

COMP-421/764 Database Systems, Winter 2021

Project 1: Database Design and Data Modelling

Due Date Feb 02, 05:00pm

In the programming project of this course, you will develop and build a database application for a **real-world** domain. Step-by-step, you will design a schema, create a database using DB2, populate your database with data, maintain, query and update your data, develop application programs, and implement a user-friendly interface. The interface can be very simple, so no requirement for web-programming, etc. You will only use a standard programming language (Java) in the last project deliverable. In the first deliverable of the project, you are only concerned with developing a data model for your application.

RAMQ has hired you to develop a database (and associated application) to manage the COVID vaccination distribution process in Quebec. Will you play a pivotal role in this process and deliver the results by the end of winter or will it be another bleak summer of lockdowns in Quebec?

To constrain to the scope of the course, we have simplified many real-world situations and procedures in the below description. Any semblance to reality is merely an accident.

Application Requirement

The COVID vaccination program is open to anyone who is a Quebec resident and is willing to sign up for it. A person who is interested in taking the vaccine has to call RAMQ (Quebec's health care system) and register their name. At this point, your application will record the person's basic information such as health insurance number, name, phone, address (city, postal code, street addr), date of birth, gender, etc., as well as the date on which the person registered.

Additionally, a priority system is setup to ensure that those among the population that is at the highest risk is vaccinated first. A person who registers is immediately assigned to the highest priority category that they are eligible for. The following list of categories summarize the current priority system (the highest on top). It is assigned based on their situation on the day that they are registering.

1. Health Care workers, Elderly (≥ 65), Immunologically Compromised
2. Teachers, Children below 10, Those in physical proximity to first priority
3. Essential Service Workers, Those in physical proximity to second priority
4. Everybody else

Be mindful of how this information is modeled and stored in the system. Your application should be capable of generating a list of registered individuals ordered based on their priority as well as identifying individuals belonging to a specific category (for example, who are the health care workers?). However, the system has to record only the category associated with the highest priority that can be assigned to that person.

The vaccination is performed through designated vaccination locations. They have a unique name (simplification). Each location can vaccinate a number of people on a given day (may vary depending on the dates) and provides a number of slots for each day that they are open for vaccination and the availability of vaccine. The city, postal code and street address of the vaccination location must be stored in the system. As vaccine shots and free slots become available, administrative workers (not tracked in the system) of RAMQ will create slots for the locations and call registered individuals based on their priority, etc. and provide them a slot at a vaccination location on a specific day. The date on which this allocation is done must be captured in the system. Only one individual maybe assigned a given slot. Keep in mind that a slot is not (only) necessarily a notion of a time as there could be multiple

concurrent slots depending on the infrastructure and personnel available at a location (e.g. there are 10 vaccination tents in a location on a certain day). Also, just because the place has the ability, does not mean it will be able to maximally utilize its space as they may not have enough vaccines available. Make sure that the system has a way of tracking the time and date associated with the slot. Allocation of a slot for each vaccine dose for a person will be done separately and not at the same time.

Vaccine shots are administered by registered nurses. Although they are always employed with a specific hospital, they may be deputed to a vaccination location (these locations need not be a hospital) that is different from their employer on specific days (an example is a nurse providing vaccination shots at a Pharmacy on a certain day). The system needs to keep track of the information about the nurse (their name, Canadian nurses license number, their actual employer, etc.) as well as where they are assigned on a specific day (even if it is their own employer's location). Multiple nurses could be on the same location on a given day. Nurses are assigned by RAMQ depending on the needs of different vaccination locations on specific days, their location assignment is not necessarily already made when slots are provided to the registered individuals.

The system should also keep a record of vaccine related information. Presently, Quebec is planning to use *Pfizer-BioNTech* and *Moderna*, but the system must be able to accommodate any new vaccines without involving data model or application changes. Each vaccine has the need for a specific number of doses to be administered to be effective and a waiting period between those doses (the same vaccine is used for all the doses of a person). Although, all the vaccines known at this time requires two doses, your system should be capable of handling any variation without having to change the model or the application. This vaccine metadata is stored in the system for reference purposes. The system of course, should have a way of tracking the number of doses already administered to an individual.

Vaccine shots are manufactured and transported in batches and our system is only interested in storing the manufacturing date, batch number, and expiry date associated with them. This information is entered into the system when the vaccine batch reaches the vaccination location. Keep in mind that two different vaccine companies may end up with identical manufacturing information for their vaccine batches. The system should also keep track of which vaccine batch was sent to which location. Vaccine shots are enclosed in vials, which can be uniquely identified within a batch by using the vial number labelled on them. For simplicity, we assume that a vial has a single dose of vaccine (meant for a single individual for one shot). The vial information is entered into the system only when it is administered to someone. However, the number of total vials in a batch must be available for reference in the system (this number can vary slightly due to various reasons). The general idea is that, using the system, we should be able to figure out the total number of vaccine shots (vials) that are available, how many got used, etc.

For medical record keeping, the system should be able to keep track of which vaccine (vial) was used to vaccinate whom, at which location, and who was the nurse that administered it. Any nurse who is on that location could be responsible for administering the shot, and are not pre-assigned to a specific individual or slot before that. Keep in mind that we need to be able to distinguish between the multiple vaccine doses that an individual is administered.

The Assignment

1. (20 Points) An ER schema/diagram including your data requirements. Be careful not to forget to underline key attributes, indicate the types of relationship sets etc. Make sure that your ER diagram has large enough fonts, etc. so that the TA can read it properly. You may use a page size larger than letter format in your drawing software. **The ER diagram you submit should be a searchable PDF** (If you export your drawing as PDF in most of the drawing softwares like Powerpoint, Google draw, etc., you should get a searchable PDF). This helps TAs quickly lookup some of the key information without having to manually look for it in the entire diagram. **Do not hand draw and scan!!** Turn in the diagram as **ER.pdf**. Use only the ER drawing notations that we discussed and covered in class. Use sensible names for your entity sets, attributes and relationships so that the TA can easily relate your model to the requirement. You may use sensible abbreviations of the name (e.g *hinsurnum* to indicate health insurance number), etc. to reduce clutter in your diagram.

The above application description may not be comprehensive in capturing every minute details. You are free to make assumptions that do not defy common sense as long as it does not go against what is already explicitly provided. Please do not add more requirements into the system. You will end up making your life difficult further down. Do not make “assumptions” because you do not know how to solve a modeling problem. If you make some assumptions which you think may not be that “intuitive”, please provide a PDF file **project.pdf** that contains those, so that the bureaucrats do not get on your case. Make sure to write them under a section heading **Assumptions**.

Further, if there are any requirements and constraints within the application that you cannot depict in the ER

diagram (there will be quite a few), you need to point them out now, so that your developer (your future self) can keep an eye out for them when writing the application code. Indicate them under a section **Restrictions** in **project.pdf**.

2. (5 Points) Use the method for translating an ER diagram to relations described in class and depict each resulting relation in the form **Relationname(attr1, attr2, attr3,...)** underlining the key attributes. Indicate when attributes are foreign keys to other relations by writing something like “attr3 foreign key referencing relation X” beside the relation. Are there opportunities to combine relations without introducing redundancy? If so, indicate which, and if not, tell us there are none. Write this under a section **Relational Translation** in **project.pdf**. Also indicate any ER model aspects that your relational model does not capture in this section.

Note: DO NOT write SQL create statements instead of the relational model. You will not be given any points. You do not have to decide on the data types either.

Submission Format: Your submission will contain **two** separate documents:

1. **ER.pdf** : The ER diagram of your project. This is a PDF file (not drawn on paper). Make sure that the ER diagram you submit is readable (such as font size, etc) and searchable (similar to how you can search for a word in the lecture slides and find it). You may use a page size larger than letter for your ER diagram page if that ensures it is readable. Make sure that the thick line vs thin lines, dotted lines, etc., are very clear in the diagram without having to zoom into it at 200x.
2. **project.pdf** Rest of your project document (assumptions, restrictions, relational translation, etc.), should be in a PDF file (again a searchable PDF).

Please turn in your submission in mycourses under project 1. Only a maximum of 3 late days allowed (15% of the maximum allocated points deducted per day, rounded up). Project grades also influence the pass/fail criteria of the course. Please review the course outline for any of these details.

Approach

Read the requirement part 2-3 times to get a general idea of what you are dealing with. Sit down with a pencil and paper (old school style), draw out just the entity sets and keys in it that evidently stand out in the requirements. Draw the relationships between them. Now go back to the requirements and see what is it that is missing in your ER model and make one more pass. Now you have a decent base ER model to work with. You can start drawing it in your drawing software and tune/adjust as required, add additional attributes that are part of the entity sets, etc.

Review your ER model. In essence, you are trying to read the model and figure out what it can do (like the last quiz question we did in class). Is there any important “information” that your system is not able to capture (e.g. what is the name of the vaccine given to a particular person?) - this is a bad situation. Take actions to address it. Does your model result in duplicate information - try to avoid this when you can through proper modeling techniques, if that is not possible, or if you have sensible reasons (as discussed in class) to keep it as such, document it in your **restrictions** so that your developer is aware of this. Can your model result in inconsistent information being stored? (e.g. a nurse working in one location in a day but giving shots in another location on the same day) - again if a proper modeling technique does not exist to solve this problem, then you should document it.

It is beneficial if you split the entire exercise over multiple days instead of cramming into one day, so that you have a little bit of time to think about it at the back of your mind as you keep refining your model. Good ideas do not pop out when you are under duress.

Remember, your ER model will not be able to capture all of the requirements. The challenge is to recognize this for specific situations. Do not over-complicate the model. It does not necessarily solve the problems, but might add more. A simple model should be preferred over a complex model with the same number of limitations.

An ideal implementation of this requirement should result in approx 10 entity sets and a similar number of relationships. Use that as a guidance to keep track of your modeling adventures.

Questions ?

Please use Piazza for any clarifications you need. Do not email the instructor or TAs as this leads to a lot of duplicate questions and responses (not an efficient system). Please check the pinned post “P1 general clarifications” before you post a new question. It might have been already addressed there, in which case we will not address it again.

Do keep in mind that clarifications are limited to the nature of the application requirement (for example you want to make an assumption that a nurse would work only in one location on a given day as it is not stated in the requirement). Do not ask for clarification on whether your ER solution/approach is correct or not. Remember, the bureaucrats know only of their application needs, not much about software development.

There will be specific office hours for the project that will be announced closer to the due date.