Prof. Dr. Stefan Brass
Institut für Informatik
MLU Halle-Wittenberg

12. Dezember 2003

Vorlesung “Datenbanken I”
— Zwischenklausur —

Name: ________________________

Matrikelnummer: ________________________

Studiengang: ________________________

Aufgabe	Punkte	Max. Punkte	Zeit
1 (Integritätsbedingungen)		4	12 min
2 (Relationale Algebra)		6	15 min
3 (SQL)		9	20 min
4 (SQL Anfragen analysieren, Theorie)		4	12 min
Summe		23	59 min

Hinweise:

- Bearbeitungsdauer: 60 Minuten
- Skript, Bücher, Notizen sind erlaubt. Notebooks, PDAs, etc. dürfen nicht verwendet werden. Mobiltelefone bitte ausschalten.
- Die Klausur hat 8 Seiten. Bitte prüfen Sie die Vollständigkeit.
- Bitte benutzen Sie den vorgegebenen Platz. Wenn Sie auf die Rückseite ausweichen müssen, markieren Sie klar, daß es eine Fortsetzung gibt.
- Sie bekommen einen Extrazettel mit dem Datenbankschema (identisch zu Seite 2 der Klausur). Auf diesem Zettel bitte nichts schreiben und mit der Klausur abgeben.
- Tauschen Sie keinesfalls irgendwelche Dinge mit den Nachbarn aus. Notfalls rufen Sie eine Aufsichtsperson zur Kontrolle.
- Bei den Aufgaben zum Ankreuzen sollten Sie wenigstens raten, wenn Sie die richtige Lösung nicht wissen (wenn Sie nichts ankreuzen, haben Sie den Punkt auf jeden Fall verloren). Es ist jeweils genau eine Antwort pro Teilaufgabe richtig.
- Fragen Sie, wenn Ihnen eine Aufgabe nicht klar ist!

Beispiel-Datenbank (für alle Aufgaben)

Der Dozent möchte sein Skript in ein Online-Lernsystem umwandeln. Dies soll u.a. erlauben, daß Studierende selbst die Teile des Skripts auswählen, die sie lernen wollen.

- Die Tabelle "ABSCHNITTE" enthält einen Eintrag für jeden Abschnitt des Skripts (Lerneinheit). Jeder Abschnitt hat eine eindeutige Nummer "AB" (Primärschlüssel). Die Überschrift "TITEL" des Abschnitts ist Sekundärschlüssel.

ABSCHNITTE			
<u>AB</u>	TITEL	URL	SEITEN
101	Einleitung	c1_intro/s1.pdf	5
201	Relationales Modell: Basis	c2_relmo/s1.pdf	3
202	Integritätsbedingungen	c2_relmo/s2.pdf	7
203	Relationale Algebra	c2_relmo/s3.pdf	12
301	Einfache Anfragen in SQL	c3_sql_i/s1.pdf	6

- Die zweite Tabelle enthält Datenbank-Begriffe, die im Skript eingeführt oder verwendet werden. Begriffe werden über eine eindeutige Nummer "BEG" identifiziert, "NAME" ist Sekundärschlüssel. Begriffe sind in eine Hierarchie (wie ein Verzeichnisbaum) eingeordnet. Das Attribut OBERBEG ist ein Fremdschlüssel, der auf die Tabelle selbst verweist. Es ist das einzige Attribut der Datenbank, das null sein kann.

BEGRIFFE		
<u>BEG</u>	NAME	OBERBEG^o
1000	Datenmodell	(null)
1001	Relationales Modell	1000
1002	Schlüssel	1001
1003	Fremdschlüssel	1001
1004	Datei	(null)
1005	Anfragesprachen	(null)
1006	Relationale Algebra	1005
1007	Projektion	1006

- Die dritte Tabelle setzt Abschnitte und Begriffe in Beziehung (wie ein Index in einem Buch). "AB" ist ein Fremdschlüssel, der auf "ABSCHNITTE" verweist, und "BEG" verweist auf "BEGRIFFE". In der Spalte "ART" steht der Wert "DEF" für ein definierendes Vorkommen des Begriffs, der Wert "BSP" für ein Beispiel, und der Wert "VORAUS" für ein Vorkommen des Begriffes, bei dem seine Kenntnis bereits vorausgesetzt wird. Den Primärschlüssel für diese Tabelle müssen Sie in Aufgabe 1a) selbst bestimmen.

VORKOMMEN		
AB	BEG	ART
101	1004	VORAUS
101	1000	DEF
101	1005	DEF
201	1000	VORAUS
202	1002	DEF
202	1002	BSP
202	1003	DEF
202	1007	VORAUS
203	1007	BSP
203	1007	DEF

Aufgabe 1 (Integritätsbedingungen) 4 Punkte

Kreuzen Sie die jeweils richtige Antwort an. Machen Sie pro Unteraufgabe nur ein Kreuz. Pro Unteraufgabe gibt es einen Punkt.

a) Was wäre der richtige Primärschlüssel für die Tabelle `VORKOMMEN`? Der Beispielzustand soll natürlich zulässig sein.

- ☐ `AB`
- ☐ `BEG`
- ☐ `ART`
- ☐ `AB, BEG`
- ☐ `AB, BEG, ART`

b) Das Attribut für die Seitenanzahl eines Abschnitts ist im gegebenen Datenbankzustand eindeutig. Kann man "`SEITEN`" als Sekundärschlüssel deklarieren?

- ☐ Ja. Das wäre hilfreich, um die Korrektheit der Daten zu überwachen.
- ☐ Nein, das würde sinnvolle Zustände ausschließen.

c) Betrachten Sie die Fremdschlüsselbedingung `VORKOMMEN(AB→ABSCHNITT,...)`. Welche der folgenden Operationen kann den Fremdschlüssel möglicherweise verletzten (d.h. das DBMS muß bei dieser Operation die Fremdschlüsselbedingung prüfen)?

- ☐ Einfügung in `ABSCHNITTE`
- ☐ Einfügung in `VORKOMMEN`
- ☐ Löschung aus `VORKOMMEN`

d) Angenommen, man würde folgendes Tupel in die Tabelle `BEGRIFFE` einfügen:

BEG	NAME	OBERBEG^o
1008	SQL	1009

Würde dies eine Integritätsbedingung verletzten?

- ☐ Der Primärschlüssel `BEG` von `BEGRIFFE` wäre verletzt.
- ☐ Der Sekundärschlüssel `NAME` wäre verletzt.
- ☐ Der Fremdschlüssel `OBERBEG` wäre verletzt.
- ☐ Der Fremdschlüssel `BEG` in `VORKOMMEN` wäre verletzt.
- ☐ Die Einfügung wird ohne Fehler ausgeführt.

Aufgabe 2 (Relationenalgebra) 6 Punkte

Formulieren Sie die folgenden Anfragen in der Relationenalgebra. 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. Um Missverständnisse zu vermeiden, sind bei Aufgabe 2 und 3 jeweils die Ergebnisse der gesuchten Anfragen im Beispiel-Zustand angegeben.

a) Geben Sie alle Begriffe aus (jeweils den Namen), die im Abschnitt mit dem Titel "Integritätsbedingungen" definiert werden (Art des Vorkommens: "`DEF`").

Anfrage-Ergebnis im Beispiel:

NAME
Schlüssel
Fremdschlüssel

b) Geben Sie alle Begriffe aus (jeweils Nummer und Name), die irgendwo im Skript vorausgesetzt werden (Art des Vorkommens "`VORAUS`"), aber nirgendwo definiert werden (Art "`DEF`").

Anfrage-Ergebnis im Beispiel:

BEG	NAME
1004	Datei

Aufgabe 3 (SQL Anfragen) 9 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.

a) In welchen Titeln von Abschnitten kommen sowohl die Teilzeichenkette “`SQL`” als auch die Teilzeichenkette “`Anfragen`” vor? Es ist nicht bekannt, ob zuerst “`SQL`” und dann “`Anfragen`” kommt, oder umgekehrt. Geben Sie jeweils die Abschnittsnummer, den Titel und die URL aus.

Anfrage-Ergebnis im Beispiel:

AB	TITEL	URL
301	Einfache Anfragen in SQL	c3_sql_i/s1.pdf

b) Geben Sie Nummer und Titel aller Abschnitte aus, die mindestens einen Begriff definieren, der in der Begriffshierarchie ganz oben steht, für den also “`OBERBEG`” einen Nullwert enthält. Im Beispiel würde Abschnitt 101 die Begriffe 1000 (Datenmodell) und 1005 (Anfragesprachen) definieren. Denken Sie wie bei allen Anfragen darüber nach, ob eine explizite Duplikateliminierung notwendig ist.

Anfrage-Ergebnis im Beispiel:

AB	TITEL
101	Einleitung

c) Es ist schlecht, wenn Begriffe definiert werden, nachdem Sie bereits vorausgesetzt wurden. Schreiben Sie eine Anfrage, die Abschnitte A und B findet, so daß B eine größere Nummer (`AB`) als A hat, aber in B ein Begriff definiert wird, der in A vorausgesetzt wird. Geben Sie den Namen des Begriffs und die Nummern der beiden Abschnitte A und B aus. Die Ausgabe-Spalte mit der Abschnittsnummer von A soll "`VORAUSGESETZT_IN`" heißen, und die mit der Abschnittsnummer von B soll "`DEFINIERT_IN`" heißen.

Anfrage-Ergebnis im Beispiel:

NAME	VORAUSGESETZT_IN	DEFINIERT_IN
Projektion	202	203

Aufgabe 4 (SQL Anfragen analysieren, Theorie) 4 Punkte

Gegeben sei die folgende Anfrage. Sie soll diejenigen Begriffe ausgeben, die dem Begriff "`Datenmodell`" direkt untergeordnet sind:

```
SELECT X.NAME
FROM   BEGRIFFE X, BEGRIFFE Y
WHERE  X.OBERBEG = Y.BEG AND Y.NAME = 'Datenmodell'
```

Kreuzen Sie die jeweils richtige Antwort an. Machen Sie pro Unteraufgabe nur ein Kreuz.

a) `OBERBEG` kann null sein. Wäre es vielleicht wichtig, die Anfrage so zu formulieren:

```
SELECT X.NAME
FROM   BEGRIFFE X, BEGRIFFE Y
WHERE  X.OBERBEG IS NOT NULL
AND    X.OBERBEG = Y.BEG AND Y.NAME = 'Datenmodell'
```

- ☐ Dies liefert immer genau das gleiche Ergebnis wie die gegebene Anfrage.
- ☐ Dies kann manchmal weniger Ergebnistupel als die gegebene Anfrage liefern (stärkere Einschränkung).
- ☐ Ohne diesen zusätzlichen Test auf Null kann die gegebene Anfrage zu einer Fehlermeldung führen.

b) Kann die gegebene Anfrage Duplikate liefern? `BEG` ist Primärschlüssel der Tabelle `BEGRIFFE`, `NAME` ist ein Sekundärschlüssel.

- ☐ Ja. Man sollte `DISTINCT` angeben.
- ☐ Nein. `DISTINCT` wäre überflüssig.

c) Kann man in der Relationenalgebra oder SQL-92 eine Anfrage formulieren, die nicht nur die direkten Unterbegriffe eines Begriffs liefert, sondern alle (auch indirekt) untergeordneten Begriffe? Die Tiefe des Baumes (Schachtelungstiefe der "Unterbegriff"-Beziehung) ist nicht bekannt.

- ☐ Nein. In einer einzigen Anfrage ist das nicht möglich.
- ☐ Ja, sicher. Dafür würden sogar schon die Operationen Selektion, Projektion, Join ausreichen.

d) Es sollen jetzt nicht nur die direkten Unterbegriffe von "`Datenmodell`" ausgeben werden, sondern auch die von "`Integritätsbedingung`". Ist die folgende Anfrage dafür korrekt?

```
SELECT X.NAME
FROM   BEGRIFFE X, BEGRIFFE Y
WHERE  Y.NAME = 'Datenmodell'
AND    Y.NAME = 'Integritätsbedingungen'
AND    X.OBERBEG = Y.BEG
```

- ☐ Korrekt.
- ☐ Die Bedingung ist inkonsistent (widersprüchlich).
- ☐ Es fehlt eine Join-Bedingung.
- ☐ Man muß hier explizit Klammern setzen.