

# **PROJECT REPORT**

## **ILSZ-511 DATABASE DESIGN**

**Spring 2022**

## **HEALTH DRUG DISPENSE SYSTEM**

### **TEAM MEMBERS:**

1. **Akhilesh Gowda**
2. **Himanshu Joshi**

## 1. Scenario and Database Requirements

The current Drug Dispense System is not able to keep a track on the people consuming it, it is harder to trace the people who are consuming the medicinal drugs. If the drugs are misused, it can result in fatality. The company, the government should know who is consuming what, and how much number of drugs. Moreover, it can also be a case of drugs hoarding. Imagine the current ongoing pandemic, where some person buys many doses of Pfizer's/Moderna just to store them and sell it in dark web/black market. In a nutshell, the current system fails to track the drug until the last mile. It is only able to track from the manufacturer to the shop/drug dispensary selling it.

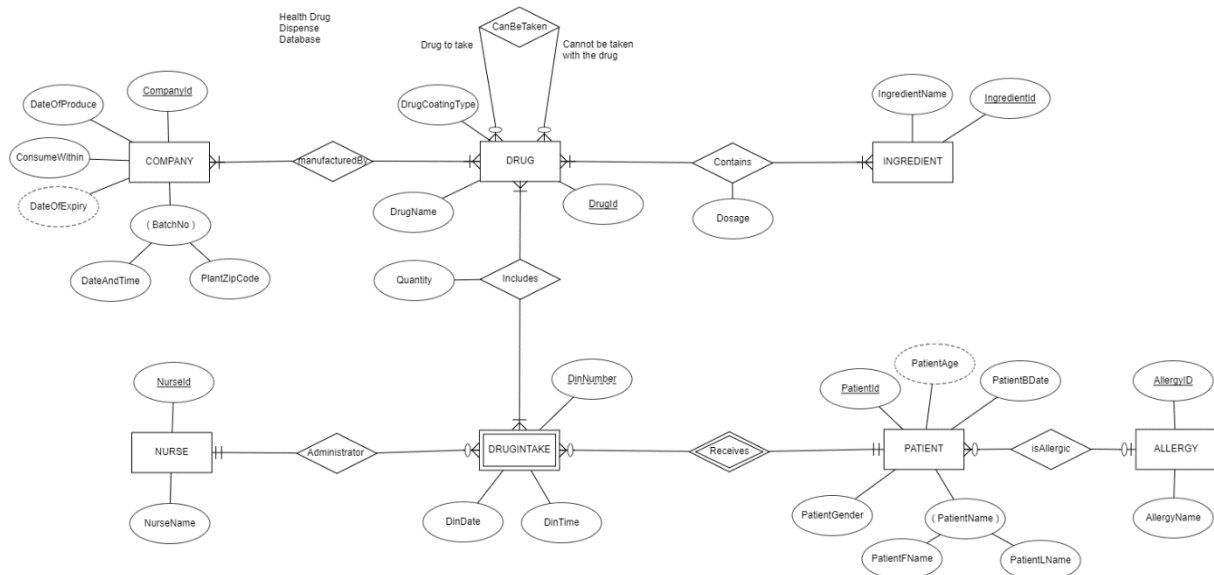
We at Health Drug Dispense Inc. (HDD Inc.) aim to keep a control on the drug intake of patients, by tracking the last mile drug delivery, i.e. by mapping the drug with each patient. With this we achieve and control in the drug dispense system, and we can have an alert if the patient is overconsuming. When a person buys a drug, he will be added in the central system, and the government or the local bodies can monitor them.

The sole purpose of this database lies in maintaining the records for each drug with a patient. Each entry will be mapped with some patient.

The intended users we plan to use our database are pharmacies, drug distributors, and hospitals. We have kept our database lucid, so that multiple users will understand the schema/working.

## 2. EERD

Let's have a look to the EERD of our schema.



The database, we have is highly scalable, and in 3-NF form.

The Entities we have are:

1. **Company**: The company we refer here is the drug manufacturing company. It consists of a unique company ID, Date of Produce of the drug, Consume Within which mentions within how many days drug has to be consumed, the derived attribute, Date of Expiry, which is derived by Adding the consume within with Date of Produce, a Batch Number which is combination of Date and Time of produce, and the Plant Zip Code where the drug was manufactured.
2. **Drug**: The Drug entity stores the data related to the drug which includes, the Drug ID which is unique, Drug Name, the coating type of Drug.
3. **Ingredient**: The ingredient entity stores the ingredient of the drug, which includes, unique Ingredient ID, and Ingredient Name.

The above 3 tables explained are related with Drugs, Company, and Ingredient i.e. producer. Now let's see the consumer part.

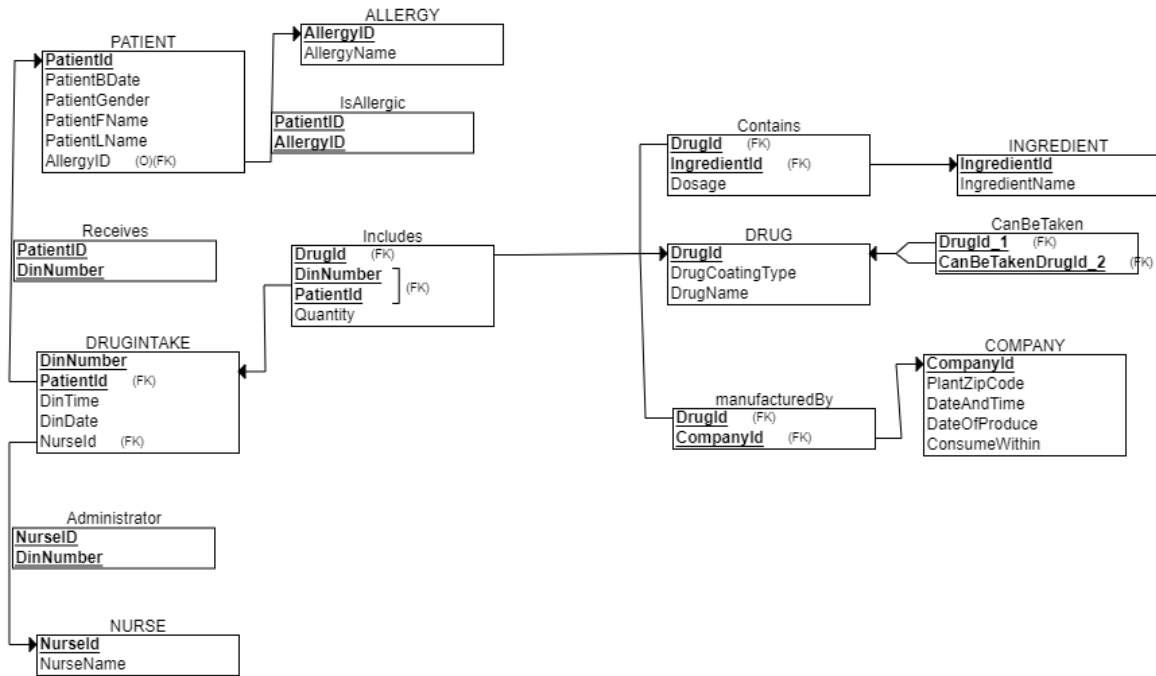
4. Nurse: This is the entity which stores the data related to the nurse which has prescribed the drug to the patient. It consists of unique Nurse ID, and the name of Nurse.
5. Drug Intake: In this entity we store, the time and date the patient has consumed the drug, and the Drug intake number.
6. Patient: This is the main consumer in the system, and for which the schema is being built. It consists of unique patient ID, Patient Gender, a derived attribute Patient Name which consists of Patient First and Last Name, Patient Date of Birth and Patient age, which is derived by Patient Date of Birth.
7. Allergy: This entity stores the allergy information of the patient. It contains a unique Allergy ID and the Allergy Name.

The Relationships we have in our schema are:

1. Manufactured By: This relationship will map Company and Drug entity. It stores the information related to the drug and the manufacturer. It maps the drug ID, with the company ID, which is compulsory and can be many to many mapping.
2. Contains: This relationship maps the relation with Drug and Ingredient entities. It stores the information related to Drug ID, Ingredient ID, and the number of Dosages. It is compulsory and can be many to many.
3. Includes: This relationship maps Drug with Drug Intake entities. It includes drug intake number, Patient ID, Drug ID, and the quantity of drugs the patient has consumed. It is compulsory and can be many to many.
4. Administrator: This relationship maps Nurse with the Drug intake entity. It stores the Nurse ID, and the Drug intake number which has prescribed the drug.
5. Receives: This relationship maps Drug intake with the Patient. It stores the Drug intake number and the patient ID which has consumed the drug.
6. Is Allergic: This relationship maps Patient with the Allergy entity. It stores Patient ID, and Allergy ID attributes. It is optional and can be many to many.

### 3. Relational Schema

The relational schema of our database is as follows:



The entities we discussed above ...

### 4. Normalization/Denormalization

We aim to keep the schema lucid and normalized. We can have a denormalized database, if let's say, a patient has multiple allergies, and the allergies are stored in patient entity. Now for each allergy he has a different record would be stored. Similarly, another patient having similar allergies will also be stored in same database. Now the data in patient entity is redundant. Multiple patients are having same allergies.

So, to avoid this redundant data, and have a normalized table, we have created a new table, allergy, which will store this data.

We didn't de-normalize any table because we planned the schema beforehand and we discussed from point of view of a database designer and the end users.

## 5. Create and populate DB

We create an imaginary scenario. We added a few imaginary patients, allergies, drugs, etc. and populated the database.

The tables we have are:

Tables_in_z511s22_amandyar
ALLERGY
Administrator
COMPANY
CanBeTaken
Contains
DRUG
DRUGINTAKE
INGREDIENT
Includes
IsAllergic
NURSE
PATIENT
Receives
manufacturedBy

The columns in each databases are as follows:

TABLE_NAME	COLUMN_NAME	ORDINAL_POSITION	COLUMN_DEFAULT	IS_NULLABLE	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH	CHARACTER_OCTET_LENGTH
NURSE	NurseId	1	NULL	NO	int	NULL	NULL
NURSE	NurseName	2	NULL	NO	varchar	100	400
ALLERGY	AllergyID	1	NULL	NO	int	NULL	NULL
ALLERGY	AllergyName	2	NULL	NO	varchar	100	400
PATIENT	PatientId	1	NULL	NO	int	NULL	NULL
PATIENT	PatientBDate	2	NULL	NO	varchar	100	400
PATIENT	PatientGender	3	NULL	NO	varchar	100	400
PATIENT	PatientFName	4	NULL	NO	varchar	100	400
PATIENT	PatientLName	5	NULL	NO	varchar	100	400
PATIENT	AllergyID	6	NULL	YES	int	NULL	NULL
DRUG	DrugId	1	NULL	NO	int	NULL	NULL
DRUG	DrugCoatingType	2	NULL	NO	varchar	100	400
DRUG	DrugName	3	NULL	NO	varchar	100	400
INGREDIENT	IngredientName	1	NULL	NO	varchar	100	400
INGREDIENT	IngredientId	2	NULL	NO	int	NULL	NULL
COMPANY	PlantZipCode	1	NULL	NO	int	NULL	NULL
COMPANY	DateAndTime	2	NULL	NO	varchar	100	400
COMPANY	DateOfProduce	3	NULL	NO	varchar	100	400
COMPANY	CompanyId	4	NULL	NO	int	NULL	NULL
COMPANY	ConsumeWithin	5	NULL	NO	varchar	100	400
Contains	Dosage	1	NULL	NO	int	NULL	NULL
Contains	DrugId	2	NULL	NO	int	NULL	NULL
Contains	IngredientId	3	NULL	NO	int	NULL	NULL
manufacturedBy	DrugId	1	NULL	NO	int	NULL	NULL
manufacturedBy	CompanyId	2	NULL	NO	int	NULL	NULL

## 6. SQL Statements

We tried to cover multiple queries covered in the class so far.

Queries:

1. Selecting the patient id, firstname and last name and his allergies

Select p.PatientId,p. PatientFName, p.PatientLName, a.AllergyName

From ALLERGY a, PATIENT p

Where a.AllergyID = p.AllergyID;

PatientId	PatientFName	PatientLName	AllergyName
2000	Mark	Walhberg	None
2001	Mark	Walhberg	Penicillin
2002	Max	Pattinson	Ibuprofen
2003	Akhilesh	Gowda	None
2004	Rebecca	Sims	Topiramate
2005	Allegra	Harvey	Nyquil
2006	Julia	Wanchic	Carbamazepine
2007	Emma	Watson	Ibuprofen
2008	John	Cena	Anticonvulsants
2009	Harry	Henderson	Antibiotics
2010	Arnold	Walhberg	Penicillin
2011	Julia	Roberts	Anticonvulsants
2012	Sandra	Bullock	Carbamazepine

Q.2. Selecting the patient id, firstname and last name having an allergy to Ibuprofen

Select p.PatientId,p. PatientFName, p.PatientLName, a.AllergyName

From ALLERGY a, PATIENT p

Where a.AllergyID = p.AllergyID and a.AllergyName = 'Ibuprofen';

PatientId	PatientFName	PatientLName	AllergyName
2002	Max	Pattinson	Ibuprofen
2007	Emma	Watson	Ibuprofen

Q.3 selecting all the patients who are not allergic to 'Nyquil' by making use of Minus

```
SELECT PatientId,PatientFName,PatientLName
from PATIENT p LEFT JOIN ALLERGY USING(AllergyID) where
AllergyName<>'Nyquil';
```

PatientId	PatientFName	PatientLName
2000	Mark	Walhberg
2001	Mark	Walhberg
2002	Max	Pattinson
2003	Akhilesh	Gowda
2004	Rebecca	Sims
2006	Julia	Wanchic
2007	Emma	Watson
2008	John	Cena
2009	Harry	Henderson
2010	Arnold	Walhberg
2011	Julia	Roberts
2012	Sandra	Bullock

Q.4 patient and their nurses

```
SELECT p.PatientId, p.PatientFName,n.NurseName
FROM DRUGINTAKE d, NURSE n, PATIENT p
```



WHERE n.Nurseld = d.Nurseld and p.PatientId = d.PatientId;

PatientId	PatientFName	NurseName
2000	Mark	Amy
2001	Mark	Cassidy
2002	Max	Lorain
2004	Rebecca	Sims
2003	Akhilesh	Jack
2005	Allegra	Cadee
2006	Julia	Becky
2007	Emma	Callie
2008	John	Natalie
2009	Harry	Bronwen
2010	Arnold	Theresa
2011	Julia	Pete
2012	Sandra	Adam

Q.5 company id , Drug name and ingredient name

SELECT c.CompanyId, d.DrugName, i.IngredientName

FROM COMPANY c, manufacturedBy m, DRUG d, Contains co, INGREDIENT i

WHERE m.DrugId = d.DrugId and c.CompanyId = m.CompanyId and co.DrugId = d.DrugId and i.IngredientId = co.IngredientId;

+ Options

CompanyId	DrugName	IngredientName
5000	Tylenol	Water
5000	Topiramate	Sulfate
5002	hydroxyzine	Magnesium
5002	Carbamazepine	Chlorine
5003	Hydrocodone	Carbonic Acid
5003	Diphenhydramine HCL	Phospate
5005	Escitalopram	Zinc
5006	Antibiotic	Phospate
5006	Anticonvulsants	Bicarbonate
5007	Penicillin	Sodium
5008	Ibuprofen	Iron

Q.6 Number of patients with Allergy

```
SELECT DISTINCT COUNT(p.PatientId) AS PatientsWithAllergy
FROM PATIENT p, ALLERGY a
WHERE p.AllergyID = a.AllergyID and a.AllergyName <>'None'
```

**PatientsWithAllergy**

11

Q.7 companies manufacturing more than one type of drugs

```
SELECT m.CompanyId,COUNT(m.DrugId) as NumberOfDrugs
FROM manufacturedBy m
GROUP by m.CompanyId
HAVING NumberOfDrugs>1;
```

CompanyId	NumberOfDrugs
5000	2
5002	2
5003	2
5006	2

Q.8 patient information for the drug taken by patient which has been manufactured by company id 5000

```

SELECT i.PatientId , p.PatientFName
FROM Includes i, PATIENT p
WHERE p.PatientId = i.PatientId and EXISTS( SELECT 1
                                           FROM manufacturedBy m
                                           where m.DrugId = i.DrugId and m.CompanyId = '5000');

```

+ Options

PatientId	PatientFName
2000	Mark
2003	Akhilesh

Q.9 Risk calculation for drugs

```

SELECT DrugId,CASE WHEN Dosage <25 THEN 'Safe to Consume'
                   WHEN Dosage >25 and Dosage <50 THEN 'Moderate Risk'
                   WHEN Dosage >50 THEN 'Highly Risky' END AS RiskTye
FROM Contains;

```

DrugId	RiskTye
3000	Safe to Consume
3001	Safe to Consume
3002	Highly Risky
3004	Moderate Risk
3005	Moderate Risk
3006	Moderate Risk
3007	Safe to Consume
3008	Moderate Risk
3009	Highly Risky
3010	Safe to Consume
3011	Highly Risky

Q.10 Joining multiple tables and fetching the drug name where Ingredient Name is "Carbonic Acid".

```
SELECT D.DrugName, I.IngredientName FROM DRUG D
      JOIN Contains C ON D.DrugId = C.DrugId
      JOIN INGREDIENT I ON I.IngredientId = C.IngredientId
WHERE I.IngredientName = "Carbonic Acid";
```

DrugName	IngredientName
Hydrocodone	Carbonic Acid

Q 11 Using union, we try to get patient name, and allergy name.

```
SELECT PatientFName FROM PATIENT
UNION
SELECT AllergyName FROM ALLERGY
```

Mark

Max

Akhilesh

Rebecca

Allegra

Julia

Emma

John

Harry

Arnold

Sandra

None

Antibiotics

Ibuprofen

Anticonvulsants

Nyquil

Carbamazepine

Topiramate

Penicillin