csc343 winter 2021
assignment #1: relational algebra
sample solution

goals

This assignment aims to help you learn to:

- read a relational scheme and analyze instances of the schema
- read and apply integrity constraints
- express queries and integrity constraints of your own
- think about the limits of what can be expressed in relational algebra

Your assignment must be typed to produce a PDF document **a1.pdf** (hand-written submissions are not acceptable). You may work on the assignment in groups of 1 or 2, and submit a single assignment for the entire group on MarkUs. You must establish your group well before the due date by submitting an incomplete, or even empty, submission.

background

You will be working on a schema for a database to track covid-19 vaccinations.[1] Vaccines batches are tracked from the factory that produces them. Their arrival in time Canada, and in the province or territory they are to be administered in, are recorded. Minimum and maximum intervals for follow-up doses are also recorded. There is a unique identifier for each vial.

Patients, vaccine administrators and attendants are each recorded, both to follow up on subsequent doses (where required by the manufacturer), and to track vaccine efficacy and safety. Each patient's covid status at the time of vaccination is recorded, and the time of the latest subsequent infection is recorded. Patients are observed by the attendants for at least 15 minutes after vaccination, and any bad reactions are treated and recorded.

relations

- Batch(<u>bID</u>, mID, productionDate, vialCount)
  Vaccine batch *bID*, manufacturer *mID*, was produced on *productionDate*, with *vialCount* vials in this batch.

[1] Details may not be accurate.

- Vial(vID, bID, thawTime, dose_count)
  Vial *vID* from batch *bID* removed from cold storage at *thawTime*, with *dose_ count* doses remaining.[2]

- Manufacturer(mID, name, thawMax, intervalMin, intervalMax)
  Manufacturer *mID*, with company *name*, *thawMax* maximum hours vaccine is usable after being removed from cold storage, *intervalMin* minimum days to second dose, *intervalMax* maximum days to second dose (both zero for a single-dose vaccine).

- Tracking(bID, canadaDate, locationDate, locationName)
  Batch *bID* arrived in Canada on *canadaDate*, shipped to province or territory *locationName* on *locationDate*.

- Vaccination(pID, date, vID, adID, atID, reaction, covidStatus)
  Patient *pID* vaccinated on *date* from vial *vID*. The dose was administered by *adID*, the patient was attended by *atID*. At vaccination time the patient had infection status *covidStatus* and reaction to vaccine *reaction*.

- Patient(pID, latestPositiveTest)
  Patient *pID* had most recent positive Covid-19 test on *latestPositiveTest* (00:00:00, January 1st, 1970 if this never happened).

- Staff(sID, pID, specialty)
  Medical staff *sID* is also patient *pID*, and has medical specialty *speciality*.

our constraints

For each of the following constraints give a one sentence explanation of what the constraint implies, and why it is required.

1. $\Pi_{pID} Staff - \Pi_{pID} Patient = \emptyset$

   sample solution: All staff are also patients, and this reference allows checking on whether they have been recently infected with covid-19

2. $(\Pi_{adID} Vaccination \cup \Pi_{atID} Vaccination) \subseteq \Pi_{sID} Staff$

   sample solution: All those who attend or administer vaccinations are also medical staff, and the reference ensures access to their specialty.

3. $\Pi_{specialty} Staff \subseteq \{'RN', 'RPN', 'MD', 'Pharmacist'\}$

   sample solution: Medical staff have one of four pre-defined specialties, ensuring that only appropriate professions that can do this job.

4. $\Pi_{pID} Vaccination \subseteq \Pi_{pID} Patient$

   sample solution: All those receiving vaccinations are patients, and the reference ensures access to their most recent covid test, if needed.

5. $\Pi_{bID} Vial - \Pi_{bID} Batch = \emptyset$

[2] A timestamp of 00:00:00, January 1st, 1970 is recorded for any events that have not happened (yet).

sample solution: All vials come from some batch, and the reference allows access to when the vial was produced, and (indirectly) to manufacturer information.

6. $\Pi_{covidStatus} Vaccination \subseteq \{'positive', 'negative'\}$

sample solution: Every vaccination patient has either positive or negative covid status, enforcing that this is known at time of vaccination.

7. $\Pi_{reaction} Vaccination \subseteq \{'true', 'false'\}$

sample solution: All vaccination patients either have, or do not have, a bad reaction to the vaccine during the 15 minutes they are attended after the show, ensuring that bad reactions can be followed up in future vaccinations.

8. $\Pi_{mID} Batch \subseteq \Pi_{mID} Manufacturer$

sample solution: All batches reference their manufacturer, ensuring that information on the maximum thaw time and when follow-up doses (if required) should be administered can be accessed.

9. $\Pi_{bID} Tracking - \Pi_{bID} Batch = \emptyset$

sample solution: Vaccine distributed to some territory/province in Canada references the batch it comes from, allowing access to when it was produced and (indirectly) to manufacturer information.

10. **UPDATED 6/2/21** $\Pi_{vID} Vaccination - \Pi_{vID} Vial = \emptyset$

sample solution: Each vaccination uses vaccine from some vial, the reference allows access to information about how long the vaccine has been thawed and how many doses are currently left to be accessed.

queries

Write relational algebra expressions for each of the queries below. You must use notations from this course and operators:

$$\pi, \sigma, \rho, \bowtie, \bowtie_{condition}, \times, \cap, \cup, -, =$$

You may also use constants:

today (for current date) $\emptyset$ (for the empty set)

In your queries pay attention to the following:

- All relations are sets, and you may only use relational algebra operators covered in Chapter 2 of the course text.
- Do not make assumptions that are not enforced by our constraints above, so your queries should work correctly for any database that obeys our schema and constraints.
- Other than constants such as 23 or "Moderna", a select operation only examines values contained in a tuple, not aggregated over an entire column.
- Your selection conditions can use arithmetic operators, such as $+, \leq, \neq, \geq, >, <$. You can use logical operators such as $\vee$, $\wedge$, and $\neg$, and treat dates and numeric attributes as numbers that you can perform arithmetic on.

- Use good variable names and provide lots of comments to explain your intentions.
- Allow the return of multiple tuples if that is appropriate for your query.

There may be a query or queries that cannot be expressed in the relational algebra you have been taught so far, in which case just write "cannot be expressed." The queries below are not in any particular order.

1. Rationale: Let's see how well we're doing.

   (a) **Query:** Find pID of all patients who have received all required doses since the beginning of December 2020.

   sample solution: This query is not expressible in relational algebra. Consider an extremely simple instance of the database where there is only data for a single patient, and a single manufacturer that requires a 2-dose sequence. To decide whether a particular vaccination is the first or second dose of the sequence can require knowing whether its predecessor is a first or second dose, and so on. But relational algebra supports neither recursion nor iteration.

   (b) **Query:** Find the names of all provinces or territories that have used vaccine from every manufacturer in their vaccinations.

   sample solution:

   Tracked territory/province name with a manufacturer whose vaccine they used:

   $S_1 := \Pi_{locationName,name}\,(Tracking \bowtie Batch \bowtie \rho_M Manufacturer \bowtie Vial \bowtie Vaccination)$

   What if every tracked territory/province used every manufacturer's vaccine:

   $$WhatIf := \rho_{locationName,name}\,(Tracking \times Manufacturer)$$

   Tracked territory/province that missed at least one manufacturer:

   $$Missed := \Pi_{locationName}\,(WhatIf - S_1)$$

   Did not miss any Manufacturer

   $$(\Pi_{locationName} S_1 - Missed)$$

2. Rationale: Let's see how badly we're doing.

   (a) **Query:** Find pID of all patients who are still waiting for a subsequent dose more than the maximum number of days recommended by the manufacturer.

   sample solution: This query is not expressible in relational algebra. Consider an extremely simple instance of the database where there is only data for a single patient, and a single manufacturer that requires a 2-dose sequence. To decide whether a particular vaccination is the first or second dose of the sequence can require knowing whether its predecessor is a first or second dose, and so on. But relational algebra supports neither recursion nor iteration.

   (b) **Query**: Find sID of all staff who administered a vaccination from a vial that had thawed longer than recommended by the manufacturer.

   sample solution:

   Administering staff, date of vaccination, and the thawTime of the vial they used, provided the thawing ever happened:

   $$VacVia := \Pi_{adID,date,thawTime,bID}\sigma_{thawTime>0}\,(Vaccine \bowtie Vial)$$

Combine with the limit of how long the vaccine should thaw:

$$VacViaM := \Pi_{adID,date,thawTime,bID,thawMax}\,(VacVia \bowtie Batch \bowtie Manufacturer)$$

Which staff administered expired vaccine?

$$Staff(sID) := \Pi_{adID}\sigma_{date-thawTime>thawMax}\,(VacViaM)$$

(c) **Query:** Find vID of all vials with 4 or fewer doses used by the time they had exceeded the maximum time recommended by the manufacturer after thawing.

sample solution: Vaccinations that did not exceed the maximum time after thawing:

$$V := \Pi_{pID,date,vID}\sigma_{date-thawTime\leq thawMax}\,(Vaccination \bowtie Vial \bowtie Batch \bowtie Manufacturer)$$

Quintuples of vaccination date and pID from same vial that did not exceed the maximum time after thawing:

$$VQ := \sigma_{V1.vID=V2.vID=V3.vID=V4.vID=V5.vID}\,(\rho_{V1}V \times \rho_{V2}V \times \rho_{V3}V \times \rho_{V4}V \times \rho_{V5}V)$$

Vials that had at least 5 distinct vaccinations without exceeding the maximum time after thawing. Distinctness relies on tuple comparison:

$$VQD(vID) := \Pi_{V1.vID}\sigma_{V1.(pID,date)<V2.(pID,date)<V3.(pID,date)<V4.(pID,date)<V5.(pID,date)}VQ$$

Vials that had 4 or fewer vaccinations without exceeding the maximum time after thawing:

$$\Pi_{vID}V - VQD$$

3. Rationale: Trace exposures.

   Query: Staff sID_1 is exposed to covid-positive staff sID_2 if:

   (a) staff sID_2 administered or attended staff sID_1's vaccination,

   (b) staff sID_1 administered or attended staff sID_2's vaccination,

   (c) or if some staff exposed to sID_2 administered or attended sID_1's, or had a vaccination administered or attended by sID_1. vaccination.

   Find sID of all staff exposed to covid-positive staff sID 42.

   sample solution: This query has a recursive structure, and is not expressible in relational algebra.

4. Rationale: Find versatile staff.

   Query: Find all staff who have worked to both administer vaccines and attend patients (not necessarily at the same vaccination).

   sample solution:

$$Result(sID) := \Pi_{adID}\sigma_{V1.adID=V2.atID}\,(\rho_{V1}Vaccination \times \rho_{V2}Vaccination)$$

5. Rationale: Quality control.

   (a) **Query:** Find the staff who gave the most recent Moderna vaccine that had a bad ('true') reaction. Keep ties.

sample solution: For brevity we use natural joins, although we know they are brittle and ideally should be replaced by θ-join.
Moderna vaccine batches, and when they were produced:

$$MB := \Pi_{bID,productionDate}\sigma_{name='Moderna'}\left(Batch \bowtie Manufacturer\right)$$

Moderna batches that are not the most recent:

$$OldMB(bID) := \Pi_{T1.bID}\sigma_{T1.productionDate<T2.productionDate}\left(\rho_{T1}MB \times \rho_{T2}MB\right)$$

Most recent Moderna batch:

$$NewestMB := \Pi_{bID}MB - OldMB$$

Staff who administered vaccinations with a bad reaction of the most recent Moderna batch:

$$\Pi_{adID}\sigma_{reaction='true'}\left(NewestMB \bowtie Vial \bowtie Vaccination\right)$$

(b) **Query:** Find all patients who did not have a positive covid status when they were vaccinated in Ontario, but did have a positive test at some later date (possibly in a different province or territory).[3]

sample solution: We use natural joins for brevity, although we know these are brittle and ought to be replaced by θ-joins.
Batch information for vaccination:

$$VacBatch := \Pi_{pID,date,covidStatus,bID}Vaccination \bowtie Vial$$

Ontario vaccinations with negative covid status:

$$NegVacOnt := \sigma_{locationName='Ontario' \wedge covidStatus='negative'}VacBatch \bowtie Tracking$$

Patients with latest positive covid status after their negative status at a vaccination.

$$\Pi_{pID}\sigma_{latestPositiveTest>date}Patient \bowtie NegVacOnt$$

your constraints

For each of these constraints you should derive a relational algebra expression of the form $R = \emptyset$, where R may be derived in several steps, by assigning intermediate results to a variable. If the constraint cannot be expressed in the relational algebra you have been taught, write "cannot be expressed."

1. No vial is in two different batches.

   sample solution: Assert that there are no vials with the same vID but differing bIDs.

$$\sigma_{V1.vID=V2.vID \wedge V1.bID<>V2.bID}\left(\rho_{V1}Vial \times \rho_{V2}Vial\right) = \emptyset$$

2. No patient receives vaccines from two different manufacturers.

[3] Not that we are advocating inter-provincial travel at this poine.

sample solution: Assert that no pair of vaccinations for the same patient have different manufacturers. This requires several joins.

Find each vaccination patient's corresponding vaccine manufacturer:

$$PM := \Pi_{pID,mID} Vaccination \bowtie Vial \bowtie Batch \bowtie Manufacturer$$

No two vaccinations of the same patient use vaccine from different manufacturers:

$$\sigma_{P1.pID=P2.pID \wedge P1.mID!=P2.mID}\left(\rho_{P1}PM \times \rho_{P2}PM\right) = \emptyset$$

3. No patient is vaccinated with more than two doses.

sample solution: Assert that there is no patient in three distinct vaccinations:

Abbreviate name of Vaccination:

$$V := Vaccination$$

No patient received three vaccinations:

$$\sigma_{(V1.pid=V2.pid=V3.pid) \wedge (V1.date<V2.date<V3.date)}\left(\rho_{V1}V \times \rho_{V2}V \times \rho_{V3}V\right) = \emptyset$$

4. All staff receive at least one vaccination dose before they either administer, or attend, vaccinations.

sample solution: Each staff may be vaccinated several times, and are also likely to administer or attend several times, so we want to show that there are no staff who administer or attend a vaccination before all their own vaccinations.

Staff and date when they administered vaccinations at least as late as some date they personally were vaccinated:

$$\begin{aligned} S_1(adID, date) \quad := \quad & \Pi_{adID,V1.date} \\ & \left(\sigma_{V1.adID=V2.pID \wedge V1.date \geq V2.date}\left(\rho_{V1}Vaccination \times \rho_{V2}Vaccination\right)\right) \end{aligned}$$

Staff who administered a vaccination earlier than all their own vaccinations. This include staff who administered but were never vaccinated themselves:

$$S_2 := \Pi_{adID}\left(\left(\Pi_{adID,date}Vaccination\right) - S_1\right)$$

Staff and date when they attended vaccinations at least as late as some date they personally were vaccinated:

$$\begin{aligned} S_3(atID, date) \quad := \quad & \Pi_{atID,V1.date} \\ & \left(\sigma_{V1.atID=V2.pID \wedge V1.date \geq V2.date}\left(\rho_{V1}Vaccination \times \rho_{V2}Vaccination\right)\right) \end{aligned}$$

Staff who attended a vaccination earlier than all their own vaccinations. This include staff who attended but were never vaccinated themselves:

$$S_4 := \Pi_{atID}\left(\left(\Pi_{atID,date}Vaccination\right) - S_1\right)$$

No staff administered nor attended a vaccination earlier than all their own vaccinations:

$$\rho_{S5(sID)}S2 \cup \rho_{S6(sID)}S4 = \emptyset$$

5. No vaccine is administered before it arrives in some Canadian territory or province.

sample solution: Get tracking information for vaccine used in each vaccination. We used natural joins for brevity, although in week 5 we discussed how these are brittle.

$$VacVialTrack := Vaccination \bowtie Vial \bowtie Tracking$$

Assert that no vaccination occurs before the tracking date for a given batch:

$$\sigma_{Vaccination.date<Tracking.date}VacVialTrack = \emptyset$$

submissions

Submit **a1.pdf** on MarkUs. One submission per group, whether a group is one or two people. You declare a group by submitting an empty, or partial, file, and this should be done well before the due date. You may always replace such a file with a better version, until the due date.

Double check that you have submitted the correct version of your file by downloading it from MarkUs.

marking

We mark your submission for correctness, but also for good form:

- For full marks you should add comments to describe the *data*, rather than *technique*, of your queries. These may help you get part marks if there is a flaw in your query.
- Please use the assignment operator, ":=" for intermediate results.
- Name relations and attributes in a manner that helps the reader remember their intended meaning.
- Format the algebraic expressions with line breaks and formatting that help make the meaning clear.