

Gruppe A

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

PRÜFUNG AUS DATENBANKSYSTEME (181.038)			22. 10. 2001
Kennnr.	Matrikelnr.	Familienname	Vorname

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

Aufgabe 1: (25)

Modellieren Sie die Datenbank für eine Laborübung an der TU. Unterstreichen Sie je Relation einen Schlüssel. Verwenden Sie nur die vorgegebenen Attributnamen. (Diese sind üblicherweise 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!

Für jede Studentin sind der Name (NAME), eine eindeutige Matrikelnummer (MNR), und eine Emailadresse (MAIL) gespeichert. Einige dieser Studentinnen absolvieren die Übung nicht mehr selbst, sondern arbeiten als Tutorinnen. Dafür werden zusätzlich Kontonummer (KONTO) und Bankleitzahl (BANK) gespeichert.

Es gibt eine Reihe von Übungsbeispielen, und bei jedem dieser Beispiele mehrere Gruppen. Für jedes Beispiel sind neben eindeutiger Nummer (NR) eine Kurzbeschreibung des Lernzieles (ZIEL) sowie der letztmögliche Abgabetermin (DL) vorge-
merkt. (Abgaben können jederzeit bis zu diesem Termin erfolgen.) Die einzelnen Angaben sind mit Beschreibung (BESCHR) sowie Beispielnnummer und -gruppe (NR, GRUPPE; zusammen eindeutig) gespeichert.

Jede Studentin muss von jedem Beispiel eine Gruppe lösen, wobei die konkrete Zuordnung zu Semesterbeginn vorgenommen und in der Datenbank gespeichert wird. Im Rahmen eines Abgabegesprächs wird das Beispiel danach bei einer Tutorin abgegeben, wobei jeder Studentin fix eine Tutorin zugeordnet ist. Von diesen Gesprächen werden weiters Datum und Uhrzeit (DATUM, ZEIT) gespeichert.

Für den Fall dass eine Abgabe negativ war, besteht die Möglichkeit eines weiteren Abgabetermins bei einer beliebigen Studienassistentin, wobei auch hier zusätzlich Datum und Uhrzeit (DATUM, ZEIT) gespeichert werden. Eine Studienassistentin ist eine Tutorin mit zusätzlicher Qualifikation (QUALI), die eine Reihe von Tutorinnen betreut und auch über eine Telefonnummer am Institut (TEL) verfügt; jede Tutorin ist genau einer Studienassistentin zugeordnet.

stud (<u>MNR</u> , NAME, MAIL, TUTOR_MNR) [3]
tutor (<u>MNR</u> , KONTO, BANK, STUDASS_MNR) [3]
studass (<u>MNR</u> , QUALI, TEL) [3]
beispiel (<u>NR</u> , ZIEL, DL) [3]
angabe (<u>NR</u> , <u>GRUPPE</u> , BESCHR) [3]
studbsp (<u>MNR</u> , <u>NR</u> , GRUPPE) [3]
termin (<u>MNR</u> , <u>NR</u> , DATUM, ZEIT) [3]
termin2 (<u>MNR</u> , <u>NR</u> , STUDASS_MNR, DATUM, ZEIT) [4]
()
()

Aufgabe 2:

(34)

In der Relationalen Datenbank des österreichischen Sportverbandes ist die folgende Information gespeichert:

```
mitgl(mnr, name, adresse, seit, snr)    sektion(snr, sname)
beitrag(bnr, jahr, beschr, euro)        bezahlt(mnr, jahr, bnr)
```

Der Sportverband ist in verschiedene Sektionen geteilt, von den Mitgliedern kennen wir die Mitgliedsnummer, Name, Adresse, Beitrittsdatum und wissen, zu welcher Sektion sie gehören. Die verschiedenen Beitragskategorien sind durchnummeriert und der Betrag pro Jahr und die Beschreibung werden gespeichert. Weiters ist bekannt, welches Mitglied wann welchen Beitrag bezahlt hat (ein Beitrag pro Jahr).

a) Schreiben Sie eine Abfrage in SQL, die Name und Adresse sowie den Namen der Sektion aller Mitglieder ausgibt, die im Jahr 2000 den höchsten Beitrag in der jeweiligen Sektion bezahlt haben, sowie diesen Beitrag. [11]

```
select m.name, m.adresse, s.sname, b.euro
from mitgl m, sektion s, beitrag b, bezahlt z
where m.snr = s.snr and
      z.mnr = m.mnr and
      z.bnr = b.bnr and
      z.jahr = b.jahr and
      z.jahr = '2000' and
      b.euro >= all (select max(b1.euro)
                    from beitrag b1, bezahlt z1, mitgl m1
                    where z1.jahr = '2000' and
                          b1.jahr = z1.jahr and
                          b1.bnr = z1.bnr and
                          z1.mnr = m1.mnr and
                          m1.snr = m.snr);
```

b) Schreiben Sie eine Abfrage in SQL, die die Namen aller Sektionen ausgibt, die im Jahr 2001 nur Mitglieder hatten, die mehr als 50 Euro Beitrag gezahlt haben. [8]

```
select s.sname
from sektion s
where s.snr not in (select m.snr
                   from mitgl m, bezahlt z, beitrag b
                   where m.mnr = z.mnr and
                         z.jahr = '2001' and
                         b.jahr = z.jahr and
                         b.bnr = z.bnr and
                         b.euro <= 50);
```

c) Schreiben Sie eine Abfrage in Relationaler Algebra, die Name und Adresse aller Mitglieder ausgibt, die im Jahr 2000 einen höheren Beitrag bezahlt haben, als im Jahr 2001. [7]

$$\pi_{name, adresse} \left[mitgl \bowtie \sigma_{(mnr=mnr') \wedge euro < euro'} \left[\pi_{mnr, euro} \sigma_{jahr=2001} (bezahlt \bowtie beitrage) \bowtie \pi_{mnr', euro'} \sigma_{jahr=2000} (bezahlt' \bowtie beitrage') \right] \right]$$

d) Schreiben Sie ein stratifiziertes Datalog-Programm, das im Prädikat **brav**(MNR, NAME, ADRESSE) alle Mitglieder ausgibt, die seit Ihrem Beitritt bis 2001 jedes Jahr den Mitgliedsbeitrag bezahlt haben. Das Attribut SUM(X,Y,Z) ($X = Y + Z$) steht zur Verfügung. [8]

```
bez(MNR,VON,VON) :- mitgl(MNR, NAME, ADRESSE, VON, SNR), bezahlt(MNR, BNR, VON).
bez(MNR,VON,BIS) :- bez(MNR,VON,BIS1), bezahlt(MNR, BNR, BIS), SUM(BIS, BIS1, 1).

brav(MNR, NAME, ADRESSE) :- mitgl(MNR,NAME, ADRESSE, VON, SNR), bez(MNR, VON, '2001').
```

Aufgabe 3:

(7)

Gegeben sind die Relationenschemata: $p(CDF)$, $q(CDF)$, $r(ABE)$, $s(AC)$. Optimieren Sie den folgenden algebraischen Ausdruck:

$$\pi_C \sigma_{D>5 \wedge F<3} \left[\left(s \bowtie (p \cap q) \right) \cup \pi_{ACDF} (q \times r) \right]$$

$$\left[\left(\pi_C s \bowtie \pi_C (\sigma_{D>5 \wedge F<3} p \cap \sigma_{D>5 \wedge F<3} q) \right) \cup \pi_C (\pi_C \sigma_{D>5 \wedge F<3} q \times r) \right]$$

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

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

$F_1 = \{C \rightarrow E, BD \rightarrow CE, F \rightarrow A\}$

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

$F_3 = \{FE \rightarrow AC, EC \rightarrow B, BD \rightarrow E\}$

RS	Schlüssel	Zerlegung in 3NF (Einen Schlüssel in jeder Relation unterstreichen)					
R1	BDF	R1-1 <u>FA</u>	R1-2 <u>CE</u>	R1-3 <u>BDC</u>	R1-4 <u>BDF</u>		
R2	AD	R2-1 <u>CB</u>	R2-2 <u>EB</u>	R2-3 <u>AFE</u>	R2-4 <u>ADC</u>		
R3	D F E, D F B .	R3-1 <u>FEAC</u>	R3-2 <u>BCDE</u>	R3-3	R3-4		

Aufgabe 5:

(6)

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

$$\mathbf{F} = \{AXL \rightarrow S, X \rightarrow AB, BA \rightarrow ZB, L \rightarrow BX\}.$$

$$L \rightarrow S, X \rightarrow A, X \rightarrow B, BA \rightarrow Z, L \rightarrow X$$

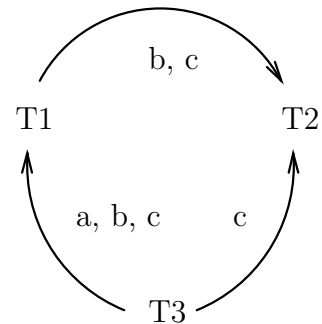
Aufgabe 6:

(7)

Gegeben sei folgende Parallelausführung der Transaktionen $T1-T3$. a) Geben Sie einen Präzedenzgraphen an: [4]

$T1$	$T2$	$T3$
		read a write c
write a read a read b		
		read b
write b read c		
		read c
	read b read c write c	

Präzedenzgraph:



b) Ist die Ausführung serialisierbar ?

[3]

☒ ja, weil der Präzedenzgraph azyklisch ist.

☐ nein, weil

Die Ausführung wird aber nach Streichen von (mindestens) Operation(en) serialisierbar, z.B. die Operation(en)

Aufgabe 7:

(6)

Ist (modulo Benennung von Relationen) die Umwandlung eines ER-Diagramms in ein Relationenschema eindeutig festgelegt?

☐ ja, weil

☒ nein, weil bei Weak Entities etwa nicht immer klar hervorgeht, welche Schlüssel verbundener Entities in den Schlüssel der aus der Weak Entity hervorgehenden Relation eingehen.

Gesamtpunkte: 100