

B461 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 assignments, please upload two files to canvas: **assignment2.sql** containing your SQL code and **assignment2.txt** containing the result of running the queries. The **assignment2.sql** file must be such that the AI's can run it in their PostgreSQL environment. 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 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 that part of SQL covered in lectures SQL Part 1 and SQL Part 2.

1. Find the ID and name of each person who works for IBM and whose salary is lower than another person who works for IBM as well and has 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.
2. Find the ID and name of each person who knows another person who works for 'Hulu', but who does not know a person who works at 'Intel' and has the 'Networks' 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 name of each company located in Bloomington, but not in Indianapolis, along with the ID, name, and salary of each person who works for that company and who has the next to highest salary (i.e. the second highest salary) at that company.
 - (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 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 cannot use GROUP BY and aggregate functions.

1.

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

Test your view

(b) Define a view Programmer that returns the set of IDs of persons whose job skill is Programming. Test your view.

(c) Using the views SalaryAbove50000 and Programmer, write the following query in SQL: 'Find the ID and name of each person who (a) works for 'Netflix', (b) has a salary which is strictly above 50000, and (c) who does not know any person whose job skill is Programming 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 sub relation of Person consisting of the employees whose salary is strictly above that of this value. Test your view for the parameter values 30000, 50000, and 55000.

(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 your view for the parameters 'Yahoo', 'Google', and 'Amazon'.

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 $(x, x^{1/3}, x^x, 10^x, x!, \log_2 x)$.

Ex: $A = \{4, 8, 12, 16, 20\}$

Reference Table will be given in expected output.