Problem 1:

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

b) $\pi_{sname}(student) - \pi_{student.sname}(\sigma_{student.age > d.age}(student \times \rho_{d}(student)))$

c) $S1 \leftarrow \rho_{S1}(\sigma_{sex=female}(student \times eroll))$

$S2 \leftarrow \rho_{S2}(S1) \qquad S3 \leftarrow \rho_{S3}(S1) \qquad S4 \leftarrow \rho_{S4}(S1)$

$\pi_{cno, cname, sectno}(\sigma_{S1.sname != S2.sname \ \& \ S1.sname != S3.sname \ \& \ S1.sname != S4.sname \ \& \ S2.sname != S3.sname \ \& \ S2.sname != S4.sname \ \& \ S1.cno = S2.cno \ \& \ S2.cno = S3.cno \ \& \ S3.cno = S4.cno}(S1 \times S2 \times S3 \times S4)$

d) In relational algebra, it's impossible to realize the count function, so there's no such answer.

e) $temp1 \leftarrow \sigma_{dname = \text{"Computer Science"}}(major) \bowtie student$

$temp2 \leftarrow \sigma_{dname = \text{"Mechanical Engineering"}}(major) \bowtie student$

$result \leftarrow \pi_{sname}(temp1 \cap temp2)$

f) In relational algebra, it's impossible to realize the count function, so there's no such answer

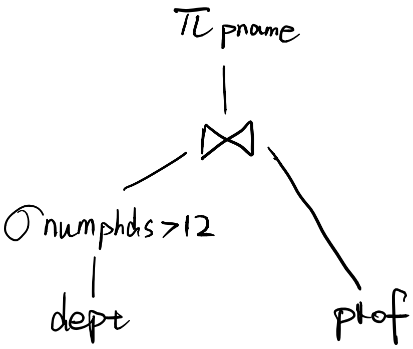
g) $temp1 \leftarrow \pi_{dname}(course) - \pi_{dname}(\sigma_{cname = \text{"Programming"}}(course))$

$result \leftarrow \pi_{dname, numphds}(dept \bowtie temp1)$

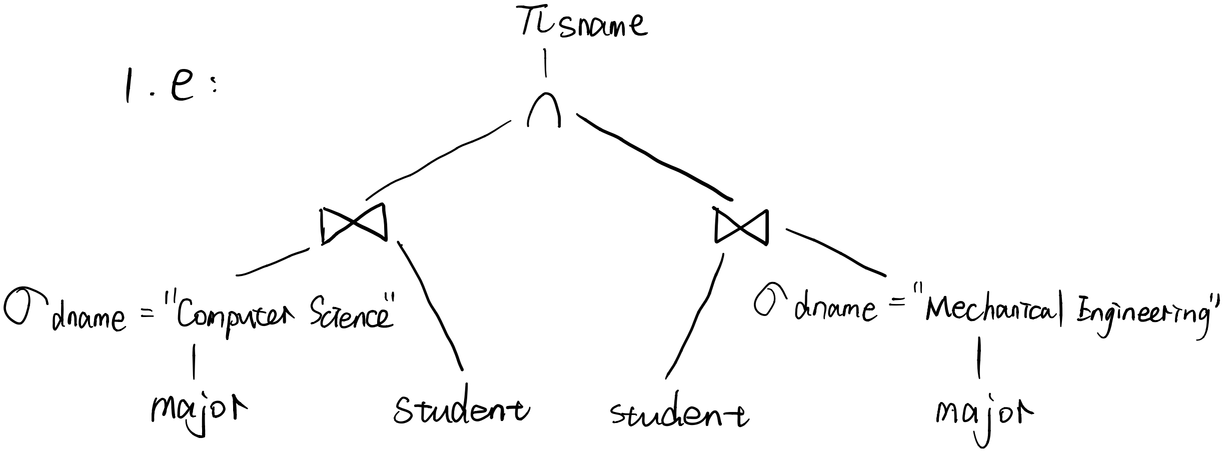
h) $\pi_{pname}(prof \bowtie \pi_{dname}(major \bowtie \pi_{std}(\sigma_{age<18}(Student))$

Problem 2:

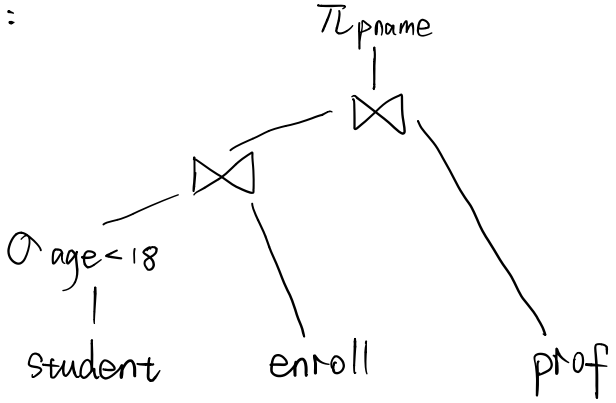
a) 1.a:



1.e:


π_{sname}
$\cap$
$\bowtie$
$\bowtie$
$\sigma_{dname = "Computer Science"}$
major
student
student
$\sigma_{dname = "Mechanical Engineering"}$
major

1.h:



b)

Print the names of suppliers who both have red and green products, and the cost of these two products is less than 100.

Problem 3:

```
CREATE TABLE Employee
(Emp_no NUMBER PRIMARY KEY,
Emp_name CHAR (20) NOT NULL,
Room_no integer NOT NULL);

CREATE TABLE Department
(Dept_no NUMBER PRIMARY KEY,
Dept_name CHAR (20) NOT NULL,
Dept_head VARCHAR (20) NULL,
FOREIGN KEY (Emp_no) REFERENCES Employee
ON DELETE SET DEFAULT ON UPDATE CASCADE);

ALTER TABLE Employee ADD
(FOREIGN KEY (Manager_id) REFERENCES Employee
ON DELETE SET DEFAULT ON UPDATE CASCADE,
FOREIGN KEY (Dept_no) REFERENCES Department
ON DELETE SET DEFAULT ON UPDATE CASCADE);

CREATE TABLE Project
(Proj_code INTEGER PRIMARY KEY,
Proj_name CHAR (20) NOT NULL,
Start_date DATE NULL,
End_date DATE NULL,
FOREIGN KEY (Emp_no) REFERENCES Employee
ON DELETE SET DEFAULT ON UPDATE CASCADE);

CREATE TABLE Works-on
(Proj_code INTEGER,
Emp_no NUMBER,
PRIMARY KEY (Proj_code, Emp_no),
FOREIGN KEY (Emp_no) REFERENCES Employee
ON DELETE CASCADE ON THE UPDATE CASCADE,
FOREIGN KEY (Proj_code) REFERENCES Project
ON DELETE CASCADE ON THE UPDATE CASCADE);

CREATE TABLE Job
(Job_code INTEGER PRIMARY KEY,
Job_title CHAR (20) DEFAULT ‘worker’ NOT NULL);


CREATE TABLE Salary
(Salary_level CHAR (20) PRIMARY KEY,
Mon_Salary INTEGER CHECK(>0),
FOREIGN KEY (Job_code) REFERENCES Job
ON DELETE CASCADE ON THE UPDATE CASCADE);

Create table Salary-hist
(Salary_level CHAR (20),
Emp_no NUMBER,
PRIMARY KEY (Salary_level, Emp_no),
FOREIGN KEY (Salary_level) REFERENCES Salary
ON DELETE CASCADE ON UPDATE CASCADE,
FOREIGN KEY (Emp_no) REFERENCES Employee
ON DELETE CASCADE ON UPDATE CASCADE);
```