Fall 2023 B561 Assignment 4

Muazzam Siddiqui, Radhika Agarwal, Nihanth Vuddanti

Released: 16 October, 2023
Due: 26 October, 2023

This assignment relies on the lectures:

- Functions and expressions in SQL;
- Aggregate functions and partitioning;
- Triggers; and
- Queries with quantifiers.

To turn in your assignment, you will need to upload to Canvas a single file with name `assignment4.sql` which contains the necessary SQL statements that solve the problems in this assignment. The `assignment4.sql` file must be so that the AI's can run it in their PostgreSQL environment. You should use the `Assignment-Script-2023-Spring-assignment4.sql` file to construct the `assignment4.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 `assignment4.txt` file that contains the results of running your queries and `assignment4.pdf` file that contains the venn diagrams with conditions.

1 Database schema and instances

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

```
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 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 (`p1`; `p2`) where `p1` and `p2` are `pids` of persons. The pair (`p1`; `p2`) indicates that the person with `pid` `p1` knows the person with `pid` `p2`. We do not assume that the relation `Knows` is symmetric: it is possible that (`p1`; `p2`) is in the relation but that (`p2`; `p1`) 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`;
- `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.

2 Solving queries using Aggregate Functions

Formulate the following queries in SQL. You must use aggregate functions in ALL these 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. Find each pair (c, pn) where c is the city and pn is the name of the person that lives in c, and earns the highest salary among all persons living in c. You must not use set predicates in this query. (5.5 Points)

2. Find the pid and name of each person who has fewer than 2 of the combined set of job skills of persons who work for Netflix. By combined set of jobskills we mean the set

$$\{s \mid s \; is \; a \; jobskill \; of \; an \; employee \; of \; Netflix \; \}$$

   Explanation: Let p1 be a person, then p1 has less than two of the skills of people working at Netflix. (5.5 Points)

3. Find each pairs (s1; s2) of skills such that the set of persons with skill s1 is the same as the set of persons with skill s2.

   Explanation: Let A be the set of people with skills s1 and B be the set of people with skill s2, find the pairs for which A and B are the same. (5.5 Points)

4. Find each pid of a person who knows at least three people who (a) work for Apple and (b) who make less than 60000. You must not use set predicates in this query. (5.5 Points)

5. Find the cname of each company, such that some person that works there knows at-least half of the people that work at Google. You must not use set predicates in this query. (5.5 Points)

6. Find each pair (c, a) where c is the cname of each company that has at least one manager, and a is the minimum salary of an employee at that company, provided that the employee is not a manager. You must not use set predicates in this query. (5.5 Points)

7. (a) Using the GROUP BY count method described in the lecture slides, define a function

```
create or replace function numberOfSkills(c text)
returns table (pid integer, salary int, numberOfSkills bigint) as
$$
...
$$ language sql;
```

that returns for a company identified by its cname, each triple (p, s, n) where (1) p is the pid of a person who is employed by that company, (2) s is the salary of p, and (3) n is the number of job skills of p. (Note that a person may not have any job skills.) (3 Points)

(b) Test your function for Problem 7a for Apple, Amazon, and ACM. (1 Point)

(c) Write the same function numberOfSkills as in Problem 7a but this time without using the GROUP BY clause. (3 Points)

(d) Test your function for Problem 7c for Apple, Amazon, and ACM. (1 Point)

(e) Using the function numberOfSkills but without using set predicates, write the following query: "Find each pair (c; p) where c is the name of a company and where p is the pid of a person who (1) works for company c, (2) makes more than 50000 and (3) has the most job skills among all the employees who work for company c." (3 Points)

3 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:

8. Find the pid and name of each person who knows all the persons who (a) live in Seattle, (b) make at least 45000, and (c) have at least one skill. (5.5 Points)

9. Find the cname of each company who only employs managers who make more than 50000. (5.5 Points)

Make the following query using the COUNT function: (Show the Venn diagrams with conditions in pdf file)

10. Find the pid and name of each person who knows at least 4 people who each have at most 2 skills. (5.5 Points)

11. Find the cname of each company that employs an odd number of persons where at least two persons have the salary greater than or equal to 55000 (5.5 Points)

12. 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. (5.5 Points)

4 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.

13. Write a trigger to check for primary key constraint. Trigger should include definition and function. (5.5 Points)

14. Write a trigger to check for referential integrity constraint. Trigger should include definition and function. (5.5 Points)

15. 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. (6.5 Points)

16. Consider applying the following constraint over the relation personSkill. "Each skill of a person who works for Apple should also be the skill of the person who works for Google". 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. (5.5 Points)

    - 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.

17. 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: (5.5 Points)

    (a) 1005 moving from Google to Apple

    (b) 1012 moving from Apple to Google