B561 Advanced Database Concepts
Assignment 2

This assignment is designed to enhance your skills in working with SQL. The lectures on which it relies are SQL Part 1, SQL Part 2, Views, and Expressions and Functions in SQL.

To turn in your assignment, you will need to upload to Canvas a single file with name `assignment2.sql` which contains the necessary SQL statements that solve the problems in this assignment. The `assignment2.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 `assignment2.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 in assignments`.)

In the database for this assignment, we maintain a set of persons `Person`, a relation `Knows`, a set of companies `Company`, a relation `WorksFor`, a set of job skills (`JobSkill`), and a relation `PersonSkills`. (The data for this database are given along with this assignment are in the `data.sql` file.) The schemas for these sets and relations are as follows (primary keys are underlined):

```
Person(pid : integer, name:text, city:  text, birthYear:  integer)
Knows(pid1 : integer, pid2 : integer)
Company(cname : text, city : text)
WorksFor(pid : integer, cname:text, salary:integer)
JobSkill(skill : text)
PersonSkill(pid : text, skill : text),
```

- The `city` and `birthYear` in `Person` specify the city in which the person lives and his or her birth year.
- 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 `city` attribute in `Company` indicates a city in which the company is located. (Companies may be located in multiple cities.)
- The relation `WorksFor` stores the unique company (identified by `cname`) for which a person works along with the salary he or she makes at that company. (Incidentally, it is possible that a person in the `Person` relation does not work for any company.)
- The relation `JobSkill` only has the attribute `skill` which is the name of a possible job skill.

- The relation `PersonSkill` provides for each person his or her job skills. A person may have multiple job skills. It is also possible that a person does not have any job skills.

We assume the following primary key and foreign key constraints:

- `pid` is the primary key of `Person`
- (`pid1`, `pid2`) is the primary key of `Knows`
- (`cname`, `city`) is the primary key of `Company`
- `pid` is the primary key of `WorksFor`
- `skill` is the primary key of `JobSkill`
- (`pid`, `skill`) is the primary key of `PersonSkill`
- `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`
- `pid` is a foreign key in `WorksFor` referencing the primary key `pid` in `Person`
- `cname` in `WorksFor` references a `cname` that appears 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`

1 Formulating queries in Pure SQL

Formulate the following queries in **Pure** SQL. Pure SQL is that fragment of SQL in which you can NOT use views, including temporary and parameterized views, user-defined functions, or `GROUP BY` and aggregate functions. Pure SQL is actually that part of SQL covered in lectures SQL Part 1 and SQL Part 2.

1. Find the pid and name of each person who (a) works for a company located in 'Bloomington' and (b) knows as person who lives in 'Chicago'.

   (a) Formulate this query in SQL without using subqueries and set predicates. You are allowed to use the SQL operators `INTERSECT`, `UNION`, and `EXCEPT`.

   (b) Formulate this query in SQL by only using the `IN` or `NOT IN` set predicates.

   (c) Formulate this query in SQL by only using the `SOME` or `ALL` set predicates.

   (d) Formulate this query in SQL by only using the `EXISTS` or `NOT EXISTS` set predicates.

2. Find the pid and name of each person who knows another person who works for 'Google', but who does not know a person who works at 'Amazon' and has the 'Programming' skill.

   (a) Formulate this query in SQL without using subqueries and set predicates. You are allowed to use the SQL operators `INTERSECT`, `UNION`, and `EXCEPT`.

   (b) Formulate this query in SQL by only using the `IN` or `NOT IN` set predicates.

   (c) Formulate this query in SQL by only using the `SOME` or `ALL` set predicates.

   (d) Formulate this query in SQL by only using the `EXISTS` or `NOT EXISTS` set predicates.

3. Find the cname of each company which employs at least two different persons who have at least one common jobskill.

   (a) Formulate this query in SQL without using subqueries and set predicates. You are allowed to use the SQL operators `INTERSECT`, `UNION`, and `EXCEPT`.

   (b) Formulate this query in SQL by only using the `IN` or `NOT IN` set predicates.

   (c) Formulate this query in SQL by only using the `EXISTS` or `NOT EXISTS` set predicates.

4. Find the pid and name of each person who works for 'IBM' and who has a higher salary that any person with the 'Databases' skill and who also works at 'IBM'.

   Replaced 'Database' with 'Databases'.

   (a) Formulate this query in SQL without using set predicates. Furthermore, for this question, you can use views.

   Replaced 'subqueries' with 'set predicates'. Furthermore, for this question, you can use views.

   (b) Formulate this query in SQL by using subqueries and set predicates.

5. Find the cname of each company along with the pids and names of the persons who work for that company and who have the next to lowest salary (i.e., the second lowest salary) at that company.

6. Find the pid and name of each person who knows at most one other person, and that other person has at least 2 job skills.

7. Find the each job skill that is not the job skill of any person who works for 'Yahoo' or for 'Netflix'.

   Replaced 'IBM' with 'Netflix'.

8. Find the pairs of job skills (s_1, s_2) such that each person with job skill s_1 also has job skill s_2.

9. Find the pairs of company names (c_1, c_2) such that no person who works for the company with cname c_1 has a higher salary than any person who works for the company with cname c_2.

10. Find the pid of each person who has all but 2 job skills. I.e., such a person lacks exactly two job skills from among the possible job skills.

11. Find each tuple (p_1, p_2, p_3) such that if the person with pid p_1 knows the person with pid p_2 then the person with pid p_3 does not know the person with pid p_2.

2 Formulating queries in SQL using views

Formulate the following queries in **Pure** SQL augmented with views, and this includes temporary and parameterized views. However, you can not use `GROUP BY` and aggregate functions.

1. (a) Define a view `SalaryAbove50000` that defines the subrelation of `Person` consisting of the employees whose salary is strictly above 50000.

   Replaced 'Employee' by 'Person'.

   Test you view.

   (b) Define a view `IBMEmployee` that returns the set of pids of persons who works for 'IBM'.

   Replaced the previous question 2.1.b with this new question.

   Test your view.

   (c) Using the views `SalaryAbove50000` and `IBMEmployee`, write the following query in SQL: 'Find the pid and name of each person who (a) works for 'Apple', (b) has a salary which is strictly above 50000, and (c) who does not know any person who works at 'IBM' with a salary strictly above 50000.'

2. (a) Define a parameterized view `SalaryAbove(amount integer)` that returns, for a given value for the `amount` parameter, the subrelation of `Person` consisting of the employees whose salary is strictly above that of this value.

   Replaced 'Employee' by 'Person'.

   Test your view for the parameter values 30000, 50000, and 70000.

   (b) Define a view `KnowsEmployeeAtCompany(cname text)` that returns the set of pids of persons who know a person who works at the company given by the value of the parameter `cname`.

   Test you view for the parameters 'Yahoo', 'Google', and 'Amazon'.

   (c) Using the parameterized views

   `SalaryAbove(amount)`

   and

   `KnowsEmployeeAtCompany(cname)`,

   find each triple (s, c, p) such that

   - s is the value of any possible salary that occurs in the `Worksfor` relation;
     Replaced 'Employee' by 'Worksfor'..
   - c is a company name that occurs in the `Company` relation; and

- p is the pid of a person who (a) works for the company with name c, (b) has a salary which is above the value s, and (c) knows a person who works at another company than that with name c and who has a salary that is not above the value s.

3 Queries with expressions and functions; Boolean queries

1. Let $A(x)$ be the relation schema for a set of positive integers. (The domain of x is `INTEGER`.)

   Write a SQL statement that produces a table which, for each $x \in A$, lists the tuple $(x, \sqrt{x}, x^2, 2^x, x!, \ln x)$.

   For example, if $A = \{1, 2, 3, 4, 5\}$ then your SQL statement should produce the following table:

```
   x |  square_root_x   | x_squared | two_to_the_power_x | x_factorial |    logarithm_x
---+------------------+-----------+--------------------+-------------+-------------------
   1 |                1 |         1 |                  2 |           1 |                 0
   2 |  1.4142135623731 |         4 |                  4 |           2 | 0.693147180559945
   3 | 1.73205080756888 |         9 |                  8 |           6 |  1.09861228866811
   4 |                2 |        16 |                 16 |          24 |  1.38629436111989
   5 | 2.23606797749979 |        25 |                 32 |         120 |   1.6094379124341
( 5 rows)
```

2. Let $A(x)$, $B(x)$ and $C(x)$ be three unary relation schemas that represent sets A, B and C of integers. The domain of x is `INTEGER`.

   Give answers to the following problems. You should provide two different SQL queries. One answer wherein you can use the set operations INTERSECT and/or EXCEPT, and a second answer wherein you can not use these operators.[1] You can use user-defined functions but you can not use aggregate functions.

   (a) Determine the truth-value of $A \cap B \neq \emptyset$. For example, if $A = \{1, 2\}$ and $B = \{1, 4, 5\}$ then the result of your SQL statements should be

```
 answer
--------
 t
(1 row)
```

   If, however, $A = \{1, 2\}$ and $B = \{3, 4\}$ then the result of your statement should be

```
 answer
--------
 f
(1 row)
```

[1] You are permitted to use the UNION operator.

(b) Determine the truth-value of $A \subseteq B$.

(c) Determine the truth-value of $(A \cup B) = C$.

(d) Determine the truth-value of $|(A - B) \cup (B - C)| = 1$.

3. For each of the following statements, write a boolean SQL query that determines the truth value of the following statements:

   (a) Each person has at least two job skills.

   (b) There exists a company all of whose employees have a salary above 55000.

   (c) There exists a pair of different persons who know the same persons.

4. Let $W(A, B)$ be a relation schema. The domain of A is `INTEGER` and the domain of B is `text`.

   Write a SQL query with returns the A-values of tuples in W if A is a primary key of W. Otherwise, i.e., if A is not a primary key, then your query should return the A-values of tuples in W for which the primary key property is violated. (In this query you should consider temporary views using the `WITH` SQL statement.)

   For example, consider the following relation instance for W:

W

A	B
1	John
2	Ellen
3	Ann

   Then your query should return the following answer since, in this case, A satisfies the primary property for W.

```
a
---
1
2
3
(3 rows)
```

   However, if we have the following relation instance for W

W

A	B
1	John
2	Ellen
2	Linda
3	Ann
4	Ann
4	Nick
4	Vince
4	Lisa

then your query should return the following answer because the primary key property of A for W is violated for the A-values 2 and 4.

```
a
---
2
4
(2 rows)
```