Task 3: Refinement, normalization, and SQL-DDL on the medical data

Dr. Rudra Pratap Deb Nath
Associate Professor
Department of computer Science and Engineering
University of Chittagong

Project Description

The overall goal in the whole mini-project is to create and work with a medical database in several incremental steps. The first step is to identify the information that is going to be represented as well as special requirements. Then, the information is modeled in an ER diagram and mapped to a relational schema. The database schema is then refined and normalized. Later on, database dumps will be provided and queried.

Refinement, normalization, and SQL-DDL

In the previous task you have created a relational schema. In this task, the main goal is to extend it and improve it by identifying functional dependencies and applying normalization. Assume you have a client who wants to use your database and is interested in the following pieces of information. You kindly agreed to check your database schema and adapt it if necessary[1]. Thus, your database should be able to store at least

- information about drugs: name, category, products, possible side effects, diseases it can treat, clinical trials
- information about diseases: name, category, possible treatments, clinical trials
- information about products: name, company, drugs, diseases
- information about clinical trials: title, start date, completion date, number of participants, status, conditions studied, main researcher, location

Report

In task 1, you have already designed this database based on what you already knew back then. Now, try it again based on what you have learned so far in the course— please use the notation introduced in the course (slides) and solve the following sub-tasks:

1. Identify functional dependencies for each of the relations you created.
2. If necessary, transform your schema into 3NF and formally show that the result is actually in 3NF.
3. Determine for each obtained relational schema the highest normal form it still supports, i.e., check if your relations are also in BCNF.
4. Describe the normalized relations with
   - their definitions (name, attribute names and types, keys)
   - information an how and why you derived them from the original relations
   - information on the highest normal form that
5. List the SQL statements that create all your tables properly and ensure the functional dependencies that you have identified. Hint: Consider UNIQUE NOT NULL constraints to enforce candidate key characteristics. Do not forget to define primary keys and foreign keys.

[1] If you have used the list of sample questions included in the previous self study, most of this information should already be in your schema.

6. Revise your ER diagram if you identify flaws in your original design
7. Compare your revised ER diagram and your normalized relations with the ones from tasks 1 and 2. Examples of questions that you might consider are:
    - What are the differences to your previous design?
    - Why did these differences occur?
    - What advantages does the new design have over the old one?

Course goals covered by this self study

- Create and evaluate a database schema that adheres to normal forms (logical design)
- Improving the quality of an existing database design
- Using SQL to create tables

Submission

Submit on the Google classroom by 4 PM Sunday (06 July 2025).