T-202-GAG1: Project 5

# Readings

Ramakrishnan & Gehrke: Chapter 19.

# The Project

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

You are given a script (CREATE.sql) which creates five relations, R1 through R5, which all have five columns, called A, B, C, D and E. Each of the tables, however, has a different primary key; you can find the primary key in the CREATE TABLE statements.

You are also given a script (FILL.sql) to populate the five relations with data. Each relation 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.

*Note: The script FILL.sql inserts almost 66 thousand lines into the five relations. You need to make sure that at least one group member has a DBMS installation capable of handling this (relatively small) database.*

# Assumptions

For this project, we consider 1NF, 2NF, 3NF, BCNF, 4NF and 5NF.

The relations R1 through R5 should be considered in isolation, as each relation describes a different hypothetical situation.

Furthermore, you can assume that the following simplifying assumptions hold for each of the relations R1 through R5:

* No join dependencies are embedded in the relation.
* All dependencies in the database have one column on both sides, such as   
  A 🡪 B or A 🡪🡪 B.
* If you find a (functional or multi-valued) dependency that holds for the instance given, then it will hold for all instances of the relation.
* The only dependencies you need to consider for decomposition are a) the dependencies that can be extracted from the data and b) the given key constraint.
* No FDs hold other than those you can detect in your data, so you can accurately categorize relations to 1NF, 2NF, 3NF or BCNF.
* Other multi-valued and join dependencies *may* hold, however, so you must be careful in your statements about 4NF and 5NF.

# Normalization

For each relation R1 through R5, you must determine the following:

1. All the dependencies represented in the relation.
2. All the keys of the relation.
3. The normal form of the relation.

Then you must decompose the relation to the highest normal form possible, and no lower than 3NF, while preserving FDs.

For each relation resulting from the decomposition, you must determine the following:

1. The key(s) of the relation.
2. Any functional dependencies that exist in the relation, aside from trivial FDs and FDs of the form *key*🡪*attribute.*
3. The highest normal form you can claim for the relation.
4. An argument for why the relation is or is not in: 3NF, BCNF, 4NF and 5NF.
   1. For 3NF and BCNF, you must state why the relation is, or is not, in that normal form (recall the assumption above about FDs).
   2. For 4NF and 5NF, you must state why the relation is in that normal form, or why you *cannot state* whether the relation is in that normal form or not.

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.

*Hint: You can use SQL queries to detect potential FDs (given the assumptions above, you can in fact use SQL queries to* determine *FDs). You can also write SQL queries to assist you in detecting MVDs, although more interpretation of results is needed. If you are clever, you can even create a script to generate all possible checks for FDs, thus largely automating the detection process.*

*Hint: To populate the new tables, use the INSERT INTO … SELECT … notation in SQL (e.g., see* [*http://www.w3schools.com/sql/sql\_insert\_into\_select.asp*](http://www.w3schools.com/sql/sql_insert_into_select.asp)*).*

# Naming Convention

Each relation that you create should be named by concatenating the name of the original relation (R1 through R5) and the names of all the columns it contains, in alphabetical order. For example, if the original relation was named R1, then a decomposed relation containing the columns A, B and C should be named R1ABC. This guarantees unique names, which can be easily and consistently checked.

# Deliverables

***The project is a group project, with three (or two) students per group. The deadline is at 23:59 on Wednesday October 15. Late submissions will not be accepted, so make sure to submit your solutions on time.***

Submit three files:

1. A PDF file (report.pdf) describing, for each relation R1 through R5, the normalization process, as described in Section 4.
2. A text file (CREATE.sql) containing SQL commands to create the resulting database tables.
3. A text file (FILL.sql) containing SQL commands to fill the resulting database tables from the tables in the original database.

Correctly normalizing each of the 5 files above gives 20% of the final grade.