UNIVERSITÄT DES SAARLANDES
Fachbereich 14 – Informatik
Lehrstuhl Prof. Weikum

Datenbanksysteme WS 1999/2000

Übung 3 — Beispiellösung

Aufgabe 1: SQL - Universitätsdatenbank

Gegeben sei das aus den letzten Übungen bekannte Schema einer Universitätsdatenbank (siehe Übungsblatt):

Formulieren Sie die folgenden Anfragen in SQL:

a) An welchen Hausmeister muß sich Professor Weikum wenden, wenn er seinen Zimmerschlüssel vergessen hat?

```
SELECT Hausmeister
FROM   Gebaeude, Professor
WHERE  P_Name = 'Weikum'
AND    Professor.Gebaeude = Gebaeude.Gebaeude
```

b) Welche Studenten (Matrikel_Nr) haben eine Prüfung beim augenblicklichen Prodekan ihres Fachbereichs abgelegt?

```
SELECT DISTINCT Student.Matrikel_Nr
FROM   Pruefung, Fachbereich, Student
WHERE  Pruefer = Prodekan
AND    Student.Matrikel_Nr = Pruefung.Matrikel_Nr
AND    Student.Fachbereich_Nr = Fachbereich.Fachbereich_Nr
```

c) Wo (Gebäude, Raum) fand die Prüfung von Hugo Meier im Fach Betriebssysteme statt? (Annahme: Professoren prüfen in ihren Büros)

```
SELECT Gebaeude, Raum
FROM   Pruefung, Student, Professor
WHERE  S_name = 'Hugo Meier'
AND    Student.Matrikel_Nr = Pruefung.Matrikel_Nr
AND    Pruefer = P_Name
AND    Fach = 'Betriebssysteme'
```

d) Welche Studenten (Matrikel_Nr) mit mindestens 4 Semestern haben noch keine Prüfung abgelegt?

```
SELECT  Matrikel_Nr
FROM    Student
WHERE   Semester > 3
AND     NOT EXISTS
   (SELECT *
            FROM Pruefung
            WHERE Student.Matrikel_Nr = Pruefung.Matrikel_Nr)
```

Alternativ mit <u>NOT IN</u>:

```
SELECT  Matrikel_Nr
FROM    Student
WHERE   Semster > 3
AND     Matrikel_Nr NOT IN
           (SELECT DISTINCT Matrikel_Nr
            FROM Pruefung)
```

Alternativ mit <u>MINUS</u>:

```
SELECT  Matrikel_Nr
FROM    Student
WHERE   Semster > 3
MINUS
SELECT  DISTINCT Matrikel_Nr
FROM    Pruefung
```

e) Welche Studenten (Matrikel_Nr) haben ausschließlich Prüfungen bei Professoren ihres Fachbereichs abgelegt?

```
SELECT Matrikel_Nr
FROM   Student
WHERE  NOT EXISTS
  (SELECT *
           FROM   Pruefung, Professor
           WHERE  Student.Matrikel_Nr = Pruefung.Matrikel_Nr
           AND    Pruefer = P_Name
           AND    Student.Fachbereich_Nr != Professor.Fachbereich_Nr)
```

Dies liefert allerdings auch Student(inn)en, die noch gar keine Prüfung haben. Besser daher:

```
SELECT DISTINCT Matrikel_Nr
FROM Student, Pruefung p1
WHERE Student.Matrikel_Nr =p1.Matrikel_Nr
  AND NOT EXISTS
 (SELECT *
          FROM Pruefung p2, Professor
          WHERE p2.Matrikel_Nr = Student.Matrikel_Nr
            AND Pruefer = P_Name
            AND Professor.Fachbereich_Nr != Student.Fachbereich_Nr)
```

f) Welche Studenten (Matrikel_Nr) haben alle ihre bisher abgelegten Prüfungen mit der Bestnote 1.0 bestanden?

```
SELECT Matrikel_Nr
FROM   Pruefung p1
WHERE  NOT EXISTS
  (SELECT *
           FROM Pruefung p2
           WHERE p1.Matrikel_Nr= p2.Matrikel_Nr
             AND Note > 1.0)
```

Einfacher wird die Lösung mit der Benutzung von MINUS:

```
SELECT  DISTINCT Matrikel_Nr
FROM    Pruefung
MINUS
SELECT  DISTINCT Matrikel_Nr
FROM    Pruefung
WHERE   Note > 1.0
```

Eine noch elegantere Lösung wäre:

```
SELECT Matrikel_Nr
FROM   Pruefung
GROUP  BY Matrikel_Nr
HAVING max(Note) = 1.0
```

g) Bestimmen Sie die Durchschnittsnote für jeden Studenten.

```
SELECT Matrikel_Nr, AVG(Note) FROM Prüfung GROUP BY Matrikel_Nr
```

h) Welche Studenten (Matrikel_Nr) haben ein Prüfungsfach (Fach) besser abgeschlossen als der Durchschnitt in diesem Fach?

```
SELECT DISTINCT Matrikel_Nr FROM Prüfung P WHERE Note > ( SELECT
AVG(Note) FROM Prüfung ps WHERE p.Fach = ps.Fach )
```

Alternativ geht hier auch eine Lösung in zwei Schritten mit einem View.

```
CREATE VIEW AVG_Fach AS SELECT Fach, AVG(Note) AVG_Note FROM Prüfung
GROUP BY Fach
```

```
SELECT DISTINCT Matrikel_Nr FROM AVG_Fach a, Prüfung p WHERE Note
> AVG_Note AND a.Fach = p.Fach
```

Eine vermeintlich einfache Lösung funktioniert leider nicht:

```
SELECT DISTINCT Matrikel_Nr FROM Prüfung GROUP BY Fach HAVING Note
> AVG(Note)
```

Hier kommen mit Matrikel_NR im SELECT und Note im HAVING zwei Nichtgruppeneigenschaften vor. Dies ist in Queries mit GROUP BY nicht erlaubt.

i) Welches Prüfungsfach (Fach) wurde mit der besten Durchschnittsnote abgeschlossen?

```
SELECT Fach FROM Prüfung GROUP BY Fach HAVING AVG(Note) ≥ All (
SELECT AVG(Note) FROM Prüfung GROUP BY Fach )
```

Infolge einer Implementationsbeschränkung von ORACLE V6 ist dieses Statment nicht erlaubt. Mit Hilfe einer View kann das Problem umgegangen werden:

```
CREATE VIEW AVG_Fach AS SELECT Fach, AVG(Note) AVG_Note FROM Prüfung
GROUP BY Fach
SELECT Fach FROM AVG_Fach WHERE AVG_Note = ( SELECT MAX ( AVG_Note
) FROM AVG_Fach );
```

Ein anderer Ansatz, der ohne View auskommt, sieht wie folgt aus:

```
SELECT Fach, AVG(Note) FROM Prüfung GROUP BY Fach Order by 2 DESC
```

Bei dieser Formulierung werden die Prüfungen nach Fächern gruppiert und nach Durchschnittsnote absteigend sortiert ausgegeben. Das erste Resultattupel ist also die gewünschte Antwort. Insofern ist dieser Antwort keine "exakte" Lösung, aber unter pragematischen Gesichtspunkten trotzdem brauchbar.

j) Welcher Student (Name) hat alle Prüfungen als Bester abgeschlossen?

```
SELECT DISTINCT S_Name FROM Student s WHERE NOT EXISTS (SELECT *
FROM Prüfung p1 WHERE s.Matrikel_Nr = p1.Matrikel_Nr AND EXISTS (SELECT
* FROM Prüfung p2 WHERE p1.Fach = p2.Fach AND p1.Note < p2.Note))
```

Aufgabe 2: Anfragen in SQL - Musikdatenbank

Gegeben sei ein gegenüber Übung 1 erweitertetes Schema der Musikdatenbank:

Disk	(DiskID, DiskTitel, **Preis**)
	78462, W. A. Mozart: Klavierkonzerte, 29.99
Musikstück	(DiskID, StückID, Titel, **Länge**)
	78462, 4, Konzert für Klavier und Orchester Nr. 21, 2732
Person	(PID, Name, Nationalität)
	9362, W. A. Mozart, Österreich
Interpret	(PID, DiskID, StückID, Funktion, Instrument)
	15267, 78462, 4, Solist, Klavier
Autor	(PID, DiskID, StückID, Tätigkeit)
	9362, 78462, 4, Komponist

Formulieren Sie die folgenden Anfragen in SQL:

a) Welche Stücke (Titel) hat F. Chopin komponiert?

```
SELECT distinct m.titel
FROM musikstueck m, person p, autor a
WHERE m.diskid=a.diskid AND m.stueckid=a.stueckid AND
      a.funktion='Komponist' AND
      p.pid=a.pid AND p.name LIKE '%Chopin%';
```

b) Welches ist die teuerste Disk und was kostet sie?

```
SELECT disktitel,preis
FROM disk d
WHERE NOT EXISTS(SELECT * FROM disk d2
                 WHERE d2.preis>d.preis);
```

c) Welche Disk enthält das längste Stück unter den Disks, die nicht mehr als 20 (Mark) kosten ?

```
SELECT d.disktitel
FROM disk d, musikstueck m
WHERE d.diskid=m.diskid AND d.preis<=30
      AND NOT EXISTS(SELECT * FROM disk d2, musikstueck m2
                     WHERE d2.diskid=m2.diskid AND d2.preis<=30
                     AND m.laenge<m2.laenge);
```

d) Welche Disks (DiskTitel) enthalten ausschließlich Stücke, die von F. Chopin komponiert wurden?

```
SELECT distinct d.disktitel
FROM disk d, person p, musikstueck m, autor a
WHERE d.diskid=a.diskid AND a.diskid=m.diskid AND
      m.stueckid=a.stueckid AND
      a.pid=p.pid AND a.funktion='Komponist' AND
      p.name LIKE '%Chopin%'
      AND NOT EXISTS
       (SELECT * FROM musikstueck m2, autor a2
        WHERE m2.diskId=m.diskId AND
              a2.diskid=m2.diskid AND
              a2.funktion='Komponist' AND
              a2.pid<>p.pid);
```

e) Welche Disks enthalten kein Stück, das länger ist als 60 Sekunden?

```
SELECT DISTINCT d.diskid, d.disktitel
FROM musikstueck m, disk d
WHERE d.diskid=m.diskid
GROUP BY d.diskid, d.disktitel
HAVING MAX(laenge)<=60;
```

f) Welchen Durchschnittspreis haben Disks, auf denen Interpreten aus über 3 Nationen zu hören sind?

```
SELECT AVG(preis)
FROM disk d, interpret i, person p
WHERE d.diskid=i.diskid AND p.pid=i.pid
GROUP BY i.diskid
HAVING COUNT(DISTINCT nationalitaet)>3;
```

g) Ermitteln Sie für jede Disk die Gesamtlänge der drei längsten Stücke!

```
SELECT d.diskid, d.disktitel, SUM(m.laenge), count(*)
FROM disk d, musikstueck m
WHERE d.diskid=m.diskid AND
      3>(SELECT COUNT(*)
         FROM disk d2, musikstueck m2
         WHERE d2.diskid=d.diskid AND
               m2.diskid=m.diskid AND
               m2.laenge<m.laenge
         )
GROUP BY d.diskid, d.disktitel;
```