Consider the following relations:

Student(snum: integer, sname: string, major: string, level: string, age: integer)
Class(name: string, meets at: string, room: string, fid: integer)
Enrolled(snum: integer, cname: string)
Faculty(fid: integer, fname: string, deptid: integer)

The meaning of these relations is straightforward; for example, Enrolled has one record per student-class pair such that the student is enrolled in the class.
Write the following queries in SQL. No duplicates should be printed in any of the answers.

1. Find the names of all Juniors (level = JR) who are enrolled in a class taught by I. Teach.

2. Find the age of the oldest student who is either a History major or enrolled in a course taught by I. Teach.

3. Find the names of all classes that either meet in room R128 or have five or more students enrolled.

4. Find the names of all students who are enrolled in two classes that meet at the same time.

5. Find the names of faculty members who teach in every room in which some class is taught.

6. Find the names of faculty members for whom the combined enrollment of the courses that they teach is less than five.

7. For each level, print the level and the average age of students for that level.

8. For all levels except JR, print the level and the average age of students for that level.

9. For each faculty member that has taught classes only in room R128, print the faculty member’s name and the total number of classes she or he has taught.

10. Find the names of students enrolled in the maximum number of classes.

11. Find the names of students not enrolled in any class.

12. For each age value that appears in Students, find the level value that appears most often. For example, if there are more FR level students aged 18 than SR, JR, or SO students aged 18, you should print the pair (18, FR).

The answers are given below:

```
1. SELECT DISTINCT S.Sname
FROM Student S, Class C, Enrolled E, Faculty F
```

```
WHERE S.snum = E.snum AND E.cname = C.name AND C.fid = F.fid AND
F.fname = ‘I.Teach’ AND S.level = ‘JR’

2. SELECT MAX(S.age)
FROM Student S
WHERE (S.major = ‘History’)
OR S.snum IN (SELECT E.snum
FROM Class C, Enrolled E, Faculty F
WHERE E.cname = C.name AND C.fid = F.fid
AND F.fname = ‘I.Teach’ )

3. SELECT C.name
FROM Class C
WHERE C.room = ‘R128’
OR C.name IN (SELECT E.cname
FROM Enrolled E
GROUP BY E.cname
HAVING COUNT (*) >= 5)
```

SQL: Queries, Constraints, Triggers 47

```
4. SELECT DISTINCT S.sname
FROM Student S
WHERE S.snum IN (SELECT E1.snum
FROM Enrolled E1, Enrolled E2, Class C1, Class C2
WHERE E1.snum = E2.snum AND E1.cname <> E2.cname
AND E1.cname = C1.name
AND E2.cname = C2.name AND C1.meets at = C2.meets at)

5. SELECT DISTINCT F.fname
FROM Faculty F
WHERE NOT EXISTS (( SELECT *
FROM Class C )
EXCEPT
(SELECTC1.room
FROM Class C1
WHERE C1.fid = F.fid ))

6. SELECT DISTINCT F.fname
FROM Faculty F
WHERE 5 > (SELECT COUNT (E.snum)
FROM Class C, Enrolled E
WHERE C.name = E.cname
AND C.fid = F.fid)

7. SELECT S.level, AVG(S.age)
FROM Student S
GROUP BY S.level

8. SELECT S.level, AVG(S.age)
FROM Student S
```

```
WHERE S.level <> ‘JR’
GROUP BY S.level

9. SELECT F.fname, COUNT(*) AS CourseCount
FROM Faculty F, Class C
WHERE F.fid = C.fid
GROUP BY F.fid, F.fname
HAVING EVERY ( C.room = ‘R128’ )

10. SELECT DISTINCT S.sname
FROM Student S
WHERE S.snum IN (SELECT E.snum
FROM Enrolled E
GROUP BY E.snum
48 Chapter 5
HAVING COUNT (*) >= ALL (SELECT COUNT (*)
FROM Enrolled E2
GROUP BY E2.snum ))

11. SELECT DISTINCT S.sname
FROM Student S
WHERE S.snum NOT IN (SELECT E.snum
FROM Enrolled E )

12. SELECT S.age, S.level
FROM Student S
GROUP BY S.age, S.level,
HAVING S.level IN (SELECT S1.level
FROM Student S1
WHERE S1.age = S.age
GROUP BY S1.level, S1.age
HAVING COUNT (*) >= ALL (SELECT COUNT (*)
FROM Student S2
WHERE s1.age = S2.age
GROUP BY S2.level, S2.age))
```