CP363 – Database I: Group Project

Inpatient Pharmacy Inventory Management System



Group #30: Daniel Nguyen, Melina De Iuliis, Daniel Faseyi, Dharmik Patel, Spencer McLean, Jacob Schwartz, Jagnoor Nijjar

2022-03-30

# **Table of Contents:**

Abstract ………………………………………………………………2

Problem Definition ………………………………………………………………2

Requirements ………………………………………………………………3

Patient ………………………………………………………………3

Doctor ………………………………………………………………3

Prescription ………………………………………………………………3

Drug Item ………………………………………………………………3

Drug Pharmacy ………………………………………………………………3

Inpatient Pharmacy ………………………………………………………………4

Pharma Company ………………………………………………………………4

Contract ………………………………………………………………4

ER Modeling ………………………………………………………………5

Dependencies………………………………………………………………6

Normalization ……………………………………………………………… 6

Conclusion ………………………………………………………………7

Abstract

The use of computerized pharmacy inventory databases can be implemented to improve patient care and decrease wait times at the hospital and pharmacy. This project helps us see the different perspectives on hospital systems by following the path of medications. Upon creating this database, we decided to complete it under a few assumptions, such as;

* drugs will be covered through the patient's insurance, meaning the transactions will not be tracked
* inpatient pharmacy will always have a stable supply of drugs, supplied by the manufacturing company through a contract
* the pharmacy only dispenses drugs to patients already registered with the hospital database

This project will not only allow us to look deeper into the structure of pharmaceutical databases, but will allow us to apply our knowledge based on what we have learned in this course and create our own functional database.

Problem Definition

Pharmacies have been running off databases for a long period of time now. It is crucial to make sure they are efficient and easy to handle while making sure all the relevant information is being recorded and easy to find. The major issue with most medical pharmacy databases is that they keep track of inventory through manual counting. They are also very difficult to navigate , hard to update information and tasks are often redundant. Our Inpatient pharmacy inventory management system would allow the collection of inventory items into a database where they can be properly tracked. This can include scanning the medication barcodes which would be updated on the database system we propose. The new database collection will allow proper inventory tracking, help the department maintain a more suitable level of inventory,and would improve patient care.

Requirements

**PATIENT**

The pharmacy dispenses drugs to the patient. Patients are already registered into the hospital system with basic information such as; first name, last name, address, contact number, room number, and is uniquely identified by ‘patient\_id’ integer.

**DOCTOR**

Doctors are also already registered in the system, and is a mandatory entity for issuing prescriptions to patients. They have a unique ‘doctor\_id’ identifier which contains basic information such as their first name, last name, contact phone number, as well as their speciality.

**PRESCRIPTION**

Prescriptions are required in order to prove validity of prescription, and update inventory when fulfilling prescription orders. Uniquely identified through “prescription\_id”, the prescription entity stores crucial information, such as; the name of the drug to be prescribed, and the identity number of that prescribed drug, the strength of the drug, number of refills needed, special instructions the consumer of the drug may need to know, the date the prescription was issued to the patient, as well as the identification of the doctor who prescribed that drug.

**DRUG\_ITEM**

Each prescription entity corresponds with one drug\_item. The drug\_item entity is used to uniquely identify each drug in the system, and contains attributes such as; trade name of the drug, strength of the drug, expiry date, a drug description (i.e., safe administration amounts, when it is most effective to use, etc.) and the identification number of the manufacturer.

**DRUG\_PHARMACY**

The drug\_pharmacy is one of the most important entities since it allows the inpatient pharmacy to keep track of the quantity of a specific drug that the pharmacy has on hand in real time. Both the pharmacy’s identification number and drug identification number are required in order to determine the inventory of a specific inpatient pharmacy for a prescribed drug.

**INPATIENT\_PHARMACY**

The inpatient\_pharmacy – uniquely identified by pharmacy\_ID, is used to indicate the pharmacy’s name, address, and contact information. It is a mandatory entity used to connect the pharmacy with its inventory as well as provide a stable supply of drugs through the contract.

**PHARMA\_COMPANY**

Pharma\_company is the entity used to identify the pharmaceutical company manufacturing and supplying the drugs that the pharmacy will be dispensing to the patients. Pharma companies usually contain a strong database of information about the medications they may produce and or use. This entity would be used to hold relative contact information such as the manufacturer's name, address, and their contact number.

**CONTRACT**

A contract is an agreement between two or more parties. In this case it would be between the inpatient pharmacy and drug manufacturer, the contract is used to view the terms and or conditions the pharmacy may have with their relation to the drug manufacturer. This would be identified by the contract\_id containing information such as the start date, the contract period which would indicate the length of time in which the manufacturer is supplying drugs to the inpatient pharmacy, the identification of the pharmacy as well as the identification of the manufacturer.

ER Modeling

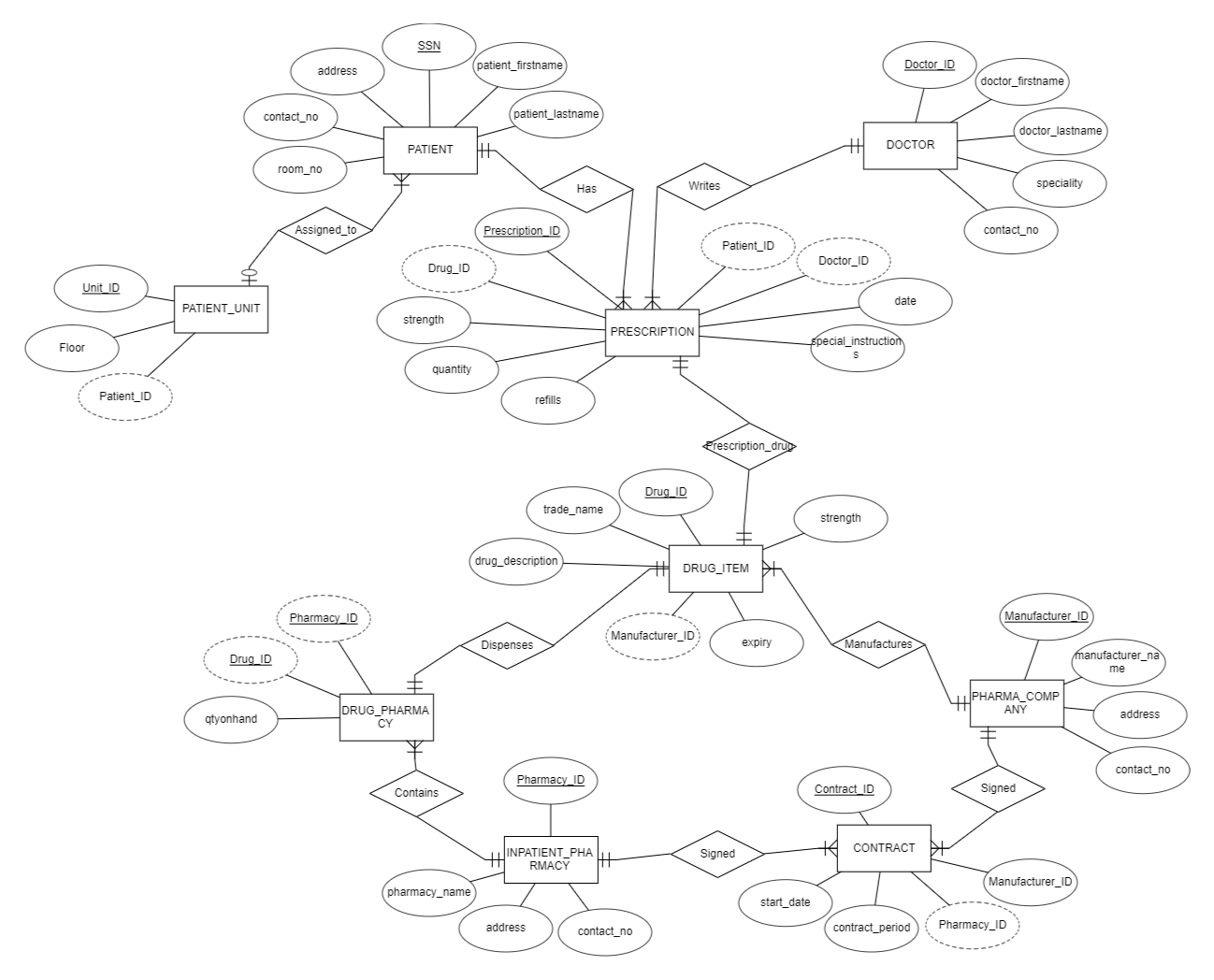
****

Figure 1. Entity-relationship diagram of inpatient pharmacy inventory management system.

As previously mentioned, the pharmacy database keeps records of drugs being dispensed from the pharmacy to the patient. In short, drugs are dispensed to the patient from the pharmacy through a prescription, and the pharmacy maintains its supply of drugs through a contract with a pharmaceutical company. As displayed in our entity-relationship diagram, the following relations are identified:

1. A single PATIENT may be assigned to one PATIENT\_UNIT, but one PATIENT\_UNIT can contain multiple PATIENTS. Thus, the relation between PATIENT\_UNIT and PATIENT is 1:N
2. A single PATIENT can have multiple PRESCRIPTIONS, thus the relation between them is 1:N
3. A single PRESCRIPTION is written by a single DOCTOR, but one DOCTOR can write many PRESCRIPTIONS. Thus, the relation between DOCTOR and PRESCRIPTION is 1:N
4. A single PRESCRIPTION prescribes a patient entity with a single DRUG\_ITEM, thus the relationship between PRESCRIPTION and DRUG\_ITEM is 1:1
5. A single PHARMA\_COMPANY can manufacture a variety of DRUG\_ITEM entities, thus the relationship is 1:N
6. A single PHARMA\_COMPANY can have multiple CONTRACTS with various pharmacies thus the relation between them is 1:N
7. A single INPATIENT\_PHARMACY can also have multiple CONTRACTS with various pharmaceutical manufacturers, thus the relation between them is 1:N
8. A single INPATIENT\_PHARMACY can have multiple DRUG\_PHARMACY entities to keep track of inventory for each individual drug item, thus the relation between them is 1:N
9. One DRUG\_PHARMACY entity corresponds to one DRUG\_ITEM thus the relationship between them is 1:1

Dependencies

**Entity:** Patient

**FD:** Patient\_id → ( Patient\_firstName, Patient\_lastName, Patient\_address, Patient\_contact, Patient\_room )

**Entity:** Doctor

**FD:** Doctor\_id → ( Doctor\_firstName, Doctor\_lastName, Doctor\_specialty, Doctor\_contact)

**Entity:** Prescription

**FD:** Prescription\_id → ( Drug\_id, Strength, Quantity, Refills, Patient\_id, Doctor\_id, date, instructions )

**Entity:** Drug\_Item

**FD:** Drug\_id→ ( Trade\_name, Description, Strength, Expiry, Manufacturer\_id)

**Entity:** Patient\_unit

**FD:** Unit\_id → ( Patient\_id, floor)

**Entity:** Drug\_pharmacy

**FD:** Pharmacy\_id, Drug\_id → ( Quantity\_on\_hand)

**Entity:** Inpatient\_pharmacy

**FD:** Pharmacy\_id → ( Pharmacy\_name, Pharmacy\_address, Pharmacy\_contact)

**Entity:** Contract

**FD:** Contract\_id → ( Start\_date, Contract\_period, Pharmacy\_id, Manufacturer\_id)

**Entity:** Pharma\_company

**FD:** Manufactuering\_company\_id → ( Company\_name, Company\_address, Company\_contact)

Normalization

All entities in our database satisfy the criteria for 3NF. None contain multi-value attributes, satisfying the criteria for 1NF. Along with being 1NF none contain any partial dependencies, satisfying the criteria for 2NF. As well, none contain any transitive dependencies which further pushes the entities past the criteria for 3NF.

An example of how we got all entities to 3NF is the drug\_pharamacy entity. Initially, we had the qtyonhand attribute from drug\_pharmacy within the drug\_item entity. Doing this however, would have required the pharmacy\_id attribute to be within drug\_item as well to distinguish the quantity within a specific pharmacy within the hospital. This would have led to a slew of partial dependencies in which only the qtyonhand attribute would depend on pharmacy\_id and drug\_id while all other attributes within drug\_item would only depend on drug\_id. We resolved this issue by splitting the attributes across two tables: drug\_item and drug\_pharmacy. Drug\_pharmacy contains drug\_id and pharmacy\_id as primary and foreign keys, and qtyonhand. All other attributes remained drug\_item along with drug\_id.

Conclusion

Creating the ‘Inpatient Pharmacy Inventory Management System’ was a great learning experience for implementing a DBMS for real world use. Throughout this project we were able to understand the full capabilities and implementations of the DBMS through the use of ERD and normalization processes. The development of the project has not only allowed us to discover the differences in pharmaceutical management system views, but has also introduced us to how complex databases can really get, and the importance of avoiding data redundancy as well as preventing anomalies for an efficient and functioning database.