

Relational Model:

Assumptions

1. If a person belongs to two categories at the same (both are their highest) priority, we are keeping track of only one (arbitrarily chosen). This is why there is a key constraint from Person to Category.
2. A nurse may work at different locations on the same date as per the ER model. Any further restrictions on that (enforcing that the nurse should not work at more than one location on a given day) should be enforced by the applications if that is required.
3. Vaccine dates and slots are created/populated by the administrators as and when required. It is assumed they take into account any availabilities, etc., before doing that.
4. It is assumed that all Hospitals (where nurses work) could also be vaccination locations. As there is no participation constraint, they are not forced to be providing vaccination facilities immediately.

Restrictions

1. The model has no way of enforcing that the same vaccine (manufacturer) is administered to a given individual for all the required doses.
2. The model has no way of enforcing that the total number of doses administered to an individual conforms to the number of required doses associated with each vaccine.
3. The model cannot enforce that the number of vials associated with a vaccine batch will be within the total number of doses recorded for a particular vaccine batch.
4. The model cannot enforce that a vial that is part of a vaccine batch shipped to a particular location will be recorded as being used for vaccine shot/slot associated with the same vaccination location.
5. The model cannot enforce that a nurse who is assigned to a particular location for a specific day will be recorded as administering vaccine shots in a slot associated with the same vaccination location for that day.
6. The model cannot enforce that a vial used for a vaccine shot at a location belongs to a batch recorded as being shipped to that location.
7. The model will allow the storage of vial information although it is not yet administered to anyone.

Relational Translation

1. Category(cname, prioritynum)
2. Person(hinsurnum, name,gender,dob,phone,city,postalcd,streetaddr,regdate, cname)
cname references Category
3. Vaccine(vname, waitperiod,doses)
4. VaccLocation(locname, lcity,lpostalcd,lstreetaddr)
5. VaccineBatch(vname,batchno, mfgdate,expydate,numdoses, locname) vname
references Vaccine, locname references VaccLocation

6. Vial(vname,batchno,vialid) vname, batchno references VaccineBatch
7. Hospital(locname) locname references VaccLocation
8. Nurse(licensenum, name, locname) locname references Hospital
9. VaccDates(locname,vdate) locname references VaccLocation
10. VaccSlot(locname,vdate,vtime,vslot, hinsurnum, asgndate, licensenum, vname, batchno, vialid) locname, vdate references VaccDates, licensenum references Nurse, hinsurnum references Person, (vname, batchno, vialid) references Vial.
11. NurseAssignments(licensenum, locname, vdate) licensenum references Nurse, (locname, vdate) references VaccDates

To be true to ER, the “used” relationship should be merged to Vial and not VaccSlot. But we can anyways only preserve one of the two key constraints in the relational model. By putting it in the VaccSlot, it might be easier to lookup some of the information (like the brand of vaccine used for a particular person).

Pending Constraint

1. The model cannot require the same vaccine to be administered to the same person.
2. The model cannot enforce the vials number be smaller or equal to the entered number of doses.
3. The model cannot enforce the vial to be assigned where its batch is sent to.
4. The model cannot enforce the nurse assigned to a location impossible to appear in reality with different slot and location.

SQL Queries

a.

```
SELECT * FROM Vaccslot WHERE hinsurnum is NULL AND locname = 'Jewish General' AND vdate = '2021-03-20';
```

db2 => SELECT * FROM Vaccslot WHERE hinsurnum is NULL AND locname = 'Jewish General' AND vdate = '2021-03-20';

LOCNAME	VDATE	VTIME	VSLLOT	HINSURNUM	ASGNDATE	LICENSENUM	VNAME	BATCHNO	VIALID
Jewish General	03/20/2021	12:00:22	1	--	--	2834298	Moderna	28322	75677
Jewish General	03/20/2021	15:00:22	1	--	--	2834296	Pfizer-BioNTech	28321	78658

2 record(s) selected.

b.

```
SELECT p.name, v.asgndate, vb.expirydate FROM VaccSlot v LEFT JOIN Person p ON v.hinsurnum = p.hinsurnum LEFT JOIN VaccineBatch vb ON vb.vname = v.vname AND vb.batchno = v.batchno WHERE p.name = 'Jane Doe' AND v.asgndate = '2021-02-06';
```

db2 => SELECT p.name, v.asgndate, vb.expirydate FROM VaccSlot v LEFT JOIN Person p ON v.hinsurnum = p.hinsurnum LEFT JOIN VaccineBatch vb ON vb.vname = v.vname AND vb.batchno = v.batchno WHERE p.name = 'Jane Doe' AND v.asgndate = '2021-02-06';

NAME	ASGNDATE	EXPIRYDATE
Jane Doe	02/06/2021	01/09/2022

1 record(s) selected.

c.

```
SELECT COUNT(*) FROM VaccSlot v LEFT JOIN VaccLocation n ON v.locname = n.locname WHERE n.lcity = 'Montreal' AND v.asgndate = '2021-02-06';
```

db2 => SELECT COUNT(*) FROM VaccSlot v LEFT JOIN VaccLocation n ON v.locname = n.locname WHERE n.lcity = 'Montreal' AND v.asgndate = '2021-02-06';

COUNT(*)
1

1 record(s) selected.

d.

```
SELECT name,phone,hinsurnum FROM Person WHERE hinsurnum in (SELECT P.hinsurnum FROM (SELECT hinsurnum FROM VaccSlot WHERE vname = 'Pfizer-BioNTech' AND asgndate <'2021-02-01' AND hinsurnum is not NULL) P GROUP BY P.hinsurnum HAVING COUNT(P.hinsurnum)<2);
```

```
db2 => SELECT name,phone,hinsurnum FROM Person WHERE hinsurnum in (SELECT P.hinsurnum FROM (SELECT hinsurnum FROM VaccSlot WHERE vname = 'Pfizer-BioNTech' AND asgndate <'2021-02-01' AND hinsurnum is not NULL) P GROUP BY P.hinsurnum HAVING COUNT(P.hinsurnum)<2);
```

NAME	PHONE	HINSURNUM
Ethan	416-4464267	2012832396

1 record(s) selected.

e.

```
SELECT c.cname ,COUNT(p.hinsurnum) FROM VaccSlot v LEFT JOIN Person p ON v.hinsurnum = p.hinsurnum LEFT JOIN Category c ON p.cname = c.cname GROUP BY c.cname HAVING COUNT(p.hinsurnum)>0;
```

```
db2 => SELECT c.cname ,COUNT(p.hinsurnum) FROM VaccSlot v LEFT JOIN Person p ON v.hinsurnum = p.hinsurnum LEFT JOIN Category c ON p.cname = c.cname GROUP BY c.cname HAVING COUNT(p.hinsurnum)>0;
```

CNAME	2
Children	1
Elderly	1
Healthcare Workers	1
Normal People	1
Teachers	1

5 record(s) selected.

Montreal Nurse

CREATION:

```
CREATE VIEW mtlnurses AS SELECT N.licenseum, N.name, N.locname AS Hospital, VA.lpostalcd, VA.lstreetaddr FROM Nurse N LEFT JOIN
```

```
NurseAssignments NA ON N.licenseum = NA.licenseum LEFT JOIN VaccLocation VA ON NA.locname = VA.locname WHERE VA.lcity = 'Montreal';
```

SELECT ALL:

```
SELECT * FROM mtlnurses LIMIT 5;
```

SELECT JEWISH GERERAL ONLY:

```
SELECT * FROM mtlnurses WHERE Hospital = 'Jewish General' LIMIT 5;
```

INSERTING:

```
INSERT INTO mtlnurses(licenseum,name,Hospital,lpostalcd,lstreetaddr) VALUES(2833290,'Tom','Central park','H3B1B4','130 City Councillor Street');
```

```
db2 => CREATE VIEW mtlnurses AS SELECT N.licenseum, N.name, N.locname AS Hospital, VA.lpostalcd, VA.lstreetaddr FROM Nurse N LEFT JOIN NurseAssignments NA ON N.licenseum = NA.licenseum LEFT JOIN VaccLocation VA ON NA.locname = VA.locname WHERE VA.lcity = 'Montreal';
DB20000I The SQL command completed successfully.
```

```
db2 => SELECT * FROM mtlnurses LIMIT 5;
```

LICENSENUM	NAME	HOSPITAL	LPOSTALCD	LSTREETADDR
2834294	Jane	Central park	H3B1B4	130 City Councillor Street
2834295	Yori	MTL Church	H3B1B5	131 City Councillor Street
2834298	Coco	Jewish General	H3B1B6	45 People Councillor Street

3 record(s) selected.

```
db2 => SELECT * FROM mtlnurses WHERE Hospital = 'Jewish General' LIMIT 5;
```

LICENSENUM	NAME	HOSPITAL	LPOSTALCD	LSTREETADDR
2834298	Coco	Jewish General	H3B1B6	45 People Councillor Street

1 record(s) selected.

```
db2 => INSERT INTO mtlnurses(licenseum,name,Hospital,lpostalcd,lstreetaddr) VALUES(2833290,'Tom','Central park','H3B1B4','130 City Councillor Street');
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
```

Check Constraints

```
ALTER TABLE VaccineBatch ADD CONSTRAINT comdate CHECK (mfgdate<expdydate);
```

```
INSERT INTO vaccinebatch VALUES ('Pfizer-BioNTech', 28320, '2021-01-23', '2020-01-23', '560', 'Central park');
```

Run successfully:

```
db2 => ALTER TABLE VaccineBatch ADD CONSTRAINT comdate CHECK (mfgdate<expdydate);
DB20000I The SQL command completed successfully.
```

Wrong entry:

```
db2 => INSERT INTO vaccinebatch VALUES ('Pfizer-BioNTech', 28320, '2021-01-23', '2020-01-23', '560', 'Central park');
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 "YXU283.VACCINEBATCH.COMDATE". SQLSTATE=23513
```