CS391 Assignment 5

Chris-Emio (chrisr98@bu.edu)

May 5, 2020

Exercise 1 Assuming $N2 = N1 > 0$, minimum and maximum possible size are the following:

a) Everything from R1 and R2 excluding duplicates.

$R1 \cup R2 \longrightarrow$ min: (if they are all duplicates) $\frac{N2+N1}{2}$ and max: (if they are all unique) $N2+N1$

b) Everything R1 and R2 have in common.

$R1 \cap R2 \longrightarrow$ min: (if they are all unique and have nothing in common) 0 and max: (if they are all duplicates and have everything in common) $\frac{N2+N1}{2}$

c) Everything in R1 but not R2.

$R1 - R2 \longrightarrow$ min: (if they are all duplicates and have everything in common) 0 and max: (if they are all unique and have nothing in common) $N1$

d) Each row of R2 paired with each row of R1

$R1 \times R2 \longrightarrow$ min: (if R2 is empty) $N1$ and max: (if R2 is not empty, number of columns will equal N2+N1) $N1 \times N2$

e) Select row where attribute 'a' is equal to 5 from R1

$\sigma_{a=5}(R1) \longrightarrow$ min: (column unchanged but rows will change to meet the 'a=5' requirement) $N1 - a(rows)$ and max: $N1$ remains unchanged

f) Keep only columns of attribute 'a' from R1

$\pi_a(R1) \longrightarrow$ min: (rows unchanged but the number of columns is the number of attributes retained) $N1 - a(columns)$ and max: $N1$ remains unchanged

g) R1/R2 contains all N1 tuples such that for every N2 tuple in R2, there is an N1N2 tuple in R1

$R2/R1 \longrightarrow$ min: (if none of R1 has N1N2 tuples) 0 and max: (if all of R1 has N1N2 tuples) $N1$

Exercise 2

Suppliers(**sid: integer**, sname: **string**, address: **string**)
Parts(**pid: integer**, pname: **string**, color: **string**)
Catalog(**sid: integer, pid: integer**, cost: **real**)

• Find the sids of suppliers who supply some red or green part.

a) $\pi_{sid}(\sigma_{color=Red}Parts \cup \sigma_{color=Green}Parts) \rtimes Suppliers)$

• Find the sids of suppliers who supply some red part and some green part.

b) $\pi_{sid}(\sigma_{color=Red}Parts \cap \sigma_{color=Green}Parts) \rtimes Suppliers)$

• Find the sids of suppliers who supply every part.

c) $\pi_{sid}((\sigma_{pname=*}Parts) \rtimes Suppliers)$

• Find the sids of suppliers who supply every red part.

d) $\pi_{sid}((\sigma_{pname=*,color=red}Parts) \rtimes Suppliers)$

• Find the sids of suppliers who supply every red or green part.

e) $\pi_{sid}((\sigma_{pname=*,color=Red}Parts \cup \sigma_{pname=*,color=Green}Parts) \rtimes Suppliers)$

• Find the sids of suppliers who supply every red part or supply every green part.

f) $\pi_{sid}(\sigma_{color=Red}Parts \rtimes Suppliers) \cup \pi_{sid}(\sigma_{color=Green}Parts \rtimes Suppliers)$

• Find pairs of sids such that the supplier with the first sid charges more for some part than the supplier with the second sid.

g) $\pi_{sid}(\sigma_{cost=100}Catalog \rtimes Suppliers) \times \pi_{sid}(\sigma_{cost=200}Catalog \rtimes Suppliers)$

• Find the pids of parts supplied by at least two different suppliers.

h) $\pi_{pid}((\sigma_{pname=*}Parts \rtimes \sigma_{sname=Bob}Suppliers) \cap (\sigma_{pname=*}Parts \rtimes \sigma_{sname=Joe}Suppliers))$

• Find the pids of the most expensive parts supplied by suppliers named Yosemite Sham .

i) $\pi_{pid}(\sigma_{cost=max}Catalog \rtimes \sigma_{sname=YosemiteSham}Suppliers)$

Exercise 3

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

I think class has a typo where name should be cname because it is written as cname in the enrolled tuple.

1. Find the names of all students who are enrolled in 0 classes.

```
SELECT [DISTINCT] S.sname
FROM Student S, Enrolled E
WHERE SUM(E.snum) = 0
```

2. Find the course with the most students enrolled that starts before 9:00AM.

```
SELECT [DISTINCT] C.cname
FROM Student S, Enrolled E1, Enrolled E2, Class C
WHERE C.time = 9AM, SUM(E1.snum) > SUM(E2.snum)
```

3. Find the number of unique students that every professor teaches.

```
SELECT [DISTINCT] SUM(S.snum)
FROM Student S
WHERE NOT EXIST (SELECT F.fid
                 FROM Faculty F
                 WHERE NOT EXIST (SELECT C.cname
                                  FROM Class C
                                  WHERE C.fid = F.fid
                                    AND (SELECT E.cname
                                         FROM Enrolled E
                                         WHERE E.snum = S.snum
                                           AND E.cname = C.cname)))
```

4. Find the name of the youngest student who is an American Studies major or in an Intro to International Relations class.

```
SELECT [DISTINCT] S.sname
FROM Student S
WHERE  MIN(S.age)
  AND  S.major = American Studies OR S.num = (SELECT E.snum
                                              FROM Enrolled E
                                              WHERE E.cname = Intro to IR)
```

5. Print the average level of students in each class, for every class.

```
SELECT S.level
FROM Student S, Enrolled E, Class C
WHERE E.snum = S.snum, E.cname = C.cname
  AND S.level = (SELECT AVG(S2.level)
                 FROM Student S2
```

6. Print the average age and average level of students in each major, for every major.

```
SELECT S.age, S.level
FROM Student S, Enrolled E, Class C
WHERE S.age = (SELECT AVG(S2.age)
                 FROM Student S2
                 WHERE S2.major = S.major
  AND S.level = (SELECT AVG(S2.level)
                 FROM Student S2
                 WHERE S2.major = S.major
```

7. Find the major in which the most students have more than one class with a given professor.

```
SELECT S.major
FROM   Student S,
WHERE  S.major = SELECT C.cname
                 FROM Class C
                 WHERE (COUNT(SELECT F.fid
                        FROM Faculty F, Class C
                        WHERE F.fid = C.fid)) > 2
```

8. Find all pairs of students taking the same courses.

```
SELECT S1.sname, S2.sname
FROM   Student S1, Student S2, Enrolled E
WHERE  S1.snum = E.snum AND S2.snum = E.snum
  AND S1.snum < S2.snum
```