Dr. Stefan Brass
Institut für Informatik
Universität Giessen

13. Februar 2001

Vorlesung Datenbanksysteme
— Endklausur —

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 (Funktionale Abhängigkeiten, BCNF)		8
Summe		35

Anleitung

- Sie haben Zeit bis 11^{45} (d.h. ca.1:20).
- 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 10 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. 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 12 Minuten insgesamt. Dies ergibt 1:12, 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 und 2

Gegeben seien drei Tabellen aus einem fiktiven Data Dictionary, die Benutzer, Tabellen und Zugriffsrechte für Tabellen beschreiben. Die erste Tabelle ist eine Liste aller Benutzer des Datenbanksystems. Sie enthält die Benutzerkennung und den vollständigen Namen:

USERS	
USER	FULL_NAME^o
SYS	Data Dictionary
SB	Stefan Brass
NINA	Nina Brass
LISA	(null)

Die zweite Tabelle enthält eine Liste aller Tabellen des Systems:

TABS(OID, OWNER→USERS, TABNAME)

Wie in Oracle hat jede Tabelle einen Besitzer. Alle Tabellen des Systems sind eindeutig über eine Objektnummer (OID) identifiziert.

TABS		
OID	OWNER	TABNAME
1	SYS	USERS
2	SYS	TABS
3	SYS	GRANTS
4	SB	STUDENTS
5	SB	RESULTS
6	NINA	OUR_BOOKS

Schließlich gibt es noch eine Tabelle, in der alle „Grants“ (vergebenen Zugriffsrechte) verzeichnet sind:

GRANTS(OID→TABS, GRANTOR→USERS, GRANTEE→USERS, PRIVILEGE, GRANTABLE).

Dabei ist GRANTOR der Benutzer, der das Recht vergeben hat, und GRANTEE ist der Benutzer, der das Recht bekommen hat. PRIVILEGE ist ein Recht, wie z.B. SELECT für Leserechte und INSERT für Einfügerechte. GRANTABLE ist „Y“ falls das Recht WITH GRANT OPTION vergeben wurde, sonst „N“.

GRANTS				
OID	GRANTOR	GRANTEE	PRIVILEGE	GRANTABLE
1	SYS	SB	SELECT	N
6	NINA	SB	SELECT	Y
6	NINA	SB	INSERT	Y
6	SB	LISA	SELECT	N

Alle Attribute außer FULL_NAME sind nicht null. Sie brauchen nicht die Möglichkeit von Grants an PUBLIC zu berücksichtigen.

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 bitte Benutzerkennung und vollständigen Namen aller Benutzer aus, mit Ausnahme des Benutzers `SYS` und von Benutzern, deren vollständiger Name ein Nullwert ist. Im Beispiel soll das Anfrage-Ergebnis so aussehen:

USER	FULL_NAME
SB	Stefan Brass
NINA	Nina Brass

b) Geben Sie alle Benutzer aus (Benutzerkennung und vollständiger Name), die Rechte an Tabellen besitzen, die `NINA` gehören (d.h. die als `GRANTEE` für solche Tabellen eingetragen sind). Im Beispiel sollte das Ergebnis so aussehen:

USER	FULL_NAME
SB	Stefan Brass
LISA	(null)

c) Geben Sie alle Benutzer aus (nur Benutzerkennung), die keine Tabellen besitzen. Im Beispiel sollte das Ergebnis so aussehen:

USER
LISA

d) Welche Benutzer (Benutzerkennung) besitzen mindestens 3 Tabellen? Im Beispiel wäre das Ergebnis (die Ergebnis-Spalte kann auch einen anderen Namen haben):

OWNER
SYS

e) Wer hat sowohl Lese-Rechte als auch Einfüge-Rechte für die Tabelle „`OUR_BOOKS`“ (außer der Besitzerin der Tabelle)? Im Beispiel wäre das Ergebnis (die Ergebnis-Spalte kann auch einen anderen Namen haben):

GRANTEE
SB

f) Geben Sie alle Tabellen aus (Benutzerkennung des Besitzers und Tabellenname), an denen der Benutzer „`SB`“ Leserechte hat, d.h. bei denen er entweder als `GRANTEE` für ein `SELECT`-Privilege eingetragen ist, oder von denen er der Besitzer ist. Sie können voraussetzen, daß der Besitzer einer Tabelle niemals selbst als `GRANTEE` eingetragen wird. Sortieren Sie das Anfrageergebnis nach dem Besitzer der Tabelle. Im Beispiel müßte das Ergebnis so aussehen:

OWNER	TABNAME
NINA	OUR_BOOKS
SB	STUDENTS
SB	RESULTS

g) Gibt es jemanden außer `NINA`, der/die Lese-Rechte für alle Tabellen hat, die `NINA` gehören? Drucken Sie ggf. die Benutzerkennungen dieser Benutzer. Natürlich kann Nina mehr als eine Tabelle besitzen. Wenn Sie wollen, können Sie voraussetzen, daß es mindestens eine ist. Sie können aber auch eine Anfrage schreiben, die alle Benutzer ausgibt, falls `NINA` keine Tabellen besitzt. Im Beispiel wäre das Ergebnis (die Ergebnis-Spalte kann auch einen anderen Namen haben):

USER
SB

Aufgabe 2 (SQL CREATE TABLE) 6 Punkte

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

- Deklarieren Sie die üblichen Integritätsbedingungen (Schlüssel, Fremdschlüssel, `NOT NULL`).
- Zusätzlich stellen Sie über `CHECK`-Constraints bitte sicher, daß `GRANTOR` und `GRANTEE` verschieden sind, und `GRANTABLE` nur die beiden Werte `Y` und `N` annehmen kann. Für `PRIVILEGE` würde man eigentlich auch einen `CHECK`-Constraint benötigen, aber Sie können ihn zur Abkürzung weglassen.
- Geben Sie den beiden `CHECK`-Constraints bitte Namen.
- Die `OID` ist eine sechsstellige Zahl. Sie können voraussetzen, daß `OID` bereits in `TABS` als positiv deklariert wurde.
- Benutzerkennungen sind Zeichenketten variabler Länge mit Maximallänge 32.
- `PRIVILEGE` sind Zeichenketten der festen Länge 6.

Aufgabe 3 (Funktionale Abhängigkeiten, BCNF) 8 Punkte

Gegeben sei eine Variante der obigen Tabelle `TABS`, die noch um das Datum der Tabellenerzeugung erweitert wurde:

TABS			
OID	OWNER	TABNAME	CREATED
1	SYS	USERS	01.10.2000
2	SYS	TABS	01.10.2000
3	SYS	GRANTS	01.10.2000
4	SB	STUDENTS	15.10.2000
5	SB	RESULTS	15.10.2000
6	NINA	OUR_BOOKS	01.01.2001

Folgende funktionalen Abhängigkeiten gelten:

$$\begin{aligned} \text{OID} &\rightarrow \text{OWNER, TABNAME} \\ \text{OID} &\rightarrow \text{CREATED} \\ \text{OWNER, TABNAME} &\rightarrow \text{OID} \end{aligned}$$

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) Ein Entwickler hat vorgeschlagen, noch die funktionale Abhängigkeit

$$\text{OWNER} \rightarrow \text{TABNAME}$$

hinzuzunehmen. Was bedeutet diese funktionale Abhängigkeit?

- ☐ Zwei verschiedene Benutzer können nicht eine Tabelle gleichen Namens haben.
- ☐ Jeder Benutzer kann nur eine Tabelle haben.
- ☐ Tabellen können über die Kombination von Besitzer und Tabellenname identifiziert werden.
- ☐ Alle obigen Aussagen sind falsch.

b) Gilt in dem obigen Beispielzustand der Tabelle die funktionale Abhängigkeit

$$\text{TABNAME, CREATED} \rightarrow \text{OID}$$

- ☐ Ja.
- ☐ Nein.

c) Was wären alle minimalen Schlüssel dieser Tabelle. Setzen Sie dabei nur die obigen drei funktionalen Abhängigkeiten voraus, nicht die Abhängigkeiten aus a) und b).

- ☐ Nur `OID` wäre ein Schlüssel.
- ☐ `OID` wäre ein Schlüssel, `OWNER`, `TABNAME`, `CREATED` wäre ein zweiter Schlüssel.
- ☐ `OID` wäre ein Schlüssel, `OWNER`, `TABNAME` wäre ein zweiter Schlüssel.
- ☐ Alle obigen Aussagen sind falsch.

d) Ist die Tabelle in BCNF?

- ☐ Ja.
- ☐ Nein, die funktionale Abhängigkeit `OID` → `OWNER, TABNAME` verletzt die BCNF-Bedingung.
- ☐ Nein, `OID` → `CREATED` verletzt die BCNF-Bedingung.
- ☐ Nein, `OWNER, TABNAME` → `OID` verletzt die BCNF-Bedingung.