

## Finalklausur zur Vorlesung Datenbanksysteme I – Wintersemester 2003/2004

Prüfer: Prof. R. Bayer, Ph.D.

Datum: 13.02.2004

Zeit: 16.15 Uhr

### Hinweis:

Die Bearbeitungszeit beträgt 90 Minuten. Bitte benutzen Sie die ausgeteilten Blätter zur Abgabe Ihrer Lösung. Bitte schreiben Sie leserlich, kennzeichnen Sie eindeutig Ihre Lösungsvorschläge und streichen Sie Irrwege oder Zwischenrechnungen durch. Die Klausur umfasst **3 Seiten** mit insgesamt **5 Aufgaben**. Die erreichbare Höchstpunktzahl ist 60 Punkte.

**Falls Sie Ihre Note über die Webpage erfahren möchten, kreuzen Sie bitte die Erklärung auf dem Umschlag an. Ansonsten erfahren Sie Ihre Note über den entsprechenden Ausgang.**

### Aufgabe 1: DB-Anwendung (10 Punkte)

Sie befinden sich in einem großen Möbelhaus. Sie möchten einen Tisch kaufen. In der Ausstellung des Möbelhauses suchen Sie den gewünschten Tisch aus. Auf das Preisschild des Tisches ist auch die Nummer des Regals gedruckt, in dem Sie den Tisch im Abholmarkt finden können. Sie notieren sich die Regalnummer, um den Tisch im Abholmarkt mitnehmen zu können. Im Abholmarkt stellen Sie fest, dass die Regalnummer des Tisches nicht existiert.

1.1 Welchen Datenbank-Fachbegriffen entsprechen **Tisch** und **Regalnummer**? (2 Punkte)

Tisch ist Teil des Schlüssels in der Ausstellung. (Alternative Lösung: Tisch bezeichnet generell Tupel)

Regalnummer ist Schlüssel im Abholmarkt und Fremdschlüssel in der Ausstellung.

1.2 Welche Probleme traten im Abholmarkt auf?

Warum traten die Probleme im Abholmarkt auf? Nennen Sie jeweils die Datenbank-Fachbegriffe! (4 Punkte)

Die Fremdschlüsselbeziehungen zwischen Ausstellung und Abholmarkt wurden nicht berücksichtigt (Referentielle Integrität verletzt, dangling reference).

Das Tupel mit dem Schlüssel Regalnummer wurde im Abholmarkt gelöscht oder der Schlüssel Regalnummer geändert. Die Löschung, bzw. Änderung wurde aber nicht in die Ausstellung propagiert.

1.3 Wie würden Sie die oben genannten Probleme in einer Datenbank lösen (formulieren Sie hierzu für das Szenario ein einfaches Schema in SQL)? (4 Punkte)

```
CREATE TABLE Ausstellung (  
bezeichnung VARCHAR(5) NOT NULL PRIMARY KEY,  
regalnummer INTEGER FOREIGN KEY REFERENCES Abholmarkt ON DELETE SET  
NULL ON UPDATE CASCADE )
```

```
CREATE TABLE Abholmarkt (  
regalnummer INTEGER NOT NULL PRIMARY KEY,  
bezeichnung varchar(5))
```

mögliche Alternativen: ON DELETE CASCADE, ON DELETE NO ACTION, ON UPDATE NO ACTION

## Aufgabe 2: Schemamodifikation (4 Punkte)

Gegeben sei folgendes Datenbankschema: T (a: integer, b:varchar(10), c:char(1))

Formulieren Sie einen oder mehrere SQL-Befehle mit denen Sie die Tabelle von T nach E umbenennen und das Attribut c entfernen können.

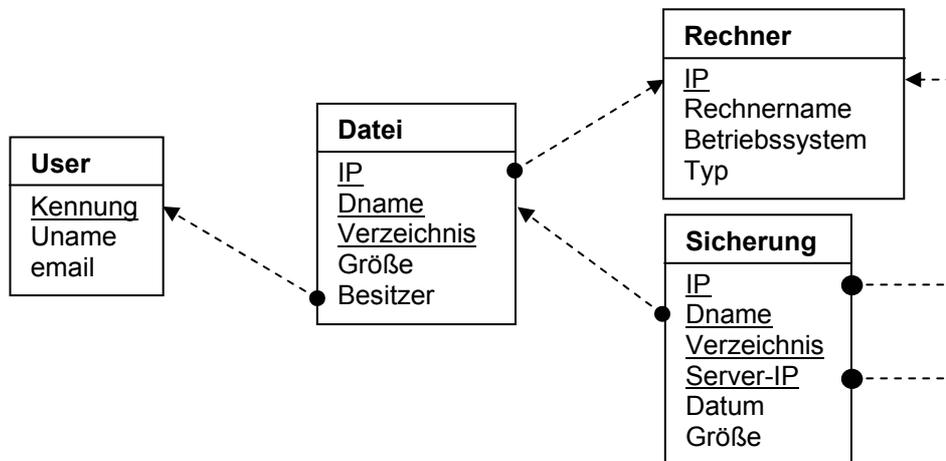
Man erzeugt eine neue Tabelle E mit den gewünschten Attributen und fügt die Tupel aus T mit einer Projektion ohne c in E ein.

```
CREATE TABLE E (  
a: INTEGER NOT NULL PRIMARY KEY,  
b: VARCHAR(10)  
);  
INSERT INTO E SELECT a,b from T;  
DROP TABLE T;
```

Natürlich kann man die Spalte c auch mit DROP COLUMN (wurde aber nicht besprochen) entfernen, umbenennen geht auch mittels RENAME TABLE T TO E (ebenfalls nicht besprochen)

### Aufgabe 3: SQL (16 Punkte)

Gegeben ist folgendes Relationenschema einer Datenbank, die Dateien eines Rechners und ihre Sicherungen auf einem Server verwaltet:



**User** (Kennung, Uname, email)

**Rechner** (IP, Rechnername, Betriebssystem, Typ) mit  $Typ \in \{ 'AR', 'Server' \}$   
AR=Arbeitsplatzrechner

**Datei** (IP, Dname, Verzeichnis, Größe, Besitzer)

**Sicherung** (IP, Dname, Verzeichnis, Server-IP, Datum, Größe)

- Benutzer, identifiziert durch ihre Kennung, besitzen Dateien (Fremdschlüssel Datei:Besitzer -> User:Kennung ).
- Dateien werden auf einem Arbeitsplatzrechner (IP) in einem Verzeichnis unter einem Dateinamen gespeichert. Ihre Größe wird in MB angegeben.
- Dateien werden auf Servern (Server-IP) gesichert, wobei auf einem Server immer nur eine Sicherung einer Datei liegt.

Formulieren Sie folgende Anfragen in SQL:

3.1. User, die am 13.2.04 Dateien auf Windows-Servern gesichert haben. (3 Punkte)

Ergebnis: Username (Uname) und email

```
SELECT DISTINCT u.Uname, u.email
FROM User u, Datei d, Sicherung s, Rechner r
WHERE s.Datum = 13.2.04 AND
      s.Server-IP = r.IP AND
      r.Betriebssystem = 'Windows' AND
      s.IP = d.IP AND
      s.Dname = d.Dname AND
      s.Verzeichnis = d.Verzeichnis AND
      d.Besitzer = u.Kennung
```

DISTINCT kann auch durch ein GROUP BY u.Uname, u.email ersetzt werden.

Alternativ mit Unteranfrage, die alle Windows-Server ermittelt:

```
SELECT DISTINCT u.Uname, u.email
FROM User u, Datei d, Sicherung s
WHERE s.Datum = 13.2.04 AND
      s.Server-IP IN
      (SELECT IP FROM Rechner r
       WHERE r.Typ = 'Server' AND r.Betriebssystem = 'Windows') AND
      s.IP = d.IP AND
      s.Dname = d.Dname AND
      s.Verzeichnis = d.Verzeichnis AND
      d.Besitzer = u.Kennung
```

3.2. Dateien, die noch nie gesichert wurden. (4 Punkte)

Ergebnis: Dname, Verzeichnis, IP

```
SELECT Dname, Verzeichnis, IP
FROM Datei d
WHERE NOT EXISTS
      (SELECT * FROM Sicherung s
       WHERE d.IP = s.IP AND d.Dname = s.Dname AND
            d.Verzeichnis = s.Verzeichnis)
```

Alternativ mit EXCEPT:

```
(SELECT Dname, Verzeichnis, IP FROM Datei d)
EXCEPT
(SELECT Dname, Verzeichnis, IP FROM Sicherung s)
```

Alternativ mit NOT IN:

```
SELECT Dname, Verzeichnis, IP FROM Datei s
WHERE (Dname, Verzeichnis, IP) NOT IN
      (SELECT (Dname, Verzeichnis, IP) FROM Sicherung s)
```

3.3. Belegter Plattenspeicher pro Arbeitsplatzrechner. (4 Punkte)

Ergebnis: Rechnername und Dateivolumen

```
SELECT r.Rechnername, tmp.Volumen
FROM Rechner r
      (SELECT IP, sum (Größe ) as Volumen FROM Datei GROUP BY IP)
WHERE r.IP = tmp.IP
```

Die folgende Lösung ist nicht korrekt, da Rechnername nach dem Schema nicht eindeutig sein muß (es gab aber trotzdem Punkte auf Teile der Lösung). Das GROUP BY kann nicht durch GROUP BY r.IP ersetzt werden, wegen der Einschränkungen der Verwendung im SELECT.

```
SELECT r.Rechnername, sum(Größe)
FROM Datei d, Rechner r WHERE d.IP=r.IP
GROUP BY r.Rechnername
```

3.4. Benutzer, die mehr als 500 MB Dateivolumen auf einem Arbeitsplatzrechner haben.

Ergebnis: Kennung und IP (5 Punkte)

```
SELECT d.Besitzer, d.IP
FROM Datei d
GROUP BY d.IP, d.Besitzer
HAVING sum(Größe) > 500
```

Die Query funktioniert, weil in Datei nur Dateien von Arbeitsplatzrechnern vermerkt sind (ein Join mit Rechner und eine Selektion auf AR ist also nicht nötig, aber auch nicht falsch).

#### Aufgabe 4: Normalformen (24 Punkte)

Gegeben sind die beiden separaten Mengen F1 und F2 funktionaler Abhängigkeiten:

F1: Autor $\rightarrow$ Verlagszugehörigkeit Autor, Verlagszugehörigkeit $\rightarrow$ Übersetzer Titel $\rightarrow$ Autor, Übersetzer Titel $\rightarrow$ Verlag	F2: Autor $\rightarrow$ Verlagszugehörigkeit, Übersetzer Titel $\rightarrow$ Autor, Verlag
---	---

4.1. Bestimmen Sie die Attributhüllen  $AttrHülle(F1, Titel)$  und  $AttrHülle(F1, Autor)$ . (2 Punkte)

$AttrHülle(F1, Titel) = (Titel, Verlag, Autor, Übersetzer, Verlagszugehörigkeit)$   
 $AttrHülle(F1, Autor) = (Autor, Verlagszugehörigkeit, Übersetzer)$

4.2. Prüfen Sie anhand der Kanonischen Überdeckung, ob F1 und F2 äquivalent sind. (5 Punkte)

Linksreduktion (nur Regeln betrachten, die auf der linken Seite mehrere Attribute haben):

Betrachtete Regel: Autor, Verlagszugehörigkeit  $\rightarrow$  Übersetzer

Betrachtetes Attribut Autor:  $\{Verlagszugehörigkeit\}^+ = (Verlagszugehörigkeit)$

Betrachtetes Attribut *Verlagszugehörigkeit*:

$\{Autor\}^+ = (Autor, Verlagszugehörigkeit, Übersetzer)$

Damit kann die Abhängigkeit reduziert werden zu: Autor  $\rightarrow$  Übersetzer

Rechtsreduktion:

Betrachtete Regel: Titel  $\rightarrow$  Autor, Übersetzer

Betrachtetes Attribut Übersetzer:

$AttrHülle(F1, Titel) = (Titel, Verlag, Autor, Übersetzer, Verlagszugehörigkeit)$

Damit kann die Abhängigkeit reduziert werden zu: Titel  $\rightarrow$  Autor

Nach Zusammenfassung ergibt sich folgende Regelmenge:

F1-neu:

Autor  $\rightarrow$  Verlagszugehörigkeit, Übersetzer

Titel  $\rightarrow$  Autor, Verlag

Damit: F1  $\equiv$  F2

4.3. Gegeben ist die Relation R und die Funktionalen Abhängigkeiten F2 (s.o.) mit  
 R (Titel, Autor, Verlag, Verlagszugehörigkeit, Übersetzer)  
 Bestimmen Sie die Schlüsselkandidaten für R anhand von F2. (2 Punkte)

Schlüsselkandidat für R bezüglich F2 ist Titel  
 (Titel → Verlag, Autor → Verlagszugehörigkeit, Übersetzer)

4.4. Gegeben ist wieder R und F2  
 Begründen Sie, welche Normalformen R erfüllt und welche nicht (NF1, NF2, NF3, BCNF).  
 Verwenden Sie dazu die Definitionen der Vorlesung. (4 Punkte)

1NF: Alle Attribute atomar  
 2NF: Alle Nichtschlüssel-Attribute sind voll funktional abhängig vom Primärschlüssel  
 (der gleichzeitig einziger Schlüsselkandidat ist)  
 3NF: nicht erfüllt, da es transitive Abhängigkeiten gibt, z.B. Titel → Autor → Übersetzer  
 BCNF nicht erfüllt, da 3NF nicht erfüllt.

4.5. Normalisieren Sie R bezüglich der dritten Normalform. (3 Punkte)

R1 (Autor, Übersetzer, Verlagszugehörigkeit)  
 R2 (Titel, Autor, Verlag)  
 R2 enthält Schlüsselkandidat, damit ist Zerlegung abgeschlossen.

4.6. Gegeben ist wieder R mit modifizierten funktionalen Abhängigkeiten F3:

R	(Titel, Autor, Verlag, Verlagszugehörigkeit, Übersetzer)
F3:	Autor → Verlagszugehörigkeit, Übersetzer Titel → Verlag

Wie lauten nun die Schlüsselkandidaten von R? (2 Punkte)  
 Welche Normalformen erfüllt R und welche nicht? (2 Punkte)

Einziger Schlüsselkandidat ist (Titel, Autor)  
 1NF: Alle Attribute atomar  
 2NF: nicht erfüllt, da nicht alle Nichtschlüssel-Attribute vom Primärschlüssel voll funktional abhängig sind  
 Bsp: Verlag nur von Titel nicht aber von Autor abhängig  
 3NF: nicht erfüllt, da nicht 2NF  
 BCNF nicht erfüllt, da 3NF nicht erfüllt.

4.7. Normalisieren Sie R bezogen auf F3 vollständig.  
 Wie wird die Verlustfreiheit der Zerlegung von R garantiert? (4 Punkte)

R1 (Autor, Übersetzer, Verlagszugehörigkeit)  
 R2 (Titel, Verlag)  
 R3 (Titel, Autor)  
 Verlustfreiheit der Zerlegung wird durch Relation R3 garantiert, die die Beziehung zwischen Titel und Autor herstellt.

### Aufgabe 5: B-Bäume (6 Punkte)

Erzeugen Sie einen B-Baum der Ordnung  $k=1$  durch schrittweises Einfügen der folgenden Sequenz  $S$  von ganzen Zahlen:  $S = \{10, 15, 20, 80, 65, 31, 53, 11, 12\}$

