T-202-GAG1: Project IV

Readings

Ramakrishnan & Gehrke: Chapters 2 and 3.

Project Outline

Disclaimer: The description of this project is entirely fictional.

A political party, the Wily And Sinister Party or WASP, has found its support waning and decided to create a database to support its operations. The WASP have already hired consultants to interview potential users, coming up with the following requirements for the database, but they need you to design their database by creating an ER-diagram and transforming it into SQL tables.

Database Requirements

The following requirements were determined through interviews with WASP:

1. For people, the database should keep track of their ID, name, address, phone number, date of birth (DOB), and date of death (DOD). The default value of DOD is NULL!

2. People are further divided up into members of WASP and enemies; each person is a member, an enemy, or possibly both. For members, the starting date of membership is registered. Note that WASP members cannot leave the party, even in death!

3. For each WASP member, a list of their assets is maintained, which could be used to achieve the WASP agenda. Each asset of a person is identified with the name of the asset, but text fields must also be maintained to a) describe the asset in more detail and b) describe how the asset could potentially be used.

4. For each WASP enemy, one or more WASP members are assigned as opponents. The opponent appointment has a start date and an end date (the latter may be unknown, in case of open-ended assignments).

5. Furthermore, events and relationships between various people (e.g. accidents, business partnership, marriage, mortal enemies, …) are registered in a fairly flexible way. The WASP party uses the general term "linkings". Each linking is an entity that is assigned an ID, name, type and a more detailed description. Multiple people may participate in each linking (the minimum number of people per linking is one, in case of an event) and each person may participate in multiple linkings with different people.

6. The WASP party has a list of roles, each with a unique ID and a unique title. (Roles include party roles, such as chairman, and external roles, such as election candidate or senator.) Members take turns filling the roles: each member may fill multiple roles at any given time and multiple members may fulfill the same role. Members may, however, only be appointed to the same role once. The start and end dates of each appointment to a role are always known in advance and recorded, as well as the monthly salary for the role.

7. The WASP party keeps track of other political parties, both domestically and abroad. For each party, the country and name are a unique identifier, but the party is given an ID as well that serves as a primary key. For each such party, one WASP member monitors the developments in the external party.

8. The WASP maintains a list of sponsors. For each sponsor, an ID is registered, as well as their name, their address and their industry.

9. Each sponsor may give grants to a number of WASP members. Each grant, however, is to a single person. For each grant, the date the grant is awarded is registered, as well as the amount and a text field called "payback" that describes what the sponsor expects in return. Each sponsor may give multiple grants to multiple members, and each member may receive multiple grants from multiple sponsors.

10. Each grant is reviewed by one member of WASP. The date of the review is decided when the grant is registered, typically one year in the future. At review time, the grant is assigned a numerical grade from 1 to 10, depending on how well the WASP member executed the payback.

Project Restrictions

The database design should follow the requirements definition strictly. For example, ID columns should only be used where specifically mentioned. Where ID columns are mentioned, however, you can assume that they are a key. Assume that columns are NOT NULL, unless the project description indicates otherwise.

There are requirements, however, which cannot be specified in the ER-diagram. Some of these can be addressed in the database creation script, while others may require more advanced techniques, such as assertions (which are not available in Postgres). When standard CREATE TABLE commands are not sufficient to implement requirements, you must note this in your report. You are not asked to write triggers to implement those restrictions. If you do that, however, you will receive extra credit.

Groups

The project is a group project. Each group must have 2-3 students, and students must form groups themselves.

Note: *It is possible to get an exemption and submit alone, if for some reason collaboration is completely impossible. In order to request an exemption, please send mail to gylfig@ru.is. No exemptions will be made for larger groups!*

Project Deliverables

Solutions must be submitted on Canvas. The deadline is at 23:59 on Thursday October 18.

Late submissions will not be accepted, so make sure to submit your solutions on time.

Submit three files:

a) A PDF file containing a complete ER-diagram. The diagram must follow the notation presented in the R&G book and in lectures, and must not be hand-drawn.

   The ER-diagram is the main deliverable of the project, as most of the entities and relationships are captured there. In order to fully specify the database design, however, the remaining two documents are also needed.

b) A query file (.sql) containing SQL commands to create the corresponding database tables.

c) A PDF file with a short report, which a) enumerates assumptions you have made that you felt were not clearly specified in the project description or cleared up in discussions on Piazza; b) enumerates all important design decisions made when converting the ER-diagram to tables; and c) enumerates all requirements which are *not* represented in either the ER-diagram or the SQL code.

   The report should have the title "Database Design for WASP", have your names, a short introduction of the contents, which should mention which software or web-site you used to generate the ER-diagram, and then three sections, one for each of the above enumeration.

Correctly supporting each of the 10 requirements above, or showing why (some part of it) cannot be supported, gives 10% of the final grade for the project. It is possible, however, to lose points due to unclear presentation in any one of the three files.