

# DBMS - Mini Project

## Pharmaceutical Database

Submitted By:

Name Naman Choudhary

SRN PES2UG20CS209

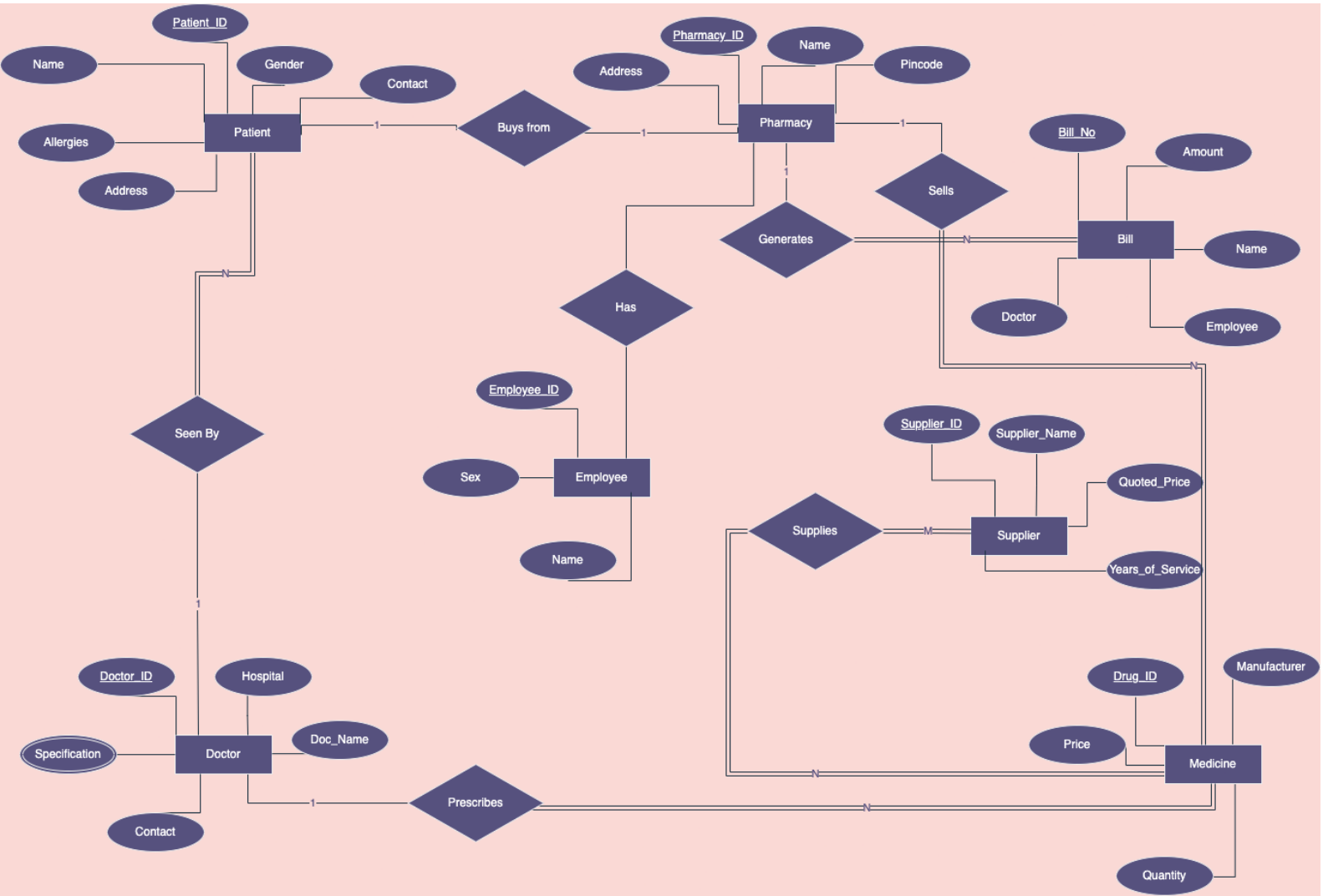
V Semester Section D

## **Short Description and Scope of the Project**

The Pharmaceutical management system will help Pharmacists across the country to manage, organise and keep track of medicine available in inventory and in the market. This will help Pharmacists to never be low on stock of medicine, as unavailability of medicine can put people suffering from long-term and short-term illnesses at risk. This system will also help in maintaining and tallying the expenditure as well as, keeping track of user information about required medicine of repeated customers

With this project, my aim was to develop a comprehensive system that could deal with challenges faced in day to day operation of a modern pharmacy.

# ER Diagram



# Relational Schema

## Relationship Schema

### Patient

<u>Patient_ID</u>	Name	Gender	Allergies	Contact	Address	Doctor_ID
-------------------	------	--------	-----------	---------	---------	-----------

### Doctor

<u>Doctor_ID</u>	Doc_Name	Hospital	Contact
------------------	----------	----------	---------

### Pharmacy

<u>Pharmacy_ID</u>	Name	Address	Pincode	<u>Patient_ID</u>	<u>Employee_ID</u>
--------------------	------	---------	---------	-------------------	--------------------

### Medicine

<u>Drug_ID</u>	Manufacturer	Quantity	Price	Doctor_ID	<u>Pharmacy_ID</u>
----------------	--------------	----------	-------	-----------	--------------------

### Supplier

<u>Supplier_ID</u>	Supplier_Name	Quoted_Price	Years_of_service
--------------------	---------------	--------------	------------------

### Employee

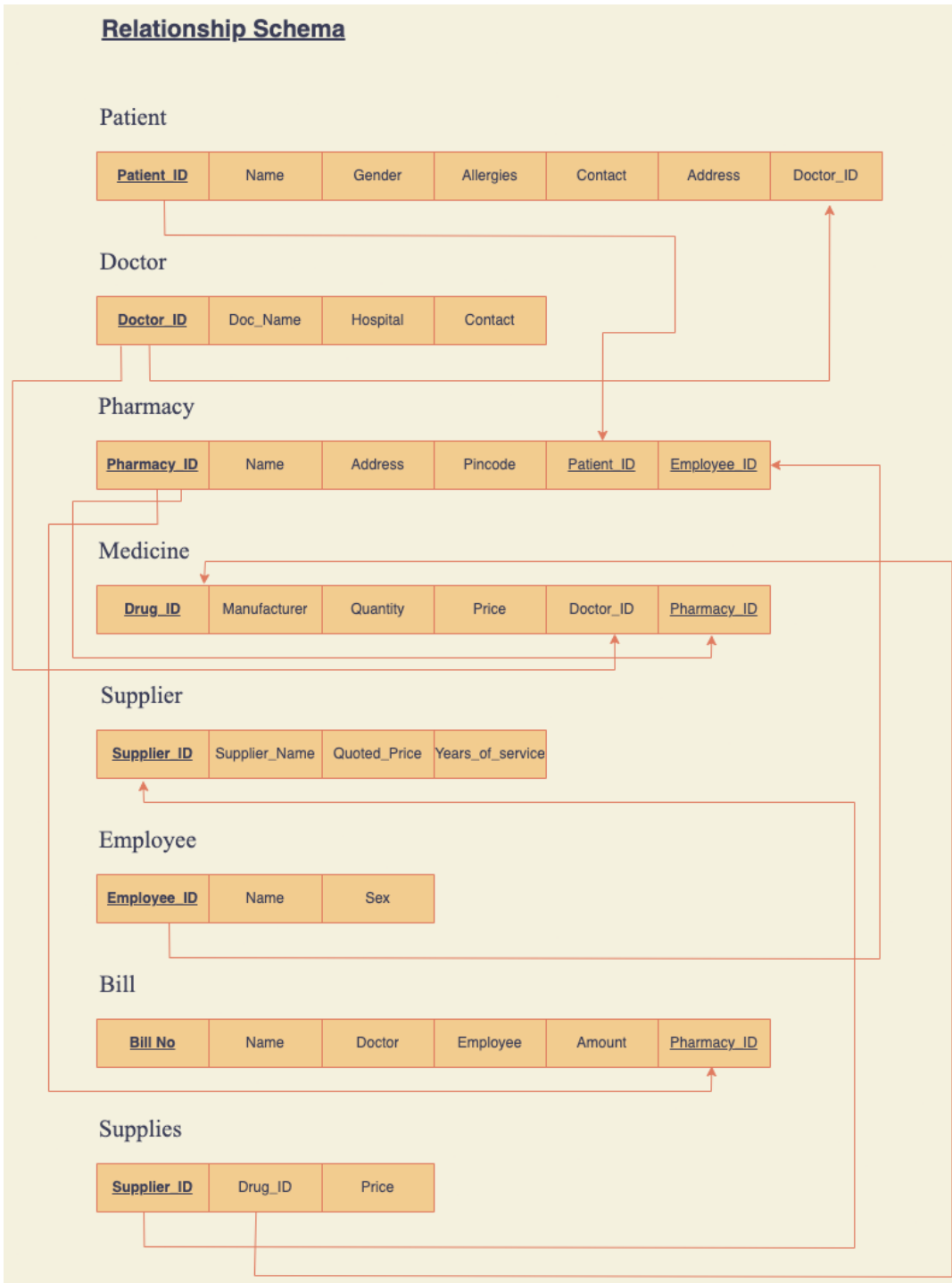
<u>Employee_ID</u>	Name	Sex
--------------------	------	-----

### Bill

<u>Bill_No</u>	Name	Doctor	Employee	Amount	<u>Pharmacy_ID</u>
----------------	------	--------	----------	--------	--------------------

### Supplies

<u>Supplier_ID</u>	Drug_ID	Price
--------------------	---------	-------



## DDL statements - Building the database

SQL

-- Table creation with DDL commands

```
CREATE TABLE CUSTOMER (  
  Aadhaar BIGINT NOT NULL,  
  first_name CHAR(255) NOT NULL,  
  last_name CHAR(255) NOT NULL,  
  phone BIGINT NOT NULL UNIQUE,  
  gender CHAR(1) NOT NULL,  
  address CHAR(255) NOT NULL,  
  date_of_birth DATE NOT NULL,  
  insurance_id BIGINT NOT NULL,  
  PRIMARY KEY (Aadhaar)  
);
```

```
ALTER TABLE Customer  
ADD CONSTRAINT insures FOREIGN KEY (insurance_id) REFERENCES Insurance  
ON DELETE CASCADE;
```

```
,CREATE TABLE Prescription (  
  prescription_id BIGINT NOT NULL,  
  Aadhaar BIGINT NOT NULL,  
  doctor_id BIGINT NOT NULL,  
  prescribed_date DATE NOT NULL,  
  PRIMARY KEY (prescription_id)  
);
```

```
ALTER TABLE Prescription  
ADD CONSTRAINT holds FOREIGN KEY (Aadhaar) REFERENCES Customer (Aadhaar)  
ON DELETE CASCADE;
```

```
CREATE TABLE PRESCRIBED_DRUGS (  
  prescription_id BIGINT NOT NULL,  
  drug_name CHAR(255) NOT NULL,  
  prescribed_quantity BIGINT NOT NULL,  
  refill_limit BIGINT NOT NULL,  
  PRIMARY KEY (prescription_id, drug_name)  
);
```

```
ALTER TABLE PRESCRIBED_DRUGS  
ADD CONSTRAINT consists_of FOREIGN KEY (prescription_id)  
REFERENCES Prescription (prescription_id) ON DELETE CASCADE;
```

```
CREATE TABLE Orders (  
  -- Table creation with DDL commands
```

```
CREATE TABLE Orders (  
  order_id BIGINT NOT NULL,  
  prescription_id BIGINT NOT NULL,  
  EmployeeID BIGINT NOT NULL,  
  order_date DATE NOT NULL,  
  PRIMARY KEY (order_id)  
);  
  
ALTER TABLE Orders  
ADD CONSTRAINT prepares FOREIGN KEY (EmployeeID) REFERENCES Employee (ID)  
  
ALTER TABLE Orders  
ADD CONSTRAINT uses FOREIGN KEY (prescription_id) REFERENCES  
Prescription (prescription_id);  
  
CREATE TABLE ORDERED_DRUGS (  
  order_id BIGINT NOT NULL,  
  drug_name CHAR(255) NOT NULL,  
  batch_number BIGINT NOT NULL,  
  ordered_quantity BIGINT NOT NULL,  
  Price BIGINT NOT NULL,  
  PRIMARY KEY (order_id, drug_name, batch_number)  
);  
  
ALTER TABLE ORDERED_DRUGS  
ADD CONSTRAINT containss FOREIGN KEY (order_id) REFERENCES Orders (order_id)  
ON DELETE CASCADE;  
  
ALTER TABLE ORDERED_DRUGS  
ADD CONSTRAINT Fulfilled_From FOREIGN KEY (drug_name, batch_number)  
REFERENCES Medicine(drug_name, batch_number);  
  
CREATE TABLE Insurance (  
  insurance_id BIGINT NOT NULL,  
  company_name CHAR(255) NOT NULL,  
  start_date DATE NOT NULL,  
  end_date DATE NOT NULL,  
  co_insurance BIGINT NOT NULL,  
  PRIMARY KEY (insurance_id)  
);  
  
CREATE INDEX Insurance_Company_Name ON Insurance (company_name);
```

```
CREATE TABLE Employee (  
  ID BIGINT NOT NULL,  
  Aadhaar BIGINT NOT NULL,  
  License BIGINT UNIQUE,  
  first_name CHAR(255) NOT NULL,  
  last_name CHAR(255) NOT NULL,  
  start_date DATE NOT NULL,  
  end_date DATE,  
  role CHAR(255) NOT NULL,  
  salary BIGINT NOT NULL,  
  phone_number BIGINT NOT NULL,  
  date_of_birth DATE NOT NULL,  
  PRIMARY KEY (ID)  
);  
  
CREATE TABLE Medicine (  
  drug_name CHAR(255) NOT NULL,  
  batch_number BIGINT NOT NULL,  
  MedicineType CHAR(255) NOT NULL,  
  Manufacturer CHAR(255) NOT NULL,  
  stock_quantity BIGINT NOT NULL,  
  expiry_date DATE NOT NULL,  
  Price BIGINT NOT NULL,  
  PRIMARY KEY (drug_name, batch_number)  
);  
  
CREATE TABLE Bill (  
  order_id BIGINT NOT NULL,  
  Customer_Aadhaar BIGINT NOT NULL,  
  total_amount BIGINT NOT NULL,  
  customer_payment BIGINT NOT NULL,  
  insurance_payment BIGINT NOT NULL,  
  PRIMARY KEY (order_id, Customer_Aadhaar)  
);  
  
ALTER TABLE Bill  
ADD CONSTRAINT makes FOREIGN KEY (order_id) REFERENCES Orders (order_id)  
  
ALTER TABLE Bill  
ADD CONSTRAINT pays FOREIGN KEY (Customer_Aadhaar)  
REFERENCES Customer (Aadhaar);
```



## Populating the Database

SQL

```
-- Insertion

-- Normal Insert into all Columns

INSERT INTO Customer(Aadhaar,first_name,last_name,
phone,gender,address,date_of_birth,insurance_id)
VALUES
(784526471835,'Shyam','Bhat',7648964389,'M',
"Hosa Road,Electronic City",'2002-09-03',55446677),
(6489075453,'Ravi','Singh',8976785649,'M',
"Nice Road,Electronic City",'2002-02-13',66554433),
(546790987890,'Reha','Lal',9966442211,'F',
"Sarjapur Road,Whitefield",'2002-10-17',99006655)

INSERT INTO Insurance(insurance_id,
company_name,start_date,end_date,co_insurance)
VALUES
(55446677,"Adani Insurance",'2002-09-03','2102-09-03',9804),
(66554433,"Reliance Insurance",'2002-02-13','2102-02-13',7593),
(99006655,"Big Insurance",'2002-10-17','2102-10-17',8694)

-- Insert into specific columns

INSERT INTO Employee(ID,Aadhaar,License,first_name,
last_name,start_date,role,salary,phone_number,date_of_birth)
VALUES
(732,658361987523,9875,'Rahul','Gupta','2021-09-24',
'intern',60000,8899786432,'1992-06-19'),
(758,658361985378,8295,'Veer','Das','2015-09-24',
'pharmacist',70000,8899786432,'1992-06-19')

-- UI based insert - depecited in image

-- Medicine table

INSERT INTO Medicine(drug_name,batch_number,MedicineType,
Manufacturer,stock_quantity,expiry_date,Price)
VALUES
('Dolo-650',788907,'Paracetamol','Micro Labs',800,'2023-02-12',31),
('Crocicn',893145,'Targetted Pain','GSK',800,'2022-01-02',20)
```



## Join Queries

Showcase at least 4 join queries

Write the query in English Language, Show the equivalent SQL statement and also a screenshot of the query and the results

## **Aggregate Functions**

Showcase at least 4 Aggregate function queries

Write the query in English Language, Show the equivalent SQL statement and also a screenshot of the query and the results

## **Set Operations**

Showcase at least 4 Set Operations queries

Write the query in English Language, Show the equivalent SQL statement and also a screenshot of the query and the results

## Functions and Procedures

Create a Function and Procedure. State the objective of the function / Procedure. Run and display the results.

SQL

```
CREATE PROCEDURE `GENERATE_BILL`(IN `order_id` BIGINT,
IN `aadhaar` BIGINT, IN `insurance_id` BIGINT)
NOT DETERMINISTIC CONTAINS SQL SQL SECURITY DEFINER BEGIN
DECLARE total_amount BIGINT;
DECLARE copayment_percentage BIGINT;
DECLARE copayment_amount BIGINT; -- this is the amount insurance company pays
DECLARE customer_payment BIGINT; -- this is the amount customer pays
-- do a total of all orders
SELECT SUM('price')
INTO total_amount
FROM ORDERED_DRUGS
WHERE 'order_id' = order_id;
-- get insurance details
SELECT co_insurance
INTO copayment_percentage
FROM INSURANCE
WHERE 'insurance_id' = insurance_id;
-- the insurance company will pay this amount
SET copayment_amount = total_amount * copayment_percentage;
-- the customer will pay this amount
SET customer_payment = total_amount * (1 - copayment_percentage);
-- Insert data
INSERT INTO BILL VALUES (order_id, aadhaar, total_amount,
customer_payment, copayment_amount);
END;
```

Objective: Generates a bill for the user and saves this bill into the BILL table

Parameters

Execute routine `GENERATE\_BILL`

Routine parameters

Name	Type	Function	Value
order_id	BIGINT		984324
aadhaar	BIGINT		784526471835
insurance_id	BIGINT		55446677

Go

Close

Output

	order_id	Customer_Aadhaar	total_amount	customer_payment	insurance_payment
<div><div></div><div>Edit</div><div>Copy</div><div>Delete</div></div>	984324	784526471835	51	40	11

## Triggers and Cursors

Create a Trigger and a Cursor. State the objective. Run and display the results.

SQL

```
CREATE OR REPLACE TRIGGER Validate_Employee BEFORE INSERT OR
UPDATE ON EMPLOYEE
FOR EACH ROW
BEGIN
IF LOWER(:NEW.role) != 'cashier' OR
LOWER(:NEW.role != 'pharmacist') OR
LOWER(:NEW.role != 'cpht') OR
LOWER(:NEW.role != 'intern') THEN
RAISE_APPLICATION_ERROR(-20000, 'Invalid role given for employee');
END IF;
IF :NEW.license := NULL AND LOWER(:new.role) != 'cashier' THEN
RAISE_APPLICATION_ERROR(-20000,
'Can not leave license blank for anyone except cashiers'); END IF;
END;
```

Objective: Checks if the employee role is valid or not

Possible values-> Pharmacist,CPhT,Intern,Cashier

### Error

SQL query: [Copy](#)

```
INSERT INTO Employee(ID,Aadhaar,License,first_name,last_name,start_date,role,salary,phone_number,date_of_birth)
VALUES
(732,658361987523,9875,'Rahul','Gupta','2021-09-24','human',60000,8899786432,'1992-06-19');
```

MySQL said: 

#2000 - Invalid role given for employee

## Developing a Frontend

The frontend should support

1. Addition, Modification and Deletion of records from any chosen table
2. There should be an window to accept and run any SQL statement and display the result

The screenshot shows a web application titled "Pharmacy Database Management System" by Naman Choudhary at PES University, Electronic City. The interface features a central form with two main sections: "Medicine INFO" and "Medicine DETAILS".

**Medicine INFO** fields:

- Drug Name:
- Batch Number:
- Medicine Type:
- Manufacturer:
- Stock Quantity:
- Expiry Date:
- Price:

**Medicine DETAILS** (displayed as a list):

- Crocin 893145 {Targetted Pain} GSK 800 2022-01-02 20
- Dolo-650 788907 Paracetamol {Micro Labs} 800 2023-02-12 31
- Test1 111 Histamine {Micro Labs} 800 2023-02-12 31
- Test2 222 Targetted GSK 300 2022-01-02 20

At the bottom, there is a navigation bar with the following buttons: Add New, Display, Clear, Delete, Search, Update, and Exit.