B461 – Database Concepts
Practice Problems with Solution
Fall 2021

For these practice problems we will use the following relation schema:

- `Person (pid, pname, city)`
- `Company (cname, city)`
- `Skill (skill)`
- `jobSkill (pid, skill)`
- `worksFor (pid, cname, salary)`
- `Manages (mid, eid)`

In this database we maintain a set of persons (Person), a set of companies (Company), and a set of skills (Skill). The city attribute in Person specifies the city in which the person lives. The city attribute in Company indicates a city in which the company is located. (Companies may be located in multiple cities.) A person can be employed by at most one company. (We permit that a person is not employed but that a company must have at least one employee.) A person can have multiple job skills (jobSkills). A job skill can be a skill of multiple persons. (A person may not have any job skills, and a skill may have no person who has that skill.) A pair (m; e) in Manages indicates that person e is a manager of person m. The attributes mid and eid refer to the pids of the manager and the employee, respectively. We permit that a manager manages multiple employees and that an employee can have multiple managers. (It is possible that an employee is not a manager and that an employee has no manager.) We further require that if a manager manages an employee, then that employee must work for the company where that manager is also an employee.)

1. Identify and specify the likely foreign keys in the above relational schema

   Solution:

   FK(pid) in jobSkill referencing key (pid) in Person
   FK(pid) in worksFor referencing key (pid) in Person
   FK(mid) in Manages referencing key (pid) in Person
   FK(eid) in Manages referencing key (pid) in Person
   FK(skill) in jobSkill referencing key (skill) in Skill
   FK(cname) in worksFor referencing key (cname) in Company

2. Draw an entity-relationship diagram that models this relational schema. Include the constraints on the relationships. (In the exam, you can either
   a. Draw the diagram on a paper, take a picture with your phone, transfer the picture to your laptop and upload it on canvas as your response OR
   b. Write the entities (with attributes), relationships (with attributes and constraints) as your answer

Formulate the following queries in the relational algebra: You can use the abbreviations P, C, S, JS, W, and M for the relations Person, Company, Skill, jobSkill, worksFor, and Manages

3. Find the pid and pname of each person who lives in the same city as one or more of his or her managers.

Solution:

$$\pi_{P_1.pid,\, P_1.pname}(\sigma_{P_1.city\, =\, P_2.city}\ ((P_1\ \bowtie_{P_1.pid\, =\, M.eid}\ M)\ \bowtie_{P_1.pid\, =\, M.eid}\ M)$$

4. Find the cname of each company that is located in `Bloomington' but in no other city.

   Solution:

$$\pi_{cname}(\sigma_{city\, =\, Bloomington}\ (C)) - \pi_{cname}(\sigma_{city\, \neq\, Bloomington}\ (C))$$

Translate each of the following queries into an equivalent relational algebra expression in RA notation. (You are not required to optimize these RA expressions.) We recommend that you show some intermediate step in the translation. This will permit us to give partial credit for correct translation steps.

5. Query 1

```
SELECT m.mid
FROM Manages m, worksFor w1, worksFor w2
WHERE m.mid = w1.pid
       AND m.eid = w2.pid
       AND w1.salary < w2.salary
       AND w1.cname = `Google'
```

   Solution:

$$\pi_{M.mid}(\sigma_c\ (M\ \times\ W_1\ \times\ W_2))$$

   Where c is the condition

$$M.mid =\ W_1.pid\ \bigwedge M.eid$$

$$=\ W_2.pid\ \bigwedge W_1.salary <\ W_2.salary\ \ \bigwedge W_1.cname = 'Google'$$

6. Query 2

```
SELECT p.pid
FROM Person p, worksFor w
WHERE p.city = `Indy'
       AND p.city <> w.city
       AND p.id NOT IN (SELECT m.mid
                        FROM manages m
                        WHERE EXISTS (SELECT 1
                                      FROM worksFor w
                                      WHERE m.eid = w.pid
                                      AND w.salary > 20000))
```

   Solution:

First we translate the above query to the following

```
SELECT p.pid
FROM Person p, worksFor w
WHERE p.city = `Indy'
       AND p.city <> w.city
      AND NOT EXISTS (SELECT m.mid
                      FROM manages m, worksFor w1
                      WHERE p.pid = m.mid
                      AND m.eid = w1.pid
                      AND w1.salary > 20000))
```

$$\pi_{P.pid}(E - F)$$

Where E and F are the following expressions

$$E\colon \sigma_{city=Indy}(P) \bowtie_{P.city \neq W.city} W$$

$$F\colon \pi_{P.*,W.*}\big(W \times P \bowtie_{P.pid=M.mid} M\big) \bowtie_{M.eid=W_1.pid} (\sigma_{salary>2000}(W_1)))$$

Formulate the following queries using the object relational SQL

7. Create a view companyEmployees(cname, employees) which associates with each company, identified by a cname, the set of pids of persons who work for that company.

   Solution:

```
create or replace view companyEmployees as
   select distinct cname, array(select w.pid
                                    from   worksfor w
                                  where w.cname = c.cname order by
pid) as employees
   from company c order by cname;
```

8. Define a view cityHasCompanies(city, companies) which associates with each city the set of cnames of companies who are located in that city.

   Solution:

```
create or replace view cityHasCompanies as
   select distinct c1.city, array(select c2.cname
                                            from   company c2
                                            where    c2.city     =
c1.city) as companies
   from   company c1
   order by 1;
```

9. Define a view personHasSkills(pid,skills) which associates with each person, identified by a pid, his or her set of job skills. Observe that a person may not have any job skills.

Solution:

```
create or replace view personHasSkills as
   select distinct p.pid, array(select s.skill
                                from   jobSkill s
                                 where s.pid = p.pid order by 1)
                                 as skills
   from person p order by pid;
```

10. Consider the following B+-tree of order n=2 that indexes records with keys 3, 8, 10, and 11

```
         -----------
         | /  10  \ |
         -----------
         /        \
        /          \
   ----------     ------------
   | 3   8   |--> | 10   11  |
   ----------      ------------
```

Show the contents of your B+-tree index after inserting records with keys 0, 7, 14, and 9 in that order.