CSC343 Introduction to Database

Assignment 1: Relational Algebra

Group member: Weizhou Wang 1004421262
Lingwei Sun 1004723276

Explanation of the Constraints:

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

This implies all the pIDs in table Staff must belong to pIDs in table Patient. It is required since a person who is both a staff and a patient is also a patient.

- $(\Pi_{adID} Vaccination \cup \Pi_{atID} Vacination) \subseteq \Pi_{sID} Staff$

This implies that adID and atID in Vaccination must belong to sID in Staff. It is required since only staff can be administrators and attendants of the vaccinations.

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

The specialty that each staff can have is one of the following: Registered Nurse (RN), Registered Practical Nurse (RPN), Doctor of Medicine (MD) or Pharmacist. It is required since only staff who belong to these four specialties are qualified to administer the vaccination and/or attend the patients who get the vaccination.

- $\Pi_{pID} Vaccination \subseteq \Pi_{pID} Patient$

The pID in Vaccination must belong to the pID in Patient. It is required since people who want to get vaccinations must register as a patient first.

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

The bID in table Vial must belong to the bID in table Batch. It is required since every vial comes from a batch.

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

The status that a patient can have for covid at vaccination is one of the following: positive(infected according to test result) or negative(not infected according to test result). It is required as the patient can only either be infected or not be infected.

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

The reaction of a patient after vaccination will be recorded as one of the following: true (patient has a bad reaction) or false (patient does not have a bad reaction). It is required as a patient can only either have a bad reaction or does not have a bad reaction after vaccination.

- $\Pi_{mID} Batch \subseteq \Pi_{mID} Manufacturer$

This implies that every mID in Batch must belong to mIDs in Manufacturer. It is required since every batch of vaccines must be produced by one of the registered manufacturers, otherwise, we cannot make sure whether this vaccine is safe.

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

It implies that the bID in table Tracking must belong to the bID in table Batch. It is required since every batch we are tracking must be one of the batches manufactured.

- $\Pi_{vID} Vaccination - \Pi_{vID} Vial = \emptyset$

This implies the vID in table Vaccination must belong to the vID in table Vial. It is required since the vial the patient vaccinated from must be one of the vials manufactured.

Queries:

1. **Rationale: Let's see how well we're doing.**
   Query: Find pID of all patients who have received all required doses since the beginning of December 2020.

   --**We define: "since the beginning of December 2020" includes "the beginning of December 2020"**

   -- Combine tables to get the manufacturer and intervals for each vaccination

   $VaccWithManu(pID,\ date,\ mID,\ thawTime,\ thawMax,\ intervalMin,\ intervalMax\ )\ :=$

   $\pi_{pID,\ date,\ mID,\ thawTime,\ thawMax,\ intervalMin,\ intervalMax}\ (Vaccination \bowtie Vial \bowtie Batch \bowtie Manufacturer)$

   -- Find vaccinations since the required date, as the superset of all required vaccinations

   $VaccAfter(pID,\ date,\ mID,\ thawTime,\ thawMax,\ intervalMin,\ intervalMax)\ :=$

   $\sigma_{date \geq 2020\text{-}12\text{-}01\ 00:00:00}\ VaccWithManu$

   -- Find all vaccinations using unexpired(thaw time) vials, those are valid vaccinations

   $VaccValid\ :=\ \pi_{pID,\ date,\ mID,\ intervalMin,\ intervalMax}\ \sigma_{(date\ -\ thawTime)\ \leq\ thawMax} VaccAfter$

   -- Find those vaccinations that only need one dose, those are finished

   $OneDose(pID,\ date,\ mID,\ intervalMin,\ intervalMax)\ :=\ \sigma_{intervalMin\ =\ 0\ \wedge\ intervalMax\ =\ 0} VaccValid$

   -- subtract the superset by the finished 1-dose vaccinations to get all two-dose vaccinations

   $TwoDose(pID,\ date,\ mID,\ intervalMin,\ intervalMax)\ :=\ VaccValid\ -\ OneDose$

   -- Pair every two vaccinations for each patient after the date, where the manufacturer should be the same one, and the interval between two shots should be within the recommended interval.

   $TwoDosePair\ (pID,\ T1.date,\ T2.date,\ intervalMin,\ intervalMax)\ :=$

   $\pi_{T1.pID,\ T1.date,\ T2.date,\ T1.intervalMin,\ T2.intervalMax}\ \sigma_{T1.pID\ =\ T2.pID\ \wedge\ T1.mID\ =\ T2.mID\ \wedge\ T1.date\ >\ T2.date}$

   $[\ (\rho_{T1} TwoDose)\ \times\ (\rho_{T2} TwoDose)\ ]$

   $TwoDoseFinish(pID,\ T1.date,\ T2.date)\ :=\ \pi_{pID,\ T1.date,\ T2.date}$

   $\sigma_{\ (T1.date\ -\ T2.date)\ >\ intervalMin\ \wedge\ (T1.date\ -\ T2.date)\ <\ intervalMax}\ TwoDosePair$

   -- Split two dates from the above relation. These are paired vaccinations

   $Vacc1(pID,\ date)\ :=\ \pi_{T1.pID,\ T1.date} TwoDoseFinish$

   $Vacc2(pID,\ date)\ :=\ \pi_{T1.pID,\ T2.date} TwoDoseFinish$

   $PairedVacc(pID,\ date)\ :=\ Vacc1\ \cup\ Vacc2$

   -- Subtract all 2-dose vaccinations after the date by the paired vaccinations and 1-dose vaccinations. Then we get vaccinations where the corresponding patients have not finished vaccinations.

   $NotFinished(pID)\ :=\ \pi_{pID}[(\pi_{pID,\ date} VaccAfter)\ -\ PairedVacc\ -(\pi_{pID,\ date} OneDose)]$

   -- Subtract all patients having vaccinations after the date by the not finished patients to get the answer

   $(\pi_{pID} VaccAfter)\ -\ NotFinished$

   However, after we wrote the above attempt, we found that the above step labelled in purple is wrong. For example, if a patient took vaccinations three times, called V1, V2, and V3, such that all of them are from the same manufacturer, and every combination of two vaccinations are in the recommended interval. Our original goal is to pair them into (V1, V2), and leave V3 alone, such that the patient is still waiting for V4 as a second dose of V3. However, this step would actually pair them into {(V1, V2),(V2, V3),(V1, V3)}, such that all of them are paired and the patient got all required doses, which is obviously wrong.

We thought we needed to count the total number of 2-dose vaccinations a patient took after the date, and classify it into odd/even to distinguish whether the patient got all required doses. But this is impossible using Relational Algebra with the current schema. So this query **cannot be expressed.**

Query: Find the names of all provinces or territories that have used vaccines from every manufacturer in their vaccinations.

-- The mID of All manufacturers

$$AllManufacturers := \pi_{mID} Manufacturer$$

-- The name of all recorded provinces or territories

$$AllLocations := \pi_{locationName} Tracking$$

-- The combination assuming all locations have used vaccines from every manufacturer

$$ShouldHaveBeen(locationName, mID) := AllLocations \times AllManufacturers$$

-- A subset from above, showing those really happens (vaccines really used)

$$LocationWithManu(locationName, mID) := \pi_{locationName, mID}(Vaccination \bowtie Vial \bowtie Batch \bowtie Tracking)$$

-- Subtract to find locations that do not meet the requirement

$$WereNotAlways := ShouldHaveBeen - LocationWithManu$$

-- Subtract the above result to get our final answer

$$(\pi_{locationName} LocationWithManu) - (\pi_{locationName} WereNotAlways)$$

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

   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.

   --For consistency, we did not count vaccines that were thawed longer than the maximum thaw time.

   --We define: "since the beginning of December 2020" includes "the beginning of December 2020"

   -- Combine tables to get the manufacturer and intervals for each vaccination

   $VaccWithManu(pID,\ date,\ mID,\ thawTime,\ thawMax,\ intervalMin,\ intervalMax) :=$

   $\pi_{pID,\ date,\ mID,\ thawTime,\ thawMax,\ intervalMin,\ intervalMax}\ (Vaccination \bowtie Vial \bowtie Batch \bowtie Manufacturer)$

   -- Find all vaccinations using unexpired(thaw time) vials, those are valid vaccinations

   $VaccValid := \pi_{pID,\ date,\ mID,\ intervalMin,\ intervalMax}\ \sigma_{(date - thawTime) \le thawMax} VaccWithManu$

   -- Find those vaccinations that only need one dose, those are finished

   $OneDose(pID,\ date,\ mID,\ intervalMin,\ intervalMax) := \sigma_{intervalMin = 0\ \wedge\ intervalMax = 0} VaccValid$

   -- subtract the superset by the finished 1-dose vaccinations to get all two-dose vaccinations

   $TwoDose(pID,\ date,\ mID,\ intervalMin,\ intervalMax) := VaccValid - OneDose$

   -- Pair every two vaccinations for each patient after the date, where the manufacturer should be the same one, and the interval between two shots should be within the recommended interval.

   $TwoDosePair\ (pID,\ T1.date,\ T2.date,\ intervalMin,\ intervalMax) :=$

   $\pi_{T1.pID,\ T1.date,\ T2.date,\ T1.intervalMin,\ T2.intervalMax}\ \sigma_{T1.pID = T2.pID\ \wedge\ T1.mID = T2.mID\ \wedge\ T1.date > T2.date}$

   $[\ (\rho_{T1} TwoDose) \times (\rho_{T2} TwoDose)\ ]$

   $TwoDoseFinish(pID,\ T1.date,\ T2.date) := \pi_{pID,\ T1.date,\ T2.date}$

   $\sigma_{(T1.date - T2.date) > intervalMin\ \wedge\ (T1.date - T2.date) < intervalMax} TwoDosePair$

   -- Split two dates from the above relation. These are paired vaccinations

   $Vacc1(pID,\ date) := \pi_{T1.pID,\ T1.date} TwoDoseFinish$

   $Vacc2(pID,\ date) := \pi_{T1.pID,\ T2.date} TwoDoseFinish$

   $PairedVacc(pID,\ date) := Vacc1 \cup Vacc2$

   -- Subtract all 2-dose vaccinations after the date by the paired vaccinations and 1-dose vaccinations. Then we get vaccinations where the corresponding patients have not finished vaccinations.

   $NotFinished(pID,\ date) := (\pi_{pID,\ date} VaccValid) - PairedVacc - (\pi_{pID,\ date} OneDose)$

   -- Get the intervalMin, and intervalMax for not finished vaccinations

   $NotFinishedInfo(pID,\ date,\ intervalMin,\ intervalMax) :=$

   $\pi_{pID,\ date,\ intervalMin,\ intervalMax}\ (NotFinished \bowtie VaccWithManu)$

   -- From the above vaccinations, find those that have already waited for the subsequent more than the intervalMax.

   -- The corresponding patients are the answer

   $\pi_{pID}\ \sigma_{(today - date) > intervalMax} NotFinishedInfo$

   Same as Query 1.1, after we wrote the above attempt, we found that the above step labelled in purple is wrong. For example, if a patient took vaccinations three times, called V1, V2, and V3, such that all of them are from the same manufacturer, and every combination of two

vaccinations are in the recommended interval. Our original goal is to pair them into (V1, V2), and leave V3 alone, such that the patient is still waiting for V4 as a second dose of V3. However, this step would actually pair them into {(V1, V2),(V2, V3),(V1, V3)}, such that all of them are paired and the patient got all required doses, which is obviously wrong.

We thought we needed to count the total number of 2-dose vaccinations a patient took after the date, and classify it into odd/even to distinguish whether the patient got all required doses. But this is impossible using Relational Algebra with the current schema. So this query **cannot be expressed.**

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

-- Find the corresponding thaw time and max thaw interval for each vial

$$VialWithThawMax\ (vID,\ thawTime,\ thawMax)\ := \pi_{vID,\ thawTime,\ thawMax}(Vial \bowtie Batch \bowtie Manufacturer)$$

-- Relate the above relation with the vaccination date to get the target staffs' sIDs

$$Answer(sID,\ date,\ thawTime,\ thawMax)\ :=\ \pi_{adID,\ date,\ thawTime,\ thawMax}(Vaccination \bowtie VialWithThawMax)$$

$$\pi_{sID}\ \sigma_{(date\ -\ thawTime)\ >\ thawMax}\ Answer$$

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.

-- Find each vial vID with the maximum hours where the vaccine is usable after removed from cold storage thawMax

$$vailWithThawMax := \pi_{vID,\ thrawMax}\ (Vial \bowtie Batch \bowtie Manufacturer)$$

-- Find each vial vID at vaccination with vaccination date and vial thaw time

$$vialWithTime := \pi_{pID,\ vID,\ date,\ thawTime}\ (Vaccination \bowtie Vial)$$

-- Relate the above relations and Find the vial vID and vaccination date that is used before the time exceed the thrawMax time after thawing

$$vialInfoCombined := \pi_{pID,\ vID,\ thrawMax,\ date,\ thawTime}\ (vialWithTime \bowtie vailWithThawMax)$$

$$vialBeforeExceed := \pi_{pID,\ date,\ vID}\ \sigma_{(date\ -\ thawTime)\ <thrawMax}\ vialInfoCombined$$

-- Get the vaccinations that are not the latest for each vial just before exceeding the max thaw time.

-- If we found multiple vaccinations that are latest, we break the tie by choosing the one with the largest pID as the latest vaccination just before exceeding the max thaw time. (We use the same logic for all the following steps)

$$pairForLoserOnce := (\rho_{v1} vialBeforeExceed) \times (\rho_{v2} vialBeforeExceed)$$

$$UsedAtLeastTwice(pID,\ date,\ vID) := \pi_{v1.pID,\ v1.date,\ v1.vID}$$
$$\sigma_{\ v1.vID = v2.vID \wedge [\ (v1.date<v2.date) \vee (v1.date=v2.date \wedge v1.pID<v2.pID)\ ]}\ pairForLoserOnce$$

-- Get the vaccinations that are not the 1st and 2nd latest for each vial just before exceeding the max thaw time.

$$pairForLoserTwice := (\rho_{v3} UsedAtLeastTwice) \times (\rho_{v4} UsedAtLeastTwice)$$

$$UsedAtLeastTriple(pID,\ date,\ vID) := \pi_{v3.pID,\ v3.date,\ v3.vID}$$
$$\sigma_{\ v3.vID = v4.vID \wedge [\ (v3.date<v4.date) \vee (v3.date=v4.date \wedge v3.pID<v4.pID)\ ]}\ pairForLoserTwice$$

-- Get the vaccinations that are not the 1st, 2nd and 3rd latest for each vial just before exceeding the max thaw time.

$$pairForLoserTriple := (\rho_{v5} UsedAtLeastTriple) \times (\rho_{v6} UsedAtLeastTriple)$$

$$UsedAtLeastFour(pID,\ date,\ vID) := \pi_{v5.pID,\ v5.date,\ v5.vID}$$
$$\sigma_{\ v5.vID = v6.vID \wedge [\ (v5.date<v6.date) \vee (v5.date=v6.date \wedge v5.pID<v6.pID)\ ]}\ pairForLoserTriple$$

-- Get the vaccinations that are not the 1st-4th latest for each vial just before exceeding the max thaw time.

-- The remainings tuples are vials used more than 4 times before expiring

$$pairForLoserFour := (\rho_{v7} UsedAtLeastFour) \times (\rho_{v8} UsedAtLeastFour)$$

$$UsedAtLeastFive\ (vID) := \pi_{v7.vID}$$
$$\sigma_{\ v7.vID = v8.vID \wedge [\ (v7.date<v8.date) \vee (v7.date=v8.date \wedge v7.pID<v8.pID)\ ]}\ pairForLoserFour$$

-- Subtract the UsedAtLeastFive, to get vials used 4 or fewer times before expiring

$$(\pi_{vID} vialBeforeExceed) - UsedAtLeastFive$$

3. **Rationale: Trace exposures.**

Query: Staff sID1 is exposed to covid-positive staff sID2 if:

(a) staff sID2 administered or attended staff sID1's vaccination,

(b) staff sID1 administered or attended staff sID2's vaccination,

(c) or if some staff exposed to sID2 administered or attended sID1's, or had a vaccination administered or attended by sID1. vaccination.

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

-- Find sIDs exposed directly to 42 which belong to type (a)

--Find all exposed patients when sID 42 administered or attended the patients' vaccination.

$$ExposedPatients(pID) := \pi_{pID}\, \sigma_{adID = 42 \vee atID = 42} Vaccination$$

--Find the corresponding sID if the patients are also staffs

$$ExposedStaffA(sID) := \pi_{sID}\, (ExposedPatients \bowtie Staff)$$

-- Find sIDs exposed directly to 42 which belong to type (b)

--Find the pID of sID 42

$$pIDsID42\ (pID) := \pi_{pID}\, \sigma_{sID = 42} Staff$$

--Find the staff who administered sID 42's vaccination

$$ExposedStaffBAD(sID) := \pi_{adID}(\ Vaccination \bowtie pIDsID42)$$

--Find the staff who attended sID 42's vaccination

$$ExposedStaffBAT(sID) := \pi_{atID}\, (Vaccination \bowtie pIDsID42)$$

--Combine them together

$$ExposedStaffB(sID) := ExposedStaffBAD \ \cup\ ExposedStaffBAT$$

--Find sIDs exposed indirectly by 42, directly by staff in type (a) and type (b)

--Find the total exposed staff in (a) and (b)

$$ExposedStaffAB(sID) := ExposedStaffA \ \cup\ ExposedStaffB$$

--Find patients who were administered or attended by exposed staff in (a) and (b)

$$PatientsC(pID) := \pi_{pID}$$

$$[(Vaccination \bowtie_{adID = sID} ExposedStaffAB) \cup (Vaccination \bowtie_{atID = sID} ExposedStaffAB)]$$

--Find the corresponding sID for the staffs who were administered or attended by exposed staffs in (a) and (b)

$$ExposedStaffC(sID) := \pi_{sID}(PatientsC \bowtie Staff)$$

--Find staffs who administered or attended the vaccination of the exposed staff in (a) and (b)

$$CorrespondPatientAB(pID) := \pi_{pID}(ExposedStaffAB \bowtie Staff)$$

$$ExposedStaffDAD(sID) := \pi_{adID}(Vaccination \bowtie CorrespondPatientAB)$$

$$ExposedStaffDAT(sID) := \pi_{atID}(Vaccination \bowtie CorrespondPatientAB)$$

$$ExposedStaffD(sID) := ExposedStaffDAD \cup ExposedStaffDAT$$

-- Combined for answer

$$ExposedStaffAB \cup ExposedStaffC \cup ExposedStaffD$$

The implementation depends on the definition of the question, especially (c). In the above steps, we define that staff infected by sID 42 only count up to the secondary infections. If the question is defined as counting all indirect infectors, then we need to use recursion, which is **inexpressible** using 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).

   -- Find all staff who have administered a vaccination

   $$Administer(sID) := \pi_{adID}Vaccination$$

   -- Find all staff who have attended a patient

   $$Attend(sID) := \pi_{atID}Vaccination$$

   -- Take the intersection

   $$Administer \cap Attend$$

5. **Rationale: Quality control.**

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

-- Find the mID of Moderna vaccine

$$Moderna\ (mID)\ :=\ \pi_{mID}\ \sigma_{name\ =\ 'Moderna'} Manufacturer$$

-- Find the vials produced by Moderna

$$VailByModerna\ (vID,\ productionDate)\ :=\ \pi_{vID,\ productionDate}(Vial \bowtie Batch \bowtie Moderna)$$

-- Find the Vaccinations using vials from Moderna that have a reaction

$$Reaction\ (pID,\ date,\ sID,\ productionDate)$$

$$:=\ \pi_{pID,\ date,\ adID,\ productionDate}\ \sigma_{reaction\ =\ 'true'}\ (Vaccination \bowtie VailByModerna)$$

-- Assume the more recent the date is the larger value attribute date has and Find vaccinations having reactions but the vial is not recently produced

$$NotRecent\ (pID,\ date,\ sID)\ :=$$

$$\pi_{R1.pID,\ R1.date,\ R1.sID}\ \sigma_{R1.productionDate<R2.productionDate}\ [(\rho_{R1} Reaction) \times (\rho_{R2} Reaction)]$$

-- Subtract to get the answer

$$\pi_{sID}\ [\ (\pi_{pID,\ date,\ sID}\ Reaction)\ -\ NotRecent\ ]$$

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).

-- Find all vials that are in Ontario

$$VialOntario\ (vID)\ :=\ \pi_{vID}\ \sigma_{loationName\ =\ 'Ontario'}(Vial \bowtie Tracking)$$

-- Find all patients who did not have a covid status when they had vaccinations from the above vials

$$StatusPatientOntario\ (pID,\ date)\ :=\ \pi_{pID,\ date}\ \sigma_{covidStatus\ =\ 'negative'}(Vaccination \bowtie VialOntario)$$

-- Find patients from above who had a positive test at some later date

$$\pi_{pID}\ \sigma_{latestPositiveTest\ >\ date}(StatusPatientOntario \bowtie Patient)$$

Constraints Expressions:

1. **No vial is in two different batches.**

   -- Find and combine two Vial relations with tuples containing the same vial vID.

   $$CrossVial := \sigma_{Vial1.vID = Vial2.vID} ((\rho_{Vial1} Vial) \times (\rho_{Vial2} Vial))$$

   -- Each tuple with two vials that has the same vID does not have the same batch bID implies 'No vial is in two different batches'.

   $$\sigma_{Vial1.bID \neq Vial2.bID} CrossVial = \emptyset$$

2. **No patient receives vaccines from two different manufacturers.**

   -- Get the batch, manufacturer that a vial belongs to.

   $$VialWithBatch := \pi_{vID,\ bID,\ mID} (Vial \bowtie Batch)$$

   -- Get the manufacturer of the vial each patient vaccinates from

   $$PatientWithManu := \pi_{pID,\ mID} (Vaccination \bowtie VialWithBatch)$$

   -- Combine two relations with tuples containing the same patient pID

   $$PatientAllManu := \sigma_{P1.pID = P2.pID} ((\rho_{P1} PatientWithManu) \times (\rho_{P2} PatientWithManu))$$

   -- Each tuple with the same pID does not have the same mID implies 'No patient receives vaccines from two different manufacturers'.

   $$\sigma_{P1.mID \neq P2.mID} PatientAllManu = \emptyset$$

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

   -- Find patient that vaccinated with at least three doses

   $$AllTriplet := (\rho_{V1} Vaccination) \times (\rho_{V2} Vaccination) \times (\rho_{V3} Vaccination)$$

   $$AtLeastThree := \pi_{V1.pID} (\sigma_{V1.pID=V2.pID \wedge V1.pID=V3.pID \wedge V1.date<V2.date \wedge V2.date<V3.date} AllTriplet)$$

   -- Set the patients that are vaccinated at least three doses to empty implies 'No patient is vaccinated with more than two doses.'

   $$AtLeastThree = \emptyset$$

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

-- Relate the staff sID with the date he/she gets vaccinated

$$StaffVacDate(sID, vacDate) := \pi_{sID, date}(Staff \bowtie Vaccination)$$

-- Find the earliest vaccination time for each staff

$$NotEarly(sID, vacDate) :=$$

$$\pi_{S1.sID, S1.vacDate}\, \sigma_{S1.sID = S2.sID \wedge S1.vacDate > S2.vacDate}\, [(\rho_{S1} StaffVacDate) \times (\rho_{S2} StaffVacDate)]$$

$$EarlyVacDate(sID, vacDate) := StaffVacDate - NotEarly$$

-- Find bad staff, who administered or attended vaccinations before they got the first vaccination.

$$WorkDate := \pi_{sID, date, vacDate}\, \sigma_{sID = adID \vee sID = atID}\, (EarlyVacDate \times Vaccination)$$

$$BadStaff := \pi_{sID}\, \sigma_{vacDate \geq date} WorkDate$$

-- There should be no bad staff

$$BadStaff = \emptyset$$

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

-- Relate the arriving date to a territory/province and the vaccination date of each vial

$$ArrivingDate := \pi_{vID, canadaDate}(Vial \bowtie Tracking)$$

$$TwoDates := \pi_{canadaDate, date}(ArrivingDate \bowtie Vaccination)$$

-- Set the vials with vaccination date earlier than the arrival date to empty implies ‘No vaccine is administered before it arrives in Canadian territory/province’.

$$\sigma_{date < canadaDate} TwoDates = \emptyset$$