SQL Examples

Murat Kantarcioglu

Exercise 5.1

Exercise 5.1 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.

Create Tables

```
create table student(
    snum numeric(9,0) primary key,
    sname varchar(30),
    major varchar(25),
    standing varchar(2),
    age numeric(3,0)
);
create table faculty(
    fid numeric(9,0) primary key,
    fname varchar(30),
    deptid numeric(2,0)
);
```

```
create table class(
      name varchar(40) primary key,
      meets_at varchar(20),
      room varchar(10),
      fid numeric(9,0),
      foreign key (fid) references faculty(fid)
);

create table enrolled(
      snum numeric(9,0),
      cname varchar(40),
      primary key(snum,cname),
      foreign key (snum) references
            student(snum),
      foreign key (cname) references
           class(name)
 );
```

Exercise 5.1

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)

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

```
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)
```

Exercise 5.1

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)

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

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

Exercise 5.1

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)

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

```
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)
```

Exercise 5.1

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)

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.

```
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' )
```

Exercise 5.1

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)

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

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

Exercise 5.1

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)

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).

Exercise 5.1

```
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))
```