Dr. Stefan Brass
Institut für Informatik
TU Clausthal

28. Januar 2003

Vorlesung “Datenbanken I”
— Endklausur —

Name: ______________________________

Matrikelnummer: ______________________________

Aufgabe	Punkte	Maximale Punktzahl
1 (SQL)		15
2 (CREATE TABLE, INSERT)		5
3 (ER-Diagramm)		6
4 (BCNF)		4
Summe		30

Anleitung

- Sie haben Zeit bis 11^{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 12 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. Tauschen Sie keinesfalls irgendwelche Papiere oder Gegenstände aus. Notfalls rufen Sie bitte den Dozenten zur Kontrolle.
- Bitte stellen Sie sicher, daß ich Ihre Handschrift lesen kann. Verwenden Sie keine rote Tinte. Sie dürfen mit Bleistift schreiben, aber dann könnte es schwieriger werden, mir Fehler bei der Korrektur nachzuweisen (fragen dürfen Sie natürlich auf jeden Fall).
- Meine Zeitschätzung ist: Aufgabe 1 a), c), d) je 10 Minuten, und Aufgabe b) und e) je 5 Minuten (d.h. insgesamt 40 Minuten für Aufgabe 1). Aufgabe 2 und 3 je 10 min, Aufgabe 4 15 Minuten. Dies ergibt 1:15, die Zeitreserve beträgt dann noch 15 Minuten. Falls Sie Schwierigkeiten mit einer Aufgabe haben, oder länger in den Materialien suchen müßten, bearbeiten Sie zunächst die anderen. Es empfiehlt sich aber, alle Aufgaben wenigstens zu versuchen (wenn Sie nichts hinschreiben, bekommen Sie für die Aufgabe sicher 0 Punkte).
- Viel Erfolg!

Beispiel-Datenbank für Aufgaben 1 und 2

Gegeben sei eine Datenbank für Bahnverbindungen.

- Die erste Tabelle enthält Daten über Züge. Züge fahren täglich, eventuell nicht Samstags und Sonntags (kompliziertere Ausnahmen seien zur Vereinfachung nicht betrachtet).

ZUEGE		
ZNR	SA	SO
RB 25786	N	N
RB 25742	Y	Y
RB 34348	Y	N
RB 34303	Y	N
RB 34350	Y	N
RB 34305	Y	N

- Die zweite Tabelle enthält Daten über Haltepunkte, d.h. wann welcher Zug in welchem Ort ankommt und abfährt. Beim Startbahnhof ist die Ankunft null, beim Zielbahnhof die Abfahrtzeit null. Die Spalten `ANKUNFT` und `ABFAHRT` seien von einem Datentyp `TIME`, auf dem die üblichen Vergleiche definiert sind (z.B. <, >). Zur Vereinfachung nehmen wir an, daß kein Zug über Mitternacht hinaus fährt und wir auch nicht an entsprechenden Umsteige-Verbindungen interessiert sind. Nur die Spalten `ANKUNFT` und `ABFAHRT` erlauben Nullwerte. `ZNR` ist ein Fremdschlüssel, der auf `ZUEGE` verweist.

FAHRPLAN			
ZNR	BAHNHOF	ANKUNFT^o	ABFAHRT^o
RB 25786	Braunschweig	(null)	6:08
RB 25786	Wolfenbüttel	6:17	6:18
RB 25786	Goslar	7:18	(null)
RB 25742	Braunschweig	(null)	7:13
RB 25742	Wolfenbüttel	7:22	7:23
RB 25742	Goslar	8:01	(null)
RB 34348	Braunschweig	(null)	8:04
RB 34348	Salzgitter-Ringelheim	8:24	8:29
RB 34348	Holzminden	9:35	(null)
RB 34303	Salzgitter-Ringelheim	8:40	8:41
RB 34303	Goslar	8:54	(null)
RB 34350	Braunschweig	(null)	10:04
RB 34350	Salzgitter-Ringelheim	10:24	10:29
RB 34350	Holzminden	11:35	(null)
RB 34305	Salzgitter-Ringelheim	10:40	10:41
RB 34305	Goslar	10:54	(null)

Aufgabe 1 (SQL Anfragen) 15 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. Sofern nicht ausdrücklich gefordert, spielt die Sortierung der Ausgabe keine Rolle. Entsprechend können die Spalten des Anfrageergebnisses auch andere Namen haben, als in den Beispielen gezeigt (wenn nicht explizit ein Name vorgeschrieben ist).

a) Geben Sie alle direkten Verbindungen (d.h. ohne Umsteigen) von Braunschweig nach Goslar aus, die am Samstag fahren (d.h. die Spalte `SA` enthält ein `Y`). Drucken Sie für jede solche Verbindung die Zugnummer, die Abfahrtszeit in Braunschweig und die Ankunftszeit in Goslar. Verbindungen von Goslar nach Braunschweig (d.h. in der umgekehrten Richtung) schließen Sie bitte aus (die Ankunftszeit in Goslar muß nach der Abfahrtszeit in Braunschweig sein). Sortieren Sie die Ausgabe nach der Abfahrtszeit in Braunschweig.

ZNR	ABFAHRT	ANKUNFT
RB 25742	7:13	8:01

b) Nehmen wir an, das Ergebnis von Aufgabe a) wurde als Sicht `BS_GO_DIREKT` deklariert:

BS_GO_DIREKT		
ZNR	ABFAHRT	ANKUNFT
RB 25742	7:13	8:01

Außerdem hat ein freundlicher Datenbank-Professor für Sie die Sicht `BS_GO_UMST` deklariert, die Verbindungen mit einmal Umsteigen enthält:

BS_GO_UMST				
ABFAHRT	ANKUNFT	ZUG1	ZUG2	UMST_IN
8:04	8:54	RB 34348	RB 34303	Salzgitter-Ringelheim
10:04	10:54	RB 34350	RB 34305	Salzgitter-Ringelheim
8:04	10:54	RB 34348	RB 34305	Salzgitter-Ringelheim

Offensichtlich war der Professor bei der Definition von Umsteigen sehr großzügig (der zweite Zug muß abfahren, nachdem der erste angekommen ist). Daher die dritte, offensichtlich schlechte Verbindung. Dieses Problem werden Sie unter c) lösen. Hier besteht Ihre Aufgabe darin, aus den beiden Sichten eine Sicht `BS_GO` zusammenzusetzen, die Verbindungen und mit höchstens einmal Umsteigen enthält. Es soll Abfahrtszeit, Ankunftszeit und der Umsteigebahnhof angegeben werden (ein Leerzeichen `' '` falls direkte Verbindung):

BS_GO		
ABFAHRT	ANKUNFT	UMST_IN
7:13	8:01	
8:04	8:54	Salzgitter-Ringelheim
10:04	10:54	Salzgitter-Ringelheim
8:04	10:54	Salzgitter-Ringelheim

c) Das Ergebnis von b) sei wie gefordert als Sicht BS_GO deklariert:

BS_GO		
ABFAHRT	ANKUNFT	UMST_IN
7:13	8:01	
8:04	8:54	Salzgitter-Ringelheim
10:04	10:54	Salzgitter-Ringelheim
8:04	10:54	Salzgitter-Ringelheim

Ihre Aufgabe besteht nun darin, alle schlechten Verbindungen zu entfernen. Eine Verbindung X zählt dabei als schlecht, wenn es eine Verbindung Y gibt, die später als X abfährt und früher als X ankommt (dann wäre es günstiger, nicht X zu nehmen, sondern Y). Dabei ist entweder bei der Abfahrt oder bei der Ankunft auch gleichzeitig erlaubt (nur so kann man die letzte Verbindung in der Tabelle ausschließen). Definieren Sie eine Sicht BS_GO_GUT, die nur die Teilmenge der Verbindungen aus BS_GO enthält, die nicht schlecht sind, d.h. nicht von einer anderen Verbindung geschlagen werden:

BS_GO_GUT		
ABFAHRT	ANKUNFT	UMST_IN
7:13	8:01	
8:04	8:54	Salzgitter-Ringelheim
10:04	10:54	Salzgitter-Ringelheim

d) Sei nun das Ergebnis aus c) gegeben:

BS_GO_GUT		
ABFAHRT	ANKUNFT	UMST_IN
7:13	8:01	
8:04	8:54	Salzgitter-Ringelheim
10:04	10:54	Salzgitter-Ringelheim

Angenommen, Sie müssen spätestens um 9:00 in Goslar sein. Sie wollen so spät wie möglich in Braunschweig losfahren. Formulieren Sie eine SQL-Anfrage, die die für Sie optimale Verbindung ausdruckt.

e) Geben Sie für jeden Bahnhof im Fahrplan aus, wie häufig er angefahren wird (d.h. wie häufig er im Fahrplan auftaucht, wobei die Ankunftszeit nicht null ist). Berücksichtigen Sie dabei nur Bahnhöfe, bei denen dieser Wert mindestens vier ist:

BAHNHOF	ZUG_ANKUENFTE
Goslar	4

Aufgabe 2 (CREATE TABLE, INSERT) 5 Punkte

Schreiben Sie eine `CREATE TABLE` Anweisung für die Tabelle `FAHRPLAN`. Schreiben Sie außerdem eine `INSERT`-Anweisung, die das Beispiel-Tupel einfügt.

FAHRPLAN			
<u>ZNR</u>	<u>BAHNHOF</u>	ANKUNFT^o	ABFAHRT^o
RB 25786	Braunschweig	(null)	6:08

- `ZNR` ist eine Zeichenkette der festen Länge 10. `ZNR` verweist auf die Tabelle `ZUEGE`.
- `BAHNHOF` ist eine Zeichenkette variabler Länge (maximal 30 Zeichen).
- `ANKUNFT` und `ABFAHRT` haben den fiktiven Datentyp `TIME`. Diese beiden Attribute können null sein.
- Formulieren Sie außerdem eine `CHECK`-Bedingung, um sicherzustellen, das `ANKUNFT` vor `ABFAHRT` liegt (benutzen Sie `<`). Für unnötige Komplikationen können Sie einen halben Punkt verlieren.
- Geben Sie der `CHECK`-Bedingung einen Namen. Die anderen Integritätsbedingungen brauchen Sie nicht zu benennen.

Aufgabe 3 (ER-Modell) 6 Punkte

Zeichnen Sie ein ER-Diagramm für eine Variante der CD-Datenbank, die Sie aus den Hausaufgaben kennen (etwas vereinfacht). Sie sollen folgende Daten speichern:

- Für Komponisten eine eindeutige Nummer, den Namen, und den Vornamen.
- Für Stücke eine eindeutige Nummer, den Titel, und optinal die Tonart.
- Welcher Komponist hat welches Stückt geschrieben? Jedes Stück stammt von maximal einem Komponisten. Es kann Stücke geben, deren Komponist nicht bekannt ist.
- Für jede CD eine eindeutige Nummer, einen Titel, und die Gesamtspielzeit in Minuten. Die Gesamtspielzeit ist eventuell nicht bekannt.
- Welches Stück ist auf welcher CD? Stücke können auf mehreren CDs enthalten sein. Es ist auch möglich, daß es noch gar keine Aufnahme eines Stückes in der Datenbank gibt. Jede CD muß mindestens ein Stück enthalten, kann aber beliebig viele enthalten.

Geben Sie Schlüssel und Kardinalitäten an und markieren Sie optionale Attribute.

Aufgabe 4 (BCNF) 4 Punkte

Kreuzen Sie die jeweils richtige bzw. beste Antwort an. Machen Sie pro Unteraufgabe nur ein Kreuz. Pro Unteraufgabe gibt es einen Punkt. Falls Sie die Lösung nicht wissen, raten Sie (wenn Sie nichts ankreuzen, haben Sie den Punkt auf jeden Fall verloren).

Die Bahn hat sich entschlossen, über Verspätungen Buch zu führen. Sie hat dazu folgende Tabelle angelegt:

FAHRTEN						
ZNR	DATUM	BAHNHOF	AN_PLAN	AN_TAT	AB_PLAN	AB_TAT
RB 25786	27.01.2003	Braunschweig	(null)	(null)	6:08	6:10
RB 25786	27.01.2003	Wolfenbüttel	6:17	6:19	6:18	6:20
RB 25786	27.01.2003	Goslar	7:18	7:23	(null)	(null)
RB 25742	27.01.2003	Braunschweig	(null)	(null)	7:13	7:13
RB 25742	27.01.2003	Wolfenbüttel	7:22	7:25	7:23	7:26
RB 25742	27.01.2003	Goslar	8:01	8:12	(null)	(null)
RB 25786	28.01.2003	Braunschweig	(null)	(null)	6:08	6:08
RB 25786	28.01.2003	Wolfenbüttel	6:17	6:17	6:18	6:18
RB 25786	28.01.2003	Goslar	7:18	7:19	(null)	(null)

`AN_PLAN` ist die geplante Ankunftszeit, und `AN_TAT` die tatsächliche Ankunftszeit. Entsprechendes gilt für die Abfahrtszeiten `AB_PLAN` und `AB_TAT`.

Gegeben seien folgende funktionale Abhängigkeiten:

- `ZNR`, `BAHNHOF` → `AN_PLAN`, `AB_PLAN`
- `ZNR`, `DATUM`, `BAHNHOF` → `AN_TAT`, `AB_TAT`
- `ZNR`, `AN_PLAN` → `BAHNHOF`
- `ZNR`, `AB_PLAN` → `BAHNHOF`

a) Implizieren die gegebenen vier funktionalen Abhängigkeiten folgende FA?

`ZNR`, `AN_PLAN` → `AB_PLAN`

Kreuzen Sie die beste Möglichkeit an:

☐ Ja.

☐ Ja, diese funktionale Abhängigkeit ist sogar trivial.

☐ Nein.

b) Ein Entwickler schlägt vor, man sollte noch zusätzliche die funktionale Abhängigkeit

`BAHNHOF, AB_PLAN` $\rightarrow$ `ZUGNR`

fordern. Was meinen Sie dazu?

- ☐ Diese funktionale Abhängigkeit ist schon von den gegebenen FAen impliziert. Man braucht sie daher nicht zusätzlich zu fordern.
- ☐ Im Prinzip wäre es durchaus sinnvoll. Es könnte aber bei größeren Bahnhöfen eventuell verletzt werden. Deswegen kann man sie nicht fordern.
- ☐ Diese funktionale Abhängigkeit macht keinen Sinn. Sie ist auch schon in den Beispieldaten verletzt.

c) Was wäre/wären der/die Schlüssel für die Tabelle `FAHRTEN` bei den gegebenen vier funktionalen Abhängigkeiten, d.h. ohne die FA aus a)? Genau eine Möglichkeit ist richtig.

- ☐ `ZNR`, `DATUM`
- ☐ `ZNR`, `BAHNHOF`
- ☐ `ZNR`, `DATUM`, `BAHNHOF`
- ☐ `ZNR`, `DATUM`, `BAHNHOF` und `ZNR`, `DATUM`, `AN_PLAN` und `ZNR`, `DATUM`, `AB_PLAN`.
- ☐ `ZNR`, `DATUM`, `BAHNHOF` und `ZNR`, `AN_PLAN` und `ZNR`, `AB_PLAN`.
- ☐ `DATUM`, `AN_PLAN`.

d) Ist die Tabelle in BCNF?

- ☐ Ja.
- ☐ Nein. Zum Beispiel verletzt `ZNR`, `BAHNHOF` $\rightarrow$ `AN_PLAN`, `AB_PLAN` die BCNF-Bedingung.
- ☐ Nein. Zum Beispiel verletzt `ZNR`, `DATUM`, `BAHNHOF` $\rightarrow$ `AN_TAT`, `AB_TAT` die BCNF-Bedingung.