Relational Model

Person(hinsurnum, name, city, phone, dateOfBirth, gender, postalCode, streetAddress, registrationDate, category, priority)

DesignatedVaccineLocation(locationName, city, postalCode, streetAddress)

Hospital(locationName) locationName references DesignatedVaccineLocation

Nurse(licenseNumber, nurseName, locationName) locationName references Hospital

VaccinationDates(date,locationName) locationName references DesignatedVaccineLocation

Vaccine(vaccineName, numberOfDoses, waitingPeriod)

Batch(batchNumber,vaccineName, numberOfVials, expiryDate, manufacturingDate, locationName) vaccineName references Vaccine, locationName references DesignatedVaccineLocation

Vial(vialNumber,batchNumber,vaccineName) batchNumber,vaccineName references Batch

Slot(slotNumber,time,date,locationName, hinsurnum, allocateDate, licenseNumber, vialNumber, batchNumber, vaccineName) date,locationName references VaccinationDates, hinsurnum references Person, licenseNumber references Nurse, (vialNumber, batchNumber, vaccineName) references Vial

NurseWorkAssignments(licenseNumber,date,locationName) date,locationName references VaccionationDates

Pending Constraints

The following constraints pertain to the Slot table:

A person's hinsurnum (health insurance number) and allocateDate is updated from NULL only when someone has been offered a specific slot. Otherwise the slot is free to be offered to anyone registered to receive a vaccine.

vialNumber, batchNumber and vaccineName are only updated from NULL once the person assigned to the slot has received their vaccination.

Administrators must manually check that a person receives the same vaccine brand if said vaccine requires more than one dose.

Administrators must manually check that the appropriate waiting time (in weeks) has surpassed before administering subsequent doses to an individual (if the vaccine being administered requires more than one dose).

The nurse licenseNumber value in a slot record may be NULL at the time of the slot's allocation as a nurse's location assignment is not necessarily already made when slots are provided to the registered individuals. If the value is NULL, it must be updated when the vaccination is to be administered.

SQL Queries

a) A SQL query that lists all the available (unassigned to any individual) slots in the vaccination location Jewish General on the day of March 20th.

```
SELECT slotNumber, time, date, locationName
FROM Slot
WHERE locationName = 'Jewish General'
AND date = '2021-03-20'
AND hinsurnum IS NULL
AND allocateDate IS NULL
;
```

```
db2 => SELECT slotNumber, time, date, locationName FROM Slot WHERE locationName = 'Jewish General' AND date =
 '2021-03-20' AND hinsurnum IS NULL AND allocateDate IS NULL;

SLOTNUMBER  TIME     DATE       LOCATIONNAME
----------- -------- ---------- ------------------------------
          2 13:45:00 03/20/2021 Jewish General
          3 14:00:00 03/20/2021 Jewish General

  2 record(s) selected.
```

b) A SQL query to find the expiry date of the vaccine dose that was administered to Jane Doe on Feb 6th, 2021. (You can assume that there is no other person with that name and that this was the only dose of vaccine that person ever received).

```
SELECT expiryDate
FROM Batch
WHERE (batchNumber, vaccineName) IN
    (SELECT batchNumber, vaccineName
    FROM Slot
    WHERE date = '2021-02-06'
    AND hinsurnum IN (
        SELECT hinsurnum
        FROM Person
        WHERE name = 'Jane Doe')
        )
;
```

```
db2 => SELECT expiryDate FROM Batch WHERE (batchNumber, vaccineName) IN (SELECT batchNumber, vaccineName FROM
 Slot WHERE date = '2021-02-06' AND hinsurnum IN (SELECT hinsurnum FROM Person WHERE name = 'Jane Doe'));

EXPIRYDATE
----------
10/18/2022

  1 record(s) selected.
```

c) A SQL query to find the total number of people who were vaccinated in Montreal (defined as received vaccination from a location based on Montreal) on Feb 6th, 2021. Assume that nobody receives two doses of vaccines on the same day.

```
SELECT COUNT(*) AS numvaccinated
FROM DesignatedVaccineLocation
WHERE locationName IN (
    SELECT locationName
    FROM Slot
    WHERE date = '2021-02-06'
    AND vialNumber IS NOT NULL
) AND city = 'Montreal'
;
```

```
db2 => SELECT COUNT(*) AS numvaccinated FROM DesignatedVaccineLocation WHERE locationName IN (SELECT locationName FROM
 Slot WHERE date = '2021-02-06' AND vialNumber IS NOT NULL) AND city = 'Montreal';

NUMVACCINATED
-------------
            2

  1 record(s) selected.
```

d) Write a SQL query that will list the name, phone number and insurance number of those individuals who got one dose of Pfizer-BioNTech vaccine before Feb 1, 2021 and have not yet received another dose of the same vaccine. (Assume people do not get vaccinated by two different brands).

```
SELECT name, phone, hinsurnum
FROM Person
WHERE hinsurnum IN
    (SELECT Slot.hinsurnum
    FROM Slot
    WHERE date < '2021-02-01'
    AND vaccineName = 'Pfizer-BioNTech'
    GROUP BY hinsurnum
    HAVING COUNT(Slot.hinsurnum) < 2
    )
;
```

```
db2 => SELECT name, phone, hinsurnum FROM Person WHERE hinsurnum IN (SELECT Slot.hinsurnum FROM Slot WHERE date <
'2021-02-01' AND vaccineName = 'Pfizer-BioNTech' GROUP BY hinsurnum HAVING COUNT(Slot.hinsurnum) < 2);

NAME                         PHONE                        HINSURNUM
---------------------------- ---------------------------- ------------------------------
Peter Finnigan               514-003-2743                 3410-002-792-DE

  1 record(s) selected.
```

e) A SQL query that lists the number of people who have been given at least one dose of vaccine for each category (Elderly, Teachers, etc.).

```
SELECT sum(case when category = 'Elderly' AND hinsurnum IN
      (SELECT hinsurnum
      FROM Slot
      WHERE vialNumber IS NOT NULL)
then 1 else 0 end) as elderly, sum(case when category = 'Health Care Worker'
AND hinsurnum IN
      (SELECT hinsurnum
      FROM Slot
      WHERE vialNumber IS NOT NULL)
then 1 else 0 end) as healthCareWorkers, sum(case when category = 'Teacher'
AND hinsurnum IN
      (SELECT hinsurnum
      FROM Slot
      WHERE vialNumber IS NOT NULL)
then 1 else 0 end) as teachers, sum(case when category = 'Essential Service
Worker' AND hinsurnum IN
      (SELECT hinsurnum
      FROM Slot
      WHERE vialNumber IS NOT NULL)
then 1 else 0 end) as essentialServiceWorkers, sum(case when category =
'Everybody else' AND hinsurnum IN
      (SELECT hinsurnum
      FROM Slot
      WHERE vialNumber IS NOT NULL)
then 1 else 0 end) as everybodyElse
FROM Person
;
```

```
db2 => SELECT sum(case when category = 'Elderly' AND hinsurnum IN (SELECT hinsurnum FROM Slot WHERE vialNumber IS NOT NULL) then 1 else 0
end) as elderly, sum(case when category = 'Health Care Worker' AND hinsurnum IN (SELECT hinsurnum FROM Slot WHERE vialNumber IS NOT NULL)
then 1 else 0 end) as healthCareWorkers, sum(case when category = 'Teacher' AND hinsurnum IN (SELECT hinsurnum FROM Slot WHERE vialNumber
IS NOT NULL) then 1 else 0 end) as teachers, sum(case when category = 'Essential Service Worker' AND hinsurnum IN (SELECT hinsurnum FROM S
lot WHERE vialNumber IS NOT NULL) then 1 else 0 end) as essentialServiceWorkers, sum(case when category = 'Everybody else' AND hinsurnum I
N (SELECT hinsurnum FROM Slot WHERE vialNumber IS NOT NULL) then 1 else 0 end) as everybodyElse FROM Person;

ELDERLY     HEALTHCAREWORKERS TEACHERS    ESSENTIALSERVICEWORKERS EVERYBODYELSE
----------- ----------------- ----------- ----------------------- -------------
          1                 1           1                       1             1

  1 record(s) selected.
```

Montreal Nurses

Create a view mtlnurses that shows the license number and name of all the nurses, the name of the hospital that (currently) employs them, postal code and street address for nurses employed by the hospitals in the city of Montreal:

```
CREATE VIEW mtlnurses AS
SELECT licenseNumber, nurseName, Nurse.locationName, postalCode,
streetAddress
FROM Nurse
JOIN DesignatedVaccineLocation DVL on Nurse.locationName =
DVL.locationName
WHERE DVL.city = 'Montreal'
;
```

Create the view in DB2:

```
db2 => CREATE VIEW mtlnurses AS SELECT licenseNumber, nurseName, Nurse.locationName, postalCode, streetAddress FROM
 Nurse JOIN DesignatedVaccineLocation DVL on Nurse.locationName = DVL.locationName WHERE DVL.city = 'Montreal';
DB20000I  The SQL command completed successfully.
```

Select everything from the view:

```
db2 => SELECT * FROM mtlnurses;

LICENSENUMBER NURSENAME                     LOCATIONNAME                  POSTALCODE                    STREETADDRESS
------------- ----------------------------- ----------------------------- ----------------------------- ----------------------------------------
        12345 John Mitchell                 Jewish General                H3T 1E2                       3755 Chemin de la Côte-Sainte-Catherine
        13579 Nancy Walker                  Montreal General              H3G 1A4                       1650 Cedar Ave

  2 record(s) selected.
```

Select records where nurses are employed by the Jewish General Hospital:

```
db2 => SELECT * FROM mtlnurses WHERE locationName = 'Jewish General';

------------- ----------------------------- ----------------------------- ----------------------------- ----------------------------------------
        12345 John Mitchell                 Jewish General                H3T 1E2                       3755 Chemin de la Côte-Sainte-Catherine

  1 record(s) selected.
```

Now try inserting a record into the view (license number, name and a hospital name, etc.,) that has valid domain values for these attributes (e.g. a new nurse, but an existing hospital). Observe what happens. Take a screenshot and turn in that along with the explanation of why this happened.

```
db2 => INSERT INTO mtlnurses VALUES (14445, 'Mike M', 'Notre Dame', 'H2L 4M1', '1560 Sherbrooke St E');
DB21034E  The command was processed as an SQL statement because it was not a
valid Command Line Processor command.  During SQL processing it returned:
SQL0150N  The target fullselect, view, typed table, materialized query table,
range-clustered table, or staging table in the INSERT, DELETE, UPDATE, MERGE,
or TRUNCATE statement is a target for which the requested operation is not
permitted.  SQLSTATE=42807
```

The insertion into the mtlnurses view fails because the view itself is not an actual table – it is instead storing data from tables Nurse and DesignatedVaccineLocation. In the mtlnurses view, there is a join between Nurse and DesignatedVaccineLocation. When new values are inserted into mtlnurses, there is no way of knowing which values came from which tables, therefore the command fails. It would be better to insert the new data into the tables directly and then create the view.

Check Constraints

Creating the table Batch in DB2 with the CHECK constraint to ensure that each batch's expiry date is past its manufacturing date:

```
 db2 (cont.) => db2 (cont.) => db2 (cont.) => CREATE TABLE Batch ( batchNumber INT NOT NULL, vaccineName VARCHAR(30) N
OT NULL, numberOfVials INT NOT NULL, expiryDate DATE NOT NULL, manufacturingDate DATE NOT NULL, locationName VARCHAR(3
0) NOT NULL, CHECK (manufacturingDate < expiryDate), PRIMARY KEY (batchNumber, vaccineName), FOREIGN KEY (vaccineName)
 REFERENCES Vaccine(vaccineName) )
DB20000I  The SQL command completed successfully.
```

Attempting to insert a new record into the Batch table where the expiry date is before the manufacturing date and receiving an error:

```
db2 => INSERT INTO Batch VALUES (5, 'Pfizer-BioNTech', 200, '2021-01-20', '2021-01-25', 'Montreal General');
DB21034E  The command was processed as an SQL statement because it was not a
valid Command Line Processor command.  During SQL processing it returned:
SQL0545N  The requested operation is not allowed because a row does not
satisfy the check constraint "ASTAHL2.BATCH.SQL210225141115810".
SQLSTATE=23513
```