http://flip2.engr.oregonstate.edu:19952/

Project step 5 Draft

a) Project Outline and Database Outline - Updated Version:

Overview

Introduction Kais works at a pharmacy right now, so we decided to create a DB for a pharmacy website. There are four entities which will be generated which are customers, drugs(medicine), pharmaceutical orders, and a sales log.

People cannot guarantee that they will be healthy all their lives. Sometimes people will buy medicines in pharmacies because they are sick or in order to prevent diseases. Other patients are regular patients and get refills on maintenance medications monthly. As such it is important to have a database to track all patients and their medications, as well as a log of transactions at the pharmacy.

Problems to be solved

Collecting patient information in order to have contact and allergy information. Patients will likely visit the pharmacy more than once a year, so it is important to have their contact information and allergy history.

Keeping track of the drugs available at the pharmacy. The database will allow users to lookup drugs and get all relevant information, name, strength, national drug code, as well as on hand inventory. This way users will know which drugs to choose when filling prescriptions for patients.

The prescription and sales log will allow users to lookup which prescriptions have been ordered, as well as its current status (finished/not finished), as well as which transactions have been made, and which prescription orders were in those transactions.

Customers purchase what they want from Pharmacy, so that the Pharmacy is between customers and drugs. If medicines are in short supply, pharmacies can connect with other pharmacies to supplement products. Which means the pharmacy will contain address, name, ID as attributes.

Outline(entities)

Patients: records users' basic information,
ID: INT not NULL AUTO_INCREMENT, PK
First_Name: VARCHAR, not NULL
Last_Name: VARCHAR, not NULL
DOB: DATE, not NULL

Gender: CHAR
Address: VARCHAR
Email: VARCHAR

Drugs: records drugs' basic information
NDC: INT UNIQUE, PK
Name: VARCHAR, not NULL
Pharmacy: FK REFERENCES Pharmacy(DEA)
Strength: INT, not NULL
Price: FLOAT
Inventory: INT

Pharmacy: record pharmacy's basic information
Name: VARCHAR, not NULL
Address: VARCHAR, not NULL
DEA id: INT, UNIQUE not NULL, PK
Phone: VARCHAR, UNIQUE
Fax: VARCHAR, UNIQUE

Orders: records users' purchase information
Order_ID: ID: INT not NULL, PK
RX_ID: FK REFERENCES Prescriptions(RX)
Time: DATE, not NULL
Status: CHAR, not NULL

Prescriptions: Records each prescription to a unique number with a patient and drug
RX: INT not NULL AUTO_INCREMENT, PK
Pharmacy: FK REFERENCES Pharmacy(DEA)
Patient: FK REFERENCES Patient(ID)
Drug: FK REFERENCES Drugs(NDC)
Price: FLOAT

Sales: records pharmacy's sales information
Sale ID: INT not NULL, PK
Pharmacy: FK REFERENCES Pharmacy(DEA)
Order: FK REFERENCES Orders(Order_ID)
Name: VARCHAR, not NULL
DOB: date, not NULL
Time: DATE, not NULL

Relationship:

Order to drugs: many to many. Everyone can create an order for what they want. Each order can have many drugs in it, and each drug can be attached to many orders.

Order to Sales: one and only one to many. A sale may have many orders, but orders can only be a part of one transaction, so that the order can then be looked at as sold.

Order to Patients: zero or many to zero or many. Patients may have no orders, while each order may have many patients or be empty.

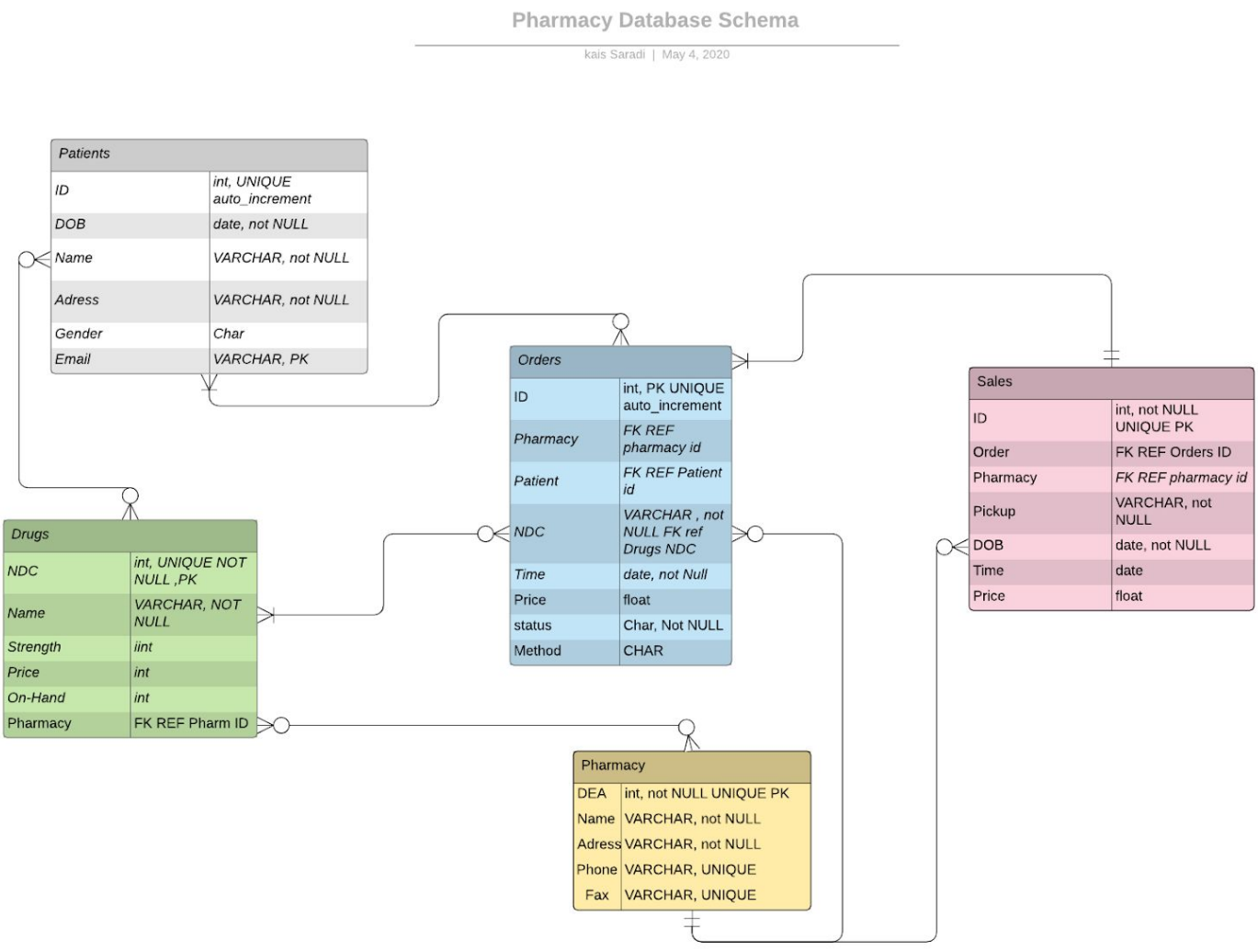
Drugs to Patients: Zero or Many to zero or many. Each patient may have many drugs they have ordered, and each drug can be attached to many patients, or none.

Orders to Pharmacy: one and only one to none or many. Each pharmacy will have none or many orders, but each order will be at exactly one pharmacy, so that the original order location can be found.

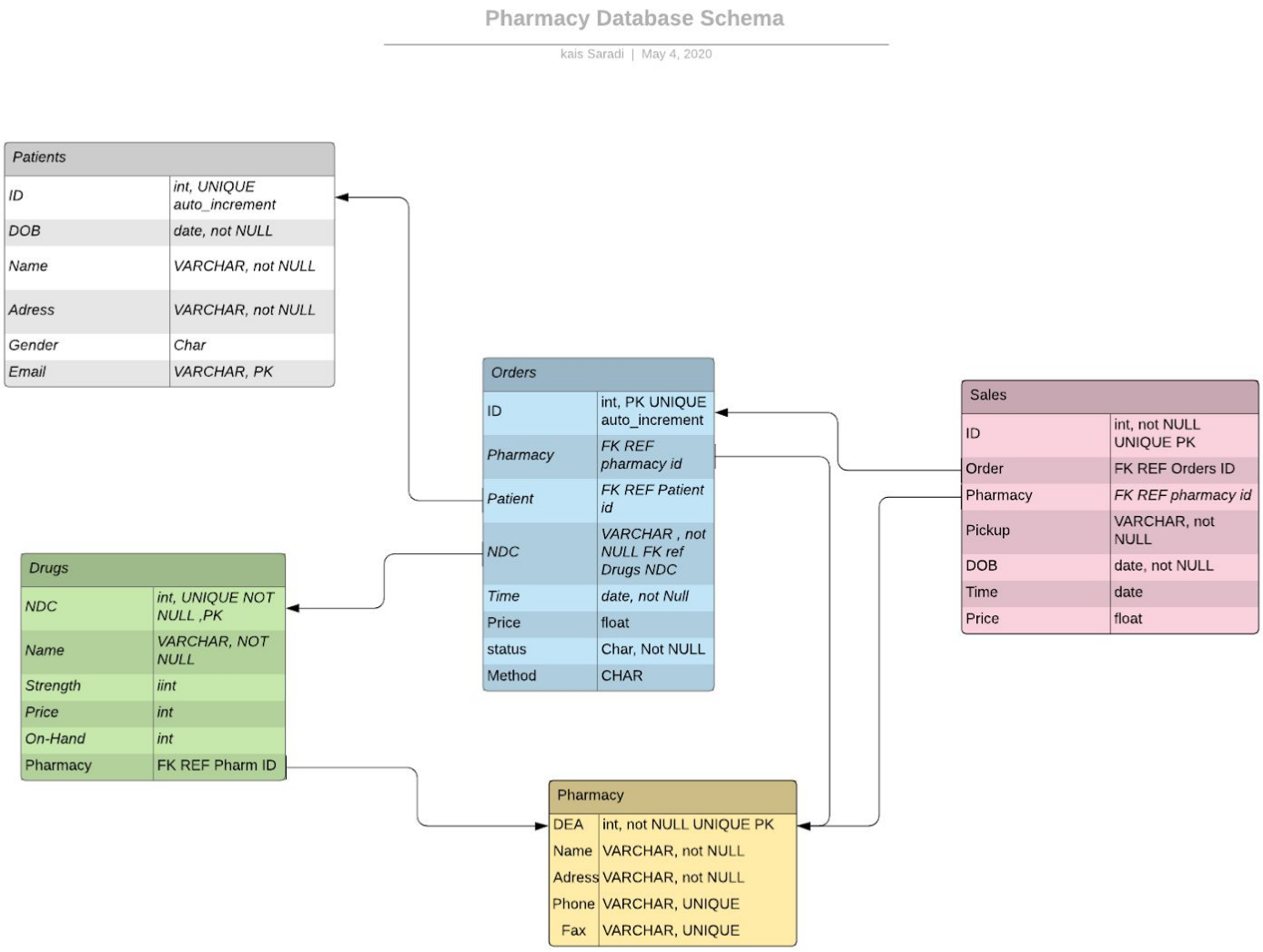
Sales to Pharmacy: one and only one to none or many. Each pharmacy will have no or many sales, but each sale will be at exactly one pharmacy, so that the pickup location can be found.

Drugs to Pharmacy: none or many to none or many: Each drug may be listed at none or many pharmacies, while each pharmacy may have no or many drugs.

Entity-Relationship Diagram:



Schema:



b) Fixes based on the feedback

Fixed DDQ file

Add foreign key in drug table reference to pharmacy

Add foreign key in orders table reference to prescription
Add foreign key in prescription table reference to drug
Add foreign key in prescription table reference to patient
Add foreign key in sales table reference to orders

Add commands for some attributes let them auto increase.