Informationssysteme - Übungsblatt 5

Jan Hendrik Dithmar - Matrikelnummer 2031259
Mi, 14-16 Uhr, Andreas Kaster

Aufgabe 5.1

(a)

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

(b)

```
SELECT DISTINCT Student.Matrikel_Nr
FROM Prüfung, Fachrichtung, Student
WHERE Prüfer = Studiendekan
AND Student.Matrikel_Nr = Prüfung.Matrikel_Nr
AND Student.Fachrichtung_Nr = Fachrichtung.Fachrichtung_Nr
```

(c)

```
SELECT Gebäude, Raum
FROM Prüfung, Student, Professor
WHERE S_Name = 'Hugo Meier'
AND Student.Matrikel_Nr = Prüfung.Matrikel_Nr
AND Prüfer = P_Name
AND Fach = 'Betriebssysteme'
```

(d)

```
SELECT Matrikel_Nr
FROM Student
WHERE Semester > 3
MINUS
SELECT DISTINCT Matrikel_Nr
FROM Prüfung
```

(e)

```
SELECT DISTINCT Matrikel_Nr
FROM Student, Prüfung pr1
WHERE Student.Matrikel_Nr = pr1.Matrikel_Nr
AND NOT EXISTS
  (SELECT *
   FROM Prüfung pr2, Professor
   WHERE pr2.Matrikel_Nr = Student.Matrikel_Nr
   AND Prüfer = P_Name
   AND Professor.Fachrichtung_Nr != Student.Fachrichtung_Nr)
```

(f)

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

(g)

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

(h)

```
SELECT DISTINCT Matrikel_Nr
FROM Prüfung pr1
WHERE Note > (SELECT AVG(Note) FROM Prüfung pr2 WHERE pr1.Fach = pr2.Fach)
```

(i)

```
SELECT Fach
FROM Prüfung
GROUP BY Fach
HAVING AVG(Note) >= ALL (SELECT AVG(Note) FROM Prüfung GROUP BY Fach)
```

(j)

```
SELECT DISTINCT S_Name
FROM Student s
WHERE NOT EXISTS
  (SELECT *
   FROM Prüfung pr1
   WHERE s.Matrikel_Nr = pr1.Matrikel_Nr
   AND EXISTS (SELECT *
               FROM Prüfung pr2
               WHERE pr1.Fach = pr2.Fach
               AND pr1.Note < pr2.Note))
```

Aufgabe 2

(a)

```
SELECT DISTINCT m.titel
FROM Musikstück m, Person p, Autor a
WHERE m.DiskID = a.DiskID
AND m.StückID = a.StückID
AND a.Funktion = 'Komponist'
AND p.PID = a.PID
AND p.Name LIKE '%Chopin%'
```

(b)

```
SELECT DiskTitel, Preis
FROM Disk disk1
WHERE NOT EXISTS (SELECT *
                  FROM Disk disk2
                  WHERE disk2.Preis > disk1.Preis)
```

(c)

```
SELECT disk1.DiskTitel
FROM Disk disk1, Musikstück m1
WHERE disk1.DiskID = m1.DiskID
AND disk1.Preis <= 20
AND NOT EXISTS (SELECT *
                FROM Disk disk2, Musikstück m2
                WHERE disk2.DiskID = m2.DiskID
                AND disk2.Preis <= 20
                AND m1.Länge < m2.Länge)
```

(d)

```
SELECT DISTINCT d.DiskTitel
FROM Disk d, Person p, Musikstück m1, Autor a1
WHERE d.DiskID = a1.DiskID
AND a1.DiskID = m1.DiskID
AND m1.StückID = a1.StückID
AND a1.PID = p.PID
AND a1.Funktion = 'Komponist'
AND p.Name LIKE '%Chopin%'
AND NOT EXISTS (SELECT *
                FROM Musikstück m2, Autor a2
                WHERE m2.DiskID = m1.DiskID
                AND a2.DiskID = m2.DiskID
                AND a2.Funktion = 'Komponist'
                AND a2.PID != p.PID)
```

(e)

```
SELECT DISTINCT d.DiskID, d.DiskTitel
FROM Musikstück m, Disk d
WHERE d.DiskID = m.DiskID
GROUP BY d.DiskID, d.DiskTitel
HAVING MAX(Länge) <= 60
```

(f)

```
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 Nationalität) > 3
```

(g)

```
SELECT d1.DiskID, d1.DiskTitel, SUM(m1.Länge), COUNT(*)
FROM Disk d1, Musikstück m1
WHERE d1.DiskID = m1.DiskID
AND 3 > (SELECT COUNT(*)
         FROM Disk d2, Musikstück m2
         WHERE d2.DiskID = d1.DiskID
         AND m2.DiskID = m1.DiskID
         AND m2.Länge < m1.Länge)
GROUP BY d1.DiskID, d1.DiskTitel
```