CS-GY 6083-A, Principles of Database Systems, Fall 2021

Homework 5: Storage and Indexing, Datalog, Recursive SQL
Due at 11:59 ET on Thursday, December 9, 2021

Description

This assignment covers the following topics:

1) Storage and indexing
2) Datalog
3) Recursive SQL

Grading

This assignment is made up of 2 problems, collectively worth 70 points, or 8% of the overall course grade. This assignment is to be completed individually. Please consult the course syllabus for a description of our academic honesty policy.

Submission instructions

- Submit your assignment on Bright Space.
- For problem 1, submit two sql files, one for part 1(a) and another for part 1(b). Name them **1a.sql** and **1b.sql**
- Do not submit anything for problem 2, we will retrieve your score directly from Gradiance.

Part 1 (20 points)

Assume that you are given the following base (EDB) relations.

- Politician (pname, party)
- Company (cname, revenue)
- Invested (pname, cname)
- Subsidiary (parent, child)

The database creation script for this schema, along with a specific instance, is available in Politicians.sql. We also present these instances here, to explain the assignment.

Politician (pname, party)

pname	party
Don	R
Ron	R
Hil	D
Bill	D

Company (cname, revenue)

cname	revenue
C1	110
C2	30
C3	50
C4	250
C5	75
C6	15

Invested (pname, cname)

pname	cname
Don	C1
Don	C4
Ron	C1
Hil	C3

Subsidiary (parent, child)

parent	child
C1	C2
C2	C3
C2	C5
C4	C6

Write SQL queries that compute the following. Keep in mind that your queries should compute the correct result on *any legal instance* of this database.

(a) **Co-investors**: Compute all pairs of politicians (pname1, pname2) such that pname1 invests in some company, and pname2 invests in a subsidiary of that company. Sort your results by pname1, breaking ties by pname2.

For the given instance, you should return (Don, Hil) because Don invests in C1, Hil invests in C3, and C3 is a subsidiary of C1. You should not return (Don, Ron), who both invest in C1, because we don't consider a company to be a subsidiary of itself.

(b) **Investment by party:** For each political party, compute the total revenue of the companies, including all subsidiaries, in which its members (politicians) invest. Result should have the schema (party, total_revenue). Sort your results by party.

For the given instance, you should return (R, 530), counting C1, C2, C3, C4, C5, C6 for Don and C1, C2, C3, C5 for Ron. Note that each company is counted once when computing the total revenue per party, no matter how many politicians from a given party invest in the company (either directly or through its parent company).

Note: We will execute the SQL files submitted by you using the below command, so please make sure these commands run and produce outputs in the files res1.txt and res2.txt. **You will not be awarded any points if your queries in 1a.sql and 1b.sql don't run.** (Replace abc123 with your own netId.)

psql -h localhost -p 5432 -U *abc123 abc123*-db < 1a.sql > res1.txt
psql -h localhost -p 5432 -U *abc123 abc123*-db < 1b.sql > res2.txt

Part 2 (50 points)

Gradiance homework "CS-GY 6083-A: HW5: Storage and Indexing, Datalog, Recursive SQL" To access Gradiance, you must create a Gradiance account at http://www.gradiance.com/services, and then use this account to log on to the system. Having logged on, sign up for a new class, specifying the following class token: **C75D9C84.**

When you enter the token above, **CS-GY 6083-A** will appear under "Your Classes". Click on the link for **CS-GY 6083-A**, and then follow the link to homeworks in the menu on the left. You will see the currently assigned homework, and will be able to open it and submit solutions. Gradiance will automatically grade your submissions and record your scores. We will retrieve your scores for this part of the assignment directly from Gradiance, there is no need to submit anything through BrightSpace for this part of the assignment.

Note that this part of the assignment is worth 24 points on Gradiance. We will calculate your score for this part of the assignment as 2 * gradiance_score + 2.