Hópverkefni 4

Readings

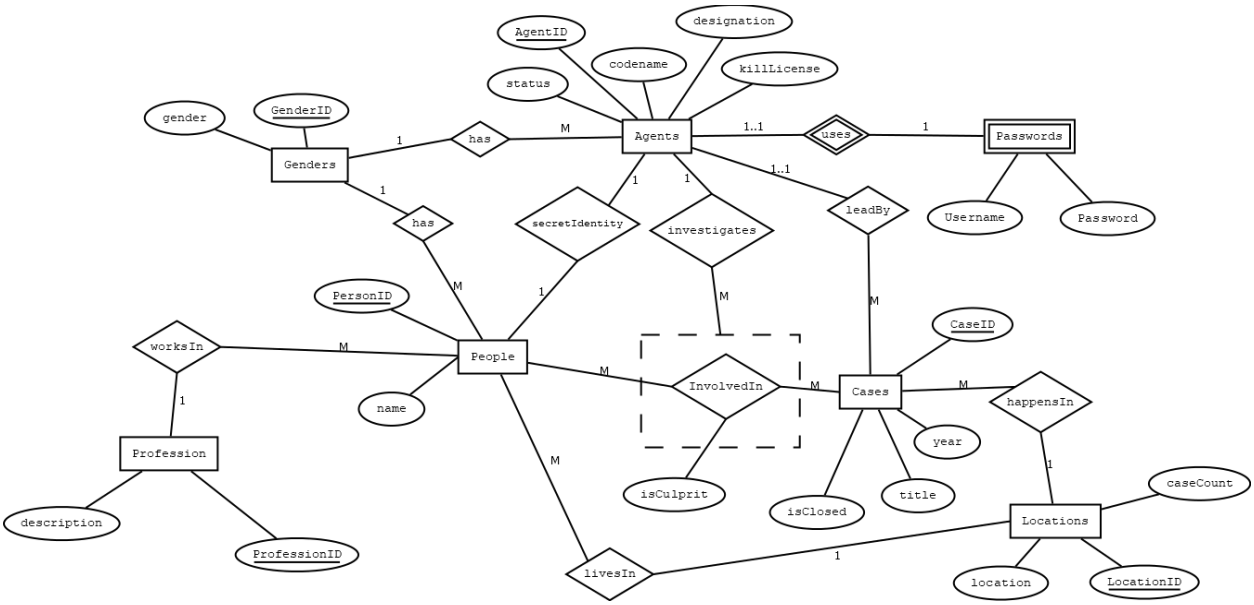
Ramakrishnan & Gehrke: Chapters 3 (esp. 3.6), 5 (esp. 5.8-5.9), and 16.6.1. Read Postgres documentation for further details, if necessary.

Project outline

Disclaimer: The description of this project is entirely fictional.

Your last meeting with the Bureau revealed that apparently there is a lot of instability in government with rumors of a revolution, and to deal with it a rock-solid database is required. The Bureau's database has a few problems with regards to agents needing to manually insert a lot of data with the possibility of mistakes ever looming.

The Director has grown quite fond of your team and has made you the head of this operation. Your task is to program some more safety and convenience into the database using views, stored procedures and triggers. "However", the Director adds, "failure is not tolerated. You don't want to experience what happened to Manny Both-Hanz after that leak last month."



Preparation and Rules

Create a new database called PIV and use the CREATE.sql file from projects 2 and 3. There is a slightly updated fill file, FILL3.sql, for this project.

Generally, your function and procedures should not rely on magic constants as input unless otherwise is clearly stated. Everything that you can do in PostgreSQL is allowed for this project, but it is up to you to figure out what is correct or incorrect.

Style is important. Do not return functions without indentation, or single line monstrosities. Remember to name your variables clearly and concisely. Helper functions are allowed, and encouraged.

Some of the problems have edge cases/corner cases that are not described here. It is important that your functions handle these by raising an exception. Generally, the later problems are more difficult than earlier ones, so if you are really quick to complete one of the later problems then it is a good idea to take a long hard look at it to make sure you really covered all possible cases.

Queries

1. *(10%)* Create a view that finds the codename and status of each agent along with the number of cases that they have led, as well as the most common location that they have led cases in. In case of a tie, display all the locations as an array.

2. *(10%)* Create a view that displays a top 3 list of suspects from 'Stokkseyri'. Those are the people involved in the most number of cases. The view should return each suspect's ID, name and the town they are from.

3. *(10%)* Create a view that finds all nemeses. When an agent finds a person to be a culprit more than once they become nemeses. The view should return the ID and codename of the agent as well as the ID and name of the person. Each person can only have at most one nemesis, and that is the agent that has busted them most often. Agents can have multiple nemeses however.

4. *(10%)* Create a procedure named InsertPerson that adds a person to the database. You should do some data validation. If the input is not correct, (empty name, gender that does not exist, or location that is not in the database) then you handle that by raising an exception. If their profession does not exist, then add it to the database.

5. *(10%)* Create a function CaseCountFixer that corrects an error in the database, currently caseCount is set to 0 for each location. The function should correct the caseCount attribute so that it is accurate for each town.

6. *(10%)* Create a trigger CaseCountTracker that whenever a new case is created, deleted or its location attribute is updated, runs CaseCountFixer.

7. *(10%)* Create a function called StartInvestigation that will open a new case for some lead agent as well as start that agent's investigation into some person for the new case. The case should take place in the person's hometown. *(Sabotage Opportunity (+5%): if the*

person's hometown happens to be the same as the agent's secret identity hometown, then you should automatically declare the person as innocent.)

8. *(10%)* When the Bureau finds defective agents working for the resistance the agent is fired immediately and removed from the database. Whenever that happens a trigger should make sure that:

    a. The cases led by the agent are distributed so that each case goes to the agent with the fewest cases to lead. In case of a tie, the case should go to the agent with a lower designation number.

    b. The people who are being investigated by that agent will still be involved in those cases, but they will not be investigated by anyone anymore.

    c. Any secret identity for that agent will be removed. Note that this has its own consequences on another table. eyda ur involved in og people toflunni

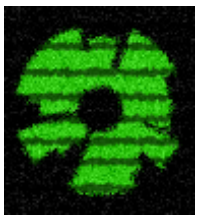
    d. *(Sabotage Opportunity (+5%): After the Agent is deleted, add them back to the database with status 'ghost', and their codename reversed. Do not use the built-in reverse function as it will trigger a security flag. Instead, you must create and use a recursive function to do the reversal. Do not undo any side effects of the deletion.)*

9. *(10%)* Create a function that returns the number of years since a case happened in a specified town. The number of years should be calculated using whatever year that the database server thinks is the current year. This function should ignore cases that happen in the future.

10. *(10%)* Create a function called FrenemiesOfFrenemies. The function takes as a parameter the ID of a person, and returns a table of all the people who that person has been involved in the same case as (we'll call them frenemies), as well as all of the people who have been involved in the same cases as the persons frenemies. Do not include the original person. Avoid duplicate results as always.

Bonus

Once again you find a surprising letter addressed to you, this time sitting on a "glámbekkur" bench in the nearby train station.



Hello Resistance member,

We are closer than ever to overthrowing the High Emperor thanks in no small part to a large combined effort to retrieve the emperor's password. We need your help one more time to prepare for the final strike to imprison the emperor and reclaim our freedom.

We need you to disable the Bureau through their database.

However, there is one small problem. If you make it too obvious then the Bureau will instantly fire you (with literal fire) and rollback the database. The changes must be small but precise. Luckily the Bureau has already tasked you with maintaining their database integrity. We've annotated the Bureau's instructions in blue with our ideas for sabotage, so please help us one more time.

Best Regards,

GAG Resistance Leader

Groups

The project is a group project. Each group should have 3 students – and you can choose your group yourself. You do not need to be in the same group as for last Hópverkefni, you are allowed to change groups if you want to, but you can also be in the same group if you would like that. Register for a group on Canvas, before handing in your group work (groups for Hópverkefni 4, start with H4). In case you do not register for a new group, your grade will be zero. We will not fix groups this time. It is not enough to have registered a group for last Hópverkefni, you need to register again, in a group starting with H4 for this particular assignment and again, if you miss this very important detail, we will not fix it for you this time.

***Note**: It is possible to get an exception and submit alone, if for some reason collaboration is completely impossible. In order to request an exception, please email me at: islind@ru.is (or on Piazza through personal message there).*

Project deliverables

The project is a group project, with three students per group. The deadline is at 23:59 on Wednesday 17. March 2021. Late submissions will not be accepted, so make sure to submit your solutions on time. A good mantra is to submit early, and submit often.

Submit one file: A text file named CODE.sql, containing the SQL code that you wrote for the project.

You should of course test your queries thoroughly by carefully inserting, deleting and updating data to test both common and boundary cases. Good luck with your project.