Eidgenössische
Technische Hochschule
Zürich

Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Swiss Federal Institute of Technology Zurich

Prof. H.-J. Schek — *A. Feßler, U. Röhm, H. Schuldt, R. Weber*

Informationssysteme IS-G SS 1997

Übung 7 — Abgabe: 27.5.1997

Aufgabe 1: Views

Gegeben sei die folgende View–Definition:

```
CREATE VIEW Lazy_Student AS
       SELECT *
       FROM   Student S
       WHERE  Semester ≥ 4
       AND    NOT EXISTS ( SELECT *
                           FROM   Prüfung P
                           WHERE  S.ETH_Nr = P.ETH_Nr)
```

a) Formulieren Sie die folgende Anfrage so um, dass Sie obige View nicht mehr verwenden.

```
SELECT S.ETH_Nr, S.S_Name, P.P_Name, P.Tel
FROM   Professor P, Abteilung A, Lazy_Student S
WHERE  S.Abt_Nr = A.Abt_Nr
AND    A.Vorsteher = P.P_Name
```

b) Sie merken, dass der Name des Studenten *Hans Maier*, der als Student im 6. Semester noch immer keine Prüfung absolviert hat, mit y statt mit i geschrieben wurde. Formulieren Sie die Änderungsoperation in SQL auf der View. Kann das System diese Operation durchführen? Begründen Sie Ihre Antwort und geben Sie gegebenenfalls Alternativen an.

c) Hans Mayer hat nun endlich seine erste Prüfung absolviert. Formulieren Sie die Änderungsoperation in SQL auf der View. Kann das System diese Operation durchführen? Begründen Sie Ihre Antwort und geben Sie gegebenenfalls Alternativen an. Probieren Sie die Änderungsoperation auf Oracle V7 aus. Verhält sich Oracle wie von Ihnen postuliert?

Aufgabe 2: Zugriffsrechte

Betrachten Sie drei Benutzer S_1, S_2 und S_3, wobei S_1 ein Recht besitzt, das er oder sie weitergeben darf. Diskutieren Sie die folgenden Autorisierungsabläufe:

a)
1. S_2 erhält das Recht von S_1 und gibt es an S_3 weiter.
2. S_3 erhält das Recht von S_1.
3. S_2 entzieht S_3 das Recht.

b)
1. S_1 gibt das Recht an S_2 und S_3 weiter.
2. S_3 erhält das Recht von S_2.
3. S_2 erhält das Recht von S_3.
4. S_1 entzieht S_2 und S_3 das Recht.

Geben Sie Algorithmen für die Rechtevergabe an, die die obigen Probleme berücksichtigen.

Aufgabe 3: Zugriffsstrukturen

Wir betrachten einen Teil unserer Musikdatenbank mit der folgenden Schemadefinition:

```
CREATE TABLE Disk
  (DiskID        NUMBER PRIMARY KEY NOT NULL,
   DiskTitel     VARCHAR(255));

CREATE TABLE Musikstueck
  (DiskID     NUMBER,
   StueckID   NUMBER,
   Titel      VARCHAR(255));

CREATE TABLE Person
  (PID           NUMBER PRIMARY KEY NOT NULL,
   Name          VARCHAR(40) NOT NULL,
   Nationalitaet CHAR(5));

CREATE TABLE Musikstueck_Autor
  (PID         NUMBER NOT NULL REFERENCES Person,
   DiskID      NUMBER NOT NULL,
   StueckID    NUMBER NOT NULL,
   Taetigkeit  VARCHAR(20) NOT NULL,
   FOREIGN KEY (DiskID, StueckID) REFERENCES Musikstueck);
```

Als Kardinalitäten und mittlere Tupellängen der verschiedenen Relationen nehmen wir an:

Disk	1000	Tupel à 100 Bytes
Musikstueck	10000	Tupel à 50 Bytes
Person	6000	Tupel à 40 Bytes
Musikstueck_Autor	30000	Tupel à 40 Bytes

Zur Effizienzsteigerung von Anfragen sind unter anderem die folgenden mittelbaren Indexe angelegt, die jeweils als B*-Baum implementiert sind:

```
CREATE UNIQUE INDEX DiskID ON Disk(DiskID);
CREATE INDEX DiskTitel ON Disk(DiskTitel);
CREATE INDEX AutorStueck ON Musikstueck_Autor (PID);
```

Nehmen Sie an, dass jede Relation und jeder Index jeweils einen eigenen Tablespace belegt, wobei die Seitengrösse 4 KBytes sei und der Seiten-Header die Länge 96 Bytes habe. Datenseiten seien zu 90 Prozent gefüllt, Indexseiten zu 70 Prozent. Der Datentyp NUMBER belege 10 Bytes, der Disktitel sei Schlüsselkandidat der Relation `Disk` und habe eine durchschnittliche Länge von 90 Bytes, ein TID oder Zeiger im B*-Baum habe die Länge 6 Bytes. Nehmen Sie zusätzlich an, dass alle Werte eines Fremdschlüsselattributes jeweils gleichhäufig vorkommen (Gleichverteilungsannahme).

Analysieren Sie die Speicherplatz- und Indexzugriffskosten dieser Datenbank. Dabei können Sie den Platzbedarf für die Slotlisten und für die Längenfelder des Datentyps VARCHAR vernachlässigen.

Aufgabe 4: Pufferverwaltung

Überlegen Sie sich, welche Informationen ein Pufferverwalter für das Bereitstellen (*fix*) und Freigeben (*unfix*) von Seiten benötigt. Unter der Annahme, dass der Datenbankpuffer beliebig gross sei, wollen wir das folgende Szenario betrachten:

Eine Datenbank habe sechs Seiten ($a \ldots f$). Drei parallele Prozesse arbeiten auf diesen Seiten und rufen den Pufferverwalter mit einer Bereitstellungsoperation (B_a = Bereitstellen der Seite a) oder einer Freigabeoperation (F_a = Freigeben der Seite a) auf. Die Folge der Operationen ist in zeitlicher Reihenfolge gegeben durch:

P_1:	B_a		B_b		F_a	F_b			B_e			
P_2:		B_b						B_e		B_b	†	
P_3:				B_c			B_f				B_e	

Zeitpunkt:	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}	T_{11}	T_{12}

a) Welche Seiten befinden sich zum Zeitpunkt T_7 im Puffer, ohne dass sie aktuell für einen Prozess bereitgestellt sind? Welche Information wird mindestens benötigt, um dies zu jedem Zeitpunkt feststellen zu können?

b) Zum Zeitpunkt T_{12} bricht der Prozess P_2 ab, ohne „seine“ Seiten freigegeben zu haben. Welche Informationen werden benötigt, um den Puffer wieder in einen „ordentlichen“ Zustand zu überführen?