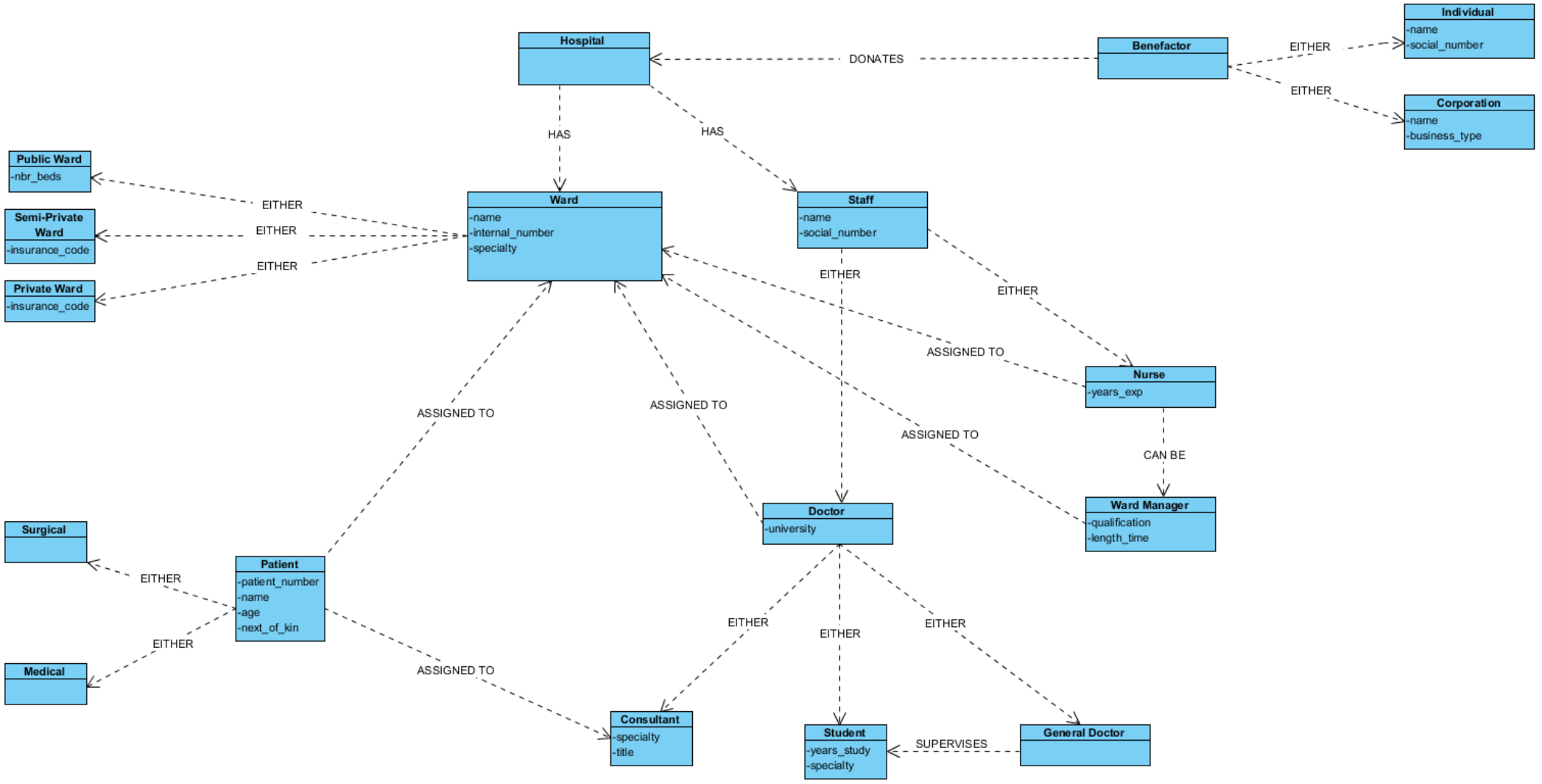
Assignment 1

Managing Big Data

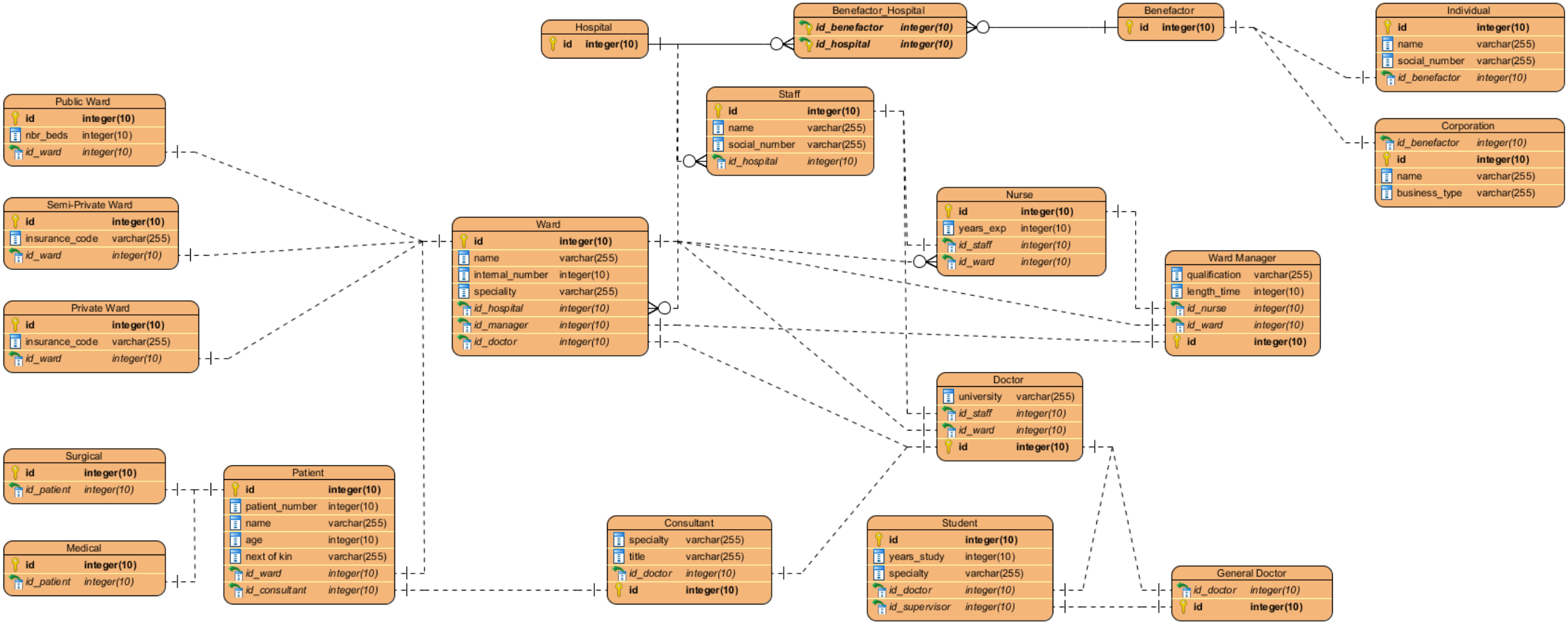
Galdan MOULINNEUF – 2927686

Part 1 – ERD and Logical Schema

a) ERD



b) Logical Schema



Part 2

a)

```
SELECT c.first_name, c.last_name
FROM Customer c
JOIN Subscription s ON s.CustomerID = c.ID
JOIN FitnessClass fc ON fc.Id = s.ClassID
WHERE fc.ClassName = 'Beginner Sumba';
```

b)

```
SELECT *
FROM FitnessClass fc
JOIN Instructor i ON i.Id = fc.InstructorId
WHERE i.FirstName = 'Fiona' AND i.LastName = 'Smith' AND fc.Cost = 7;
```

c)

d)

```
SELECT i.Id, i.LastName, AVG(fc.Cost)
FROM Instructor i
JOIN FitnessClass fc ON fc.InstructorId = i.Id
GROUP BY i.Id;
```

e)

```
SELECT fc.Id, fc.ClassName
FROM FitnessClass fc
HAVING fc.Cost > (MIN(fc.Cost) + 0.5) ;
```

f)

Part 3

To normalize the database, we will go through 4 steps :

1NF :

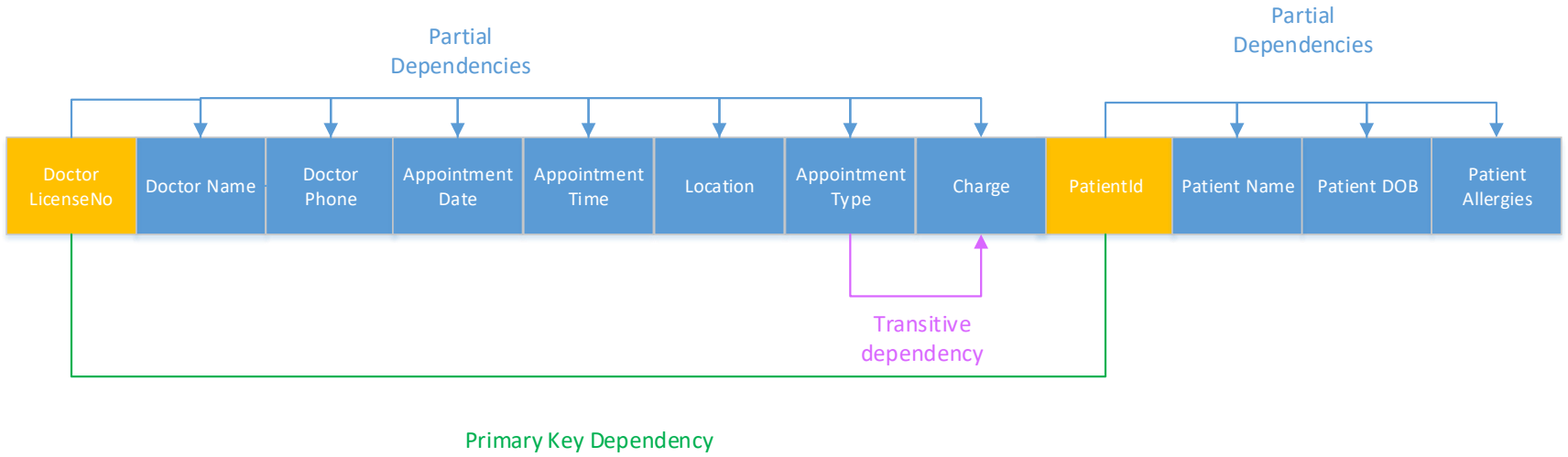
The first step is to check if there is repeting groups and separate them. There are none in the database.

Then, we need to check if there are non-atomics values and separate them as well. There is none in the database.

Then, for each group created, we need to create columns that will be our primary keys to identify an entity. We can distinguish 2 entities :

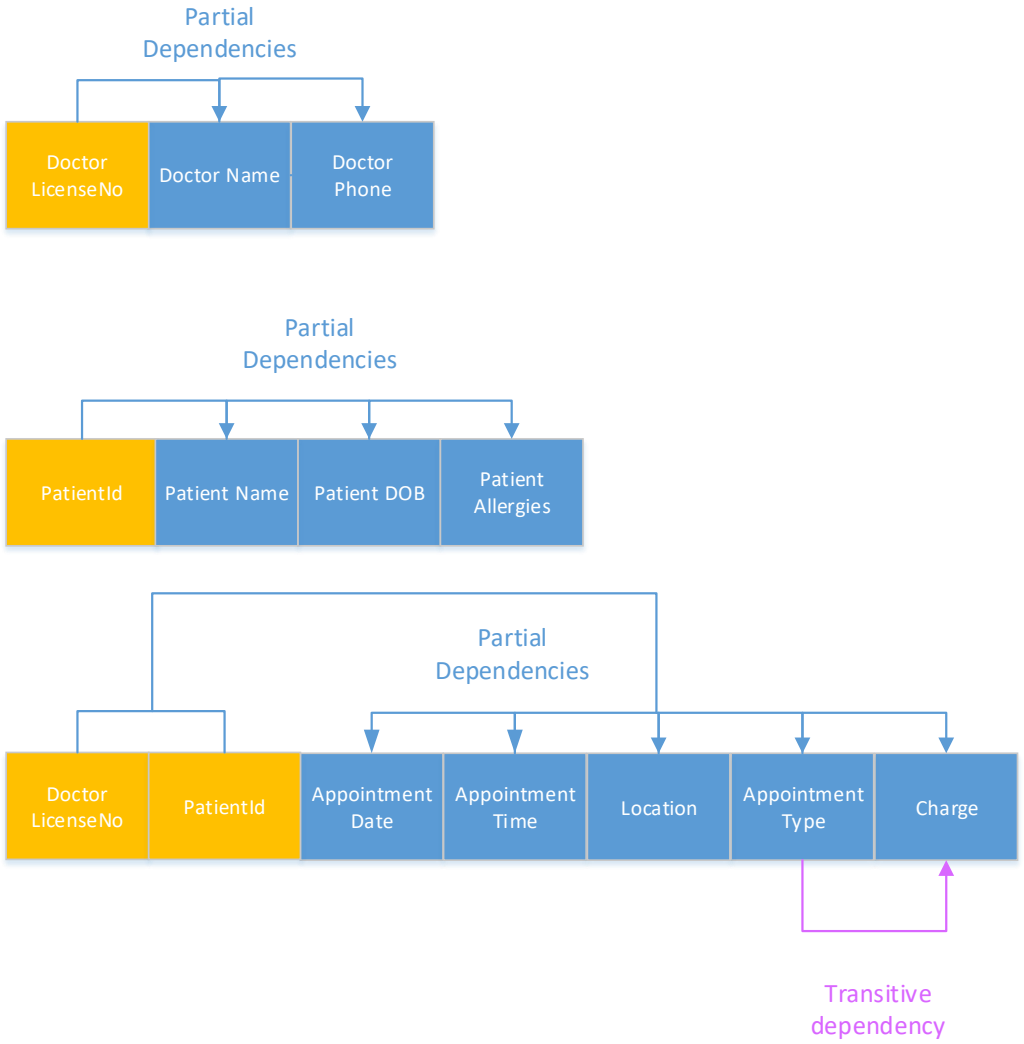
- Doctor
- Patient

Doctors already have a primary key which is « Doctor LicenseNo », same goes for Patients with « PatientId ».



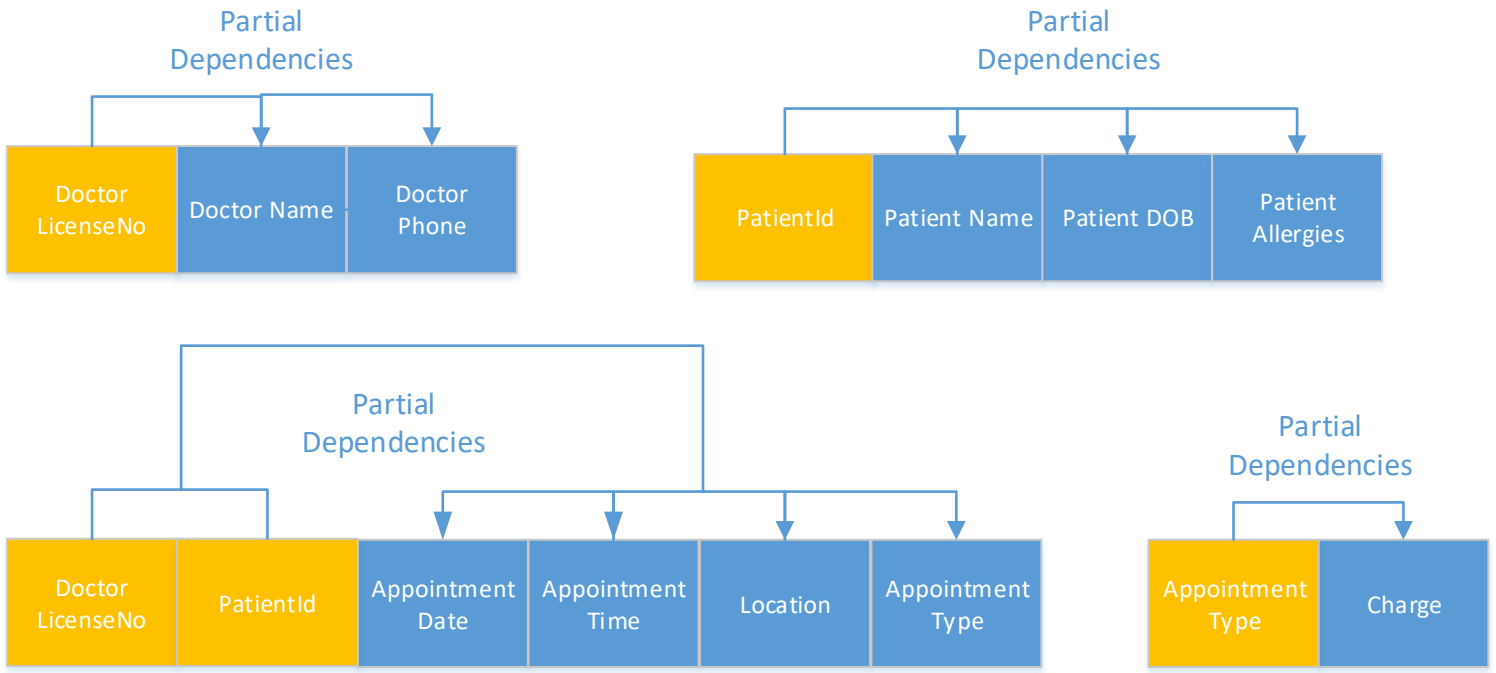
2NF :

The second step will be to divide this table into multiple tables. We already saw that there was 2 entities and we need a table to link them. So there will be a table for the doctors, a table for patients and a table for appointments.



3NF :

Third step consists of removing « Transitive dependencies ». Indeed, if there is a transitive dependency, it can be transformed into a table. The « Appointment Type » and « Charge » columns can be turned into a table of « Jobs » that for each « Job » (the appointment type) there will be a corresponding « Charge ».



BCNF :

The last step is not relevant here because the 3NF is already in the best for mit can be.