Fall 2023 B561 Assignment 1
Expressing Queries and Constraints in SQL and in Tuple Relational Calculus (TRC)

Pavan Kalyan Thota, Nihanth Vuddanti

Released: September 1, 2023
Due: September 14, 2023

1   Introduction

The goals for this assignment are to

1. become familiar with the PostgreSQL system[1];

2. create a relational database and populate it with data;

3. examine the side-effects on the state of the database caused by inserts and deletes in the presence or absence of primary and foreign key constraints;

4. formulate some queries in SQL and evaluate them in PostgreSQL; and

5. translate TRC queries to SQL and formulate queries and constraints in TRC.[2]

To turn in your assignment, you will need to upload to Canvas a single file with name `assignment1.sql` which contains the necessary SQL statements that solve the problems in this assignment. The `assignment1.sql` file must be such that the AI's can run it in their PostgreSQL environment. In addition, you will need to upload a separate `assignment1.txt` file that contains the results of running your queries. We have posted the exact requirements and an example for uploading your solution files. (See the module `Instructions for turning`

---

[1]To solve this assignment, you will need to download and install PostgreSQL (version 12 or higher) on your computer.

[2]To solve problems related to TRC, follow the syntax and semantics described in the `TRC_SQL.pdf` document in the module *Tuple Relational Calculus and SQL (lecture 4)*. That document contains multiple examples of TRC queries and constraints and how they can be translated to SQL.

`in assignments`.) Finally, you will need to upload an `assignment1.pdf` file that contains the solutions for problems related to TRC.[3]

For the problems in this assignment we will use the following database schema:[4]

$$
\begin{aligned}
&\texttt{Person}(\underline{\texttt{pid}}, \texttt{ pname, city}) \\
&\texttt{Company}(\underline{\texttt{cname}}, \texttt{headquarter}) \\
&\texttt{Skill}(\underline{\texttt{skill}}) \\
&\texttt{worksFor}(\underline{\texttt{pid}}, \texttt{ cname, salary}) \\
&\texttt{companyLocation}(\underline{\texttt{cname}}, \underline{\texttt{city}}) \\
&\texttt{personSkill}(\underline{\texttt{pid}}, \underline{\texttt{skill}}) \\
&\texttt{hasManager}(\underline{\texttt{eid}}, \underline{\texttt{mid}}) \\
&\texttt{Knows}(\underline{\texttt{pid1}}, \underline{\texttt{pid2}})
\end{aligned}
$$

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 (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.

---

[3]It is strongly recommended that you use Latex to write TRC formulas and queries. For a good way to learn about Latex, look at https://www.overleaf.com/learn/latex/Free_online_introduction_to_LaTeX_(part_1). You can also inspect the Latex source code for this assignment as well as the document TRC_SQL.tex provided in Module 4.

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

The domain for the attributes `pid`, `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`; and

- `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 `data.sql` contains the data supplied for this assignment.

2 Database creation and impact of constraints on insert and delete statements.

Create a database in PostgreSQL that stores the data provided in the `Assignment1Script.sql` file. Make sure to specify primary and foreign keys.

1. Provide 4 conceptually different examples that illustrate how the presence or absence of primary and foreign keys affect insert and deletes in these relations. To solve this problem, you will need to experiment with the relation schemas and instances for this assignment. For example, you should consider altering primary keys and foreign key constraints and then consider various sequences of insert and delete operations. You may need to change some of the relation instances to observe the desired effects. Certain inserts and deletes should succeed but other should generate error conditions. (Consider the lecture notes about keys, foreign keys, and inserts and deletes as a guide to solve this problem.) (5 Points)

3   Formulating queries in SQL

For this assignment, you are required to use tuple variables in your SQL statements. For example, in formulating the query "Find the pid and pname of each person who lives in Bloomington" you should write the query

```
SELECT  p.pid, p.pname
FROM    Person p
WHERE   p.city = 'Bloomington'
```

rather than

```
SELECT  pid, pname
FROM    Person
WHERE   city = 'Bloomington'
```

Write SQL statements for the following queries. Make sure that each of your queries returns a set but not a bag. In other words, make appropriate use of the `DISTINCT` clause where necessary.

You can **not** use the SQL JOIN operations or SQL aggregate functions such as `COUNT`, `SUM`, `MAX`, `MIN`, etc in your solutions.

2. Find the pid, pname of each person who (a) lives in MountainView, (b) works for a company which is headquartered in MountainView , and (c) has a salary less than or equal to 60000. (5 Points)

3. Find the name of each company with atleast 2 employees. (5 Points)

4. Find the ID of each person who knows someone living in Bloomington and having Networks skill (If there is a person p1, we print p1 if he/she knows atleast one person p2 such that p2 lives in Bloomington and has the Networks skill). (5 Points)

5. Find the pairs of pids (p_1, p_2) such that p1 and p2 both have atleast one common skill. (5 Points)

6. Find the ID, name, and salary of each person who works for company located in Cupertino and who earns the second lowest salary at that company.(5 Points)

7. Find the ID of managers with AI skill who are managing atleast 2 people. (5 Points)

8. Find the name of each company that has atleast two employees along with lowest salary made by an employee in that company. (The query is returns the company name and the lowest salary made by any of its employee as output). (5 Points)

9. Find the Id, Name and City pf a person who knows atleast one person who knows another person that earns more than 65000.
Let us consider 3 people p1,p2 and p3. p1 knows p2 and p2 knows p3. p3 earns more than 65000. The query returns the ID,name and city of p1. (5 Points)

4 Translating TRC queries to SQL

Consider the following queries formulated in TRC. Translate each of these queries to an equivalent SQL query.[5]

This underscores the close correspondence between TRC and SQL.

The SQL queries should be included in the `assignment1.sql` file and their outputs should be reported in the `assignment.txt` file.

10.

$\{p.pid, p.pname, w.cname, w.salary \mid Person(p) \wedge worksFor(w) \wedge Company(c) \wedge p.pid = w.pid$
$p.city = \text{'Bloomington'} \wedge 40000 \leq w.salary \wedge w.cname = c.cname \wedge c.headquarter = \text{'Seattle'}\}.$
$(5 Points)$

11.

$\{ps.pid \mid personSKill(ps) \wedge hasManager(hm) \wedge Person(m) \wedge ps.skill = \text{'Programming'}$
$\wedge hm.eid = ps.pid \wedge hm.mid = m.pid \wedge \exists ms(psersonSkill(ms) \wedge ms.pid = m.pid$
$\wedge ms.skill! = \text{'Programming'})\}.(5 Points)$

12.

$\{ps.skill \mid Person(p) \wedge personSkill(ps) \wedge p.pid = ps.pid \wedge p.city! = \text{'Bloomington'}$
$\wedge \exists m(Person(m) \wedge hasManager(hm) \wedge hm.eid = p.pid \wedge hm.mid = m.pid \wedge m.city = \text{'Bloomington'})\}.$
$(5 Points)$

13.

$\{m.pid, m.pname \mid Person(m) \wedge$
$\forall hm((hasManager(hm) \wedge hm.mid = m.pid) \rightarrow \exists e(Person(e) \wedge hm.eid = e.pid \wedge e.city = m.city))\}$
$(5 Points)$

---

[5]You can not use SQL `JOIN` operations or aggregate functions.

5 Formulating queries in the Tuple Relational Calculus

The solutions of these problems should be included in the `assignment1.pdf` file.

14. Find the pid, pname of each person who (a) lives in Bloomington, (b) works for a company where he or she earn a salary that is in range [30000, 50000] (inclusive) (c) has at least one manager. It should be a single query. (5 Points)

15. Find the pid and pname of each person who lives in a city that is different than each city in which his or her managers live. (Persons who have no manager should also be included in the answer.) (5 Points)

16. Find the pid, pname, and salary of each employee who has at least two managers such that these managers have a common job skill but provided that it is not the 'Programming' skill. (5 Points)

17. For each company, list its name along with the highest salary made by employees who work for it. (5 Points)

6 Formulating constraints in the Tuple Relational Calculus

Formulate the following constraints in TRC and as boolean SQL queries.

The TRC solutions of these problems should be included in the `assignment1.pdf` file and the SQL solutions should be included in the `assignment1.sql` file.

Here is an example of what is expected for your answers.

Example 1 *Consider the constraint "Each skill is the skill of a person." In TRC, this constraint can be formulated as follows:*

$$\forall s \; Skill(s) \rightarrow \exists ps \, (personSkill(ps) \wedge ps.skill = s.skill)$$

or, alternatively

$$\neg\exists s(Skill(s) \wedge \neg\exists ps(personSkill(ps) \wedge ps.skill = s.skill)).$$

This constraint can be specified using the following boolean SQL query.

```
select not exists (select 1
                   from   Skill s
                   where  not exists (select 1
                                      from   personSkill ps
                                      where  ps.skill = s.skill));
```

18. Each Manager manages at least two people. (5 Points)

19. Some person has a salary that is strictly lower than the salary of each of his or her managers. (5 Points)

20. Each employee and his or her managers work for the same company. (5 Points)