B561: Advanced Database Concepts
Assignment 5
Spring 2022

Due: Friday, April 1st, 11:59pm EST

This assignment covers:

- Triggers
- Queries with quantifiers

To turn in your assignment, you will need to upload to Canvas a single file with name assignment5.sql which contains the necessary SQL statements that solve the problems in this assignment. The assignment5.sql file must be so that the AI's can run it in their PostgreSQL environment. You should use the script file to construct the assignment5.sql file. (Note that the data to be used for this assignment is included in this file.) In addition, you will need to upload a separate assignment5.txt file that contains the results of running your queries and a pdf file to show your Venn diagrams.

Database schema and instances

For the problems in this assignment we will use the following database schema:[1]

```
Person(pid, pname, city)
Company(cname, headquarter)
Skill(skill)
worksFor(pid, cname, salary)
companyLocation(cname, city)
personSkill(pid, skill)
hasManager(eid, mid)
Knows(pid1, pid2)
```

In this database we maintain a set of persons (`Person`), a set of companies (`Company`), and a set of (job) skills (`Skill`). The `pname` attribute in `Person` is the name of the person. The `city` attribute in `Person` specifies the city in which the person lives. The `cname` attribute in `Company` is the name of the company. The `headquarter` attribute in `Company` is the name of the city wherein the company has its headquarter. The `skill` attribute in `Skill` is the name of a (job) skill. A person can work for at most one company. This information is maintained in the `worksFor` relation. (We permit that a person does not work for any company.) The `salary` attribute in `worksFor` specifies the salary made by the person. The `city` attribute in `companyLocation` indicates a city in which the company is located. (Companies may be located in multiple cities.) A person can have multiple job skills. This information is maintained in the `personSkill` relation. A job skill can be the job skill of multiple persons. (A person may not have any job skills, and

[1] The primary key, which may consist of one or more attributes, of each of these relations is underlined.

a job skill may have no persons with that skill.) A pair (e, m) in `hasManager` indicates that person e has person m as one of his or her managers. We permit that an employee has multiple managers and that a manager may manage multiple employees. (It is possible that an employee has no manager and that an employee is not a manager.) We further require that an employee and his or her managers must work for the same company. The relation `Knows` maintains a set of pairs (p_1, p_2) where p_1 and p_2 are pids of persons. The pair (p_1, p_2) indicates that the person with pid p_1 knows the person with pid p_2. We do not assume that the relation `Knows` is symmetric: it is possible that (p_1, p_2) is in the relation but that (p_2, p_1) is not. The domain for the attributes `pid`, `pid1`, `pid2`, `salary`, `eid`, and `mid` is `integer`. The domain for all other attributes is `text`. We assume the following foreign key constraints:

- `pid` is a foreign key in `worksFor` referencing the primary key `pid` in `Person`;
- `cname` is a foreign key in `worksFor` referencing the primary key `cname` in `Company`;
- `cname` is a foreign key in `companyLocation` referencing the primary key `cname` in `Company`;
- `pid` is a foreign key in `personSkill` referencing the primary key `pid` in `Person`;
- `skill` is a foreign key in `personSkill` referencing the primary key `skill` in `Skill`;
- `eid` is a foreign key in `hasManager` referencing the primary key `pid` in `Person`;

- `mid` is a foreign key in `hasManager` referencing the primary key `pid` in `Person`;
- `pid1` is a foreign key in `Knows` referencing the primary key `pid` in `Person`; and
- `pid2` is a foreign key in `Knows` referencing the primary key `pid` in `Person`

The file `Assignment4Script.sql` contains the data supplied for this assignment.

1 Triggers

Formulate the following queries in SQL.You can use aggregate functions in your queries and must not use set predicates where it is mentioned explicitly. You can use views, temporary views, parameterized views, and user-defined functions.

1. Explain how triggers can be used to implement the Primary key Constraint, with an example.(You are not allowed to use postgres cascade)

2. Explain how triggers can be used to implement the Referential Integrity Constraint, with an example.(You are not allowed to use postgres cascade)

3. Consider two relations R(A:integer,B:integer) and S(B:integer) and a view with the following definition:

   ```
   select distinct r.A
   from R r, S s
   where r.A > 10 and r.B = s.B;
   ```

   Suppose we want to maintain this view as a materialized view called V(A:integer) upon the insertion of tuples in R and in S. (You do not have to consider deletions in this question.)
   Define SQL insert triggers and their associated trigger functions on the relations R and S that implement this materialized view. Write your trigger functions in the language 'plpgsql.'
   Make sure that your trigger functions act in an incremental way and that no duplicates appear in the materialized view.

4. Consider applying the following constraint over the relation personSkill. "Each skill of a person who works for Google should also be the skill of the person who works for Apple".

   Write a trigger that maintains the constraint when inserting new pairs of (pid,skill) into the personSkill relation.(You can ignore the constraint restriction to hold upon the already existing previous records).

   Refer to section Q4 in the data file.

   - Consider adding the data records given in the data file.
   - Return the personSkill records across each of the newly added PIDs given in the data file.
   - Drop the records and retain the original data as provided in the data file.

5. Consider applying the following constraint over the relation knows. "Whenever a person moves from a company A to a company B, he/she should know all the managers working at the new company B."

   Refer to section Q5 from the data file to retain the data to the original state.

   Test your trigger across the below updates:

   (a) `1005 moving from Google to Apple`

   (b) `1012 moving from Apple to Google`

2 Queries with quantifiers

Using the method of **Venn diagrams with conditions** (Show these venn diagrams with conditions in pdf file), write SQL queries for the following queries with quantifiers.

To get full credit in these problems, you must write appropriate views and parameterized views for the sets A and B that occur in the Venn diagram with conditions (Show these venn diagrams with conditions in pdf file) for these queries. (See the lecture on Queries with Quantifiers.)

Hint: You can create views, functions and then use them in your query to find the answer.

Make the following two queries **without using the COUNT function**:

6. Find the pid and name of each person who knows all the persons who (a) live in Bloomington, (b) make at least 55000, and (c) have at least one skill.

7. Find the cname of each company who only employs managers who make more than 50000.

Make the following three queries **using the COUNT function**: (Show the venn diagrams with conditions in pdf file)

8. Find the pid and name of each person who knows at least 3 people who each have at most 2 managers.

9. Find the cname of each company that employs an **even** number of persons who have at least 2 skills.

10. Find the pairs (p1, p2) of different person pids such that the person with pid p1 and the person with pid p2 have the same number of skills.