

Gruppe B

Bitte **wählen sie welches Zeugnis Sie benötigen**, tragen Sie **sofort** und **leserlich** Namen und Matrikelnr. ein, und legen Sie Ihren Studentenausweis bereit.

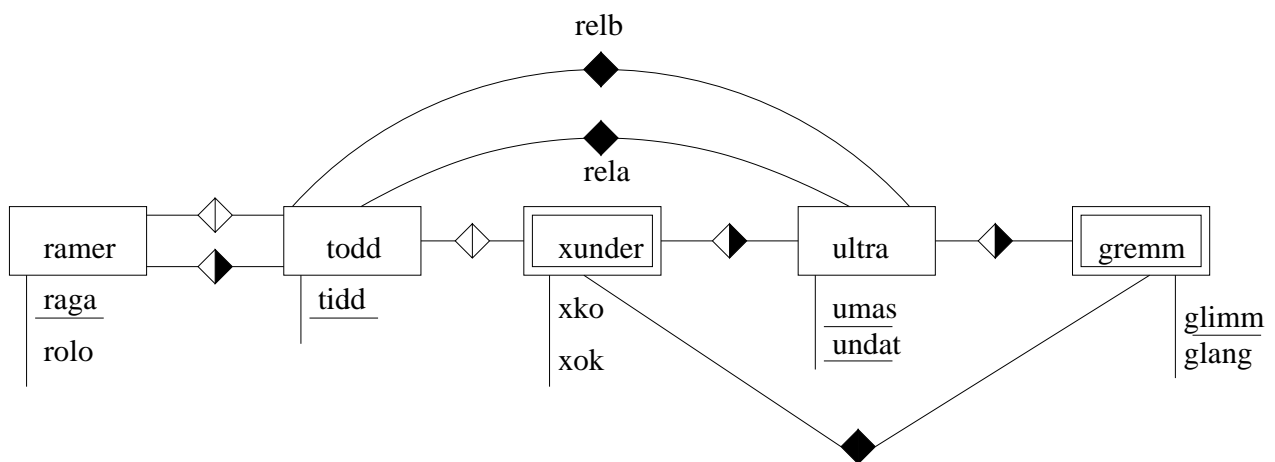
PRÜFUNG AUS DATENBANKSYSTEME		<input type="radio"/> 181.038 ("alt")	<input type="radio"/> 181.146 ("neu")	13.3.2003
Kennnr.	Matrikelnr.	Familiename		Vorname

Arbeitszeit: 90 Minuten. Aufgaben sind auf den Angabeblättern zu lösen; Zusatzblätter werden nicht gewertet.

Aufgabe 1:

(22)

Erstellen Sie aufgrund folgenden EER-Diagramms ein Relationenschema in 3NF.



Unterstreichen Sie je Relation einen Schlüssel. Führen Sie keine zusätzlichen Attribute ein und verwenden Sie möglichst wenige Relationen. Die Datenbank unterstützt keine *Nullwerte*. Achten Sie darauf dass Fremdschlüssel eindeutig der anderen Relation zugeordnet werden können.

```

    ramer ( raga, rolo, tidd* )
    todd ( tidd, raga, raga2* )
    ultra ( umas, undat, tidd )
    xunder ( tidd, xko, xok )
    gremm ( umas, undat, glimm, glang )
    rela ( tidd, umas, undat )
    relb ( tidd, umas, undat )
    xunder2gremm ( tidd, umas, undat, glimm )
    
```

* - alternative Varianten

Gegeben sei eine Datenbank mit den folgenden Relationen:

bank(<u>blz</u> , name, adresse)	transaktion(<u>code</u> , datum, wert)
konto(<u>blz</u> , <u>nummer</u> , rahmen, name)	einzahlung(<u>code</u> , blz, nummer)
person(<u>name</u> , adresse, alter)	abhebung(<u>code</u> , blz, nummer)
verdacht(<u>blz</u> , <u>nummer</u> , grund)	

Eine Bank ist durch ihre Bankleitzahl (BLZ) identifiziert. Sie hat einen Namen und eine Adresse.

Konten werden innerhalb einer Bank durch ihre Kontonummern (**nummer**) unterschieden. Sie haben einen Überziehungsrahmen (**rahmen**) und gehören einer Person (**name**). Personen werden durch Ihren Namen identifiziert. Sie wohnen an einer bestimmten Adresse. Des weiteren ist ihr Alter vermerkt.

Transaktionen auf Konten werden folgendermassen verwaltet: jede Transaktion hat einen eindeutigen **code** und findet zu einem bestimmten Zeitpunkt (**datum**) statt wobei ein Geldbetrag (**wert**) bewegt wird. Des weiteren wird das Konto vermerkt, auf das der Geldbetrag einbezahlt (Relation **einzahlung**), resp. von dem dieser abgehoben (Relation **abhebung**) wurde. Wird Geld von einem Konto auf ein anderes überwiesen, so gibt es sowohl eine Abhebung, als auch eine Einzahlung mit demselben **code**.

Zusätzlich wird eine Liste verdächtiger Konten verwaltet, wobei stets der Grund des Verdachtes angegeben ist.

a) Schreiben Sie eine Abfrage in SQL, die alle verdächtigen Konten ausgibt, von denen bereits mehr als 150.000 abgehoben wurde und die wegen "Betrug" verdächtig sind. Geben Sie jeweils Bankleitzahl, Kontonummer, die Summe der abgehobenen Beträge und die Anzahl der Abhebungen aus. Sortieren Sie das Ergebnis nach der Summe der abgehobenen Beträge. [8]

```
SELECT      t.blz, t.nummer, SUM(t.wert) AS Summe, COUNT(*) AS Anzahl
FROM        transaktion t, abhebung b, verdacht v
WHERE       b.code = t.code
           AND b.blz = v.blz
           AND b.nummer = v.nummer
           AND v.grund = 'Betrug'
GROUP BY    t.blz, t.nummer
HAVING      SUM(t.wert) > 150.000
ORDER BY    Summe DESC;
```

b) Schreiben Sie eine Abfrage in SQL, die Bankleitzahl und Kontonummer aller Konten ausgibt von denen noch nie Geld auf ein verdächtiges Konto überwiesen wurde. [8]

(SELECT blz, nummer FROM konto)	Oder:
MINUS	SELECT blz, nummer
(SELECT a.blz, a.nummer FROM einzahlung e, abhebung a, verdacht v WHERE a.code = e.code AND e.blz = v.blz AND e.nummer = v.nummer);	FROM konto WHERE NOT EXISTS (SELECT * FROM abhebung a, verdacht v, einzahlung e WHERE k.blz = a.blz AND k.nummer = a.nummer AND a.code = e.code AND e.blz = v.blz AND e.nummer = v.nummer);
statt MINUS geht auch:	
... WHERE blz, nummer NOT IN ...	

c) Schreiben Sie eine Abfrage in Relationaler Algebra, welche den Namen und die Adresse von allen Personen ausgibt, die zwar Konten besitzen die wegen "Erpressung" verdächtig sind, aber keine Konten besitzen, die wegen "Waffenhandel" verdächtig sind. [8]

$$\pi_{name, adresse}(person \bowtie ((\pi_{name}(konto \bowtie (\sigma_{grund="Erpressung"}(verdacht))) - (\pi_{name}(konto \bowtie (\sigma_{grund="Waffenhandel"}(verdacht)))))$$

Aufgabe 3:

(8)

Gegeben ist ein Graph bestehend aus gefärbten Knoten und gerichteten Kanten. Dabei beschreibt der Fakt **knoten**(k1, rot) dass der Knoten k1 die Farbe rot hat. Der Fakt **kante**(k1, k2) besagt, dass es vom Knoten k1 zum Knoten k2 eine gerichtete Kante gibt.

Schreiben Sie eine Abfrage in Datalog, die alle Knoten ausgibt, die nicht **gelb** gefärbt sind und die vom Knoten k7 aus erreicht werden können.

```
erreichbar(X) :- kante(k7,X).
erreichbar(X) :- erreichbar(Y), kante(Y,X).
ergebnis(X) :- erreichbar(X), not knoten(X, gelb).
```

Aufgabe 4:

(24)

Bestimmen Sie für jedes der Relationenschemata $R_i(R_i, F_i)$, $i \in \{1,2,3\}$, eine minimale Überdeckung.

Bestimmen Sie weiters alle Schlüssel und geben Sie eine verlustfreie und abhängigkeitstreue Zerlegung in 3NF mit möglichst wenig Relationen an. Unterstreichen Sie in jeder Relation der Zerlegung einen Schlüssel.

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

$F_1 = \{ MN \rightarrow OPQ, PQ \rightarrow O, QOP \rightarrow MR \}$

$F_2 = \{ MN \rightarrow QR, Q \rightarrow MNO, QO \rightarrow P \}$

$F_3 = \{ MN \rightarrow QR, R \rightarrow OP, O \rightarrow Q \}$

RS	minimale Überdeckung
R1	$PQ \rightarrow O, PQ \rightarrow R, MN \rightarrow P, MN \rightarrow Q, PQ \rightarrow M$
R2	$MN \rightarrow Q, MN \rightarrow R, Q \rightarrow M, Q \rightarrow N, Q \rightarrow O, Q \rightarrow P$
R3	$MN \rightarrow R, R \rightarrow O, R \rightarrow P, O \rightarrow Q$

RS	Schlüssel	Zerlegung in 3NF (Einen Schlüssel in jeder Relation unterstreichen)			
R1	MN, NPQ ..	R1-1 <u>OPQR</u>	R1-2 <u>MNPQ</u> <u>MNPQ</u>	R1-3	R1-4
R2	MN, Q	R2-1 <u>MNOPQR</u> ...	R2-2	R2-3	R2-4
R3	MN	R3-1 <u>MNR</u>	R3-2 <u>ROP</u>	R3-3 <u>OQ</u>	R3-4

Aufgabe 5:

(5)

Beachten Sie die Angabe von Beispiel 2 (Bankenbeispiel). Dabei werden für Geldtransaktionen 3 Relationen verwendet. Kann man diese Relationen zu zwei oder einer zusammenfassen oder sollte man dies nicht tun? Begründen Sie Ihre Antwort. (Falls man die Relationen zusammenfassen kann, geben Sie diese auch an.)

3 Relationen beibehalten, weil:

transaktion(<u>code</u> , datum, wert, blz1, nummer1, blz2, nummer2)	schlecht wegen Nullwerten
transaktion(<u>code</u> , <u>einaus</u> , datum, wert, blz, nummer)	geht nicht wegen Datenanomalien
abhebung(<u>code</u> , <u>blz</u> , <u>nummer</u> , datum, wert)	
einzahlung(<u>code</u> , <u>blz</u> , <u>nummer</u> , datum, wert)	geht nicht wegen Datenanomalien
transaktion(<u>code</u> , datum, wert, blz, nummer)	
transaktion2(<u>code</u> , blz, nummer)	sehr unpraktisch

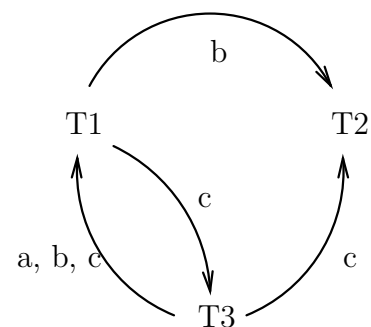
Aufgabe 6:

(8)

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

	$T1$	$T2$	$T3$
1			read a
2			write c
3	write a		
4	read a		
5	read b		
6			read b
7	write b		
8	read c		
9			write c
10		read b	
11		read c	
12		write c	

Präzedenzgraph:



b) Ist die Ausführung serialisierbar ?

[4]

- ☐ ja, weil
- ☒ nein, weil der Präzedenzgraph zyklisch ist.

Die Ausführung wird aber nach Streichen von (mindestens) 1 Operation(en) serialisierbar, z.B. die Operation(en) in der Zeile/den Zeilen 8 oder 9

Aufgabe 7:

(8)

Angenommen wir können jegliche Hardwarefehler (und Stromausfall, Terroranschläge, Atomwaffeneinsätze,...) sowie Bugs im Betriebssystem und dem DBMS unseres Datenbankservers ausschließen. Könnten wir auf die Recovery-Komponente des verwendeten DBMS verzichten (bzw. diese deaktivieren)?

[4]

- ☐ ja, weil
- ☒ nein, weil Transaktionen auch von Usern oder der Concurrency-Komponente abgebrochen werden können

Angenommen wir können Software- und Hardwareprobleme jeglicher Art (sowie Stromausfall, Terroranschläge, Atomwaffeneinsätze,...) ausschließen, ist ein regelmäßiges Backup unserer Datenbank dennoch notwendig?

[4]

- ☒ ja, weil all das nicht vor Fehlern der User oder Admins schützt
- ☐ nein, weil