University of Toronto
CSC343, Fall 2022

Assignment 1

Declare your group on MarkUs (even if working alone): as soon as you begin working together
Assignment due: Wednesday, October 5th at 4:00 pm sharp!

Learning Goals

By the end of this assignment you should be able to:

- Read a new relational schema and determine whether or not a particular instance is valid with respect to that schema.
- Apply the individual techniques for writing relational algebra queries and integrity constraints that we learned in class.
- Combine the individual techniques to solve complex problems.
- Identify problems that cannot be solved using relational algebra.

These skills will leave you well prepared to be a strong SQL programmer.

Schema

Our schema for this assignment is for a pharmacy.

Relations

- Product(DIN, name, manufacturer, form, schedule, route)
  A tuple in this relation represents a **brand-name** drug product, that can be purchased from the pharmacy. *DIN* is the Drug Identification Number, *name* is the name of the drug, *manufacturer* is the name of the manufacturer, *form* is the form in which the drug product is produced (e.g., "capsule"), *schedule* is the category in which the federal government places the drug (e.g., "narcotic"), and *route* is the route of administration of a drug product (e.g., "oral" or "intravenous").
  The possible values for *schedule* are defined in an integrity constraint below.
- Generic(DIN, brand, name, manufacturer)
  A tuple in this relation represents a **generic** drug product, that can be purchased from the pharmacy. *DIN* is the Drug Identification Number, *brand* is the DIN of the corresponding brand-name drug, *name* is the name of the generic drug, *manufacturer* is the name of its the manufacturer. All the information about the form, schedule, and route of the corresponding brand-name drug applies to a generic alternative. For example, if the brand-name drug is a capsule that is a narcotic and is taken orally, so is the corresponding generic drug.
- Price(DIN, price)
  A tuple in this relation represents the price of a drug product. *DIN* is the Drug Identification Number, and *price* is its price.
- ActiveIngredient(name)
  A tuple in this relation represents an active ingredient, that may be used in the formulation of a drug product. *name* is the name of the active ingredient.
- Contains(DIN, ingredient, strength, unit)
  A tuple in this relation represents that an active ingredient is used in the formulation of a drug product. *DIN* is the Drug Identification Number of a brand-name drug, *ingredient* is the name of the active ingredient, *strength* is the strength of the active ingredient (e.g., 200), and *unit* is the units in terms of which the strength is expressed (e.g., "mg").

- Interaction(ingredient1, ingredient2)
  A tuple in this relation represents the fact that active ingredients *ingredient1* and *ingredient2* may result in adverse effects if consumed together.

- Patient(OHIP, name, dob, phone, address)
  A tuple in this relation represents a patient. *OHIP* is the patient's OHIP number, *name* is the patient's name, *dob* is the patient's date of birth, *phone* is the patient's phone, and *address* is the patient's address.

- Pharmacist(OCP, name, registered)
  A tuple in this relation represents the fact that a pharmacist is registered with the Ontario College of Pharmacists. *OCP* is their Ontario College of Pharmacists identification number, *name* is their name, and *registered* is the date on which they were registered.

- Prescription(RxID, date, patient, drug, doctor, dosage, note)
  A tuple in this relation represents a prescription. *RxID* is the prescription ID, *date* is the date on which it was written, *patient* is the OHIP number of the patient, for whom this prescription was issued, *drug* is the drug product it is a prescription for, *doctor* is the identification number of the doctor who wrote it, and *dosage* is the dosage of the prescription.

- Filled(RxID, date, pharmacist)
  A tuple in this relation represents the fact that a prescription was filled. *RxID* is the prescription ID, *date* is the date on which the prescription was filled, and *pharmacist* is the OCP number of the pharmacist that filled the prescription.

Integrity constraints

- $\pi_{\text{DIN}}\text{Product} \cap \pi_{\text{DIN}}\text{Generic} = \phi$
- $\text{Generic[brand]} \subseteq \text{Product[DIN]}$
- $\pi_{\text{DIN}} Price - (\pi_{\text{DIN}}\text{Product} \cup \pi_{\text{DIN}}\text{Generic}) = \phi$
- $\text{Contains[DIN]} \subseteq \text{Product[DIN]}$
- $\pi_{\text{drug}}\text{Prescription} - (\pi_{\text{DIN}}\text{Product} \cup \pi_{\text{DIN}}\text{Generic}) = \phi$
- $\text{Contains[ingredient]} \subseteq \text{ActiveIngredient[name]}$
- $\text{Interaction[ingredient1]} \subseteq \text{ActiveIngredient[name]}$
- $\text{Interaction[ingredient2]} \subseteq \text{ActiveIngredient[name]}$
- For any two active ingredients A and B, if A interacts with B then B interacts with A.
  (You will express this formally in Part 2. Assume it holds when writing queries in Part 1.)
- $\text{Product[DIN]} \subseteq \text{Contains[DIN]}$
- $\text{Prescription[patient]} \subseteq \text{Patient[OHIP]}$
- $\text{Filled[RxID]} \subseteq \text{Prescription[RxID]}$
- $\text{Filled[pharmacist]} \subseteq \text{Pharmacist[OCP]}$
- $\pi_{\text{schedule}}\text{Product} \subseteq \{$ “prescription”, “narcotic”, “OTC”, “homeopathic” $\}$
- $\sigma_{\text{Prescription.RxID=Filled.RxID}and\text{Prescription.date>Filled.date}}(Prescription \times Filled) = \phi$

Part 1: Queries

Write the queries below in relational algebra. There are a number of variations on relational algebra, and different notations for the operations. You must use the same notation as we have used in this course. You may use assignment, and the operators we have used in class: $\Pi, \sigma, \bowtie, \bowtie_{condition}, \times, \cap, \cup, -, \rho$. Assume that all relations are sets (not bags), as we have done in class, and do not use any of the extended relational algebra operations from Chapter 5 of the textbook (for example, do not use the division operator).

Some additional points to keep in mind:

- Do not make any assumptions about the data that are not enforced by the original constraints given above. Your queries should work for any database that satisfies those constraints.
- Assume that every tuple has a value for every attribute. For those of you who know some SQL, in other words, there are no null values.
- Remember that the condition on a select operation may only examine the values of the attributes in one tuple, not whole columns. In other words, to use a value (other than a literal value such as 100 or “Adele”), you must get that value into the tuples that your select will examine.
- The condition on a select operation can use comparison operators (such as $\leq$ and $\neq$) and boolean operators ($\vee$, $\wedge$ and $\neg$). Simple arithmetic is also okay, *e.g.*, attribute1 $\leq$ attribute2 + 5000.
- In your select conditions, you may refer to the year component of a date attribute d using the notation d.year, and you can compare date attributes using comparison operators such as $<$.
- You are encouraged to use assignment to define intermediate results.
- The order of the columns in the result doesn’t matter.
- If there are ties, report all of them.
- When we talk about something happening, for instance, 3 times, we mean 3 or more times. If we mean *exactly* 3 times, we will say so.

At least one of the queries and/or integrity constraints in this assignment cannot be expressed in the language that you are using. In those cases, simply write “cannot be expressed”.

Note: The queries are not in order according to difficulty.

1. *Frugal doctors:* Find all doctors who have only prescribed drugs that are either (a) the cheapest generic alternative of some brand-name drug (if a brand-name drug has multiple generic alternatives tied for lowest price, prescribing any one of them satisfies this criterion), or (b) a brand-name drug with no generic alternative.

   Only consider drugs, whether brand or generic, for which a price is recorded. Exclude doctors who haven’t prescribed at least two different drugs, i.e., given prescriptions with at least two different DINs. Report the doctor’s identification number.

2. *Price gougers:* Find all pharmacists who have never filled a prescription for a generic product. Exclude pharmacists who have never filled any prescription at all. Report the pharmacist’s OCP number, name, and date of registration.

3. *Potential doctor shopping:* Two medications are equivalent if (a) they have the same DIN, (b) they are a brand-name medication and a generic equivalent, or (c) they are two generic drugs that share the same brand-name equivalent.

   Find all patients who have been prescribed equivalent medications by two different doctors. That is, doctor 1 prescribed medication A, doctor 2 prescribed medication B, and medications A and B are equivalent. Report the OHIP number, name, and phone number of the patient.

4. *Safest ingredient:* Find the active ingredient that interacts with the fewest other ingredients. Report just the name of the ingredient.

   If there is a tie for fewest interactions, report all tied ingredients. An ingredient that interacts with no other ingredients, if there is one, will definitely be included in the answer.

5. *Drug shortage:* Find all drugs, whether brand-name or generic, for which there are more than two unfilled prescriptions and where the unfilled prescriptions were written for at least two different patients. Report the DIN and manufacturer.
6. *Protecting drug patents:* Find all pairs of brand name drugs that have the exact same active ingredients (disregarding strength and unit). Report the DIN and name for each. Do not include "pseudo-duplicates". That is, if you report that drug 123 has the same active ingredients as drug 987, do not also report that drug 987 has the same active ingredients as 123.
7. *Recent narcotics:* For each pharmacist, find the prescription for a narcotic, whether brand-name or generic, that they have filled most recently. Report the pharmacist's name and OCP number, the name of the narcotic, and the date of that most recent filling. If there is a tie for most recent, report them all.

   If a pharmacist has never filled a prescription for a narcotic, they will not appear in the result.
8. *Patients at risk:* Find every doctor who has given a patient prescriptions on the same day for multiple drugs such that two or more of the drugs, whether brand-name or generic, interact with each other. Two drugs are considered to have an interaction if any of their active ingredients interact.

   Report the doctor's identification number and the date on which the interacting prescriptions were given. If a doctor has done this more than once, include a tuple for each relevant date.
9. *Many generics:* Find the pharmacist who has filled the largest number of prescriptions for generic drugs. If there are ties, report them all. Report the pharmacist's OCP number, and the filling date of the last prescription they have filled.
10. *Long-time customers:* A most senior pharmacist is one with the oldest date registered; a most junior pharmacist is analogous. (There could be ties for most senior or most junior, which is why we said "a" rather than "the".)

    Find every patient who has had a prescription filled by a most senior pharmacist and a prescription filled by a most junior pharmacist. Report the OHIP number and name of the patient, the earliest date on which they had a prescription filled by a most junior pharmacist, and the earliest date on which they had a prescription filled by a most senior pharmacist. These dates could potentially be the same.
11. *Lots of competition:* Find manufacturers for whom the following is true: (1) they make one or more brand-name drugs, (2) they themselves manufacture a generic drug alternative for each brand-name drug they make, and (3) every one of their brand-name drugs also has a generic alternative that is manufactured by some other company (not necessarily all by the same company). Report the manufacturer name.

Part 2: Additional Integrity Constraints

Express the following integrity constraints with the notation $R = \emptyset$, where R is an expression of relational algebra. If a constraint cannot be expressed using this notation, simply write "cannot be expressed". You are welcome to define intermediate results with assignment and then use them in an integrity constraint.

1. *Symmetry:* If ingredient A interacts with ingredient B, then ingredient B interacts with ingredient A.
2. *Don't surpass those with seniority:* A pharmacist cannot fill more prescriptions in a year than a pharmacist who is senior to them fills in that year. (Pharmacist A is senior to pharmacist B if A's registration date is before B's.) Remember that you may refer to the year component of a date attribute d using the notation d.year.
3. *Brand-name first:* A doctor cannot write a prescription for a generic product unless they have already written a prescription for its brand-name equivalent on an earlier date.
4. *Don't over-prescribe narcotics:* On any one day, a doctor cannot write more than one prescription for the same brand-name narcotic for the same patient.

When writing your queries for Part 1, don't assume that these additional integrity constraints hold (except for the symmetry constraint, which was noted in the schema).

Formatting instructions

Your assignment must be typed; handwritten assignments will not be marked. You may use any word-processing software you like. Many academics use LaTeX. It produces beautifully typeset text and handles mathematical notation well. If you would like to learn LaTeX, there are helpful resources online. Many people use `overleaf.com` to do their LaTeX work in the cloud. It also makes co-editing a document with your partner easy. If you want to work locally, `TeXShop` is a good option.

Whatever you choose to use, you need to produce a final document in pdf format.

If you use software that lets you choose a font size, it must be at least 10. If you use LaTeX, the default font size (or larger) is acceptable.

Submission instructions

You must declare your team (whether it is a team of one or two students) and hand in your work electronically using the MarkUs online system. Instructions for doing so are posted on the Assignments page of the course website. Well before the due date, you should declare your team and try submitting with MarkUs. You can submit an empty file as a placeholder, and then submit a new version of the file later (before the deadline, of course); look in the "Replace" column.

For this assignment, hand in just one file: A1.pdf. If you are working in a pair, only one of you should hand it in.

Check that you have submitted the correct version of your file by downloading it from MarkUs; new files will not be accepted after the due date.

How your assignment will be marked

Most of the marks will be for the correctness of your answers. However, there will be additional marks allocated to each of these:

- Comments:
  Does every assignment statement have a comment above it specifying clearly exactly what rows get to be in this relation? Comments should describe the data, (*e.g.*, "The student number of every student who has passed at least 4 courses.") not how to find it (*e.g.*, "Find the student numbers by self-joining …"). Put comments *before* the assignment, and two dashes on each line of your comment.
- Attribute names given on the LHS:
  Does every assignment statement name the attributes on the LHS? This should be done even if the names are not being changed. Together with the comments, it allows you to understand what an intermediate results contains without reading the algebra on the RHS. Think of this as analogous to good parameter names and good comments on a function.
- Relation and attribute names:
  Does every relation and every attribute have a name that will assist the reader in understanding the query quickly? Apply the same high standards you would when writing code.
- Formatting:
  Is the algebra formatted with appropriate line breaks and indentation for ease of reading and ease of understanding?

Final advice

These are my top pieces of advice for doing a great job of A1, painlessly:

- Have the summary of specific techniques beside you like a ”cheat sheet” (and have mastered understanding each one)
- Make a concrete instance of the relevant relations and what the result of the query should be for this instance. Write it down. Think like a computer scientist and make sure it tests out a few good conditions.
- Write down the LHS of the last step. Name the relation and attributes well and write a comment explaining what it takes to get into that relation. Don’t bother with the RHS yet. Then imagine some intermediate result that would make that last step easy. Don’t worry about how to create it, just write down the LHS and the comment. Keep reasoning backwards. Don’t write down any RHSs at all — no algebra! — until you have the whole thing broken down.
- Leave plenty of time for typing up your answers; the formatting will take longer than you may realize.