

# Hópverkefni 2

## Readings

Ramakrishnan & Gehrke: Sections 5.1-5.2, 5.5. Read Postgres documentation for further details, if necessary.

## Project outline

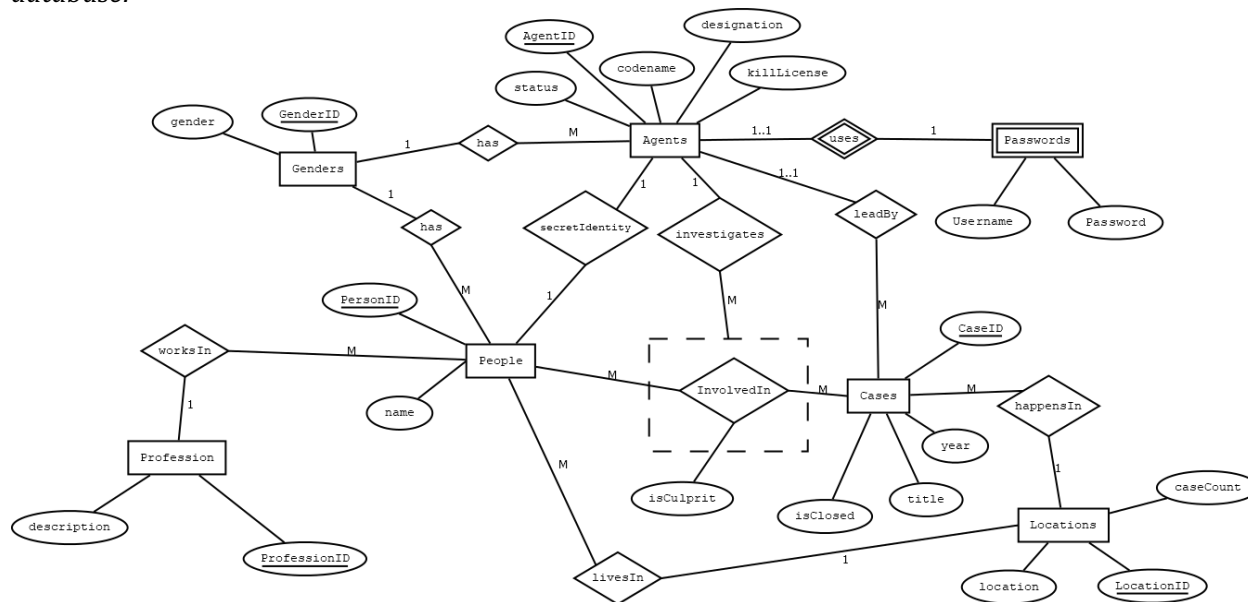
*Disclaimer: The description of this project is entirely fictional.*

After your (mostly) successful job at the Great Lava prison, you and your team have finally been approached by your true target: The Bureau. The Bureau is a shadowy organization known to be under direct command of the High Emperor himself. You didn't expect this to happen so soon, but in your mailbox you find the following:

*Hello database experts,*

*We have analyzed you and found you suitable for our purposes. You will solve the following problems or face termination. You will find all of the required documents attached. Those documents include: CREATE.sql, FILL.sql, DROP.sql and a template QUERIES.sql. It is your job to fill in the QUERIES.sql file with your solutions.*

*The database you will find attached is an older version of our current database, scrubbed of all important security details. Here you can see an ER diagram of the database:*



Once you have solved these problems you will hear from us again.

-The Bureau.

## Preparation

You need to create a database called PII and run the commands in the files CREATE.sql and FILL.sql. You can use the commands in the DROP.sql file to clean up the database at any time, and then re-run CREATE.sql and FILL.sql to set it up again. The template solution QUERIES.sql is provided to simplify starting the project.

The database seems to hold a record of People, their professions, and cases that they have been involved in. Hopefully your ER diagram skills can help you better understand how the database is set up. The data appears to be purely fictional and randomly generated, except that the locations of the cases actually exist.

## Your task

Write SQL commands to retrieve the information requested below from your PostgreSQL database. Each piece of information should be retrieved by a single query. You should make the queries as simple as you possibly can.

The output should, for example, only have the columns and rows asked for, only select from the tables required, ordering should only be used if required, and so on. Generally, the output should contain unique rows; yet, the keyword DISTINCT should be used only when needed. Note, in particular, that the construct ORDER BY ... LIMIT 1 cannot be used to find highest or lowest values. For this project, neither nested queries nor combined queries are required.

The queries should also be well and consistently formatted, and as readable as possible, as SQL queries are generally part of your code base. You should create a script and use comments to tag the SQL queries to the information requests.

Note that some queries could return empty sets. In those cases, you should of course insert sufficient data to test the queries adequately. Also note that when “gender” is requested in the queries, this refers to the description of the gender (“male”, “female” or “other”) from the Gender table. Beware the magic constant!

Always use the query “SELECT <number> AS Query;” to separate the results, as in the template solution QUERIES.txt, to make grading easier for the TAs.

## Queries

1. (10%) The ID and name of all locations, sorted descending by name
2. (10%) The name of all locations with at least one case, sorted in ascending order.
3. (10%) The number of people that are female.
4. (10%) The secret identity name of agents who have investigated people more than ten times.
5. (10%) The PersonID, name, and case title of culprits that live in the same place they committed their crime.
6. (10%) The ID, name and gender of all people who are involved in a case in Selfoss.
7. (10%) For each Person that has a profession that ends with “therapist” and is involved with an open case, list the ID, name and profession of the person and how many open cases they are involved in; the last column should be named “numcases”.

8. (10%) The codename, gender and password of all agents that have a weak password. A password is weak if the agent's codename can be found in their password.
9. (10%) The ID and name of each person involved in at least two cases in a town whose name ends in “vogur”. Additionally, a row called “hasbeenculprit” should say “guilty” if they have ever been the culprit in any of those cases, otherwise it should say “not guilty”.
10. (10%) For people that have been investigated by exactly 3 different agents, select their ID, name and gender as well as how many years it has been since they were last investigated. The last column should be named “yearsSinceLastInvestigation” Please note that some people involved in a crime are not investigated, and that The Bureau is operating in the current year: 2045.

## 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 2, start with H2). It is not enough to have registered a group for last Hópverkefni, you need to register again, in a group starting with H2 for this particular assignment.

**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](mailto:islind@ru.is) (or on Piazza through personal message there).

## Project deliverables

The project is a group project, with three students per group (you can choose your groups). The deadline is at 23:59 on Friday 12. February. Late submissions will not be accepted, so make sure to submit your solutions on time.

Submit two files:

- a) A text file named QUERIES.sql, containing all the SQL queries used to retrieve the data.
- b) A text file named RESULTS.txt, containing the results of queries in the script.

You should of course test your queries thoroughly by inserting/deleting data to test boundary cases, but that data need not be returned and the file RESULTS.txt should contain results of queries run against the original database in FILL.sql. *Note that for this project, you do NOT need to return a report.*