

Klausur zur Veranstaltung Datenbanksysteme I

Wintersemester 2002
Hessische VWA
Bernd Ulmann

14. November 2002

Hinweise:

- Die Klausur besteht aus 3 Aufgaben
- Insgesamt sind 56 Punkte erreichbar.
- Als Hilfsmittel sind alle „unbelebten Hilfsmittel“, d.h. insbesondere Ihre Vorlesungsmitschriften sowie das Skript, zugelassen.
- Schreiben Sie bitte auf jedes Blatt Ihrer Lösungen Ihren Namen und Ihre Matrikelnummer!
- Bitte schreiben Sie leserlich! (Dankeschön! :-)
- Verwenden Sie bitte keinen (!) Bleistift!
- Vergessen Sie nicht, Primärschlüsselbezeichnungen zu unterstreichen!

Viel Erfolg!

Aufgabe 1

Es seien zwei Relationen **R1** und **R2** folgender Gestalt gegeben:

R1		
A	B	C
U	N	V
V	R	A
3	-	C
7	M	B
T	E	X

R2		
D	E	F
-	X	U
I	K	B
P	C	A
K	V	H
A	H	C

1. Welche Ausgabe erzeugt das folgende SQL-Statement, angewandt auf die beiden Relationen **R1** und **R2**? (3 Punkte)

```
SELECT A, B, D
FROM R1, R2
WHERE C = E
ORDER BY C
```

- Schreiben Sie das SQL-Statement aus der vorangegangenen Aufgabe (ohne die ORDER-BY-Klausel!) mit den Mitteln der Relationenalgebra, d.h. unter Zuhilfenahme von Zeichen wie σ , π , \bowtie , etc. auf. (2 Punkte)
- Welches SQL-Kommando erzeugt einen Join zwischen R1 und R2 (das Statement ist einschließlich der benötigten ORDER-Klausel anzugeben), der folgende Tabelle zur Ausgabe hat? (3 Punkte)

D	B
P	R
I	M
A	-

- Wieviele Ergebniszeilen hätte die Ausführung des SQL-Statements

```
SELECT A
FROM R1, R2
```

zur Folge? (1 Punkt)

- Formulieren Sie ein SQL-Statement, das Ihnen alle Zeilen aus R1 zurückliefert, denen bezüglich der Bedingung $C = E$ kein Eintrag in R2 entspricht. Diese Aufgabe soll nicht mit Joins, sondern ausschließlich mit Hilfe von Subselects gelöst werden! (3 Punkte)

Aufgabe 2:

Ein Musikwissenschaftler interessiert sich ausschließlich für Solo-Klaviermusik und möchte für statistische Untersuchungen eine Datenbank mit Pianisten, Komponisten und Klavierwerken aufbauen, um beispielsweise herausfinden zu können, welche Werke von Horowitz, nicht jedoch von Gould eingespielt wurden.

Sein erster Datenbankentwurf verwendet die folgenden Tabellen:

PIANIST						
PNR	NAME	VORNAME	KUENSTLERNAME	GEBDAT	GESTDAT	GEBURTSLAND
...

WERK					
WNR	TITEL	KATEGORIE	KOMPOSITIONSJAHR	KNR	PNR
...

KOMPONIST						
KNR	NAME	VORNAME	KUENSTLERNAME	GEBDAT	GESTDAT	GEBURTSLAND
...

Offensichtlich ist dieses Datenmodell nicht nur suboptimal, sondern schwer verbesserungsbedürftig – eine Aufgabe, die sich nun Ihnen stellt. Entwerfen Sie ein verbessertes Modell, das die folgenden Punkte behandelt:

- Es soll keine Trennung mehr zwischen Tabellen für Komponisten und Pianisten mehr stattfinden – führen Sie also eine neue, gemeinsame Tabelle namens PERSON ein, die alle Einträge aus den beiden vorher verwendeten Tabellen beinhaltet. Vorsicht: Wie machen Sie kenntlich, welcher Eintrag einem Pianisten beziehungsweise einem Komponisten (oder gar einer Person, die sowohl Komponist als auch Pianist war) entspricht? Verwenden Sie keine magischen Werte, sondern eher eine zusätzliche Tabelle für diese Information!

- Die Handhabung der Geburtsländer ist, zumindest im vorliegenden Modell, nicht korrekt normalisiert – korrigieren Sie dies ebenfalls. Das gleiche gilt übrigens für die Spalte **KATEGORIE** in den Kompositionen!
 - Die Verknüpfung zwischen Pianisten und Werken ist offensichtlich unbrauchbar – beheben Sie dies ebenfalls¹.
1. Entwickeln Sie nun ein verbessertes Datenmodell, das allen oben genannten Punkten Rechnung trägt und stellen Sie es als Entity-Relationship-Modell graphisch dar. (10 Punkte)
 2. Schreiben Sie die benötigten Tabellen für ein normalisiertes Modell, das auf Ihrem ER-Diagramm basiert, mit all' ihren zugehörigen Spalten auf. (6 Punkte)
 3. Formulieren Sie ein SQL-Statement, das Ihnen die Anzahl russischer Pianisten in der Datenbank ausgibt. Verwenden Sie hierbei ausschließlich Joins². (4 Punkte)
 4. Schreiben Sie Ihr eben entwickeltes SQL-Statement nun noch einmal – verwenden Sie nun jedoch keine Joins, sondern ausschließlich Subselects! (4 Punkte)

Aufgabe 3:

Haben Sie sich je gefragt, was mit den Leichen Verstorbener, die nicht komplett an einem Ort bestattet werden können, geschieht? Nein?

Dann ist es an der Zeit, daß Sie das Traditionsunternehmen **Rest in Pieces Ltd.** kennenlernen, das sich auf partielle Bestattungen und Rückführungen spezialisiert hat und naturgemäß – denken Sie nur an den in manchen Kulturen blühenden Handel mit Schrumpfköpfen – weltweit tätig ist.

Ein solches Unternehmen kann natürlich nicht ohne Datenbankunterstützung geführt werden. Neben einfachen Fällen wie dem Massenmörder Haarmann, dessen Kopf in Formalin in einem Kabinett steht, während der Rest seiner sterblichen Hülle verbrannt und verstreut wurde, so daß es sich hierbei um einen relativ einfachen Fall handelt, müssen Sie auch mit Härtefällen wie beispielsweise Albert Einstein, dessen Gehirn in hunderte handlicher Würfel zerschnitten an Dutzenden verschiedener Orte lagert oder Lenin, den ein ähnliches Schicksal ereilt hat, umgehen können.

Aus diesem Grunde verwendet **Rest in Pieces Ltd.** folgendes Datenbankmodell, um den Überblick über die Aufbewahrungsorte der einzelnen Körperteile der Verstorbenen zu behalten:

PERSON			KOERPERTEIL		AUFBEWAHRUNGORT	
<u>PNR</u>	NAME	VORNAME	<u>KNR</u>	BESCHREIBUNG	<u>ANR</u>	BESCHREIBUNG
...

TEIL_BEFINDET_SICH_IN		
<u>KNR</u>	<u>ANR</u>	<u>PNR</u>
...

1. Geben Sie ein SQL-Statement an, das Ihnen die Anzahl von Personen in der oben dargestellten Datenbank ausgibt. (1 Punkt)
2. Formulieren Sie eine SQL-Abfrage, die Ihnen ausgibt, wieviele einzelne Körperteile von Albert Einstein sich in der Datenbank von **Rest in Pieces Ltd.** befinden. (4 Punkte)

¹Die Zuordnung zwischen Komponisten und Werken können Sie hingegen beibehalten – ignorieren Sie einfach Kompositionen, die mehr als einen Komponisten besitzen!

²Sie sollten hierfür 3 Tabellen benötigen!

3. Schreiben Sie ein SQL-Statement, das Ihnen eine Liste aus Nachnamen von Personen ausgibt, von denen Teile im Frankfurter Max-Planck-Institut für Hirnforschung liegen. Achten Sie darauf, daß die Namensliste aufsteigend sortiert wird. Verwenden Sie ausschließlich Joins zur Lösung dieser Aufgabe. (6 Punkte)
4. Nun formulieren Sie die obigen SQL-Abfrage erneut, jedoch ohne Joins, sondern ausschließlich mit Hilfe von Subselects! (3 Punkte)
5. Geben Sie eine SQL-Abfrage an, die Ihnen eine sortierte Liste von Aufbewahrungsorten ausgibt, an denen sich Gehirne von Verstorbenen, die in der **Rest in Pieces Ltd.**-Datenbank erfaßt sind, befinden. Verwenden Sie ausschließlich Joins zur Lösung dieser Aufgabe. (6 Punkte)