EECS 495
Introduction to Database Systems
Fall 2018
Instructor: Mas-ud Hussain
Solution: Homework Assignment No. 2

Problem No. 1:

a) Print the names of professors who work in departments that have more than 12 PhD students.
Sol:

$$\pi_{pname}((\sigma_{numphds>12}dept) \bowtie prof)$$

b) Print the name(s) of the youngest student(s).
Sol:

$$\rho(s1, student)$$
$$\rho(s2, student)$$
$$\rho(s3, \pi_{s2.sname}(s1 \bowtie_{s1.age<s2.age} s2))$$
$$(\pi_{sname}s1) - s3$$

c) Print the course names, course numbers and section numbers of all classes with more than 3 female students enrolled in them.
Sol:

$$\rho(s1, \sigma_{sex='f'}student)$$
$$\rho(e1, enroll \bowtie s1)$$
$$\rho(e2, e1)$$
$$\rho(e3, e1)$$
$$\rho(e4, e1)$$
$$\rho(e5, \sigma_{((*1)\wedge(*2)\wedge(*3))\wedge(*4)}(e1 \times e2 \times e3 \times e4)$$
$$\pi_{cname,cno,sectno}(course \bowtie e5)$$

$$*1\text{: e1.cno} = \text{e2.cno} = \text{e3.cno} = \text{e4.cno}$$
$$*2\text{: e1.dname} = \text{e2.dname} = \text{e3.dname} = \text{e4.dname}$$
$$*3\text{: e1.sectno} = \text{e2.sectno} = \text{e3.sectno} = \text{e4.sectno}$$
$$*4\text{: e1.sid} \neq \text{e2.sid} \neq \text{e3.sid} \neq \text{e4.sid}$$

d) For each Mathematics class, print the cno, sectno, and the number of enrolled students with GPAs below 3.5.
Sol:
This cannot be expressed in relational algebra because it requires a *Group By* statement or a *Count* function to find the total number of enrolled students with GPAs below 3.5.

e) Print the names of students who are majoring in both Computer Science and Mechanical Engineering.
Sol:

$$\rho(s1, \pi_{sid}(\sigma_{dname='ComputerScience'} major) \cap \pi_{sid}(\sigma_{dname='MechanicalEngineering'} major))$$
$$\pi_{sname}(s1 \bowtie_{sid} student)$$

f) Print the absolute difference in average age between Computer Science majors and Mathematics majors.

Sol:

This cannot be expressed in relational algebra because it requires an AVG function.

g) For those departments that have no majors taking the "Programming" courses, print the department name and the number of PhD students in the department.

Sol:

$$\rho(c1, \sigma_{cname = `Programming'} course)$$
$$\rho(d1, \pi_{dname}(dept) - \pi_{dname}(student \bowtie enroll \bowtie c1 \bowtie major \bowtie dept))$$
$$\pi_{dname, numphds}(dept \bowtie_{dname} d1)$$

h) Print the names of professors in departments where those departments have one or more majors who are under 18 years old.

Sol:

$$\pi_{pname}(prof \bowtie \pi_{dname}(major \bowtie \sigma_{age<18} student))$$

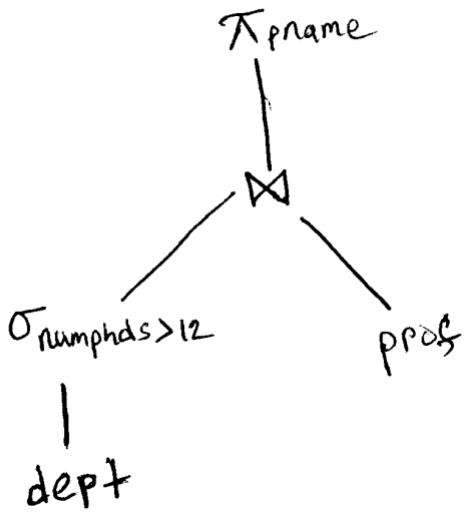
Problem No. 2:

a)

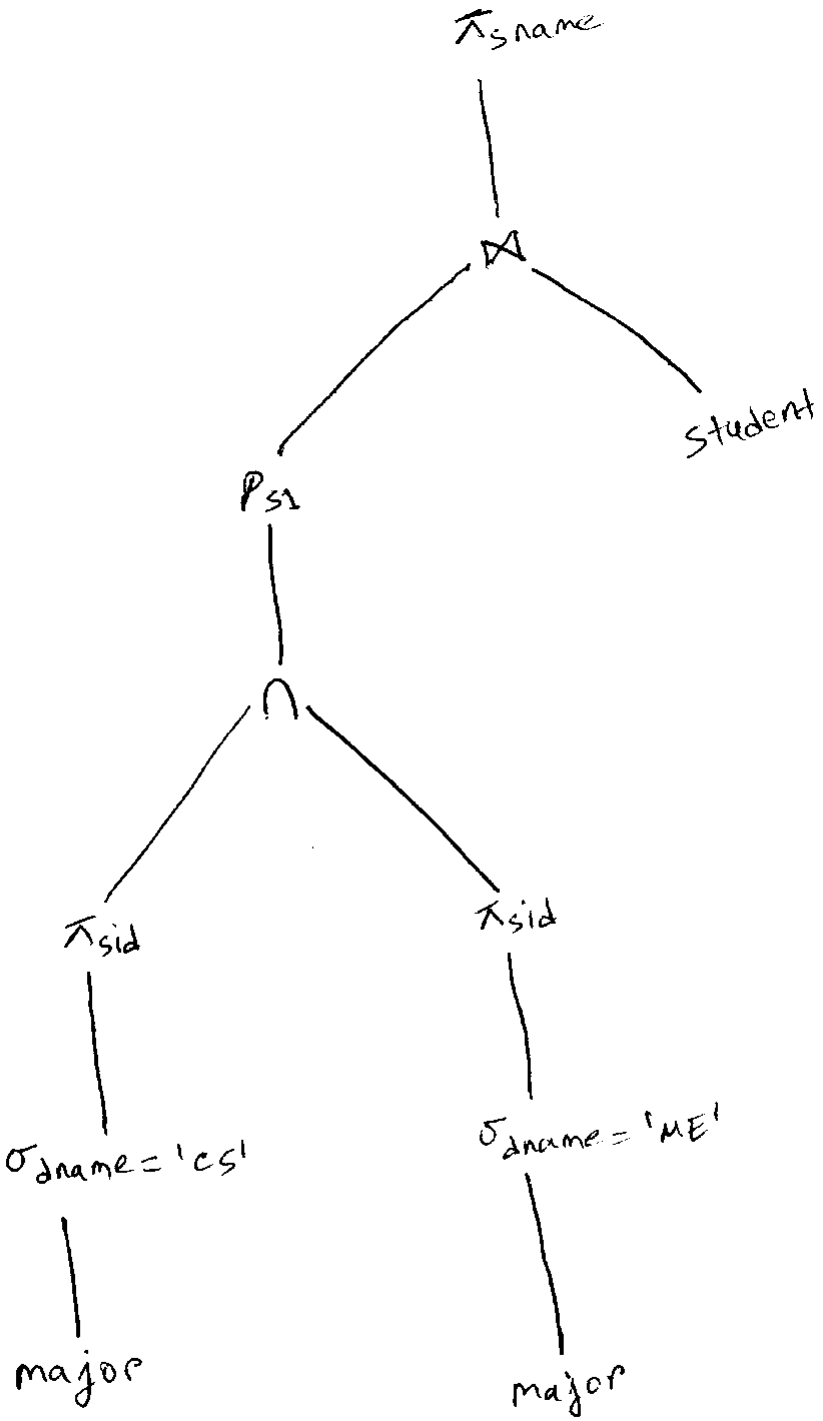
Sol:

The expression trees provided here are based on the solutions given for problem 1 in this document. Thus, the trees might be different for some students based on their own answers.

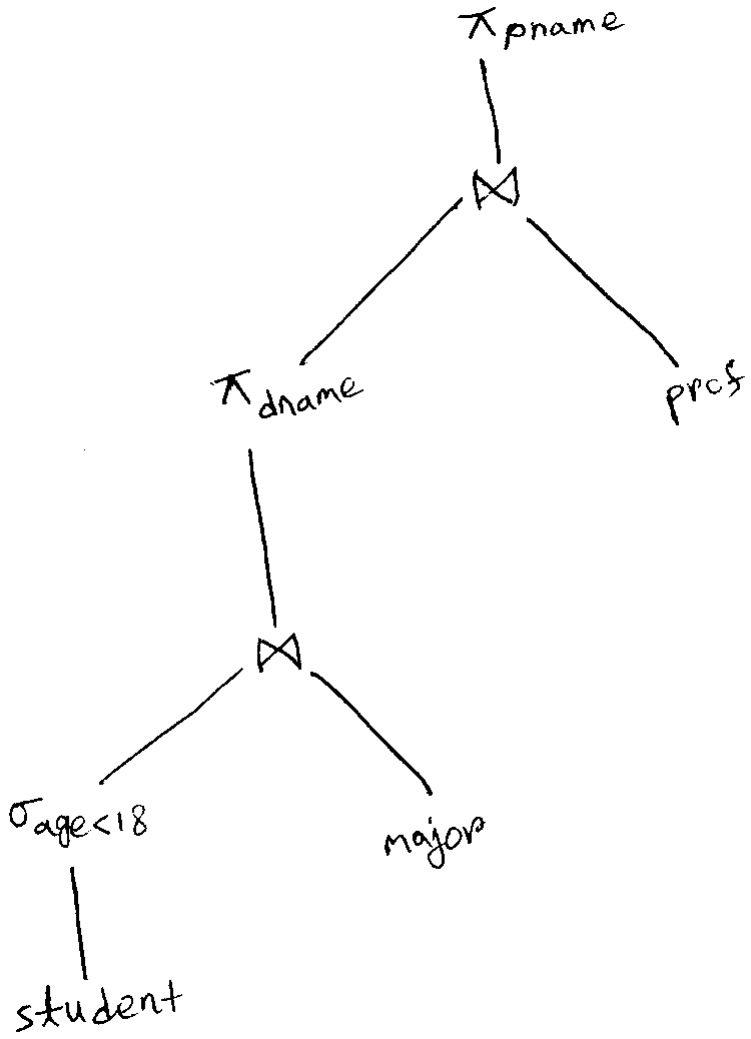
For a,



For e,


π_{sname}
$\bowtie$
Student
ρ_{s1}
$\cap$
π_{sid}
π_{sid}
$\sigma_{dname='CS'}$
$\sigma_{dname='ME'}$
major
major

For h,



b)

Sol:

The names of all suppliers who sell at least one red and one green part costing less than $ 100.

Problem No. 3:

create table **dept** (dept_no char(6),

dept_name char(20),

dept_head char(10),

primary key (dept_no));

create table **employee** (emp_no char(10),

emp_name char(20),

room_no char(8),

```
                    dept_no   char(6),
                    manager   char(10) not null,
                    primary key (emp_no),
                    foreign key (dept_no) references dept
                            on delete set null  on update cascade,
                     foreign key (manager) references employee
                            on delete cascade  on update cascade);
create table salary (salary_level   integer,
                    mon_salary   float,
                    primary key (salary_level));
create table job (job_code   char(5),
                    job_title  char(15),
                    primary key (job_code));
create table salary_hist (emp_no   char(10),
                    salary_level   integer,
                    job_code   char(5),
                    start_date   char(8),
                    end_date   char(8),
                    primary key (emp_no, salary_level),
                    foreign key (emp_no) references employee
                            on delete cascade  on update cascade,
                    foreign key (job_code) references job
                            on delete cascade on update cascade,
                    foreign key (salary_level) references salary
                            on delete cascade  on update cascade);
create table project (proj_code   char(6),
                    proj_name  char(15),
```

```
                start_date   char(8),
                end_date   char(8),
                proj_mgr   char(10),
                primary key (proj_code),
                foreign key (proj_mgr) references employee
                        on delete set default  on update cascade);
create table workson (emp_no   char(10),
                proj_code   char(6),
                primary key (emp_no, proj_code),
                foreign key (emp_no) references employee
                        on delete cascade  on update cascade,
                foreign key (proj_code) references project
                        on delete cascade  on update cascade);
```