$S$		
	A	B	C	D		E	F	G
1	$a_1$	$b_1$	$c_1$	$d_1$	4	$e_1$	$a_1$	$g_1$
2	$a_2$	$b_2$	$c_2$	$d_2$	5	$e_2$	$a_1$	$g_2$
3	$a_3$	$b_3$	$c_3$	$d_3$	6	$e_3$	$a_1$	$g_2$
					7	$e_4$	$a_2$	$g_3$
					8	$e_5$	$a_2$	$g_4$

Angenommen die Schemata  $\mathcal{R}$  und  $\mathcal{S}$  sind folgendermaßen definiert:

```
create table R(
  A integer primary key,
  B, C, D integer);
```

```
create table S(
  E integer primary key,
  F integer references R on update cascade,
  G integer);
```

Was geschieht bei der Ausführung folgender SQL-Befehle? (Geben Sie bei eventuellen Änderungen die betroffenen Tupelnummern an und beschreiben Sie die Änderungen.)

(b) (3 Punkte)

```
update R
set A = a4
where A = a1
```

(c) (3 Punkte)  
update R  
set  $A = a_5$   
where  $A = a_3$

(d) (3 Punkte)  
insert into S  
values  $(e_6, a_6, g_5)$

(e) (3 Punkte)  
delete from R  
where  $A = a_1$

5. Gegeben seien die Transaktionen  $T_1$  und  $T_2$  mit folgenden Operationen:

- $T_1$  :  $w_1[y]$   $w_1[x]$   $c_1$
- $T_2$  :  $r_2[z]$   $w_2[x]$   $r_2[y]$   $c_2$

(a) (8 Punkte)

	(konflikt-)		vermeidet kask.	
	serialisierbar	rücksetzbar	Rücksetzen	strikt
$w_1[y]$ $w_1[x]$ $r_2[z]$ $c_1$ $w_2[x]$ $r_2[y]$ $c_2$	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
$r_2[z]$ $w_1[y]$ $w_1[x]$ $w_2[x]$ $r_2[y]$ $c_2$ $c_1$	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
$r_2[z]$ $w_2[x]$ $w_1[y]$ $r_2[y]$ $w_1[x]$ $c_1$ $c_2$	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
$r_2[z]$ $w_1[y]$ $w_2[x]$ $w_1[x]$ $c_1$ $r_2[y]$ $c_2$	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

(b) (3 Punkte)

Geben Sie ein Beispiel für eine nicht-serielle Historie (mit  $T_1$  und  $T_2$ ) an, die von einem strengen 2-PL Scheduler erzeugt werden kann. Geben Sie dabei explizit die Sperr- und Entsperrbefehle an. Schreiben Sie Befehle, die gleichzeitig ausgeführt werden, auf gleicher Höhe. Verwenden Sie dabei folgende Syntax:

- $rl_i[a]$   $T_i$  setzt Lesesperre auf  $a$
- $wl_i[a]$   $T_i$  setzt Schreibsperre auf  $a$
- $ul_i[a]$   $T_i$  gibt aktuelle Sperre auf  $a$  frei

(c) (3 Punkte)

Geben Sie ein Beispiel für eine nicht-serielle Historie (mit  $T_1$  und  $T_2$ ) an, die von einem (normalen) 2-PL Scheduler **nicht** aber von einem strengen 2-PL Scheduler erzeugt werden kann. (Geben Sie auch hier wieder die Sperr- und Entsperrbefehle an.)

6. Die Informationen über das Streckennetz einer Bahngesellschaft werden in einer relationalen Datenbank gehalten. Die Datenbank besitzt folgendes Schema:

- Bahnhof(Bahnhofname, AnzahlGleise, Stadt)
- Strecke(StreckenNr, vonBahnhof, nachBahnhof, Entfernung)
- Zug(ZugNr, Zugtyp)
- Verbindung(StreckenNr, ZugNr, Kosten, Fahrkartenpreis, PersonenProJahr, Fahrzeit)

Die Relation *Bahnhof* enthält allgemeine Informationen über die angefahrenen Bahnhöfe der Gesellschaft. In *Strecke* werden für jede Strecke die Endbahnhöfe und die Entfernung gespeichert. *vonBahnhof* und *nachBahnhof* sind Fremdschlüssel in *Strecke*. *Zug* gibt an, um welchen Typ es sich handelt (z.B. Schnellzug, IC, etc.). In *Verbindung* wird gespeichert, welche Strecken von welchen Zügen gefahren werden (ein Zug fährt in der Regel mehrere Strecken während einer Reise, z.B.: Hamburg-Hannover, Hannover-Kassel, Kassel-Frankfurt). Außerdem sind hier die Kosten zum Unterhalten des Zuges (pro Jahr), der Preis einer Fahrkarte für diese Verbindung, die Anzahl der beförderten Personen (pro Jahr) und die Fahrzeit (in Minuten) zu finden. *StreckenNr* und *ZugNr* sind Fremdschlüssel.

Formulieren Sie folgenden Anfragen in SQL:

(a) (5 Punkte)

Geben Sie die Namen aller Städte aus, die von der Bahngesellschaft angefahren werden (dabei soll jede Stadt maximal einmal in der Liste auftauchen!).

Bahnhof(Bahnhofname, AnzahlGleise, Stadt)

Strecke(StreckenNr, vonBahnhof, nachBahnhof, Entfernung)

Zug(ZugNr, Zugtyp)

Verbindung(StreckenNr, ZugNr, Kosten, Fahrkartenpreis, PersonenProJahr, Fahrzeit)

---

(b) (8 Punkte)

Geben Sie zu jeder Strecke die schnellste Zugverbindung (mit StreckenNr, ZugNr, Zugtyp und verbundenen Bahnhöfen) aus.

Bahnhof(Bahnhofname, AnzahlGleise, Stadt)

Strecke(StreckenNr, vonBahnhof, nachBahnhof, Entfernung)

Zug(ZugNr, Zugtyp)

Verbindung(StreckenNr, ZugNr, Kosten, Fahrkartenpreis, PersonenProJahr, Fahrzeit)

---

(c) (8 Punkte)

Geben Sie für jeden Zugtyp die Personenkilometer (beförderte Personen mal Kilometer) pro Jahr an.

Bahnhof(Bahnhofname, AnzahlGleise, Stadt)

Strecke(StreckenNr, vonBahnhof, nachBahnhof, Entfernung)

Zug(ZugNr, Zugtyp)

Verbindung(StreckenNr, ZugNr, Kosten, Fahrkartenpreis, PersonenProJahr, Fahrzeit)

---

(d) (8 Punkte)

Im Rahmen von Rationalisierungsmaßnahmen sollen unrentable Züge stillgelegt werden. Eine Zug gilt als unrentabel, wenn die Kosten zur Unterhaltung des Zuges die Einnahmen übersteigen. Geben Sie alle unrentablen Züge mit ZugNr und Zugtyp aus.

## Teil II (für die Hörer von Prof. Dr. Majster-Cederbaum)

7. (5 Punkte)

Gegeben seien die folgenden Relationenschemata:

- $\text{liegt\_in}(\underline{\text{Ort}}, \text{Land})$
- $\text{Dolmetscher\_wohnt\_in}(\underline{\text{Name}}, \text{Ort})$
- $\text{kann\_Sprache}(\underline{\text{Name}}, \underline{\text{Sprache}})$
- $\text{Landessprache}(\underline{\text{Land}}, \underline{\text{Sprache}})$

wobei die Schlüssel durch Unterstreichung gekennzeichnet sind.

Das Heimatland eines Dolmetschers ist das Land, in dem sich der Wohnort befindet. Gehen Sie außerdem davon aus, daß jeder Ort, der Wohnort eines Dolmetschers ist in der Relation  $\text{liegt\_in}$  aufgeführt ist. Länder können mehrere Landessprachen haben. Jedes Land hat mindestens eine Landessprache und jeder Dolmetscher kann mindestens zwei Sprachen.

Beschreiben Sie die folgende Anfrage in der Relationenalgebra:

Geben Sie die Namen aller Dolmetscher aus, die nicht jede Sprache ihres Heimatlandes sprechen.

8. Seien  $T_1, T_2, \dots, T_n$  Transaktionen und  $H$  eine Historie (die die Transaktionen  $T_1$  bis  $T_n$  enthält?).

(a) (2 Punkte)

Geben Sie die Definition des Konfliktgraphen von  $H$ .

(b) (5 Punkte)

Zeigen Sie: Wurde die Historie  $H$  mittels des Zweiphasenprotokolls erstellt, so ist sie konfliktserialisierbar.

9. Gegeben sei ein Relationenschema  $\mathcal{R}$ . Seien  $K_1$  und  $K_2$  zwei (verschiedene) Schlüssel für  $\mathcal{R}$ . Beweisen bzw. widerlegen Sie:

(a) (2 Punkte)  
 $K_1 \cap K_2$  ist ein Schlüssel.

(b) (2 Punkte)  
 $K_1 \cup K_2$  ist kein Schlüssel.

10. (5 Punkte)

Gegeben sei ein Relationenschema  $\mathcal{R}$  mit funktionalen Abhängigkeiten  $\mathcal{F}$ .

Zeigen Sie: Wenn  $X \subseteq Y \subseteq \mathcal{R}$ , dann  $X_{\mathcal{F}}^+ \subseteq Y_{\mathcal{F}}^+$ .

- Um Speicherplatz zu sparen und indirekte Zugriffe zu reduzieren, wurden in einem Datenbanksystem Relationen folgendermaßen auf den Hintergrundspeicher abgebildet. Es wurde auf die interne Datensatztabelle (Slots) verzichtet, ein Tupelidentifikator (TID) besteht aus einer Seitennummer und einer Position, an der das Tupel auf der Seite zu finden ist (siehe Abbildung 2 für ein Beispiel).

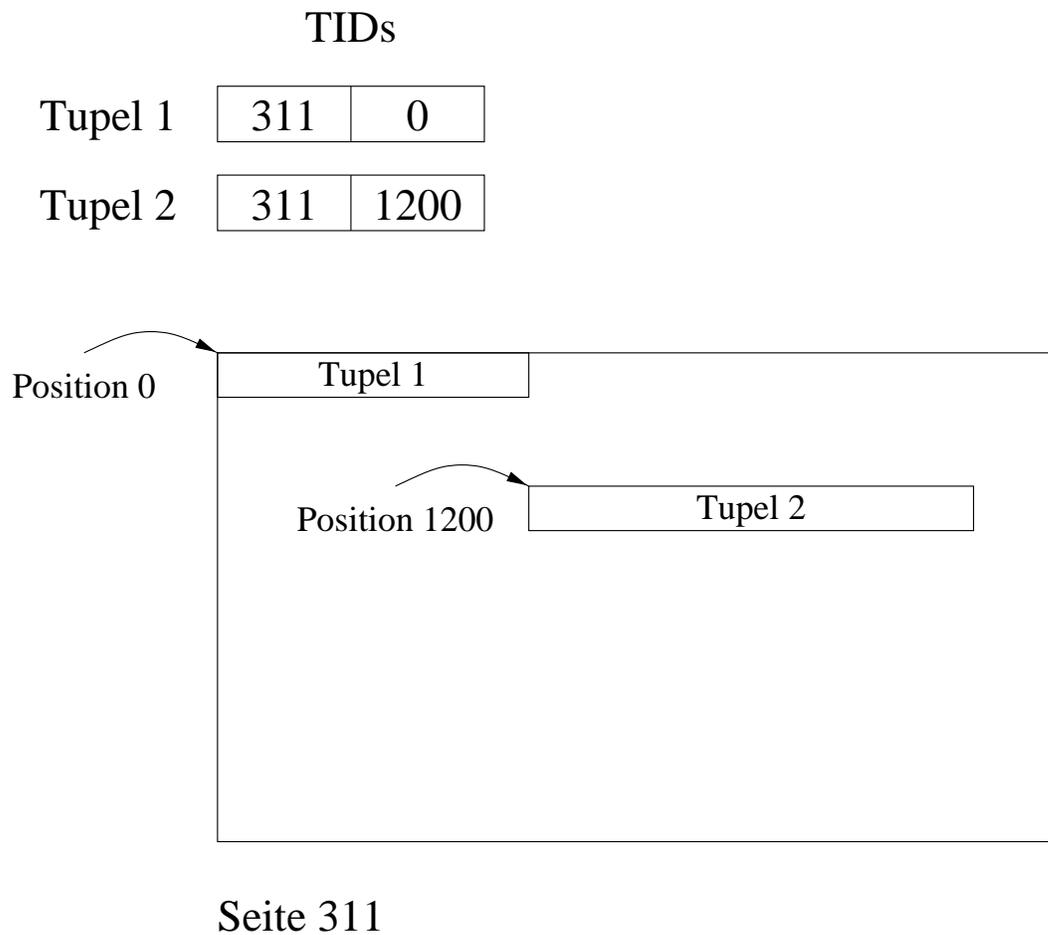


Abbildung 2: Beispiel für speicherplatzsparende TIDs

(a) (3 Punkte)

Können Tupel von einer Seite auf eine andere verschoben werden, ohne die TID zu ändern? Begründen Sie Ihre Antwort.

(b) (3 Punkte)

Können Tupel innerhalb einer Seite verschoben werden, ohne die TID zu ändern? Begründen Sie Ihre Antwort.

(c) (3 Punkte)

In welchem Fall ist das obige Verfahren dem Verfahren mit Datensatztable überlegen?

12. Die Produktdaten einer Firma werden in einer deduktiven Datenbank mit folgendem Relationenschema gehalten:

- Bauteil(Bauteiltyp, Gewicht, KonstrukteurID)
- besteht\_aus(Bauteil, Komponente, Menge)
- Konstrukteur(KonstrukteurID, Name, Geburtsdatum)

Die Relation *Bauteil* beschreibt das Gewicht eines Bauteiltyps und gibt den Konstrukteur an, der diesen Bauteiltyp entworfen hat. Die Relation *besteht* gibt an, aus welchen und jeweils wievielen Einzelkomponenten ein Bauteil besteht. In *Konstrukteur* sind persönliche Daten zu den Konstrukteuren gespeichert.

Formulieren Sie folgende Anfragen in DATALOG:

(a) (5 Punkte)

Geben Sie alle Bauteile an, aus denen ein Fahrgestell besteht.

(b) (5 Punkte)

Geben Sie alle Bauteile an, an denen der Konstrukteur Schmidt direkt oder indirekt (er hat eine Komponente davon entworfen) beteiligt ist.

13. Die folgenden Fragen beziehen sich auf das ARIES-Protokoll. (Zur Erinnerung: ARIES ist das im Buch vorgestellte Protokoll für den Wiederanlauf.)

(a) (1 Punkt)

Wodurch wird die Idempotenz in der Redo-Phase sichergestellt?

- durch Anlegen eines Log-Records in der Redo-Phase (Redo-Log-Record)
- durch Schreiben der Log-Sequence-Number (LSN) auf die betreffende Seite
- durch Anlegen eines Sicherungspunktes
- durch einen Vermerk im Log-Record der ursprünglichen Operation

(b) (1 Punkt)

Wodurch wird die Idempotenz in der Undo-Phase sichergestellt?

- durch Anlegen eines Log-Records in der Undo-Phase (Compensation-Log-Record)
- durch Schreiben der Log-Sequence-Number (LSN) auf die betreffende Seite
- durch Anlegen eines Sicherungspunktes
- durch einen Vermerk im Log-Record der ursprünglichen Operation