

Gruppe A

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") 20.6.2002			
Kennnr.	Matrikelnr.	Familienname	Vorname

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

Aufgabe 1: (20)

Modellieren Sie folgende Datenbank, die eine Reihe von Genossenschaftswohnungen verwaltet. 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!

Die Genossenschaft operiert nur in Österreich und besitzt eine Reihe von Wohnungen, von denen wir eine auf das jeweilige Gebäude bezogen eindeutige Kurzbezeichnung (KURZ), die Größe in m² (M2), eine eindeutige Nummer (NR) und den Namen der Genossenschaft (GENOSS) kennen. Wohnungen liegen entweder in Gebäuden, die ganzheitlich der Genossenschaft gehören, oder in normalen Mietshäusern, wo sie von der Genossenschaft dauergemietet sind.

Jedes Mietshaus hat einen Namen (NAME) und eine eindeutige Adresse bestehend aus Straßennamen/-nummer (STR) und Postleitzahl (PLZ) sowie eine Hausverwaltung. Weiters kennen wir die Betriebskosten pro m² (KPM2). Von Gebäuden im Besitz von Genossenschaften führen wir dieselben Daten wie bei Miethäusern, anstelle der Betriebskosten jedoch den Genossenschaftsanteil pro m² (GPM2).

Von jeder Hausverwaltung kennen wir ihre österreich-weit eindeutige Registriernummer (HRNR), ihre Telefonnummer (TEL) und Postanschrift (ADDR, PLZ); auch unsere Genossenschaft selbst ist als Hausverwaltung registriert.

Jede dieser Wohnungen kann bis zu drei Hauptmieter und beliebig viele Bewohner haben, wobei jede Person nur eine Hauptmiete eingehen darf, aber in mehreren Wohnungen als Bewohner gemeldet sein kann. Von allen Bewohnern (auch Hauptmieter zählen als solche) kennen wir Namen (VORNAME, NACHNAME) und Geburtsdatum (GEBURT), wobei wir diese drei Attribute zusammen als eindeutig annehmen; von Hauptmietern kennen wir zusätzlich noch ihre Kontonummer (KONTO).

wohnung (KURZ, M2, <u>NR</u> , GENOSS, STR, PLZ) [3]
mietshaus (NAME, <u>STR</u> , <u>PLZ</u> , HRNR, KPM2) [3]
gebäude (NAME, <u>STR</u> , <u>PLZ</u> , HRNR, GPM2) [3]
verwaltung (<u>HRNR</u> , TEL, ADDR, PLZ) [2]
person (<u>VORNAME</u> , <u>NACHNAME</u> , <u>GEBURT</u>) [3]
hauptmieter (<u>VORNAME</u> , <u>NACHNAME</u> , <u>GEBURT</u> , KONTO, NR) [3]
bewohner (<u>VORNAME</u> , <u>NACHNAME</u> , <u>GEBURT</u> , NR) [3]
()
()
()
(alternativ bei wohnung auch anderer Key!)

In der relationalen Datenbank eines Internet Buchgeschäfts werden folgende Informationen gespeichert:

buch(<u>isbn</u> , titel, preis)	autorbuch(<u>anr</u> , <u>isbn</u>)	autor(<u>anr</u> , name)
kunde(<u>knr</u> , name, email)	order(<u>knr</u> , <u>lnr</u> , isbn)	rezension(<u>isbn</u> , <u>knr</u> , sterne, text)

Jedes Buch wird durch seine International Standard Book Number (isbn) eindeutig identifiziert, durch seinen Titel und Preis beschrieben und kann beliebig viele Autoren haben. Jeder Autor wird durch seine Autorennummer (anr) eindeutig identifiziert, durch seinen Namen beschrieben und kann beliebig viele Bücher geschrieben haben. Kunden werden durch ihre Kundennummer (knr) eindeutig identifiziert und durch ihren Namen und ihre Email-Adresse beschrieben. Jede Bestellung *eines* Buches (order) wird durch die Kundennummer und eine laufende Nummer (lnr) eindeutig identifiziert und durch die ISBN des bestellten Buches beschrieben. Kunden können zu einem Buch eine Rezension (0 – 5 Sterne, Rezensionstext) verfassen.

a) Schreiben Sie eine SQL-Abfrage, die die Namen von Autoren ausgibt, deren Bücher von Kunden bestellt wurden, die auch ein Buch des Autors „Knuth“ bestellt haben. Die Ausgabe der Namen soll dabei absteigend in Bezug auf die Anzahl der Kombinationen von jeweils zwei Bestellungen erfolgen, bei denen ein Kunde einerseits ein Buch von Knuth, andererseits ein (anderes) Buch des jeweils auszugebenden Autors (der verschieden von Knuth sein soll) bestellt hat. Gibt es für einen Autor insgesamt nur eine solche Kombination von Bestellungen, so soll der Name dieses Autors nicht ausgegeben werden.

[10]

```
SELECT a2.name
FROM autor a1, autorbuch ab1, order o1, order o2, autorbuch ab2, autor a2
WHERE a1.name='Knuth' AND a1.anr=ab1.anr AND ab1.isbn=o1.isbn AND o1.knr=o2.knr AND
      o1.isbn<>o2.isbn AND o2.isbn=ab2.isbn AND ab2.anr=a2.anr AND a2.name<>'Knuth'
GROUP BY a2.name
HAVING count(*) > 1
ORDER BY count(*) DESC
```

b) Schreiben Sie eine Abfrage in SQL, die für den Kunden mit der Email-Adresse „a@b.com“ die Titel und ISBNs von bisher von ihm noch nicht bestellten Büchern von Autoren ausgibt, von denen er bereits Bücher bestellt hat.

[10]

```
SELECT DISTINCT b2.isbn, b2.titel
FROM kunde k, autorbuch ab1, autorbuch ab2, buch b2, order o
WHERE k.email='a@b.com' AND k.knr=o.knr AND o.isbn=ab1.isbn AND
      ab1.anr=ab2.anr AND ab2.isbn=b2.isbn AND
      b2.isbn NOT IN ( SELECT o1.isbn
                      FROM order o1
                      WHERE o1.knr=k.knr );
```

c) Schreiben Sie dieselbe Abfrage wie in b) in Relationaler Algebra.

[8]

$$X = \pi_{isbn}(\sigma_{email='a@b.com'}(kunde) \bowtie order)$$

$$Y = \pi_{isbn}((X \bowtie autorbuch)[anr = anr']autorbuch')$$

Resultat: $\pi_{isbn,titel}(buch \bowtie (Y - X))$

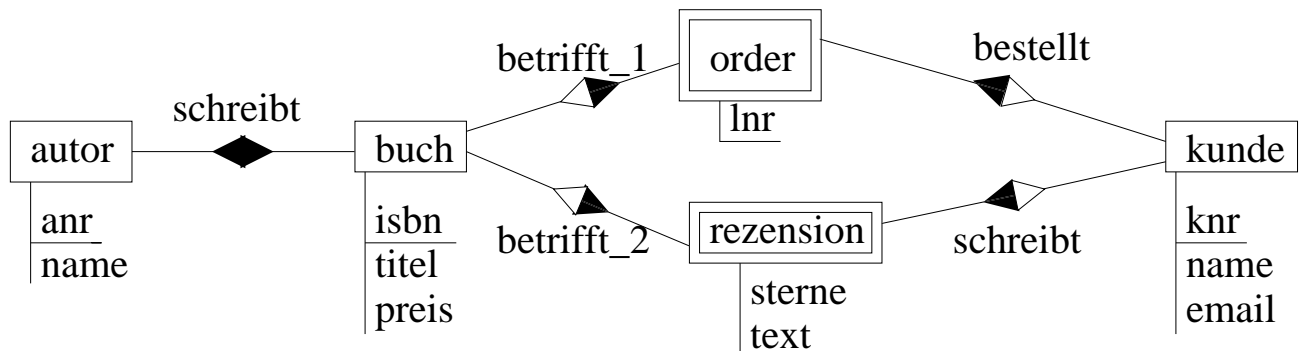
d) Schreiben Sie ein stratifiziertes Datalog Programm, das die Relation `minimale_distanz(B1,B2,D)` zwischen zwei Büchern, gegeben durch ihre ISBNNummern B1 und B2, und ihrer „minimalen Distanz“ D so berechnet, dass man von B1 zu B2 mindestens D „Schritte“ benötigt, wobei ein Schritt durch einen Kunden definiert ist, der sowohl B1 als auch B2 bestellt hat (wenn B1=B2 soll D=0 berechnet werden). Das Prädikat `sum(X,Y,Z) ≡ X=Y+Z` steht zur Verfügung.

[8]

```
d(B,B,0) :- buch(B,_,_).
d(B1,B2,1) :- order(K,_,B1,_), order(K,_,B2,_).
d(B1,B2,D) :- d(B1,B,D1), d(B,B2,D2), sum(D,D1,D2).
minimale_distanz(B1,B2,D) :- d(B1,B2,D), sum(D,D1,D2), D2 > 0, non d(B1,B2,D1).
```

e) Zeichnen Sie ein EER-Diagramm für die beschriebene Datenbank.

[10]



Aufgabe 3:

(6)

Gegeben ist das Relationenschema ABCDVW und die Menge **F** von funktionalen Abhängigkeiten. Bestimmen Sie eine minimale Überdeckung. **F** = { $A \rightarrow BC$, $BC \rightarrow ACD$, $AD \rightarrow V$, $ABC \rightarrow W$ }

$A \rightarrow B$, $A \rightarrow C$, $BC \rightarrow A$, $BC \rightarrow D$, $A \rightarrow V$, $A \rightarrow W$ oder $BC \rightarrow W$

Aufgabe 4:

(14)

Gegeben ist das Relationenschema ABUVWX mit den funktionalen Abhängigkeiten
 $F = \{ A \rightarrow B, B \rightarrow A, AB \rightarrow UV, U \rightarrow W, X \rightarrow W \}$.

Bitte markieren Sie jene Zerlegungen, die in 3NF sind, bzw. begründen Sie (etwa durch Angabe einer FD, welche die Bedingungen der 3NF verletzt), warum eine Zerlegung nicht in 3NF ist.

Zerlegung	3NF?	Begründung, falls nicht in 3NF
<u>ABUVWX</u>	<input type="radio"/>	$A \rightarrow U, A \rightarrow V, B \rightarrow U, B \rightarrow V, U \rightarrow W, X \rightarrow W$
<u>AB</u> , <u>BUVWX</u>	<input type="radio"/>	$B \rightarrow U, B \rightarrow V, U \rightarrow W$
<u>ABUVW</u> , <u>WX</u>	<input type="radio"/>	$U \rightarrow W$
<u>ABUV</u> , <u>UWX</u>	<input type="radio"/>	$U \rightarrow W$
<u>AB</u> , <u>BUV</u> , <u>UXW</u>	<input type="radio"/>	$U \rightarrow W, X \rightarrow W$
<u>ABUV</u> , <u>UW</u> , <u>XW</u>	<input checked="" type="radio"/>	
<u>AB</u> , <u>BA</u> , <u>BU</u> , <u>BV</u> , <u>UW</u> , <u>XW</u>	<input checked="" type="radio"/>	

Aufgabe 5:

(8)

Die nachfolgende Parallelausführung von Transaktionen wird mithilfe von Zeitstempeln synchronisiert.

a) Geben Sie die Werte der Zeitstempel an (soweit möglich); markieren Sie einen Transaktionsabbruch mit '*'. [5]

T_1	T_2	T_3	a	b	c
1	20	5	RT=WT=0	RT=WT=0	RT=WT=0
		read a	RT= 5		
write b				WT= 1	
	write c				WT= 20
		write c			WT= 20
	write b			WT= 20	
read b				RT= *	
	write b			WT= 20	
		write a	WT= 5		
	read a		RT= 20		

b) Welche Reihenfolge wird zwischen den Transaktionen durch die Zeitstempel festgelegt ($T_i < T_j \equiv "T_i \text{ vor } T_j"$)? Annahme: Abgebrochene Transaktionen werden sofort ihrem dem Abbruch wieder gestartet.

[3]

$T_3 < T_2 < T_1$

Aufgabe 6:

(6)

Sind die folgenden Paare von Ausdrücken in Relationaler Algebra über den Relationen $f(ABC)$ und $g(BCD)$ äquivalent? Falls nicht, begründen Sie Ihre Antwort **kurz/intuitiv**!

Ausdruck 1	Ausdruck 2	Äquivalent?	kurze Begründung
$\sigma_{B < C}[f \bowtie g]$	$(\sigma_{B < C} f) \bowtie (\sigma_{B < C} g)$	<input checked="" type="radio"/>	
$\sigma_{B < C}[f \bowtie g]$	$(\sigma_{B < C} f) \bowtie (\pi_C g)$	<input type="radio"/>	Durch π_C treten mehr Elemente von g in den Join ein.
$(\pi_{BC} f) \bowtie (\pi_{BC} g)$	$\pi_{AD}(f \times g)$	<input type="radio"/>	Die beiden Ausdrücke liefern unterschiedliche Schemata!

Gesamtpunkte: 100