

Gruppe A

Bitte tragen Sie **SOFORT** und **LESERLICH** Namen und Matrikelnr. ein und legen Sie Ihren Studentenausweis bereit.

PRÜFUNG AUS DATENBANKSYSTEME			18. 10. 1999
Kennnr.	Matrikelnr.	Familienname	Vorname

Arbeitszeit: 120 Minuten. Die Aufgaben sind auf den Angabeblättern zu lösen; Zusatzblätter werden nicht gewertet.

**Aufgabe 1:**

(25)

Für eine Behörde, die geheime Dokumente verwaltet, ist eine Datenbank zu erstellen.

Unterstreichen Sie je Relation einen Schlüssel. Verwenden Sie nur die vorgegebenen Attributnamen. (Diese sind nur bei ihrer jeweils ersten Erwähnung angeführt.) Führen Sie keine zusätzlichen Attribute ein und verwenden Sie möglichst wenige Relationen. Die Datenbank unterstützt keine Nullwerte. Für eine Relation, die nicht in 3NF ist, gibt es keine Punkte!

Jeder Beamte der Behörde, der zu den Dokumenten Zugang hat, hat eine eindeutige Personalnr (PNR), einen Namen (NAME) und ein bestimmtes Bearbeitungslevel. Von den Beamten wird außerdem jeder Zutritt zur Behörde vermerkt, indem gespeichert wird, wann er die Behörde betritt (BEGINN) und wieder verlässt (ENDE).

Jedes Level hat eine Bezeichnung (LBEZ) sowie eine eindeutige Nummer (LNR) und beinhaltet verschiedene Rechte. Diese Rechte (z. B. Leserecht, Schreibrecht, Löschrecht,...) haben eine Bezeichnung (RBEZ) und eine eindeutige Nummer (RNR) und dienen dazu, den Zugriff der Beamten zu reglementieren.

Die geheimen Dokumente haben eine eindeutige Dokumentennummer (DNR), einen Textteil (TEXT) und können auf andere Dokumente verweisen. Weiters hat jedes Dokument ein Erstelldatum (EDAT) und ein Datum, das den letzten Zugriff angibt (ZDAT). Es muß ersichtlich sein, wann (ZDAT) und in welcher Form (AKTION) wie z.B. Lesen, Schreiben, Löschen.. ein Beamter auf ein Dokument Zugriff gehabt hat.

beamter ( <u>PNR</u> , NAME, LNR	)
level ( <u>LNR</u> , LBEZ	)
rechte ( <u>RNR</u> , RBEZ	)
rechtezuordnung ( <u>LNR</u> , <u>RNR</u>	)
zutritt ( <u>PNR</u> , <u>BEGINN</u> , ENDE	)
dokument ( <u>DNR</u> , TEXT, EDAT, ZDAT	)
verweis ( <u>DNR</u> , <u>DNR1</u>	)
zugriff ( <u>PNR</u> , <u>DNR</u> , <u>ZDAT</u> , AKTION	)
(	)

**Aufgabe 2:**

(34)

In einer Relationalen Datenbank der Bundestheater wird der Verkauf der Theaterkarten organisiert.

<b>saal</b> ( <u>snr</u> , sname)	<b>platz</b> ( <u>snr</u> , <u>lage</u> , <u>reihe</u> , <u>pnr</u> ,preis)
<b>vorstellung</b> ( <u>snr</u> , <u>datum</u> ,titel)	<b>belegt</b> ( <u>snr</u> , <u>lage</u> , <u>reihe</u> , <u>pnr</u> , <u>datum</u> )

Jeder Saal wird identifiziert durch eine Nummer (**snr**) und hat einen Namen (z.B. Burgtheater). Zu jedem Platz **platz** im Saal (Parkett\_recht, Reihe 3, Platz 5) wird der Preis des Platzes gespeichert. Weiters werden Vorstellungen **vorstellung**, identifiziert durch Aufführungsort und Datum, mit ihrem Titel (z.B. Don Carlos) gespeichert. Zu jeder Vorstellung werden die verkauften Karten in der Relation **belegt** gespeichert.

a) Schreiben Sie eine Abfrage in SQL, die zu den Vorstellungen vom 26.10.1999 die freien Plätze und deren Preis ausgibt.

[8]

```
select *
from platz p, vorstellung v
where p.snr = v.snr and
      v.datum = 26.10.1999 and
      p.lage, p.reihe, p.pnr not in
      (select b.lage, b.reihe, b.pnr
       from belegt b
       where b.datum = 26.10.1999 and
            b.snr = p.snr);
```

b) Schreiben Sie eine Abfrage in SQL, die Titel und Datum aller ausverkauften Vorstellungen des Akademietheaters (AK) in der Spielzeit 1998/99 (von 1.9.98 bis 30.6.99) ausgibt.

[10]

```
select titel, datum
from vorstellung v, saal s, belegt b
where s.snr = v.snr and
      s.sname = 'AK' and
      v.datum = b.datum and
      v.snr = b.snr and
      v.datum between 1.9.1998 and 30.6.1999
group by titel,datum,s.snr
having count(*) = select count(*)
                   from platz
                   where snr = v.snr
```

c) Schreiben Sie für die Abfrage b) eine Abfrage in Relationaler Algebra

[7]

$$\pi_{titel, datum, lage, reihe, pnr} \sigma_{sname=AK \wedge 1.9.1998 \leq datum \leq 30.6.1999} (saal \bowtie vorstellung \bowtie belegt) \div \pi_{lage, reihe, pnr} \sigma_{sname=AK} (saal \bowtie platz)$$

d) Schreiben Sie ein stratifiziertes Datalog-Programm, das im Prädikat  $f(Reihe, von, bis)$  alle jene Reihen im Parkett rechts im Burgtheater ausgibt, für die für die Vorstellung vom 1.1.2000 mehr als 4 Plätze durchgehend frei sind. (Die Prädikate  $SUM(X, Y, Z) \ X=Y+Z$  und  $GR(X, Y) \ X>Y$  stehen zur Verfügung.)

[9]

```
frei(Reihe, PNR) :- saal(SNR, burgtheater), platz(SNR, parkett_rechts, Reihe, PNR, P),
                  non belegt(SNR, parkett_rechts, Reihe, PNR, 1.1.2000).

freireihe(Reihe, PNR, PNR) :- frei(Reihe, PNR).
freireihe(Reihe, von, bis) :- frei(Reihe, bis), freireihe(Reihe, von, bis'),
                              SUM(bis, bis', 1).

f(Reihe, von, bis) :- freireihe(Reihe, von, bis), SUM(bis, von, X), GR(X, 4).
```

### Aufgabe 3:

(7)

Gegeben sind die Relationenschemata  $p(CDE)$ ,  $q(ACDE)$  und  $r(BCD)$ .

Optimieren Sie den nachfolgenden algebraischen Ausdruck:

$$\pi_{AC} \sigma_{(A=3 \vee B=6) \wedge C=1} \left[ (p \cap \pi_{CDE} q) \bowtie (q \bowtie r) \right]$$

$$\pi_{AC} \left( (\sigma_{C=1} p \cap \pi_{CDE} \sigma_{C=1} q) \bowtie \pi_{ACDE} \sigma_{A=3 \vee B=6} (\sigma_{C=1} q \bowtie \sigma_{C=1} r) \right)$$

### Aufgabe 4:

(15)

Bestimmen Sie für jedes der Relationenschemata  $R_i(R_i, F_i)$ ,  $i \in \{1, 2, 3\}$ , alle Schlüssel. Geben Sie eine verlustfreie und abhängigkeits-treue Zerlegung in 3NF mit möglichst wenig Relationen an. Unterstreichen Sie in jeder Relation der Zerlegung einen Schlüssel.

Es gelte:  $R_i = ABCDEF$ ,  $i \in \{1, 2, 3\}$

$F_1 = \{ CD \rightarrow A, AB \rightarrow F, AF \rightarrow E \}$

$F_2 = \{ A \rightarrow DEF, B \rightarrow C, E \rightarrow B \}$

$F_3 = \{ AF \rightarrow C, CF \rightarrow D, CD \rightarrow E, BC \rightarrow F \}$

RS	Schlüssel	Zerlegung in 3NF (Einen Schlüssel in jeder Relation unterstreichen)			
R1	BCD .....	R1-1 <u>CDA</u> .....	R1-2 <u>ABF</u> .....	R1-3 <u>AFE</u> .....	R1-4 <u>BCD</u> .....
R2	A .....	R2-1 <u>BC</u> .....	R2-2 <u>EB</u> .....	R2-3 <u>ADEF</u> .....	R2-4 .....
R3	ABC, ABF ..	R3-1 <u>CDE</u> .....	R3-2 <u>CFD</u> .....	R3-3 <u>ABCF</u> .....	R3-4 .....

**Aufgabe 5:**

(6)

Gegeben ist das Relationenschema ABCDEF und die Menge **F** von funktionalen Abhängigkeiten. Bestimmen Sie eine minimale Überdeckung.

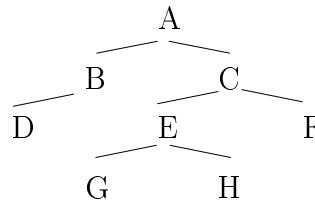
$$\mathbf{F} = \{ AD \rightarrow B, A \rightarrow CDF, C \rightarrow D, CE \rightarrow E \}.$$

$A \rightarrow B, A \rightarrow C, A \rightarrow F, C \rightarrow D$

**Aufgabe 6:**

(7)

Concurrency Control:



Erfüllen die folgenden Transaktionen  $T_1$  und  $T_2$  das Baumprotokoll? Wenn nicht, geben Sie das erste Objekt an, das falsch gesperrt wurde.

$T_1$ : lock C, lock E, lock H, unlock E, lock F, unlock C, unlock H, unlock F

$T_2$ : lock A, lock B, lock D, unlock D, unlock B, unlock A

[4]

- ☒ ja  
☐ nein, weil

Sind alle möglichen gültigen Ausführungen von  $T_1$  und  $T_2$  serialisierbar? (mit Begründung !)

[3]

- ☒ ja, weil sie das Baumprotokoll erfüllen.  
☐ nein, weil

**Aufgabe 7:**

(6)

a) Nennen Sie 2 Anforderungen bei der Modellierung von heutigen DBS, bei denen die klassischen Datenmodelle an ihre Grenzen stoßen:

[3]

1) Modellierung zusammengesetzter Objekte wie z.B. rekursive oder überlappende Objekte  
2) Modellierung von Beziehungen beliebiger Art zwischen verschiedenen Elementen  
3) Modellierung spezieller Datentypen wie z.B. Vektoren, Matrizen, geometrische Figuren

b) Charakterisieren Sie ein voll objektorientiertes DBS (nach K.Dittrich):

[3]

DBS, das alle Eigenschaften eines klassischen DBS aufweist und zusätzlich die folgenden Konzepte unterstützt: Objektidentität, Zusammensetzung von Objekten, Klassen insbesondere auch systembenutzerdefinierte Klassen, Berechnungsvollständigkeit, Einkapselung, Klassenhierarchien und Vererbung sowie Konzepte wie Überladen, Überschreiben und spätes Binden.

Gesamtpunkte: 100