INF1343, Winter 2012, Answers to Week 6 Exercises

Part 1. Simple Normalization

Determine whether each of the following two tables is in 1NF, 2NF or 3NF. For tables that do not reach 3NF provide an alternative relational design that satisfies 3NF.

a) A table that a dentist users to keep track of her work on her patients teeth:

> (first_name, last_name, insurance_company_name, patient_insurance_number, insurance_company_phone_number, visit_date, which_tooth, symptoms, diagnosis, treatment_notes, material_used, fee, amount_billed, amount_currently_owed)

There are no 1NF violations here. For further analysis we would have to make some assumptions about what would work as keys. (With a table so deeply broken it is sometimes hard to determine this.) Almost any reasonable assumption, however, will give us a 2NF violation. For example, if we assume that (first_name, last_name, insurance_company_name, patient_insurance_number, visit_date, symptoms) is a key then we have a partial dependency from (first_name, last_name, insurance_company, patient_insurance_number) → amount_currently_owed.

Here is a redesign satisfying 3NF:

> *insurance_company (<u>insurance_company_id</u>, insurance_company_name, insurance_company_phone_number)*
>
> - *Note that (insurance_company_phone_number) is a probably unique and thus works as a key, though it wouldn't be a good choice for the PK.*
>
> *patient (<u>patient_id</u>, first_name, last_name, insurance_company_id^{FK}, patient_insurance_number, amount_currently_owed)*
>
> - *One could argue that (insurance_company_id, patient_insurance_number) is a key, though this would require assuming that all companies assign a different number to each patient. It is, possible, however, that they do not. (For example, they might cover dependents under the same number.) In either case this would not be a good choice for the PK.*
>
> *visit (<u>visit_id</u>, patient_id^{FK}, visit_date)*
>
> *issue (<u>issue_id</u>, visit_id^{FK}, which_tooth, symptoms, diagnosis)*
>
> *treatment (<u>treatment_id</u>, visit_id^{FK}, treatment_notes, material_used, fee, amount_billed)*
>
> - *It would seem to make sense that material_used should be in a separate table to allow multiple materials to be associated with the same treatment. However, as it is the table does satisfy 3NF, so I am leaving it at that.*

b) A table containing information about people who registered for events at a community center:

> (title, first_name, last_name, email, home_telephone, work_telephone, event_name, event_date, event_price, spouse_name, child_name_1, child_name_2, child_name_3)

Identify primary and foreign keys in your alternative design.

The last three attributes represent the wrong solution to a 1NF violation. (Though, note that this is not, technically speaking a 1NF violation.) Let's start by fixing this:

> *person_event (person_id, title, first_name, last_name, email, home_telephone, work_telephone,*

event_name, event_date, event_price, spouse_name)

child (person_id, child_number, child_name)

Assuming that event names cannot repeat within the same date (otherwise this table wouldn't work at all), we have (person_id, event_name, event_date) as one of our keys and a partial dependency: person_id → first_name. (There are several other partial dependencies as well.) Since we fail 2NF we also automatically fail 3NF.

Here is a redesign that satisfies 3NF:

event (event_id, event_name, event_date, event_price)

person (person_id, title, first_name, last_name, email, home_telephone, work_telephone, spouse_name)

child (person_id^{FK}*, child_number, child_name)*

registration (person_id^{FK}*, event_id*^{FK}*)*

Part 2. Normalization in the Wild

Go to the Millenium Development Goals dataset in the World Bank Data Catalog website:

http://data.worldbank.org/data-catalog/millennium-development-indicators

Download the Excel version of the datafile:

http://databank.worldbank.org/databank/download/MDG_Excel.zip

The files contain 5 sheets, not counting the "Description" sheet.

a) For each of the five tables, determine whether the table satisfies the requirements for 1NF, 2NF and 3NF.

MDG_Data: The table has separate columns for each year. This is not technically a 1NF violation, though it represents the wrong way of dealing with a 1NF issue. Further analysis depends on whether we assume that different countries can have the same "Country name" and whether different series can have the same "Series name." In this context of this data set this is not the case. This means that "Country name" and "Series name" can be used in keys. This means that dependencies such as "Country_Code" → "Country Name" do not "count" as partial dependencies. So, 2NF is satisfied. However, if your intuition is telling you that this table shouldn't have "Country name" and "Series name" in it, you are correct. The presence of those attributes is a problem whether or not the letter of 2NF is violated. There are no transitive dependencies in this table, so if 2NF is satisfied then so is 3NF.

MDG_Country: 1NF is not satisfied because of the multiple values in "Latest household survey" and "Alternative conversion factor." Therefore, 2NF and 3NF are also violated automatically. If we ignored this 1NF violation, though, then the same analysis applies in regards to "Long name" and "Short name" as does to "Country name" in MDG_Data.

MDG_Series:This satisfies 3NF.

MDG_Country_Series_Notes: This table satisfies 3NF.

MDG_Footnotes: This table satisfies 3NF.

b) Show how the same data can be represented as a set of tables that satisfy 3NF.

If we think that no two countries can have the same name and that no two series can have the same name, then we can just fix 1NF violations and consider ourselves done. However, let's do a proper job and bring it to BCNF:

country (country_code, country_name, region, income_group, region, lending_category, other_groups, currency_unit, latest_population_census, special_notes, national_accounts_base_year, national_accounts_reference_year, system_of_national_accounts, sna_price_valuation, ppp_survey, balance_of_payments_manual_in_use, external_debt_reporting_status, system_of_trade, government_accounting_concept, imf_data_dissemination_standard, source_of_most_recent_income_and_expenditure_data, vital_registration_complete, latest_agricultural_census, latest_industrial_data, latest_trade_data, latest_water_withdrawal_data, two_alpha_code, wb2_code, table_name, short_name)

- *The large number of attributes in this table should make us wonder if we are looking at some variation of "wrong solution to 1NF" problem. Many of the attributes basically state the source of different kinds of data. So, this table would probably benefit from being broken up further into two tables:*

  > *country (country_code, country_name, region, income_group, region, lending_category, other_groups, currency_unit, special_notes, … latest_water_withdrawal_data, two_alpha_code, wb2_code, table_name, short_name)*
  >
  > *country_data_source (country_code, data_type, source)*

  If we do this, we might be able to get rid of the two tables we had to create to deal with repeated groups. We must be asking here, though, whether it would make more sense to associate this information with a combination of country_code and series code in a "country_series" table.

latest_household_survey (country_code^{FK}*, value)*

alternative_conversion_factor(country_code^{FK}*, value)*

series (series_code, indicator_name, short_definition, source, topic, periodicity, base_period, aggregation_method, conceptual_implications, general_comments)

country_series_notes (country_code^{FK}*, series_code*^{FK}*, note)*

footnote (country_code^{FK}*, series_code*^{FK}*, year, footnote)*

data (country_code^{FK}*, series_code*^{FK}*, year, value)*