Hópverkefni 5

Readings

Ramakrishnan & Gehrke: Chapter 19 and lectures: Module 08.

Project outline

Disclaimer: The description of this project is entirely fictional.

After the Great Database Collapse of 2045 and the GAG Revolution, the High Empire was finally abolished and the emperor imprisoned. A new democratic government has been formed and you, the database experts have been deemed heroes of the people!

You decide that the best thing to do with your time is to finally take that well deserved vacation. Packing your things, you head out to the wonderful Nauthólsvík beach in Reykjavík to catch some rays.

You put out your beach towel and are prepping to set up your sunscreen when you are approached by the newly repurposed delivery SNATI (Super Neat And Timely Infobot). The SNATI starts playing an audio recording from its mouth-like speaker.

Hello database expert,

This is the new President of Iceland: Stella, I'm sure you voted for me. I'm very sorry to interrupt your vacay but the old empire left a big mess in our databases. We need you to sort out this mess. The government currently has only 4 tables: A lot of people are depending on these (what is government without coffee anyway?)

Once you are finished with this we'll be sure not to bother you again. Good luck.

Best regards,

President Stella (no longer campaigning)

The project

The goal of this project is normalization: to transform a set of relations into the highest normal form possible, while preserving all existing dependencies.

You are given a script (CREATE.sql) to create four independent relations (CivilServices, Projects, Citizens and Coffees), each of which has seven columns and a primary key. You are also given a script (FILL.sql) to fill the four relations with data; on average each relation contains little over nine thousand rows.

Each relation models a potential real-life database design situation, but with some design problems that must be addressed. In short, each of the four relations has embedded a set of functional dependencies and/or multi-valued dependencies. You must a) find these dependencies and b) use them to guide the decomposition of the relations.

Assumptions

For this project, we consider 1NF, 2NF, 3NF, BCNF (based on functional dependencies) and 4NF (based on multi-valued dependencies).

Furthermore, in this project the following simplifying assumptions hold for each of the relations:

- The four relations must each be considered in isolation. The columns have short names that hint at their meaning, but you should not count on implicit FDs derived from the expected meaning of the columns. In short, the column names may trick you!

- Assume that all functional dependencies in each relation (aside from primary key dependencies and dependencies derived from that) can be found by checking only FDs with one column on each side, such as A → B.

- Assume that all multi-valued dependencies have only one column on each side. - Note: You need only seek MVDs in relations with three columns in the key.

- Assume that no join dependencies are embedded in the relations; we do not consider 5NF in this project.

- If you find a (functional or multi-valued) dependency that holds for the instance given, assume it will hold for all instances of the relation.

  Exception: When an ID column and a corresponding string column both determine each other, consider that only the ID column determines the string column, not vice versa.

  Example: If CID → CN and CN → CID, then consider only CID → CN valid.

- The only dependencies you need to consider for decomposition are a) the dependencies that can be extracted from the data based on the assumptions above, and b) the given key constraints.

As discussed in the book and in the online lectures, you can use SQL queries to detect potential FDs and MVDs. Using the assumptions above, you can indeed create a script to generate **all** possible checks for FDs and MVDs, thus automating the detection process. This is the role of the dæmatímaverkefni.

Normalization process

For each relation, you must determine the following:

1) All the (functional and multi-valued) dependencies represented in the relation (based on the assumptions above);
2) All the keys of the relation (based on the detected dependencies);
3) The minimal cover of the set of functional dependencies; and
4) The current normal form of the relation.

Then you must decompose the relation until each sub-relation is in either 3NF or BCNF, as well as in 4NF, while preserving all FDs of the minimal cover.

For each relation resulting from the decomposition, you must determine the same four items as above.

You must document all these steps and outcomes and write a report. Then you must create the resulting tables and populate them, by extracting the relevant data from the original relations.

Naming and filling relations

For easier grading, you must assign unique and consistent names to decomposed relations, which can then be easily and consistently checked. Here is how!

Each relation that you create must be named by concatenating the name of the original relation and the names of all the columns it contains, with "_" between, in creation order. For example, if an original relation Person (not part of this project) is partially decomposed into a relation containing the columns ID, DID and DN, the resulting relation must be named Persons_ID_DID_DN.

To populate the new relations, use the INSERT INTO … SELECT … notation in SQL (e.g., see http://www.w3schools.com/sql/sql_insert_into_select.asp).

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 5, start with H5). 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 H5 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 31. March 2021.
Late submissions will not be accepted, so make sure to submit your solutions on time. And note that, as before, one-student "groups" are only allowed in exceptional cases with prior permission.

Submit three files:

a) A PDF file (REPORT.pdf) describing, for each relation, the normalization process, as described in Section 4 above.

b) A text file (DECOMPOSE.sql) containing SQL commands to create the resulting database tables.

c) A text file (POPULATE.sql) containing SQL commands to fill the resulting database tables from the tables in the original database.

Good luck with this final group assignment.