

Gruppe A

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

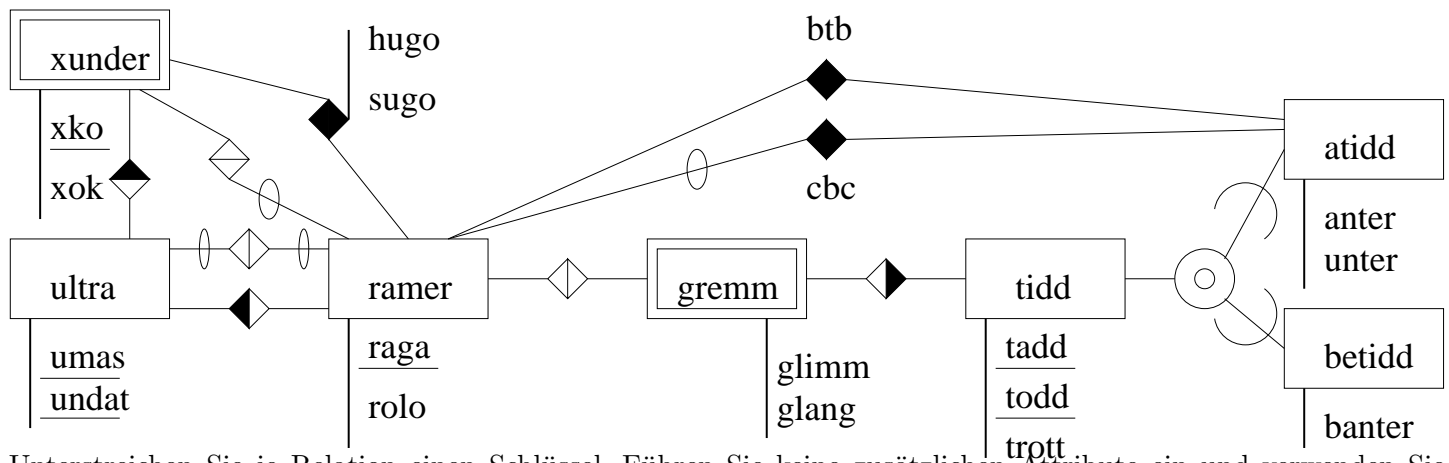
PRÜFUNG AUS DATENBANKSYSTEME <input type="radio"/> 181.038 ("alt") <input type="radio"/> 181.146 ("neu") 15. 10. 2003				
Kennzahl	Matrikelnr.	Familienname	Vorname	

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

Aufgabe 1:

(20)

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.

ultra	(<u>umas, undat</u> , raga,)
ramer	(<u>raga</u> , rolo, umas, undat, xko)
tidd	(<u>tadd, tott</u> , trott, raga)
atidd	(<u>tadd, tott</u> , anter, unter)
betidd	(<u>tadd, tott</u> , banter)
xunder	(<u>umas, undat</u> , xko, xok)
greemm	(<u>raga</u> , glimm, glang)
xund2ram	(<u>umas, undat</u> , <u>xko</u> , <u>raga</u> , hugo, sugo)
ult2ram	(<u>umas, undat</u> , raga)
btb	(<u>raga</u> , <u>tadd, tott</u>)
cbc	(<u>raga</u> , <u>tadd, tott</u>)
	()

Aufgabe 2:

(24)

Gegeben sei die Datenbank eines Handelsunternehmens mit den folgenden Relationen:

`lieferant(LID, name, adresse)`

`produkt(LID, PID, name)`

`lieferung(LID, PID, datum, menge, preis)`

Jeder Lieferant des Unternehmens hat eine eindeutige Kennung (LID), einen Namen und eine Adresse.

Von einem Lieferanten werden verschiedenen Produkte angeboten. Diese haben einen (allgemein gebräuchlichen) Namen und eine für den jeweiligen Lieferanten eindeutige (und nur für ihn geltende) Produktnummer (PID). Natürlich kann dasselbe Produkt bei verschiedenen Lieferanten auch verschiedene Produktnummern haben.

Für jede Lieferung, die das Handelsunternehmen erhält, wird das Datum, der Lieferant und die entsprechende Produktnummer, sowie die gelieferte Menge und der bezahlte Preis gespeichert.

a) Geben Sie die Namen und Adressen der Lieferanten sowie die Namen jener Produkte aus, die nur von ihm, aber von keinem anderen Lieferanten angeboten werden. [8]

```
SELECT  l.name, l.adresse, p.name AS Produkt
FROM    lieferant l, produkt p
WHERE   l.LID = p.LID
      AND p.name NOT IN ( SELECT p2.name
                          FROM   produkt p2
                          WHERE  p2.LID <> p.LID );
```

b) Geben Sie die Namen aller jener Lieferanten aus, von denen an einem Tag schon einmal Waren um insgesamt mehr als 100.000 Euro geliefert wurden. Sortieren Sie das Ergebnis nach dem Namen der Lieferanten. [8]

<code>richtig gewertete, aber nicht ganz exakte Lösung:</code>	<code>Wirklich genaue Lösung (Gibt Lieferanten mit gleichem Namen getrennt, aber jeweils nur einmal aus):</code>
<pre>SELECT DISTINCT l.name FROM lieferant l, lieferung b WHERE l.LID = b.LID GROUP BY l.LID, l.name, b.datum HAVING SUM(b.preis) > 100.000 ORDER BY l.name;</pre>	<pre>SELECT name FROM lieferant WHERE LID IN (SELECT LID FROM lieferung GROUP BY LID, datum HAVING SUM(preis) > 100.000) ORDER BY name;</pre>

c) Schreiben Sie eine Abfrage in Relationaler Algebra, welche Namen und Adresse jener Lieferanten ausgibt, die alle jemals vom Handelsunternehmen bestellten Produkte anbieten. [8]

$bestellte_produkte = \pi_{name}(lieferung \bowtie produkt)$
 $angebotene_produkte = \pi_{LID, name}(produkt)$
 $alle_produkte = anbotene_produkte \div bestellte_produkte$
 $alles_lieferer = \pi_{name, adresse}(lieferant \bowtie alle_produkte)$

Oder, zusammengefasst:

$\pi_{name, adresse}(lieferant \bowtie ((\pi_{LID, name}(produkt)) \div (\pi_{name}(lieferung \bowtie produkt))))$

Aufgabe 3:

(12)

Ein (monotoner) logischer Schaltkreis ist ein Netzwerk bestehend aus And-, Or- und Input-Gates, wobei jedes Gate durch eine eindeutige Gate-Id identifiziert wird. Input-Gates haben entweder den **wert** "true" oder "false". Der Wert eines And-Gates errechnet sich als die logische Konjunktion (\wedge) seiner *genau zwei* Vorgänger-Gates; der Wert eines Or-Gates ist die logische Disjunktion (\vee) seiner ebenfalls genau zwei Vorgänger (d.h. alle And- und Or-Gates in unserem Schaltkreis haben *fan-in* 2). Der Schaltkreis wird durch die Relationen

Input(gid, wert)

And(gid, in1gid, in2gid)

Or(gid, in1gid, in2gid)

repräsentiert, wobei gid die Gate-Id eines Gates und in1gid und in2gid die Gate-Ids der Vorgänger-Gates von And- bzw. Or-Gates repräsentieren.

Schreiben Sie ein stratifiziertes Datalog-Programm das die Menge der Gates berechnet (mittels eines unären Prädikats **resultat**), deren Wert zu "true" evaluiert und die keine Input-Gates sind.

```

true(G) :- Input(G, true).
true(G) :- And(G, G1, G2), true(G1), true(G2).
true(G) :- Or(G, G1, G2), true(G1).
true(G) :- Or(G, G1, G2), true(G2).
isinput(G) :- Input(G, _).
resultat(G) :- true(G), not isinput(G).

```

Anmerkung: die alternative Version ohne Negation ist etwas umfangreicher.

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 wenigen Relationen an. Unterstreichen Sie in jeder Relation der Zerlegung einen Schlüssel.

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

$F_1 = \{ C \rightarrow AE, E \rightarrow B, B \rightarrow ADF, D \rightarrow B \}$

$F_2 = \{ F \rightarrow DE, E \rightarrow CB, B \rightarrow AF \}$

$F_3 = \{ AF \rightarrow C, CF \rightarrow A, C \rightarrow B, BC \rightarrow DE, E \rightarrow D \}$

RS	minimale Überdeckung
R_1	$C \rightarrow E, E \rightarrow B, B \rightarrow A, B \rightarrow D, B \rightarrow F, D \rightarrow B$
R_2	$F \rightarrow D, F \rightarrow E, E \rightarrow B, E \rightarrow C, B \rightarrow A, B \rightarrow F$
R_3	$AF \rightarrow C, CF \rightarrow A, C \rightarrow B, C \rightarrow E, E \rightarrow D$

RS	Schlüssel	Zerlegung in 3NF (Einen Schlüssel in jeder Relation unterstreichen)
R_1	$C \dots\dots\dots$	R1-1 <u>B</u> ADF R1-2 <u>C</u> E R1-3 <u>E</u> B R1-4
R_2	$B, E, F \dots$	R2-1 <u>F</u> ABCDE ... R2-2 R2-3 R2-4
R_3	$AF, CF \dots$	R3-1 <u>A</u> FC <u>A</u> CF R3-2 <u>D</u> E R3-3 <u>B</u> CE R3-4

Aufgabe 5:

(7)

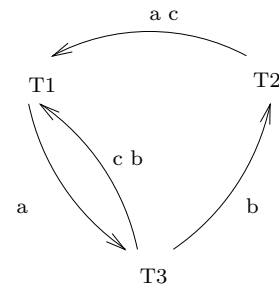
Gegeben sei folgende Parallelausführung der Transaktionen $T1-T3$.

a) Geben Sie einen Präzedenzgraphen an:

[4]

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

Präzedenzgraph:



b) Ist die Ausführung serialisierbar ?

[3]

☐ 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) read a in T3

Aufgabe 6:

(7)

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

$$\pi_{BC}\sigma_{(C=c)\wedge((B=b)\vee(D=d))}\left[\pi_{BCD}(q \bowtie r) \cup \left(\pi_{BCD}(p \bowtie q) \cap r\right)\right]$$

$$\pi_{BC}\left(\pi_{DQ} \bowtie \sigma_{(C=c)\wedge(B=b\vee D=d)}r\right) \cup \pi_{BC}\left[\left(\sigma_{B=b\vee D=d}(\sigma_{(C=c)}p \bowtie \pi_{DQ})\right) \cap \sigma_{(C=c)\wedge(B=b\vee D=d)}r\right]$$

Aufgabe 7:

(6)

Gegeben seien die folgenden beiden Relationen:

$RelA(\underline{A}, B, C)$

$RelB(\underline{B}, C, D, E)$

Schreiben Sie eine Abfrage in SQL, die genau dasselbe Ergebnis liefert wie folgender Ausdruck in Relationaler Algebra:

$$\pi_{B,C}(\sigma_{D>100}(RelA \bowtie RelB))$$

```

SELECT DISTINCT RelB.B, RelB.C
FROM RelA, RelB
WHERE RelA.B = RelB.B
AND RelA.C = RelB.C
AND RelB.D > 100;

```

Oder:

```

SELECT DISTINCT B, C
FROM RelA NATURAL JOIN RelB
WHERE RelB.D > 100;

```

Oder:

```

SELECT RelB.B, RelB.C
FROM RelA, RelB
WHERE RelA.B = RelB.B
AND RelA.C = RelB.C
AND RelB.D > 100
GROUP BY RelB.B, RelB.C;

```

Gesamtpunkte: 100