

FUNDAMENTALS OF DATABASE

PROJECT PART-2

HETERO PHARMACEUTICALS DATABASE

Group – 3

Team Members:

- Sreekar Thanda
- Shravan Kumar Nuka
- Manideep Renikindi
- Varsha Umannagari
- Rama Venkat Sumith Thota

Hetero Pharmaceuticals is a service-oriented company for medicinal drugs. Hetero provides pharma services across metro cities in INDIA. The Headquarters is located at Hyderabad. Each pharmacy has details of patients, prescriptions, doctors, day-to-day transactions, Inventory of Medicines, Insurance details. Additionally, its database holds information about the nearby clinics, which serve as entities in the whole database and have their respective attributes.

There are several entities used in this project and those are listed below,

Entity:

Hetero has enormous database as its spread across several locations in India. These entities comprise of strong and weak entities which are as listed below,

Constraints:

- We are using various constraints, they are primary key, foreign key, check and unique in this database.
- Primary keys are Employee id, Medicine code, Drug id etc.
- Foreign keys are Patient id, doctor name etc.
- Check constraint is used to check the inventory stock.

1. HETERO

Hetero is an Indian pharmacy around which our project revolves and here are the important aspects that are to be considered with Hetero. Each pharmacy has a set of attributes.

Attributes: **ph_code, Address, Contact, Zip Code.**

Primary key: **Pharmacy code (ph_code).** Here, each pharmacy store has a unique code to identify them among the listed pharmacies of that company.

Relationship: Hetero Pharmacy Store is in One-Many relationship with prescription as every pharmacy can receive multiple prescriptions from patients, One-Many relationship with employee since there are multiple employees in a single pharmacy and Many-Many

relationship with inventory because there are several Medicines/Supplies in an inventory and store needs, several of such supplies.

2. **Employees data:** Employees are individuals working across several stores and they include cashiers, store managers etc... Each employee has a set of attributes.

Attributes: Employee Name, Designation, Employee Contact, Salary, Gender, Pharmacy Code.

Primary key: Employee Identity Number (Employee_ID). Each employee would have unique Employee_ID when compared to other employees in the store.

Relationship: This Employees entity shares Many to One relationship with Pharmacy store as there are several employees working in a single store.

shares Many-One relationship with Pharmacy Locations. Employees data is considered as strong entity as this an independent to pharmacy.

3. **Inventory:** Inventory is the excess stock/surplus supplies present in warehouse or storage unit for every pharmacy company from where the supplies are sent out to every store depending upon the requirement. Inventory stores the data of the quantity of medicines and related information with respect to medicines, vaccines etc. Each inventory possess a specific set of attributes.

Attributes: Item identity Number, Item name, Quantity, Availaility.

Primary Key: Item identity Number (Item_Id). Each medicine is identified by specific identity number for identification as several medicines from several companies have same or similar compositions and this id is used to distinguish them easily.

Relationship: This entity shares Many-Many relationship with Pharmacy since the inventory holds data for several medicines and the pharmacy possess several medicines and requires similar medicines.

4. **Prescription:** Prescription is a list of medicine that a patient need to use and that varies with person to person and disease to disease as doctor recommends to the patient.

Attributes: Prescription Identification Number (Prescription_Id), Date, Pharmacy Code.

Primary Key: Prescription Identification Number (Prescription_Id).

Relationship: It shares Many-One relationship with Pharmacy since there is a single is a single prescription and there can be many pharmacy stores to purchase that medicine. Prescription shares Ternary Relationship with Doctor and Patient and Prescription shares ternary relationship with Billing and patient.

5. **Patients:** Patients are those that suffer from any ailments and they are crucial for any pharmacy because patients give complete analysis to most of the entities in this E-R representation.

Attributes: Patient Identification Number(Patient_ Id), Patient Name, Date of Birth, Disease, Gender, Contact,AGE

With the Date Of Birth we could derive the age, Hence age could be a derived attribute.

Note: AGE is a derived Attribute here.

Primary key: Patient Identification Number (Patient_ Id).

Relationship: Patient has ternary relationship with Prescription and doctor and patient possess ternary relationship with billing and prescription.

6. **Doctor:** Doctor may or might not be present in the pharmacy, but he is the only authorized person to prescribe medication to patient. The following are the attributes.

Attributes: Doctor Registration Identity Number, Specialization,Contact.

Primary Key: Doctor Registration Identity Number (Doctor Reg_No).

Relationship: Doctor has only one Ternary Relationship with prescription and patient.

7. **Insurance:** Every patient has medical insurance that covers partial or full payment for the patient or their dependents. Each patient can have one or many insurances depending upon their necessity and preference. The attributes are as follows.

Attributes: Insurance Company, Amount, Insurance Number.

Primary Key: Insurance Number.

Relationship: Insurance has many to one relation with billing and Insurance possess Many-Many relationship with Patient.

8. **Billing:** This Entity describes about the transactions made by patient at pharmacy during purchase. Each medicine has a price depending on the drug and dosage and so this entity stores the cost price of the medicines and stores the information of day-to-day payments related to buying and selling the medicines.

Attributes: Date of Billing, Billing Amount, Transaction Identity Number, Transaction Number.

Primary Key: Transaction Identity Number (Transaction_Id).

Relationship: Billing entity has many to one relationship with insurance and billing possess ternary relationship with prescription and patient.

RELATIONSHIP TABLES

TERNARY RELATIONS.

- 1.TREATMENT: This is ternary relationship table for patient, doctor, and prescription.
2. PURCHASE: This is ternary relationship for patient, billing & Prescription.

MANY TO MANY RELATIONS.

1. INSURANCE CLAIM:

2. STOCK:

CONSTRAINTS:

The set of rules, ensures that when an authorized user modifies the database, they do not disturb the data consistency, A constraint is a rule that is used for optimization purposes.

Primary key: A primary key constraint is a column or combination of columns that has the same properties as a unique constraint. You can use primary key and foreign key constraints to define relationships between tables.

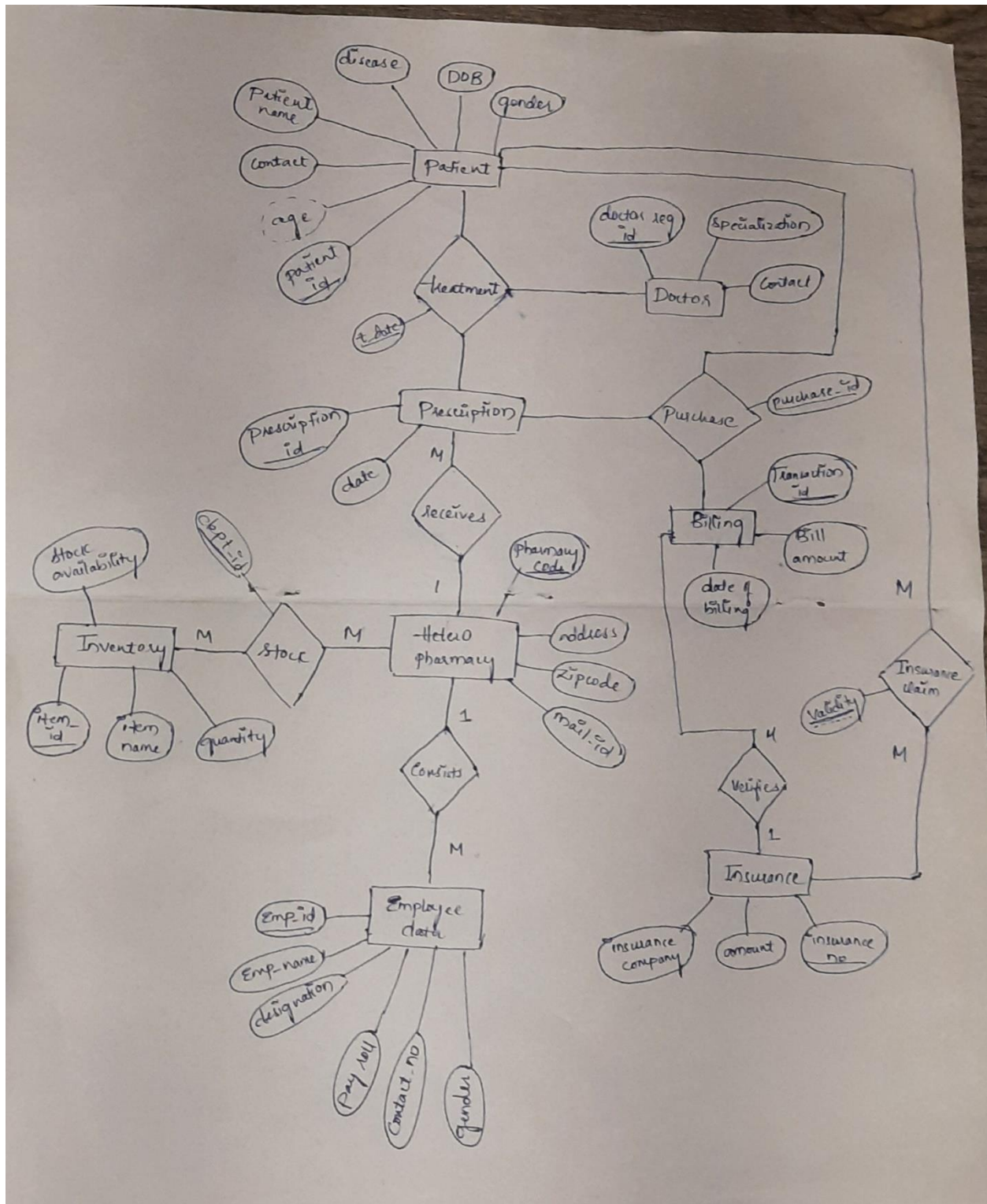
Foreign key: (referential constraint or a referential integrity constraint) is a logical rule about values in one or more columns in one or more tables. For example, a set of tables shares information about a corporation's suppliers. Occasionally, a supplier's name changes. You can define a referential constraint that states the ID of the supplier in a table must match a supplier ID in the supplier information. This constraint prevents insert, update, or delete operations that would otherwise result in missing supplier information.

Unique (unique key constraint): This is a rule that forbids duplicate values in one or more columns within a table. Unique and primary keys are the supported unique constraints. For example, a unique constraint can be defined on the supplier identifier in the supplier table to ensure that the same supplier identifier is not given to two suppliers.

Check: A check constraint (also referred to as a **table check constraint**) is a database rule that specifies the values allowed in one or more columns of every row of a table. Specifying check constraints is done through a restricted form of a search condition.

Not null: A NOT NULL constraint is a rule that prevents null values from being entered into one or more columns within a table.

E_R Diagram of Hetero Pharmaceutical:



Transformation of E_R Diagram into relations:

DB Schema of the above E-R diagram:

Hetero (Ph_code, address, zipcode, contact)

Prescription (prescription_id, date, ph_code)

Patient (Patient_id, patient name, disease, gender, dob, contact, age)

Doctor (Doctor_Id, Specialization, contact)

Employee (Employee_id, employee_name, designation, salary, contact, gender, ph_code)

Insurance (Insurance_No, amount, Insurance Company)

Inventory (Item_id, Item_name, Quantity, Availability)

Bills(Date_of_Billing, Bill_amount, insurance_no)

Treatment (token_no, patient_id, prescription_id, doctor_id)

Stock (ph_code, item_id, dept)

Purchase (transaction_id, prescription_id, patient_id, purchase_id)

Insurance claim (patient_id, insurance_no, validity)

Tables creation and insertion of atleast 10 tuples:

1. Creation and insertion of Hetero table:

```
drop table hetero;
```

```
create table hetero(
```

```
    ph_code varchar2(20) not null primary key,
```

```
    ph_contact char(20) not null,
```

```
    address varchar2(20) not null,
```

```
    zipcode int not null unique
```

```
);
```

```
insert into hetero values('A1234','932-121-1223','5432 Mean green st ', 76200);
```

```
insert into hetero values('A1235','637-223-1231','2500 W hickory st ', 76207);
```

insert into hetero values('A1236','912-222-1221','3500N Bonniebrae st ', 76205);

insert into hetero values('A1237','839-283-2822','1500 oak st', 76247);

insert into hetero values('A1238','927-282-2922','1000 East hickory st', 76244);

insert into hetero values('A1239','973-282-2929','3000 bernerd st ', 76278);

insert into hetero values('A1240','828-292-2822','6550 stella st', 76289);

insert into hetero values('A1241','228-282-2222','1230 north elem st ', 76223);

insert into hetero values('A1242','222-222-8282','4000 southloop st ', 76245);

insert into hetero values('A1243','123-182-1292','7000 north loop st ', 76254);

select * from hetero;

```
SQL*Plus: Release 21.0.0.0.0 - Production on Mon Mar 6 16:29:59 2023
Version 21.3.0.0.0
```

```
Copyright (c) 1982, 2021, Oracle. All rights reserved.
```

```
Enter user-name: system
```

```
Enter password:
```

```
Last Successful login time: Sun Mar 05 2023 13:57:38 -06:00
```

```
Connected to:
```

```
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0
```

```
SQL> drop table hetero;
```

```
drop table hetero
```

```
*
```

```
ERROR at line 1:
```

```
ORA-00942: table or view does not exist
```

```
SQL> create table hetero(
```

```
2   ph_code varchar2(20) not null primary key,
```

```
3   ph_contact char(20) not null,
```

```
4   address varchar2(20) not null,
```

```
5   zipcode int not null unique
```

```
6 );
```

```
Table created.
```

```
SQL> S|
```

```
SQL> insert into hetero values('A1234','932-121-1223','5432 Mean green st ', 76200);  
1 row created.
```

```
SQL> insert into hetero values('A1235','637-223-1231','2500 W hickory st ', 76207);  
1 row created.
```

```
SQL> insert into hetero values('A1236','912-222-1221','3500N Bonniebrae st ', 76205);  
1 row created.
```

```
SQL> insert into hetero values('A1237','839-283-2822','1500 oak st', 76247);  
1 row created.
```

```
SQL> insert into hetero values('A1238','927-282-2922','1000 East hickory st', 76244);  
1 row created.
```

```
SQL> insert into hetero values('A1239','973-282-2929','3000 bernerd st ', 76278);  
1 row created.
```

```
SQL> insert into hetero values('A1240','828-292-2822','6550 stella st', 76289);  
1 row created.
```

```
SQL> insert into hetero values('A1241','228-282-2222','1230 north elem st ', 76223);  
1 row created.
```

```
SQL> insert into hetero values('A1242','222-222-8282','4000 southloop st ', 76245);  
1 row created.
```

```
SQL> insert into hetero values('A1243','123-182-1292','7000 north loop st ', 76254);  
1 row created.
```

```
SQL> select * from hetero;
```

PH_CODE	PH_CONTACT	ADDRESS	ZIPCODE
A1234	932-121-1223	5432 Mean green st	76200
A1235	637-223-1231	2500 W hickory st	76207
A1236	912-222-1221	3500N Bonniebrae st	76205
A1237	839-283-2822	1500 oak st	76247
A1238	927-282-2922	1000 East hickory st	76244
A1239	973-282-2929	3000 bernerd st	76278
A1240	828-292-2822	6550 stella st	76289
A1241	228-282-2222	1230 north elem st	76223
A1242	222-222-8282	4000 southloop st	76245
A1243	123-182-1292	7000 north loop st	76254

2. Creation of Employee table and insertion of tuples into it:

```
drop table employee_data;

create table employee_data(

    emp_id number not null primary key,

    emp_name varchar2(30),

    designation varchar2(50),

    emp_contact varchar2(20) not null,

    salary NUMBER not null,

    gender varchar2(8),

    ph_code varchar2(20),

    FOREIGN KEY(ph_code) REFERENCES hetero(ph_code),

    constraint salary_check CHECK (salary > 0),

    constraint emp_gender_check CHECK (gender in('male','female','TS'))

);

insert into employee_data values(201,'Arun','Manager','922-2922-222',6000,'male','A1234');

insert into employee_data values(202,'Tarun','Assistant Manager','234-234-6464',5000,'male','A1235');

insert into employee_data values(203,'Varun','Receptionist','123-643-1234',2500,'male','A1236');

insert into employee_data values(204,'Varsini','Stock Manager','342-282-9382',4000,'female','A1237');

insert into employee_data values(205,'Fahien','Sales Manager','975-222-2348',4500,'male','A1238');

insert into employee_data values(206,'Vignesh','Sales_Man-1','123-346-3411',2000,'male','A1239');
```

```
insert into employee_data values(207,'Charitha','Sales_Man-2','321-067-3933',2000,'female','A1240');
```

```
insert into employee_data values(208,'Vinisha','Sales_Man-3','985-363-2933',2000,'female','A1241');
```

```
insert into employee_data values(209,'Varsha','Sales_Man-4','322-229-2223',2000,'female','A1242');
```

```
insert into employee_data values(210,'Auish','Sales_Man-5','763-272-2721',2000,'TS','A1243');
```

```
select * from employee_data;
```

```
SQL> create table employee_data(  
  2     emp_id number not null primary key,  
  3     emp_name varchar2(30),  
  4     designation varchar2(50),  
  5     emp_contact varchar2(20) not null,  
  6     salary NUMBER not null,  
  7     gender varchar2(8),  
  8     ph_code varchar2(20),  
  9     FOREIGN KEY(ph_code) REFERENCES hetero(ph_code),  
 10     constraint salary_check CHECK (salary > 0),  
 11     constraint emp_gender_check CHECK (gender in('male','female','TS'))  
 12 );
```

Table created.

```
SQL> |
```

```

11      constraint emp_gender_check CHECK (gender in('male','female','TS'))
12 );

Table created.

SQL> insert into employee_data values(201,'Arun','Manager','9222922222',6000,'male','A1234');

1 row created.

SQL> insert into employee_data values(202,'Tarun','Assistant Manager','234-234-6464',5000,'male','A1235');

1 row created.

SQL> insert into employee_data values(203,'Varun','Receptionist','123-643-1234',2500,'male','A1236');

1 row created.

SQL> insert into employee_data values(204,'Varsini','Stock Manager','342-282-9382',4000,'female','A1237');

1 row created.

SQL> insert into employee_data values(205,'Fahien','Sales Manager','975-222-2348',4500,'male','A1238');

1 row created.

SQL> insert into employee_data values(206,'Vignesh','Sales_Man-1','123-346-3411',2000,'male','A1239');

1 row created.

SQL> insert into employee_data values(207,'Charitha','Sales_Man-2','321-067-3933',2000,'female','A1240');

1 row created.

SQL> insert into employee_data values(208,'Vinisha','Sales_Man-3','985-363-2933',2000,'female','A1241');

1 row created.

SQL> insert into employee_data values(209,'Varsha','Sales_Man-4','322-229-2223',2000,'female','A1242');

1 row created.

SQL> insert into employee_data values(210,'Auish','Sales_Man-5','763-272-2721',2000,'TS','A1243');

1 row created.

```

3. Creation of Inventory Table:

drop table inventory;

create table inventory(

item_id number not null primary key,

item_name varchar2(20) not null,

availability varchar2(19) not null,

quantity NUMBER,

constraint availability check (availability in ('available','unavailable'))),

```
        constraint quantity_check check (quantity >100)

);

insert into inventory values(1000,'Dolo','available',20000);

insert into inventory values(1001,'Loperamide','unavailable',15000);

insert into inventory values(1002,'Paracetamol','available',18000);

insert into inventory values(1003,'Acetaminophen','unavailable',27000);

insert into inventory values(1004,'Ketoconazole','available',10000);

insert into inventory values(1005,'Diltiazem','unavailable',33000);

insert into inventory values(1006,'cetirizine','available',73000);

insert into inventory values(1007,'Brinzolamide','unavailable',29000);

insert into inventory values(1008,'Naproxen','available',23000);

insert into inventory values(1009,'Tylenol','available',22000);
```

```
SQL> create table inventory(
 2     item_id number not null primary key,
 3     item_name varchar2(20) not null,
 4     availability varchar2(19) not null,
 5     quantity NUMBER,
 6     constraint availability check (availability in ('available','unavailable')),
 7     constraint quantity_check check (quantity >100)
 8 );

Table created.

SQL> |
```

Insertion of tuples into Inventory table:

```

8 );
Table created.
SQL> insert into inventory values(1000,'Dolo','available',20000);
1 row created.
SQL> insert into inventory values(1001,'Loperamide','unavailable',15000);
1 row created.
SQL> insert into inventory values(1002,'Paracetamol','available',18000);
1 row created.
SQL> insert into inventory values(1003,'Acetaminophen','unavailable',27000);
1 row created.
SQL> insert into inventory values(1004,'Ketoconazole','available',10000);
1 row created.
SQL> insert into inventory values(1005,'Diltiazem','unavailable',33000);
1 row created.
SQL> insert into inventory values(1006,'cetirizine','available',73000);
1 row created.
SQL> insert into inventory values(1007,'Brinzolamide','unavailable',29000);
1 row created.
SQL> insert into inventory values(1008,'Naproxen','available',23000);
1 row created.
SQL> insert into inventory values(1009,'Tylenol','available',22000);
1 row created.
- --- -----
SQL> select * from inventory;

  ITEM_ID  ITEM_NAME  AVAILABILITY  QUANTITY
-----
    1000   Dolo      available      20000
    1001 Loperamide   unavailable    15000
    1002 Paracetamol available     18000
    1003 Acetaminophen unavailable    27000
    1004 Ketoconazole available     10000
    1005 Diltiazem   unavailable    33000
    1006 cetirizine  available     73000
    1007 Brinzolamide unavailable    29000
    1008 Naproxen    available     23000
    1009 Tylenol     available     22000

10 rows selected.

SQL> |

```

4. Creation and insertion of Prescription table:

create table prescription

(

prescription_id varchar2(20) not null primary key,

p_date date not null,

ph_code varchar2(20),

foreign key(ph_code) references hetero(ph_code)

);

insert into prescription values('301',TO_DATE('03-06-2023','DD-MM-YYYY'),'A1234');

insert into prescription values('302',TO_DATE('03-06-2023','DD-MM-YYYY'),'A1235');

insert into prescription values('303',TO_DATE('04-03-2023','DD-MM-YYYY'),'A1236');

insert into prescription values('304',TO_DATE('05-05-2023','DD-MM-YYYY'),'A1237');

insert into prescription values('305',TO_DATE('06-07-2023','DD-MM-YYYY'),'A1238');

insert into prescription values('306',TO_DATE('07-08-2023','DD-MM-YYYY'),'A1239');

insert into prescription values('307',TO_DATE('09-09-2023','DD-MM-YYYY'),'A1240');

insert into prescription values('308',TO_DATE('13-10-2023','DD-MM-YYYY'),'A1241');

insert into prescription values('309',TO_DATE('23-11-2023','DD-MM-YYYY'),'A1242');

insert into prescription values('310',TO_DATE('30-12-2023','DD-MM-YYYY'),'A1243');

insert into prescription values('301',TO_DATE('03-06-2023','DD-MM-YYYY'),'A1234');

```
SQL> create table prescription
2      (
3      prescription_id varchar2(20) not null primary key,
4      p_date date not null,
5      ph_code varchar2(20),
6      foreign key(ph_code) references hetero(ph_code)
7  );
```

Table created.

SQL> |

Table created.

```
SQL> insert into prescription values('301',TO_DATE('03-06-2023','DD-MM-YYYY'),'A1234');
```

1 row created.

```
SQL> insert into prescription values('302',TO_DATE('03-06-2023','DD-MM-YYYY'),'A1235');
```

1 row created.

```
SQL> insert into prescription values('303',TO_DATE('04-03-2023','DD-MM-YYYY'),'A1236');
```

1 row created.

```
SQL> insert into prescription values('304',TO_DATE('05-05-2023','DD-MM-YYYY'),'A1237');
```

1 row created.

```
SQL> insert into prescription values('305',TO_DATE('06-07-2023','DD-MM-YYYY'),'A1238');
```

1 row created.

```
SQL> insert into prescription values('306',TO_DATE('07-08-2023','DD-MM-YYYY'),'A1239');
```

1 row created.

```
SQL> insert into prescription values('307',TO_DATE('09-09-2023','DD-MM-YYYY'),'A1240');
```

1 row created.

```
SQL> insert into prescription values('308',TO_DATE('13-10-2023','DD-MM-YYYY'),'A1241');
```

1 row created.

```
SQL> insert into prescription values('309',TO_DATE('23-11-2023','DD-MM-YYYY'),'A1242');
```

1 row created.

```
SQL> insert into prescription values('310',TO_DATE('30-12-2023','DD-MM-YYYY'),'A1243');
```

1 row created.

```
SQL> select * from prescription;
```

PRESCRIPTION_ID	P_DATE	PH_CODE
301	03-JUN-23	A1234
302	03-JUN-23	A1235
303	04-MAR-23	A1236
304	05-MAY-23	A1237
305	06-JUL-23	A1238
306	07-AUG-23	A1239
307	09-SEP-23	A1240
308	13-OCT-23	A1241
309	23-NOV-23	A1242
310	30-DEC-23	A1243

10 rows selected.

```
SQL> |
```

5. Patient table creation and insertion of tuples:

```
create table patient(  
  
    patient_name varchar2(20) not null,  
  
    patient_id varchar2(20) not null primary key,  
  
    disease varchar2(30),  
  
    date_of_birth date,  
  
    contact varchar2(25) not null unique,  
  
    gender char(10) check (gender in ('male','female','TS'))  
  
);  
  
insert into patient values('John','2012', 'heart disease' ,TO_DATE('01-01-1998','DD-MM-YYYY') ,  
9329222222,'male' );  
  
insert into patient values('Taylor','2013', 'skin cancer' ,TO_DATE('02-03-1987','DD-MM-YYYY')  
,8327372722,'female' );  
  
insert into patient values('pearson','2014', 'chronic respiratory disease',TO_DATE('04-05-1996','DD-MM-YYYY')  
,962123452,'male' );  
  
insert into patient values('olivea','2015', 'alzheimers disease' ,TO_DATE('02-08-1897','DD-MM-YYYY') ,  
9372827282,'female' );  
  
insert into patient values('Jordan','2016', 'chronic kidney disease' ,TO_DATE('05-09-1999','DD-MM-YYYY')  
,9332129432,'male' );  
  
insert into patient values('bill hunt','2017', 'liver disease' , TO_DATE('07-03-1997','DD-MM-YYYY'),  
937283383,'male' );  
  
insert into patient values('Jesika','2019', 'cardiovascular disease' , TO_DATE('14-02-1999','DD-MM-YYYY'),  
9272293829,'female' );  
  
insert into patient values('watson','2020', 'cerebrovasculsar disease' , TO_DATE('17-06-1998','DD-MM-YYYY'),  
9383928392,'male' );  
  
insert into patient values('robinson','2021', ' scleroderma' , TO_DATE('15-09-1999','DD-MM-YYYY'),  
938293222,'male' );  
  
insert into patient values('williamson','2022', 'valve disease' , TO_DATE('12-07-1997','DD-MM-YYYY'),  
9493939393,'male' );
```



```
SQL> create table patient(
2     patient_name varchar2(20) not null,
3     patient_id varchar2(20) not null primary key,
4     disease varchar2(30),
5     date_of_birth date,
6     contact varchar2(25) not null unique,
7     gender char(10) check (gender in ('male','female','TS'))
8 );
```

Table created.

SQL> |

```

Table created.
SQL> insert into patient values('John','2012', 'heart disease' ,TO_DATE('01-01-1998','DD-MM-YYYY') , 932922222,'male' );
1 row created.
SQL> insert into patient values('Taylor','2013', 'skin cancer' ,TO_DATE('02-03-1987','DD-MM-YYYY') ,8327372722,'female' );
1 row created.
SQL> insert into patient values('pearson','2014', 'chronic respiratory disease',TO_DATE('04-05-1996','DD-MM-YYYY') ,962123452,'male' );
1 row created.
SQL> insert into patient values('olivea','2015', 'alzheimers disease' ,TO_DATE('02-08-1897','DD-MM-YYYY') , 9372827282,'female' );
1 row created.
SQL> insert into patient values('Jordan','2016', 'chronic kidney disease' ,TO_DATE('05-09-1999','DD-MM-YYYY') ,9332129432,'male' );
1 row created.
SQL> insert into patient values('bill hunt','2017', 'liver disease' , TO_DATE('07-03-1997','DD-MM-YYYY') , 937283383,'male' );
1 row created.
SQL> insert into patient values('Jesika','2019', 'cardiovascular disease' , TO_DATE('14-02-1999','DD-MM-YYYY') , 9272293829,'female' );
1 row created.
SQL> insert into patient values('watson','2020', 'cerebrovascular disease' , TO_DATE('17-06-1998','DD-MM-YYYY') , 9383928392,'male' );
1 row created.
SQL> insert into patient values('robinson','2021', ' scleroderma' , TO_DATE('15-09-1999','DD-MM-YYYY') , 938293222,'male' );
1 row created.
SQL> insert into patient values('williamson','2022', 'valve disease' , TO_DATE('12-07-1997','DD-MM-YYYY') , 9493939393,'male' );
1 row created.
SQL> |
```

6. Creating Doctor table and inserting tuples into it:

create table Doctor

(

doctor_name varchar2(20) not null ,

specialization varchar2(20),

doctor_id number not null primary key,

contact number unique not null,

```
constraint specialization check (specialization in ('cardiologist','ENT','oncologist','dentist', 'neurologist'))
);

insert into Doctor values ('John', 'cardiologist',700, 2837489655);

insert into Doctor values ('Peter', 'ENT',701,8786431467);

insert into Doctor values ('Cinderella', 'oncologist',702,9387328964);

insert into Doctor values ('Bill Hunter', 'dentist',703 ,7859766658);

insert into Doctor values ('Alex', 'neurologist',704, 2787666439);

insert into Doctor values ('Joe', 'dentist', 705,5387668736);

insert into Doctor values ('Olivia', 'ENT',706, 9787666666);

insert into Doctor values ('Mark Henry', 'neurologist',707, 6855437466);

insert into Doctor values ('Elizabeth', 'dentist',708, 9087666666);

insert into Doctor values ('Ashley', 'ENT',709, 4238366346);
```

```
SQL> create table Doctor
2 (
3   doctor_name varchar2(20) not null ,
4   specialization varchar2(20),
5   doctor_id number not null primary key,
6   contact number unique not null,
7   constraint specialization check (specialization in ('cardiologist','ENT','oncologist','dentist', 'neurologist'))
8 );

Table created.

SQL> |
```

```

Table created.
SQL> insert into Doctor values ('John', 'cardiologist',700, 2837489655);
1 row created.
SQL> insert into Doctor values ('Peter', 'ENT',701,8786431467);
1 row created.
SQL> insert into Doctor values ('Cinderella', 'oncologist',702,9387328964);
1 row created.
SQL> insert into Doctor values ('Bill Hunter', 'dentist',703 ,7859766658);
1 row created.
SQL> insert into Doctor values ('Alex', 'neurologist',704, 2787666439);
1 row created.
SQL> insert into Doctor values ('Joe', 'dentist', 705,5387668736);
1 row created.
SQL> insert into Doctor values ('Olivia', 'ENT',706, 9787666666);
1 row created.
SQL> insert into Doctor values ('Mark Henry', 'neurologist',707, 6855437466);
1 row created.
SQL> insert into Doctor values ('Elizabeth', 'dentist',708, 9087666666);
1 row created.
SQL> insert into Doctor values ('Ashley', 'ENT',709, 4238366346);
1 row created.

```

```
SQL> select * from Doctor;
```

DOCTOR_NAME	SPECIALIZATION	DOCTOR_ID	CONTACT
John	cardiologist	700	2837489655
Peter	ENT	701	8786431467
Cinderella	oncologist	702	9387328964
Bill Hunter	dentist	703	7859766658
Alex	neurologist	704	2787666439
Joe	dentist	705	5387668736
Olivia	ENT	706	9787666666
Mark Henry	neurologist	707	6855437466
Elizabeth	dentist	708	9087666666
Ashley	ENT	709	4238366346

10 rows selected.

7. Creation and insertion of tuples in insurance tables:

create table insurance(

insurance_id int not null primary key,

insurance_company varchar2(20) not null,

amount float

);

insert into insurance values(601,'hdfc credila',1000);

insert into insurance values(602,'hdfc credila',1000);

insert into insurance values(603,' united health',2000);

insert into insurance values(604,'molina healthcare',3000);

insert into insurance values(605,'care source',4000);

insert into insurance values(606,' metropolitan',5000);

insert into insurance values(607,'point32health',7000);

insert into insurance values(608,'blue cross',6000);

insert into insurance values(609,'upmc health care ',5000);

insert into insurance values(610,'highmark group',4000);

```
SQL> create table insurance(  
2     insurance_id int not null primary key,  
3     insurance_company varchar2(20) not null,  
4     amount float not null  
5 );
```

Table created.

```
SQL> |
```

```

SQL> insert into insurance values(601,'hdfc credila',1000);
1 row created.

SQL> insert into insurance values(602,'hdfc credila',1000);
1 row created.

SQL> insert into insurance values(603,' united health',2000);
1 row created.

SQL> insert into insurance values(604,'molina healthcare',3000);
1 row created.

SQL> insert into insurance values(605,'care source',4000);
1 row created.

SQL> insert into insurance values(606,' metropolitan',5000);
1 row created.

SQL> insert into insurance values(607,'point32health',7000);
1 row created.

SQL> insert into insurance values(608,'blue cross',6000);
1 row created.

SQL> insert into insurance values(609,'upmc health care ',5000);
1 row created.

SQL> insert into insurance values(610,'highmark group',4000);
1 row created.

```

```
SQL> select * from insurance;
```

INSURANCE_ID	INSURANCE_COMPANY	AMOUNT
601	hdfc credila	1000
602	hdfc credila	1000
603	united health	2000
604	molina healthcare	3000
605	care source	4000
606	metropolitan	5000
607	point32health	7000
608	blue cross	6000
609	upmc health care	5000
610	highmark group	4000

```
10 rows selected.
```

8. Creating bills table and insertion of tuples in it:

create table bills

(

trans_id varchar2(10) primary key,

bill_amount char(10),

insurance_id int not null,

foreign key(insurance_id) references insurance(insurance_id)

);

```
SQL> create table bills
  2  (
  3  trans_id varchar2(10) primary key,
  4  bill_amount char(10),
  5  insurance_id int not null,
  6  foreign key(insurance_id) references insurance(insurance_id)
  7  );
```

Table created.

```
SQL> insert into bills values('234567','234','605');
```

1 row created.

```
SQL> insert into bills values('234578','500','601');
```

1 row created.

```
SQL> insert into bills values('234589','489','607');
```

1 row created.

```
SQL> insert into bills values('234512','122','602');
```

1 row created.

```
SQL> insert into bills values('234523','523','601');
```

1 row created.

```
SQL> insert into bills values('234534','100','603');
```

1 row created.

```
SQL> insert into bills values('234545','443','607');
```

1 row created.

```
SQL> insert into bills values('234389','235','608');
```

1 row created.

```
SQL> insert into bills values('234553','765','609');
```

1 row created.

```
SQL> insert into bills values('234999','565','602');
```

9. Creating the stock table and inserting tuples into it:

```
create table stock(  
    dept_id int NOT NULL primary key,  
    ph_code varchar2(20) NOT NULL,  
    item_id NUMBER NOT NULL,  
    foreign key(ph_code) references hetero(ph_code),  
    foreign key(item_id) references inventory(item_id)  
);
```

```
insert into stock values('01','A1234','1000');
```

```
insert into stock values('02','A1236','1003');
```

```
insert into stock values('03','A1239','1002');
```

```
insert into stock values('04','A1237','1001');
```

```
insert into stock values('05','A1240','1000');
```

```
insert into stock values('06','A1242','1005');
```

```
insert into stock values('07','A1241','1007');
```

```
insert into stock values('08','A1243','1008');
```

```
insert into stock values('09','A1236','1009');
```

```
insert into stock values('10','A1235','1006');
```

```
SQL> create table stock(  
 2     dept_id int NOT NULL primary key,  
 3     ph_code varchar2(20) NOT NULL,  
 4     item_id NUMBER NOT NULL,  
 5     foreign key(ph_code) references hetero(ph_code),  
 6     foreign key(item_id) references inventory(item_id)  
 7 );
```

Table created.

```
SQL> insert into stock values('01','A1234','1000');
```

1 row created.

```
SQL> insert into stock values('02','A1236','1003');
```

1 row created.

```
SQL> insert into stock values('03','A1239','1002');
```

1 row created.

```
SQL> insert into stock values('04','A1237','1001');
```

1 row created.

```
SQL> insert into stock values('05','A1240','1000');
```

1 row created.

```
SQL> insert into stock values('06','A1242','1005');
```

1 row created.

```
SQL> insert into stock values('07','A1241','1007');
```

1 row created.

```
SQL> insert into stock values('08','A1243','1008');
```

1 row created.

```
SQL> insert into stock values('09','A1236','1009');
```

1 row created.

```
SQL> insert into stock values('10','A1235','1006');
```

1 row created.

```
SQL> |
```


10. Creating the purchase table and inserting tuples into it:

```
create table purchase(  
    purchase_id int NOT NULL primary key,  
    trans_id varchar2(10) NOT NULL,  
    prescription_id varchar2(20) NOT NULL ,  
    patient_id varchar2(20) NOT NULL,  
    foreign key(trans_id) references bills(trans_id),  
    foreign key(prescription_id) references prescription(prescription_id),  
    foreign key(patient_id) references patient(patient_id)  
);
```

```
SQL> create table purchase(  
2     purchase_id int NOT NULL primary key,  
3     trans_id varchar2(10) NOT NULL,  
4     prescription_id varchar2(20) NOT NULL ,  
5     patient_id varchar2(20) NOT NULL,  
6     foreign key(trans_id) references bills(trans_id),  
7     foreign key(prescription_id) references prescription(prescription_id),  
8     foreign key(patient_id) references patient(patient_id)  
9 );  
  
Table created.
```

```
insert into purchase values('11','234567','302','2020');  
  
insert into purchase values('22','234589','305','2012');  
  
insert into purchase values('33','234523','307','2019');  
  
insert into purchase values('44','234534','301','2017');  
  
insert into purchase values('55','234545','304','2013');  
  
insert into purchase values('66','234567','310','2015');  
  
insert into purchase values('77','234553','301','2014');  
  
insert into purchase values('88','234999','309','2020');  
  
insert into purchase values('99','234999','307','2022');  
  
insert into purchase values('100','234567','303','2015');
```

```

SQL> insert into purchase values('11','234567','302','2020');
1 row created.

SQL> insert into purchase values('22','234589','305','2012');
1 row created.

SQL> insert into purchase values('33','234523','307','2019');
1 row created.

SQL> insert into purchase values('44','234534','301','2017');
1 row created.

SQL> insert into purchase values('55','234545','304','2013');
1 row created.

SQL> insert into purchase values('66','234567','310','2015');
1 row created.

SQL> insert into purchase values('77','234553','301','2014');
1 row created.

SQL> insert into purchase values('88','234999','309','2020');
1 row created.

SQL> insert into purchase values('99','234999','307','2022');
1 row created.

SQL> insert into purchase values('100','234567','303','2015');
1 row created.

SQL> |

```

11. Creating the treatment table and inserting tuples into it:

create table treatment(

token_number NUMBER primary key,

prescription_id varchar2(20) NOT NULL,

patient_id varchar2(20) NOT NULL,

doctor_id NUMBER NOT NULL,

foreign key(prescription_id) references prescription(prescription_id),

foreign key(patient_id) references patient(patient_id),

foreign key(doctor_id) references Doctor(doctor_id)

);

```
SQL> create table treatment(  
2     token_number NUMBER primary key,  
3     prescription_id varchar2(20) NOT NULL,  
4     patient_id varchar2(20) NOT NULL,  
5     doctor_id NUMBER NOT NULL,  
6     foreign key(prescription_id) references prescription(prescription_id),  
7     foreign key(patient_id) references patient(patient_id),  
8     foreign key(doctor_id) references Doctor(doctor_id)  
9 );
```

Table created.

```
SQL> )|
```

```
7     foreign key(patient_id) references patient(patient_id)  
8 );
```

Table created.

```
SQL> insert into treatment values (TO_DATE('03-06-2023','DD-MM-YYYY'), 700, 2012, 301);
```

1 row created.

```
SQL> insert into treatment values (TO_DATE('04-03-2023','DD-MM-YYYY'), 701, 2013, 302);
```

1 row created.

```
SQL> insert into treatment values (TO_DATE('05-05-2023','DD-MM-YYYY'), 702, 2014, 303);
```

1 row created.

```
SQL> insert into treatment values (TO_DATE('06-07-2023','DD-MM-YYYY'), 703, 2015, 304);
```

1 row created.

```
SQL> insert into treatment values (TO_DATE('07-08-2023','DD-MM-YYYY'), 704, 2016, 305);
```

1 row created.

```
SQL> insert into treatment values (TO_DATE('09-09-2023','DD-MM-YYYY'), 705, 2017, 306);
```

1 row created.

```
SQL> insert into treatment values (TO_DATE('13-10-2023','DD-MM-YYYY'), 706, 2019, 307);
```

1 row created.

```
SQL> insert into treatment values (TO_DATE('23-11-2023','DD-MM-YYYY'), 707, 2020, 308);
```

1 row created.

```
SQL> insert into treatment values (TO_DATE('30-12-2023','DD-MM-YYYY'), 708, 2021, 309);
```

1 row created.

12. Creating the insurance_claim table and inserting tuples into it:

```
create table insurance_claim(  
patient_id varchar2(20) not null,  
insurance_id int not null,  
validity varchar2(20),  
foreign key(patient_id) references patient(patient_id),  
foreign key(insurance_id) references insurance(insurance_id)  
);
```

```
insert into insurance_claim values(2012, 601, Yes);  
insert into insurance_claim values(2013, 602, Yes);  
insert into insurance_claim values(2015, 603, No);  
insert into insurance_claim values(2014, 604, Yes);  
insert into insurance_claim values(2016, 605, Yes);  
insert into insurance_claim values(2017, 606, Yes);  
insert into insurance_claim values(2019, 607, No);  
insert into insurance_claim values(2020, 608, Yes);  
insert into insurance_claim values(2022, 609, Yes);  
insert into insurance_claim values(2021, 610, No);
```

```
SQL> create table insurance_claim(  
  2   patient_id varchar2(20) not null,  
  3   insurance_id int not null,  
  4   validity varchar2(20),  
  5   foreign key(patient_id) references patient(patient_id),  
  6   foreign key(insurance_id) references insurance(insurance_id)  
  7   );
```

Table created.

Table truncated.

```
SQL> insert into insurance_claim values('2012', '601', 'Yes');
```

1 row created.

```
SQL> insert into insurance_claim values('2013', '602', 'Yes');
```

1 row created.

```
SQL> insert into insurance_claim values('2015', '603', 'No');
```

1 row created.

```
SQL> insert into insurance_claim values('2014', '604', 'Yes');
```

1 row created.

```
SQL> insert into insurance_claim values('2016', '605', 'Yes');
```

1 row created.

```
SQL> insert into insurance_claim values('2017', '606', 'Yes');
```

1 row created.

```
SQL> insert into insurance_claim values('2019', '607', 'No');
```

1 row created.

```
SQL> insert into insurance_claim values('2020', '608', 'Yes');
```

1 row created.

```
SQL> insert into insurance_claim values('2022', '609', 'Yes');
```

1 row created.

```
SQL> insert into insurance_claim values('2021', '610', 'No');
```

1 row created.

```
SQL> |
```

Contribution towards the Project:

- My Contribution towards this project is that I helped the team in updating the description and developing the ER Diagram and Database Schema.
- I took up the data related to tables of Treatment, Prescription, Purchase.
- I created the tables Treatment, Prescription, Purchase and inserted atleast 10 tuples in each of the table.
- As this is an overall team work, I contributed by best to help others complete their tasks as well to complete the project part -2.

Prescription:

- I assumed that Prescription table stores information of the prescription given by a patient.
- Prescription has attributes Prescription_id, Date, where prescription_id is assumed as the primary Key.
- Prescription is in many to one relationship with the Hetero Pharmacy, Hence a foreign key from Hetero Table is carried to Prescription. Therefore, The pharmacy registration code “Ph_Code” is the foreign key in Prescription.

Table Treatment”

- Table Treatment is a ternary relationship table. Treatment table is formed as due to ternary relationship between prescription, patient and doctor.
- Treatment has the attribute as “t_date” which is assumed as the primary key
- Treatment table stores the information of patient, doctor, and the date on which the patient consults doctor.
- Hence, the patient_id and doctor_id are assumed as the Foreign Keys in treatment table.

Purchase:

- Purchase is also another ternary table which is formed from ternary relationship between prescription, bills, patient.
- Purchase stores the information related to the purchases made by the patient.
- Here, Purchase has attribute “Purchase_id” which is assumed as the Primary Key.
- Purchase has foreign keys transaction_id from bills, patient_id, prescription_id.