CS-A1153 Project – Designing a Relational Database

TOPIC: Modelling the distribution of vaccines

Introduction

Assuming that you are responsible for Corona vaccine distribution and treatment in Finland, you need to build a database to keep track of the different vaccine types, transportation of vaccine batches, treatment plans, and patient data.

(1/5) Description: vaccine manufacturers and vaccine batches

- A pharmaceutical company that has designed a vaccine type can grant licences to many manufacturers, who produce this vaccine type in batches and send them to Finland. Each manufacturer has (1) a unique ID, (2) countries of origin, and (3) contact information (phone number), and (4) ID of the vaccine they manufacture.
- The number of batches varies depending on the order number as well as availability. We need to store the data about received vaccine batches, including (1) batch ID, (2) number of vaccines in the batch, (3) vaccine type, (4) the date the vaccine was produced, and (5) the expiration date.
- There are different types of vaccines, and each batch only contains one type of vaccine. The vaccine data should include (1) the ID of the vaccine, (2) how many doses of this vaccine should be given, and (3) critical temperature.

(2/5) Description: Transportation log

- Once the vaccine batches arrive to Finland, they will be delivered to different hospitals and clinics around the country. The data about hospitals and clinics needs to be stored in the database. These include (1) names, (2) addresses, and (3) telephone numbers. The names of the hospitals and clinics are unique.
- Vaccines can be transported between hospitals and clinics. The transportation log of vaccine batches needs to be stored, and should include the departure and arrival dates, and the departure and arrival hospitals or clinics.

(3/5) Description: Vaccination shifts

- Each hospital and clinic have their own vaccination shifts for the workers. The shifts will specify which staff members are working on which weekday. The shifts repeat weekly. There can be at most one vaccination shift plan for each weekday for each hospital or clinic.
- Data about staff members should be stored, including (1) social security number (2) name, (3) birthday, (4) phone number, (5) vaccination status, and (6) role ("nurse" or "doctor").

(4/5) Description: Vaccination events

- Vaccinations will happen at hospitals and clinics. The dates of the vaccinations should be stored, and the workers are determined by the vaccination shift of the hospital for that weekday.
- Around 50-100 patients will get vaccinated in one vaccination event.
- Each hospital or clinic may organize only one vaccination event a day.
- Only one vaccine batch can be used in each vaccination event. One batch can be used in multiple vaccination events.

(5/5) Description: patients and symptoms

- Patients' information should include (1) social security number, (2) name, (3) date of birth, (4) gender, and (5) vaccination status.
- The data about which patients attended which vaccinations has to be stored in the database. Each patient needs to attend one or two vaccinations, depending on the vaccine given.
- We want to keep track of symptoms that are reported by a certain patient as well as the date when the symptom was diagnosed.
- Data of symptoms will include its name and a boolean attribute that marks whether the symptom is critical or not. Diagnoses of critical symptoms may lead to doctor's appointments or clinic visits, or to prioritizing the patient in the vaccination queue.

Requirements for Part 1

Using the description above, design a database using the UML class diagrams as specified below.

- Draw an UML diagram for the vaccine database based on the information defined in this document. Use the notations taught in the course.
- Convert the UML diagram to the relational data model. Present the schemas of the relations and underline the attributes which form the key for each relation.
- Provide answers to the following questions: What are the functional dependencies of the database? Are there any form of redundancy or other anomalies in the database structure? Is your database in the Boyce-Codd Normal Form? If it is not, use the decomposition algorithm (submit both original and decomposed version).