Dr. Stefan Brass
Institut für Informatik
Universität Giessen

5. Februar 2002

Vorlesung Dokumentation und Datenbanken
— Klausur —

Name: ______________________________

Geburtsdatum: ______________________________

Geburtsort: ______________________________

(Diese Daten werden zur Ausstellung des Leistungsnachweises benötigt.)

Aufgabe	Punkte	Maximale Punktzahl
1 (SQL Anfragen)		21
2 (SQL CREATE TABLE)		6
3 (Integritätsbedingungen)		6
Summe		33

Anleitung

- Sie haben Zeit bis 17^{45} (d.h. ca.1:30).
- Bitte schreiben Sie die Antworten in den vorgegebenen Platz. Falls der Platz nicht ausreicht, können Sie weitere Blätter oder die Rückseite verwenden, aber bitte markieren Sie klar, daß es noch eine Fortsetzung gibt. Schreiben Sie Ihren Namen und die Nummer der Aufgabe auf alle zusätzlichen Blätter.
- Bitte fragen Sie, wenn Sie die Aufgabe nicht verstehen, oder die Aufgabe Ihnen mehrdeutig erscheint.
- Die Klausur hat 9 Seiten. Bitte prüfen Sie die Vollständigkeit.
- Bitte nehmen Sie keine Klausur mit und auch keine Notizen über die Aufgaben. Sie bekommen die Klausur zurück sobald sie korrigiert ist.
- Sie können alle schriftlichen Aufzeichnungen verwenden, die vor Beginn der Klausur entstanden sind. Dagegen sind Computer (Laptops etc.) leider nicht zulässig.
- Bitte vermeiden Sie jedes Verhalten, das als Täuschungsversuch mißverstanden werden könnte. Achten Sie auf einen ausreichenden Abstand zu Ihren Nachbarn und setzen Sie sich nicht direkt neben die Teilnehmer, mit denen Sie vorher immer die Hausaufgaben zusammen abgegeben haben. Schauen Sie nicht in Richtung der Klausuren anderer Teilnehmer und sprechen Sie nicht mit anderen Teilnehmern, bis alle Klausuren abgegeben sind.
- Bitte stellen Sie sicher, daß ich Ihre Handschrift lesen kann. Verwenden Sie keine rote Tinte.
- Meine Zeitschätzung ist: Aufgabe 1 a) bis d) je 5 Minuten, e) bis g) je 10 Minuten, d.h. insgesamt sollten 50 Minuten für Aufgabe 1 ausreichen. Für Aufgabe 2 brauchen Sie ca. 10 Minuten. Für Aufgabe 3 brauchen Sie für jede Teilaufgabe ca. 3 Minuten, also 9 Minuten insgesamt. Dies ergibt 1:09, die Zeitreserve ist also nicht sehr groß. Falls Sie Schwierigkeiten mit einer Aufgabe haben, bearbeiten Sie zunächst die anderen.
- Viel Erfolg!

Beispiel-Datenbank für Aufgaben 1, 2 und 3

Ein Fan vom Silvesterfeuerwerk hat folgende Datenbank angelegt:

- Die erste Tabelle enthält die Grunddaten der Feuerwerkskörper. `NAME` ist als alternativer Schlüssel deklariert, d.h. ebenfalls eindeutig.

ARTIKEL			
NO	ANBIETER	NAME	ART
1	Comet	Purpur-Perl-Fontäne	Fontäne
2	Comet	4-Farben-Fontäne	Fontäne
3	Comet	Primero 5	Batterie-Feuerwerk
4	Weco	Feuerblumen-Vulkan	Fontäne
5	Weco	Super-Albatros	Feuerwirbel
6	Nico	Silbersonne	Sonne

- Die zweite Tabelle die Information, welcher Artikel in welchem Jahr gekauft wurde. Einige Artikel sind Bestandteil von Sortimenten, dann ist der Einzelpreis nicht bekannt und daher null. `NO` ist ein Fremdschlüssel, der auf `ARTIKEL` verweist.

KAUF			
NO	JAHR	PREIS^o	SORTIMENT^o
1	2000		Feuerzauber
2	2000		Feuerzauber
3	2000	19.95	
4	2001	3.50	
5	2001	4.00	
6	2000	5.00	
6	2001	6.00	

- Schließlich gibt es noch eine Tabelle mit Bewertungen der Feuerwerkskörper. Jedes Familienmitglied kann seine/ihre eigene Bewertung abspeichern. `NO` und `JAHR` zusammen sind ein Fremdschlüssel, der auf `KAUF` verweist.

BEWERTUNG				
NO	JAHR	BENUTZER	NOTE	TEXT
1	2000	SB	3.0	···
3	2000	SB	1.0	···
4	2001	SB	1.7	···
4	2001	PB	1.3	···
5	2001	SB	1.0	···
6	2000	SB	1.0	···
6	2001	SB	1.0	···
6	2001	PB	1.0	···

Aufgabe 1 (SQL Anfragen) 21 Punkte

Formulieren Sie die folgenden Anfragen in SQL. Sie bekommen drei Punkte für jede korrekte Anfrage. Natürlich sollen Ihre Anfragen nicht nur mit den obigen Beispiel-Daten funktionieren, sondern für beliebige Tabelleninhalte. Beachten Sie, daß auch für unnötige Komplikationen Punkte abgezogen werden können. Die Anfragen sollen keine Duplikate liefern, aber für ein unnötiges `DISTINCT` werden auch Punkte abgezogen. Die Spalten des Anfrageergebnisses können auch andere Namen haben, als in den Beispielen gezeigt.

a) Geben Sie die Namen aller Anbieter aus, von denen Feuerwerkskörper in der Datenbank stehen. Die Liste soll alphabetisch sortiert sein. Im Beispiel soll das Anfrage-Ergebnis so aussehen:

ANBIETER
Comet
Nico
Weco

b) Geben Sie Anbieter und Namen aller Fontänen und Sonnen aus. Die Ausgabe soll nach Anbietern sortiert sein, bei gleichem Anbieter nach dem Namen. Im Beispiel sollte das Ergebnis so aussehen:

ANBIETER	NAME
Comet	4-Farben-Fontäne
Comet	Purpur-Perl-Fontäne
Nico	Silbersonne
Weco	Feuerblumen-Vulkan

c) Geben Sie die Namen aller Artikel aus, die 2001 gekauft wurden, und mit 1.0 bewertet wurden. Es kann mehrere Bewertungen für den gleichen Artikel geben, dann ist nur wichtig, daß wenigstens eine Bewertung die Note 1.0 enthält. Die Bewertung soll allerdings auch aus dem Jahr 2001 stammen. Eine besondere Sortierung ist nicht notwendig. Im Beispiel sollte das Ergebnis so aussehen:

NAME
Super-Albatros
Silbersonne

d) Für welche Artikel in der Datenbank gibt es keine Bewertung? Im Beispiel sollte das Ergebnis so aussehen:

NAME
Vier-Farben-Fontäne

e) Geben Sie für jeden Artikel Anbieter, Name und die durchschnittliche Bewertung aus. Natürlich sollen nur Artikel berücksichtigt werden, für die es mindestens eine Bewertung gibt. Die Spalte mit der durchschnittlichen Bewertung soll `DURCHSCHNITTSNOTE` genannt werden. (Hinweis: In der Vorlesung wurde einmal gezeigt, wie man Durchschnitt mit Summe und Anzahl simulieren kann, aber hier können Sie natürlich die in SQL eingebaute Durchschnittsfunktion verwenden.) Im Beispiel wäre das Ergebnis:

ANBIETER	NAME	DURCHSCHNITTSNOTE
Comet	Purpur-Perl-Fontäne	3.0
Comet	Primero 5	1.0
Weco	Feuerblumen-Vulkan	1.5
Weco	Super-Albatros	1.0
Nico	Silbersonne	1.0

f) Welche Artikel wurden sowohl 2000 als auch 2001 gekauft und kosteten 2001 mindestens 10% mehr? Im Beispiel müßte das Ergebnis so aussehen:

NAME
Silbersonne

g) Was ist der teuerste Artikel, der jemals gekauft wurde? Geben Sie Anbieter, Name, Jahr und Preis aus.

ANBIETER	NAME	JAHR	PREIS
Comet	Primero 5	2000	19.95

Aufgabe 2 (SQL CREATE TABLE) 6 Punkte

Schreiben Sie ein `CREATE TABLE` Statement für die Tabelle „`KAUF`".

- Deklarieren Sie die üblichen Integritätsbedingungen (Schlüssel, Fremdschlüssel, `NOT NULL`).
- Zusätzlich stellen Sie über `CHECK`-Constraints bitte sicher, daß genau eins der Attribute "`PREIS`" und "`SORTIMENT`" null ist. D.h. wenn `PREIS` nicht null ist, muß `SORTIMENT` null sein, und umgekehrt.
- Geben Sie dem oder den `CHECK`-Constraints bitte Namen.
- Die Attribute `NO` und `JAHR` sollen vierstellige Zahlen sein. `PREIS` soll drei Stellen vor dem Komma und 2 hinter dem Komma haben. `SORTIMENT` soll eine Zeichenkette variabler Länge von bis zu 30 Zeichen sein.
- Zur Vereinfachung brauchen Sie keine `CHECK`-Constraints zu formulieren, die sicherstellen, daß `NO`, `JAHR` und `PREIS` nicht-negativ/sinnvoll sind.

Aufgabe 3 (Integritätsbedingungen) 6 Punkte

Bitte kreuzen Sie jeweils die richtige Antwort an. Pro Teilaufgabe gibt es nur genau eine richtige Antwort. Sie bekommen 2 Punkte pro richtige Antwort. Wenn Sie die richtige Antwort nicht wissen, sollten Sie wenigstens raten.

a) Jemand schlägt vor, als Schlüssel von `BEWERTUNG` nicht die Kombination von `NO`, `JAHR` und `BENUTZER` (wie bisher) zu wählen, sondern nur `NO` und `BENUTZER` zusammen. Was würde das bedeuten?

- ☐ Es wäre nicht möglich, daß zwei verschiedene Benutzer Bewertungen für den gleichen Artikel eintragen.
- ☐ Jeder Benutzer kann jeden Artikel höchstens einmal bewerten.
- ☐ Für jedes Jahr könnte dann höchstens eine Bewertung eingetragen werden.
- ☐ Alle obigen Aussagen sind falsch.

b) Das Attribut `NO` der Tabelle `KAUF` ist ein Fremdschlüssel, der auf `ARTIKEL` verweist. Für das Attribut `NO` in `ARTIKEL` ist ein `CHECK`-Constraint deklariert, der sicherstellt, daß `NO` dort keine negativen Zahlen enthält. Wenn man möchte, daß `NO` auch in `KAUF` keine negativen Zahlen enthält, muß man dann auch dort einen entsprechenden `CHECK`-Constraint formulieren? (In Aufgabe 2 brauchen Sie sich um dieses Problem nicht zu kümmern.)

- ☐ Ja. Ohne den `CHECK`-Constraint kann `NO` in `KAUF` negative Zahlen enthalten.
- ☐ Nein. Der `CHECK`-Constraint in `ARTIKEL` und die Fremdschlüssel-Bedingung in `KAUF` reichen aus.

c) Die Tabelle `ARTIKEL` soll nur Feuerwerkskörper enthalten, die auch wenigstens einmal gekauft wurden. Die bisherigen Integritätsbedingungen stellen das nicht sicher. Kann man das Problem lösen, indem man `NO` in `ARTIKEL` als Fremdschlüssel deklariert, der `KAUF` referenziert?

- ☐ Ja.
- ☐ Nein. Das wäre kein gültiger Fremdschlüssel.
- ☐ Man kann den Fremdschlüssel deklarieren, aber er würde das Problem nicht lösen.