Digital Assignment - I

ITA5008
Database Technologies

Submitted by: -

Moeenul Islam

Submitted to: -

Dr Brijendra Singh

School of Computer Science and Engineering

Vellore Institute of Technology, Vellore

02 January, 2021

Scenario :- Pharmacy management System.

Functional requirements :-

- ↳ Login page
- ↳ Home page.
- ↳ Company
- ↳ Purchase.
- ↳ Drugs
- ↳ Sales.
- ↳ User / Settings.
- ↳ Messaging.

Objective

↳ To develop an application that deals with the day to day requirement of any pharmacy.

↳ To develop easy management of the medicines/drugs

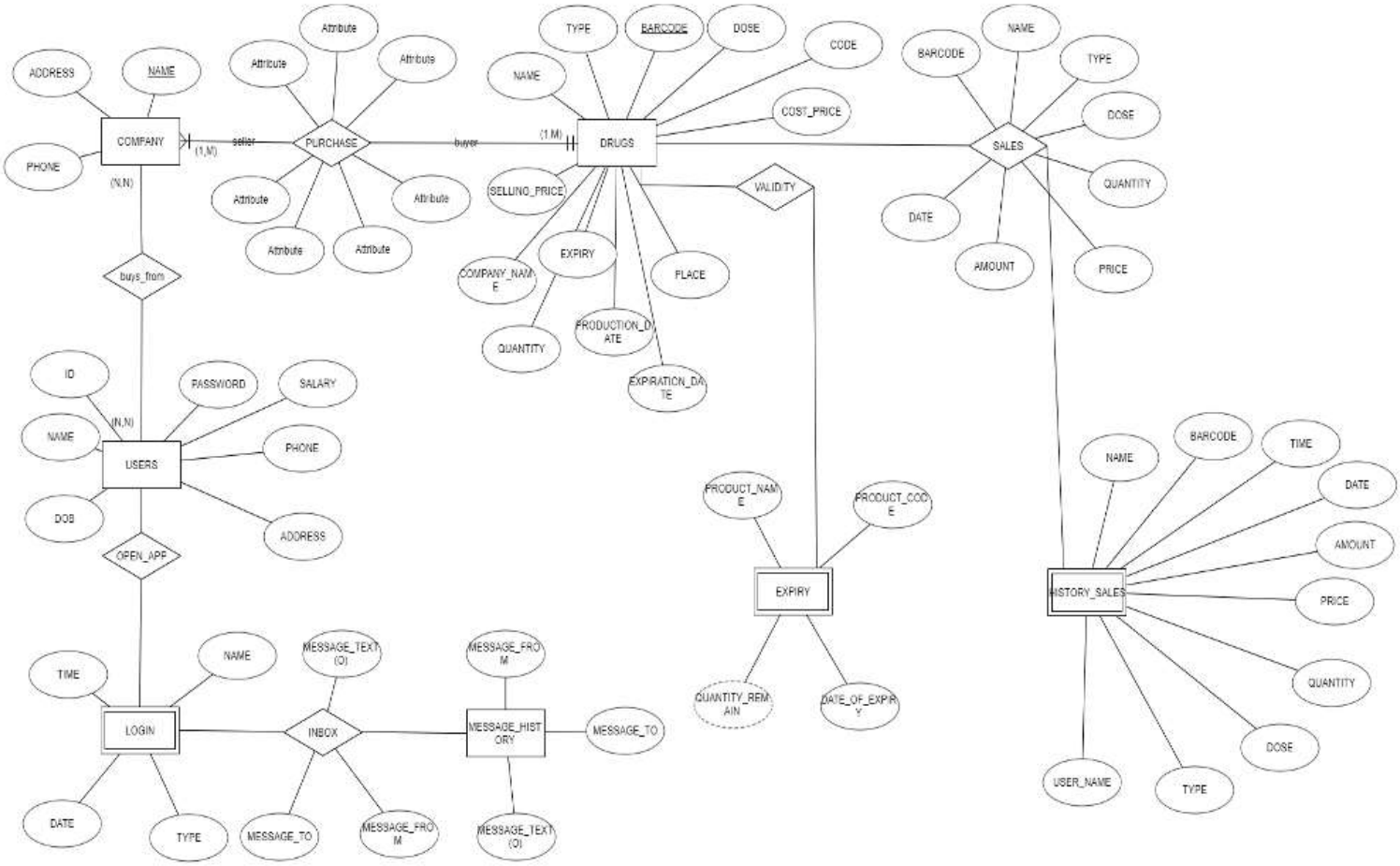
↳ To handle the inventory details, purchase details and stock expiry and quantity.

↳ To provide competitive advantage to the pharmacy

↳ To provide details, info about the stock on demand necessary and help locate it in shop easy.

↳ To make the stock manageable and simplify the use of inventory in the pharmacy.

ER DIAGRAM


ADDRESS
NAME
PHONE
COMPANY
(N,N)
buys_from
(1,M)
seller
PURCHASE
buyer
(1,M)
Attribute
Attribute
Attribute
Attribute
Attribute
Attribute
TYPE
BARCODE
DOSE
CODE
NAME
COST_PRICE
DRUGS
SELLING_PRICE
VALIDITY
EXPIRY
COMPANY_NAME
PLACE
QUANTITY
PRODUCTION_DATE
EXPIRATION_DATE
BARCODE
NAME
TYPE
DOSE
SALES
QUANTITY
DATE
AMOUNT
PRICE
ID
PASSWORD
SALARY
NAME
(N,N)
USERS
PHONE
DOB
ADDRESS
OPEN_APP
PRODUCT_NAME
PRODUCT_CODE
EXPIRY
QUANTITY_REMAIN
DATE_OF_EXPIRY
NAME
BARCODE
TIME
DATE
AMOUNT
PRICE
QUANTITY
DOSE
TYPE
USER_NAME
HISTORY_SALES
TIME
NAME
LOGIN
MESSAGE_TEXT (O)
INBOX
MESSAGE_FROM
MESSAGE_HISTORY
MESSAGE_TO
DATE
TYPE
MESSAGE_TO
MESSAGE_FROM
MESSAGE_TEXT (O)

4. Converting ERD into respective relational Schemas

Entity - Company:

↳ Attributes:

1. Name - primary key
2. Address
3. Phone

Entity - Drug:

↳ Attributes:

1. Name
2. Type
3. Barcode - Primary key
4. Dose
5. Cost-price
6. Sales-price
7. Expiry
8. Company-name - foreign key (references 'Name' in Company table)
9. Production-date
10. Expiration-date
11. Place
12. Quantity

History-Sale :

↳ Attributes :

1. Username.
2. Barcode – foreign key refrences barcode in Drug table.
3. Dose.
4. Type.
5. Price.
6. Amount.
7. Date.
8. Time.
9. Name.
10. Quantity.

Entity - Purchase :

↳ Attributes.

1. Company-name - foreign key refrences Name in Compa
2. Barcode – foreign key refrencing Barcode in drug table.
3. Type.
4. Price.
5. Amount.
6. Name.
7. Quantity.
8. ~~Sale~~
9.

Entity - Sale:

↳ Attributes:

1. Barcode - foreign key refrence Barcode in Drug table.
2. Dose
3. Type.
4. Price.
5. Amount.
6. Name.
7. Quantity.
8. Date.

Entity - User:

↳ Attributes:

1. ID - Primary key.
2. Name
3. DOB
4. Phone
5. Addres
6. Salary.
7. Password.

Entity - Login

↳ Attributes:

1. Name.
2. Type.
3. Date.
4. Time.
5. ID - foreign key refrence 'ID' in user.

Entity - Inbox

↳ Attributes

1. Message - from.
2. Message - text.
3. Message - to.
4. Sender - id.

Normalisation

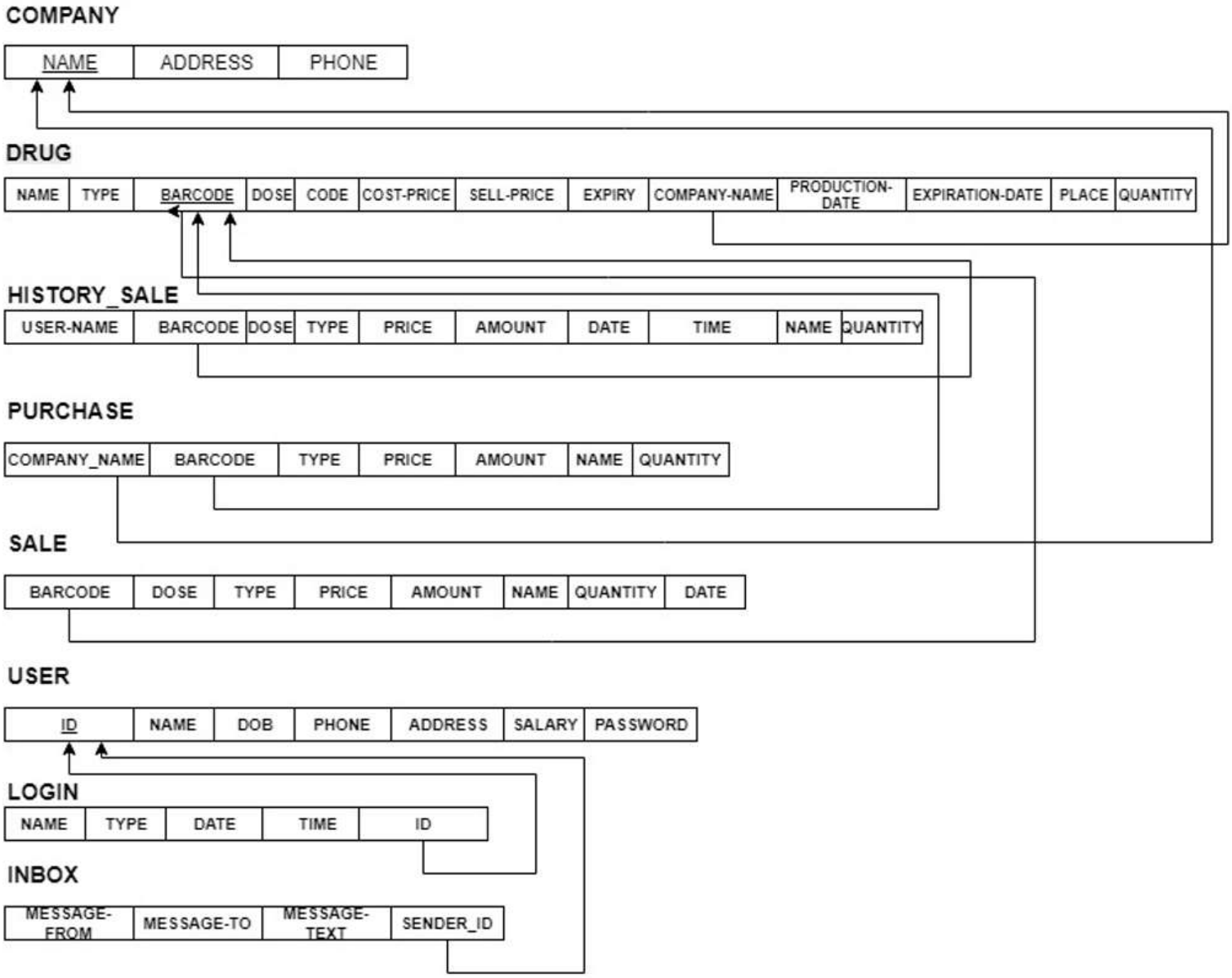
1NF :-
According to first normal form every attribute in the relation should be singled Valued.
↳ All of the tables are following 1NF

2NF :- It says a relation must be in 1NF and also must not contain any partial dependency. i.e., no non-prime attributes are dependent on any proper subset of any Candidate of any table.

3NF :- It says a relation must be in 2NF and also there should not be any transi-tive dependency for non-prime attributes as well.

Conclusion :- All of our tables follow and do not violate any of the normal forms therefore all of the relations are efficiently created.

RELATIONAL SCHEMA



Oracle Implementation

```
SQL> CREATE TABLE company (
  2    NAME varchar(50) NOT NULL PRIMARY KEY,
  3    ADDRESS varchar(50) NOT NULL,
  4    PHONE varchar(20) NOT NULL
  5  );

Table created.
```

```
SQL> CREATE TABLE drugs (
  2    NAME varchar(50) NOT NULL,
  3    TYPE varchar(20) NOT NULL,
  4    BARCODE varchar(20) NOT NULL PRIMARY KEY,
  5    DOSE varchar(10) NOT NULL,
  6    CODE varchar(10) NOT NULL,
  7    COST_PRICE FLOAT NOT NULL,
  8    SELLING_PRICE NUMBER(10,2) NOT NULL,
  9    EXPIRY varchar(20) NOT NULL,
 10    COMPANY_NAME varchar(50) NOT NULL,
 11    PRODUCTION_DATE date NOT NULL,
 12    EXPIRATION_DATE date NOT NULL,
 13    PLACE varchar(20) NOT NULL,
 14    QUANTITY NUMBER(11) NOT NULL
 15  );

Table created.
```

```
SQL> CREATE TABLE history_sales (
  2    USER_NAME varchar(20) NOT NULL,
  3    BARCODE varchar(20) NOT NULL references drugs(barcode),
  4    NAME varchar2(50) NOT NULL,
  5    TYPE varchar2(10) NOT NULL,
  6    DOSE varchar2(10) NOT NULL,
  7    QUANTITY NUMBER(10) NOT NULL,
  8    PRICE NUMBER(10,2)NOT NULL,
  9    AMOUNT NUMBER(10,2) NOT NULL,
 10    Date_ varchar2(15) NOT NULL,
 11    TIME varchar2(20) NOT NULL
 12  );

Table created.
```

```
SQL> CREATE TABLE purchase (
  2    BARCODE varchar(20) NOT NULL references drugs(BARCODE),
  3    NAME varchar(50) NOT NULL references company (NAME),
  4    TYPE varchar(20) NOT NULL,
  5    COMPANY_NAME varchar(20) NOT NULL,
  6    QUANTITY number(11) NOT NULL,
  7    PRICE number(10,2) NOT NULL,
  8    AMOUNT number(10,2) NOT NULL
  9  );

Table created.
```

```
SQL> CREATE TABLE sales (
  2    BARCODE varchar(20) NOT NULL references drugs(BARCODE),
  3    NAME varchar(50) NOT NULL,
  4    TYPE varchar(10) NOT NULL,
  5    DOSE varchar(10) NOT NULL,
  6    QUANTITY number(11) NOT NULL,
  7    PRICE number(10,2) NOT NULL,
  8    AMOUNT number(10,2) NOT NULL,
  9    Date_ varchar(15) NOT NULL
 10  );

Table created.
```

```
SQL> CREATE TABLE users (
  2    ID number(11) NOT NULL PRIMARY KEY,
  3    NAME varchar(50) NOT NULL,
  4    DOB varchar(20) NOT NULL,
  5    ADDRESS varchar(100) NOT NULL,
  6    PHONE varchar(20) NOT NULL,
  7    SALARY number(10,2) NOT NULL,
  8    PASSWORD varchar(20) NOT NULL
  9  );

Table created.
```

```
SQL> CREATE TABLE login (
  2    NAME varchar(50) NOT NULL,
  3    TYPE varchar(20) NOT NULL,
  4    Date_ varchar(20) NOT NULL,
  5    TIME varchar(20) NOT NULL,
  6    id number(11) NOT NULL references users(ID)
  7  );

Table created.
```

```
SQL> CREATE TABLE inbox (
  2    MESSAGE_FROM varchar(20) NOT NULL,
  3    MESSAGE_TO varchar(20) NOT NULL,
  4    MESSAGE_TEXT varchar(200) NOT NULL
  5  );

Table created.
```