

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") 13. 12. 2001			
Kennnr.	Matrikelnr.	Familienname	Vorname

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

Aufgabe 1:

(25)

Modellieren Sie die Datenbank eines Forschungsinstitutes. 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!

Das Institut verfügt über eine Reihe von Mitarbeitern, von denen Name (NAME), Geschlecht (SEX) und Gehalt (GEHALT) gespeichert sind, und die über eine eindeutige Mitarbeiternummer (MNR) verfügen.

Weiters besitzt das Institut eine Reihe von Rechnern mit einer eindeutigen Inventarnummer (NR) und einer Typ- sowie einer Herstellerbezeichnung (TYP, HERST). Jeder Mitarbeiter kann beliebig viele Rechner haben; jeder Rechner ist allerdings höchstens einem Mitarbeiter zugeordnet (muss allerdings nicht unbedingt zugeordnet sein).

Die Rechner werden von Technikern gewartet, wobei jeder Techniker für mehrere Rechner zuständig ist, und ein Rechner von mehreren Technikern gewartet werden kann. Techniker sind Mitarbeiter, für die zusätzlich eine Handynummer (HANDY) gespeichert ist, sowie eine beliebige Menge von Betriebssystemen (OS), die sie kennen.

Server sind Rechner, für die zusätzlich ein Wartungsvertrag abgeschlossen wird. Für sie wird daher auch die Telefonnummer der jeweiligen Wartungsfirma (TEL) und die Nummer des Wartungsvertrages (VERTRAG) gespeichert; auch ist jedem Server fix ein hauptverantwortlicher Techniker zugeteilt.

Zuletzt gibt es auch noch Assistenten, das sind Mitarbeiter im Bereich der Forschung, für die wir zusätzlich das Forschungsgebiet (GEBIET) und den akademischen Grad (GRAD) speichern müssen. Jeder Server ist einem Assistenten zugeordnet, und jeder Assistent ist für eine Reihe von Technikern verantwortlich (wobei ein Techniker auch mehreren Assistenten zugeteilt sein kann).

mitarbeiter (<u>MNR</u> , NAME, SEX, GEHALT) [2]
rechner (<u>NR</u> , TYP, HERST) [2]
rechner2mitarbeiter (MNR, <u>NR</u>) [3]
wartet (<u>MNR</u> , <u>NR</u>) [3]
techniker (<u>MNR</u> , HANDY) [3]
techniker2os (<u>MNR</u> , <u>OS</u>) [3]
server (<u>NR</u> , TEL, VERTRAG, MNR_TECH, MNR_ASS) [3]
assistent (<u>MNR</u> , GEBIET, GRAD) [3]
verantwortlich (<u>MNR_ASS</u> , <u>MNR_TECH</u>) [3]
()

In der relationalen Datenbank einer Universität werden folgende Informationen gespeichert:

```
lva(lvanr, titel, professor, studienrichtung)
student(mnr, vorname, nachname, gdatum)          zeugnis(mnr, lvanr, pdatum, note, zdatum)
```

Jede Lehrveranstaltung (kurz lva) wird durch ihre Lehrveranstaltungsnummer (lvanr) eindeutig identifiziert, und durch ihren Titel, den Namen des Professors, und die Studienrichtung beschrieben. Alle Studierenden haben eine eindeutige Matrikelnummer (mnr), einen Vornamen, Nachnamen und ein Geburtsdatum (gdatum). Jedes Zeugnis wird durch die Matrikelnummer, die Lehrveranstaltungsnummer, und das Prüfungsdatum identifiziert, und enthält die Note (eine Zahl zwischen 1 und 5), sowie das Datum der Ausstellung des Zeugnisses (zdatum).

a) Prof. Maier soll evaluiert werden. Schreiben Sie eine Abfrage in SQL, die unter den Prüfungsterminen 2000 bei Prof. Maier diejenigen Prüfungstermine mit den *schlechtesten Durchschnittsnoten* selektiert, und diese Termine mit gemeinsam mit dem Titel der Lehrveranstaltung und dem Notendurchschnitt ausgibt.

[10]

```
select z.pdatum, l.titel, avg(z.note)
from zeugnis z, lva l
where l.professor = 'Maier' and
      z.lvanr = l.lvanr and
      z.pdatum between '1.1.2000' and '31.12.2000'
group by z.pdatum, l.lvanr, l.titel
having avg(z.note) >= all (select avg(z1.note)
                           from zeugnis z1, lva l1
                           where l1.professor = 'Maier' and
                                 z1.lvanr = l1.lvanr and
                                 z1.pdatum between '1.1.2000' and '31.12.2000'
                           group by z1.pdatum);
```

b) Schreiben Sie eine Abfrage in SQL, die die Nachnamen und Matrikelnummern jener Studenten ausgibt, die schon in jeder Studienrichtung ein Zeugnis mit einer positiven Note (d.h. einer Note verschieden von 5) bekommen haben. (Sie können annehmen, dass in allen Studienrichtung schon Prüfungen stattgefunden haben.)

[9]

```
select s.nachname, s.mnr
from student s
where not exists
  ( select l.studienrichtung
    from lva l
    where l.studienrichtung not in
      ( select l2.studienrichtung
        from zeugnis z, lva l2
        where z.lvanr = l2.lvanr and
              z.mnr = s.mnr and
              z.note <> 5
      )
  );
```

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

[7]

Abkürzungen:

$X = zeugnis \bowtie lva \bowtie student$

$Y = \pi_{studienrichtung}(lva), Z = \pi_{mnr, studienrichtung, nachname}(X)$

Resultat: $Z \div Y =$

$\pi_{mnr, studienrichtung, nachname}(zeugnis \bowtie lva \bowtie student) \div \pi_{studienrichtung}(lva)$

d) Schreiben Sie dieselbe Abfrage wie in b) als stratifiziertes Datalog Programm. Das Ergebnis soll im Prädikat genie(NACHNAME, MNR) ausgegeben werden. Das Prädikat $X \neq Y$ steht zur Verfügung.

[8]

```
genie(NACHNAME, MNR)      :- non keinePruefungStudent(MNR),
                             student(MNR, _, NACHNAME, _).
keinePruefungStudent(MNR) :- keinePruefung(MNR, _).
keinePruefung(MNR, STUDIUM) :- non positiveszeugnis(MNR, STUDIUM),
                                student(MNR, -, -, -), lva(-, -, -, STUDIUM).
positiveszeugnis(MNR, STUDIUM) :- zeugnis(MNR, LVANR, -, NOTE, -),
                                   lva(LVANR, -, -, STUDIUM),
                                   NOTE  $\neq$  5.
```

Aufgabe 3:

(7)

Gegeben sind die Relationenschemata: $p(WY)$, $q(WXYZ)$, und $r(XYZ)$. Optimieren Sie den folgenden algebraischen Ausdruck:

$$\pi_{YZ} \sigma_{Y=12 \vee Z=12} [(q \bowtie p) \cup (q - (q \bowtie r))]$$

$$\pi_{YZ} (\pi_{WYZ} \sigma_{Y=12 \vee Z=12} q \bowtie p) \cup \pi_{YZ} [\sigma_{Y=12 \vee Z=12} q - (\sigma_{Y=12 \vee Z=12} q \bowtie \sigma_{Y=12 \vee Z=12} r)]$$

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

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

$F_1 = \{E \rightarrow ACD, A \rightarrow B, C \rightarrow F, BC \rightarrow D\}$

$F_2 = \{AB \rightarrow E, E \rightarrow D, D \rightarrow BF, AF \rightarrow C\}$

$F_3 = \{F \rightarrow A, E \rightarrow CDB, A \rightarrow B\}$

RS	Schlüssel	Zerlegung in 3NF (Einen Schlüssel in jeder Relation unterstreichen)			
R1	E	R1-1 <u>E</u> AC	R1-2 <u>A</u> B	R1-3 <u>C</u> F	R1-4 <u>B</u> CD
R2	A[D E B] .	R2-1 <u>A</u> FC	R2-2 <u>D</u> F	R2-3 <u>A</u> DEB	R2-4
R3	FE	R3-1 <u>A</u> B	R3-2 <u>F</u> A	R3-3 <u>E</u> CDB	R3-4 <u>F</u> E

Aufgabe 5:

(6)

Gegeben ist das Relationenschema $ABCDE$ und die Menge \mathbf{F} von funktionalen Abhängigkeiten. Bestimmen Sie eine minimale Überdeckung.

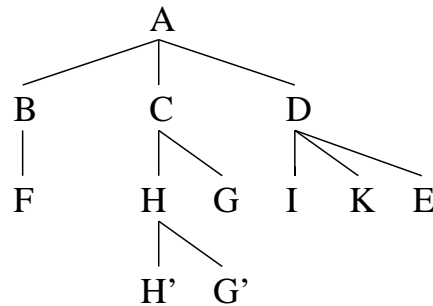
$$\mathbf{F} = \{AB \rightarrow CE, BC \rightarrow DE, B \rightarrow BAD\}.$$

$$B \rightarrow A, B \rightarrow C, B \rightarrow D, B \rightarrow E$$

Aufgabe 6:

(7)

Gegeben seien die Objekte $A, B, C, D, E, F, G, G', H, H', I, K$ in folgender Struktur:



a) Geben Sie für die Transaktionen T1 und T2 an, ob sie das hierarchische Sperrprotokoll erfüllen und gegebenenfalls die erste Operation, die das Protokoll verletzt. [4]

T1: *ilock A, ilock C, ilock D, ilock H, xlock H',*

unlock H', unlock C, unlock D, unlock A ☐ ja ☒ nein *unlock C*

T2: *ilock A, ilock B, ilock D, ilock I, xlock I, xlock K*

unlock I, unlock K, unlock B, unlock D, unlock A ☒ ja ☐ nein

b) Sind alle möglichen gültigen Ausführungen von T1 und T2 serialisierbar (mit Begründung)? [3]

- ☒ ja, weil es zu keinen Konflikten zwischen xlocks kommen kann, da diese
☐ nein, weil nur auf disjunkte Objekte gesetzt werden.

Aufgabe 7:

(6)

Wieviele Relationen werden jeweils benötigt, wenn wir folgende ER-Diagramme ins Relationenschema überführen? Und wieviele Möglichkeiten (modulo Benennungen und Reihenfolgen) gibt es dabei? (Denken Sie an die jeweiligen Fremdschlüssel!)

Entity-Relationship-Diagramm	Anzahl Relationen	Anzahl Übersetzungen
	2	2
	3	3
	3	1

Gesamtpunkte: 100