

Gruppe A

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

PRÜFUNG AUS DATENBANKSYSTEME			30. 11. 2000
Kennnr.	Matrikelnr.	Familienname	Vorname

Arbeitszeit: 120 Minuten. Die Aufgaben sind auf den Angabebältern zu lösen.

**Aufgabe 1:**

(25)

Es ist eine Datenbank eines internationalen Seegerichtshofs zu modellieren. 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!

Parteien sind Firmen oder Personen. Sie haben eine Anschrift (ANSCHRIFT) und einen Namen (NAME), die zusammen eindeutig sind, sowie eine Bankverbindung (KNR,BANK).

Verträge haben eine eindeutige Vertragsnummer (VNR) sowie eine Bezeichnung (BEZ) und ein Datum (DATUM). Sie bestehen aus pro Vertrag durchnumerierten Paragraphen (PNR), für die jeweils der entsprechende Vertragstext gespeichert ist (TEXT). Für jeden Vertrag ist vermerkt, welche Länder (LAND) ihm beigetreten sind (mit Beitrittsdatum BDATUM).

Verfahren (Gerichtsfälle oder Schlichtungsverfahren) haben eine laufende Nummer (GNR) pro Einreichungstag, ein Datum (TAG), sowie (möglicherweise mehrere) beteiligte Parteien, einen zuständigen Richter (RICHTER) und Staatsanwalt (SANW). Für Schlichtungsverfahren ist zusätzlich gespeichert, auf welche Paragraphen eines Vertrages sie sich beziehen und innerhalb welcher Frist (FRIST) eine Entscheidung vorliegen muss.

Urteile beziehen sich auf ein Verfahren (und es wird nur ein Urteil pro Verfahren gespeichert). Vermerkt ist das Datum (DATUM) des Urteils sowie sein Text (UTEXT).

partei ( <u>ANSCHRIFT</u> , <u>NAME</u> , KNR, BANK	)
vertrag ( <u>VNR</u> , BEZ, DATUM	)
paragraph ( <u>VNR</u> , <u>PNR</u> , TEXT	)
beitritt ( <u>VNR</u> , <u>LAND</u> , DATUM	)
verfahren ( <u>TAG</u> , <u>GNR</u> , RICHTER, SANW	)
partei2verf ( <u>ANSCHRIFT</u> , <u>NAME</u> , TAG, GNR	)
schlichtung ( <u>TAG</u> , <u>GNR</u> , FRIST	)
schlichtung2par ( <u>TAG</u> , <u>GNR</u> , <u>VNR</u> , <u>PNR</u>	)
urteil ( <u>TAG</u> , <u>GNR</u> , DATUM, UTEXT	)
(	)

**Aufgabe 2:**

(34)

In einer Relationalen Datenbank der Wiener Philharmoniker werden die folgenden Informationen über die Musiker und Konzerte gespeichert:

<code>konzert(<u>datum</u>,saal)</code>	<code>musiker(<u>svnr</u>,name,geburt,gruppe)</code>
<code>stück(<u>datum</u>,<u>titel</u>,<u>komponist</u>)</code>	<code>spielt(<u>datum</u>,<u>titel</u>,<u>komponist</u>,<u>svnr</u>)</code>

Zu jedem Konzert wird gespeichert, wann (`datum`) und wo es stattgefunden hat, Mitarbeiter haben eine Sozialversicherungsnummer, einen Namen, ein Geburtsdatum und gehören einer Instrumentengruppe an. Weiters wird gespeichert, welche Werke (`titel`, `komponist`) in einem Konzert aufgeführt werden und welche Musiker diese Werke bei einem bestimmten Konzert spielen.

a) Schreiben Sie eine Abfrage in SQL, die zu jedem Konzert im Musikverein das Werk angibt, bei dem die meisten Musiker mitgespielt haben sowie die Anzahl der Musiker. [10]

```
select k.datum, sp.titel, sp.komponist, count (*)
from konzert k, spielt sp
where k.datum = sp.datum and
      k.saal='Musikverein'
group by k.datum, sp.titel, sp.komponist
having count(*) >= all (select count (*)
                        from spielt
                        where datum = k.datum
                        group by titel, komponist);
```

b) Schreiben Sie eine Abfrage in SQL, die alle Konzerte (Datum und Saal) ausgibt, bei denen mindestens ein Werk aufgeführt wurde, in dem keine Streicher gespielt haben. [8]

```
select k.datum, k.saal
from konzert k, spielt s
where k.datum = s.datum and
      s.datum, s.titel, s.komponist not in (select datum, titel, komponist
                                           from spielt, musiker m
                                           where spielt.svnr = m.svnr and
                                           m.gruppe = 'streicher');
```

c) Schreiben Sie die Abfrage b) in Relationaler Algebra.

[7]

$$\pi_{\text{datum,saal}} \left[ \text{konzert} \bowtie \left( \pi_{\text{datum,komponist,titel}} \text{spielt} - \pi_{\text{datum,komponist,titel}} \sigma_{\text{gruppe='streicher'}} (\text{spielt} \bowtie \text{musiker}) \right) \right]$$

d) Schreiben Sie ein stratifiziertes Datalog-Programm, das im Prädikat STR(T,K) alle Werke ausgibt, bei denen entweder keine Streicher oder nur Streicher spielen. (Die Prädikate = und <> stehen zur Verfügung.).

[9]

```

ein_streicher(D,T,K)  :- spielt(D,T,K,S), musiker(S,_,_,streicher).
kein_streicher(D,T,K) :- stueck(D,T,K), not ein_streicher(D,T,K).

nicht_streicher(D,T,K) :- spielt(D,T,K,S), musiker(S,_,_,G), G <> streicher .
nur_streicher(D,T,K)   := stueck(D,T,K), not nicht_streicher(D,T,K).

STR(T,K) :- kein_streicher(D,T,K).
STR(T,K) :- nur_streicher(D,T,K).

```

### Aufgabe 3:

(7)

Gegeben sind die Relationenschemata:  $p(ACD)$ ,  $q(AD)$ ,  $r(CB)$ . Optimieren Sie den folgenden algebraischen Ausdruck:

$$\pi_{AD} \sigma_{(A=a) \wedge ((D=d) \vee (C=c))} \left[ \pi_{ACD} (p \bowtie r) \cup (p \cap \pi_{ACD} (q \bowtie r)) \right]$$

$$\pi_{AD} \left( \sigma_{(A=a) \wedge (D=d \vee C=c)} p \bowtie \pi_C r \right) \cup \pi_{AD} \left[ \sigma_{(A=a) \wedge (D=d \vee C=c)} p \cap \left( \sigma_{D=d \vee C=c} (\sigma_{(A=a)} q \bowtie \pi_C r) \right) \right]$$

### 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 abhängigkeitsreue Zerlegung in 3NF mit möglichst wenig Relationen an. Unterstreichen Sie in jeder Relation der Zerlegung einen Schlüssel.

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

$F_1 = \{AB \rightarrow C, CD \rightarrow E, EF \rightarrow G, G \rightarrow A\}$

$F_2 = \{A \rightarrow B, A \rightarrow C, DE \rightarrow C, C \rightarrow FG\}$

$F_3 = \{A \rightarrow B, B \rightarrow CD, D \rightarrow EF, E \rightarrow FG\}$

RS	Schlüssel	Zerlegung in 3NF (Einen Schlüssel in jeder Relation unterstreichen)			
R1	BDF[A C E G]	R1-1 <u>ABDF</u> CEG ..	R1-2 .....	R1-3 .....	R1-4 .....
R2	ADE .....	R2-1 <u>C</u> FG .....	R2-2 <u>A</u> BC .....	R2-3 <u>D</u> EC .....	R2-4 <u>ADE</u> .....
R3	A .....	R3-1 <u>E</u> FG .....	R3-2 <u>D</u> E .....	R3-3 <u>B</u> CD .....	R3-4 <u>AB</u> .....

**Aufgabe 5:**

(6)

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

$$\mathbf{F} = \{ABC \rightarrow DEG, B \rightarrow C, C \rightarrow F, C \rightarrow H, H \rightarrow E\}.$$

$$AB \rightarrow D, AB \rightarrow G, B \rightarrow C, C \rightarrow F, C \rightarrow H, H \rightarrow E$$

**Aufgabe 6:**

(7)

Die nachfolgende parallele Ausführung von Transaktionen wird mit Hilfe von Zeitstempeln synchronisiert.

a) Geben Sie die Werte der Zeitstempel an (soweit möglich); markieren Sie einen Transaktionsabbruch mit '\*'. [4]

$T_1$	$T_2$	$T_3$	a	b	c
10	20	30	RT=WT=0	RT=WT=0	RT=WT=0
read b				RT= 10	
	read a		RT= 20		
		write b		WT= 30	
read a			RT= 20		
	read c				RT= 20
	write b			WT= 30	
		write a	WT= 30		
		write b		WT= 30	
write b				WT= 30	

b) Welche Reihenfolge wird zwischen den Transaktionen durch die Zeitstempel festgelegt ( $T_i < T_j \equiv "T_i \text{ vor } T_j"$ )? Annahme: Eine abgebrochene Transaktion wird sofort nach dem Abbruch wieder gestartet. [3]

$$T_1 < T_2 < T_3$$

**Aufgabe 7:**

(6)

Können Sie die folgende SQL Abfrage so umschreiben, dass keine Rekursion verwendet wird? Wenn ja, schreiben Sie die Abfrage, wenn nein, erklären Sie warum nicht.

```
with recursive X as
  ((select svnr, name from musiker where gruppe='streicher')
   union
   (select m.svnr, X.name from musiker m, X where m.name=X.name))
select * from X;
```

Diese Abfrage selektiert den Namen und die SVN-R aller Musiker, die entweder Streicher sind oder denselben Namen wie ein Streicher haben. Die Verwendung der Rekursion ist überflüssig, da in der Rekursion keine neuen Namen dazukommen, die die Rekursion notwendig machen würden:

```
select m2.svnr, m2.name
from musiker m1, musiker m2
where m1.gruppe= 'streicher' and
      m1.name=m2.name;
```

Gesamtpunkte: 100