Thapar Institute of Engineering and Technology, Patiala

Computer Science and Engineering Department

Database Management System (UCS310)

Project Report

PHARMACY MANAGEMENT SYSTEM

Submitted to - Dr. Ranjit

Sub-Group - 2CO6

TEAM

Muskan Chalana	102203274
Asmi Gaurav	102203253
Ritish Mahajan	102203250
Pranav Khurana	102203236

INDEX

Contents

Introduction

The primary objective of the Pharmacy Management System (PMS) project is to develop a comprehensive database management system tailored specifically for pharmacies. This system aims to streamline inventory management, prescription tracking, patient information management, and sales monitoring processes. By providing a centralized platform for storing and accessing crucial data, the PMS enhances operational efficiency, minimizes errors, and improves the overall quality of service provided by pharmacies.

Pharmacies operate under unique regulations governing drug sales and management. For instance, many medications require prescriptions, with limits on purchase quantities. Pharmacists must conduct background checks on customers' medical histories to prevent drug abuse. Additionally, pharmacies must adhere to laws mandating safe disposal of expired medicine and licensing requirements for employees involved in drug preparation.

Thus, developing a Pharmacy Management System involves understanding both operational and legal aspects. Our project involved extensive research into relevant laws to ensure compliance at both Federal and State levels. The system we've developed is equipped to handle the complexities and challenges of modern pharmacy operations while meeting regulatory requirements.

Laws Affecting Pharmacies

There are several laws and regulations in India that affect pharmacies and the practice of pharmacy. Some of the key laws and regulations are:

1. *Drugs and Cosmetics Rules, 1945*: These rules provide detailed guidance on the provisions of the Drugs and Cosmetics Act, including requirements for the labeling, packaging, and advertising of drugs, and the licensing of manufacturers, importers.

2. *Pharmacy Act, 1948*: This act governs the education, registration, and practice of pharmacists in India. It establishes the Pharmacy Council of India (PCI) and the State Pharmacy Councils, which are responsible for maintaining the standards of pharmacy education and practice.

3. *Narcotic Drugs and Psychotropic Substances (NDPS) Act, 1985*: This act regulates the manufacture, distribution, and use of narcotic drugs and psychotropic substances in India.

4. *Clinical Establishment Act, 2010*: This act regulates the establishment and operation of clinical establishments, including pharmacies, in India. It sets out standards for the quality of care provided by clinical establishments and establishes the National Council for Clinical Establishments (NCCE), which is responsible for enforcing the provisions of the act.

Objective

The primary objective of the Pharmacy Management System project is to develop a comprehensive database management system tailored specifically for pharmacies. This system will streamline the processes involved in inventory management, prescription tracking, patient information management, and sales monitoring. By providing a centralized platform for storing and accessing crucial data, the PMS will enhance operational efficiency, minimize errors, and ultimately improve the overall quality of service provided by pharmacies.

Key Features

1. **Inventory Management**: The system will enable pharmacists to efficiently manage their inventory by keeping track of stock levels, expiration dates, and reordering thresholds. Automatic alerts and notifications will be implemented to ensure timely replenishment of stock and prevent shortages.

2. **Prescription Tracking**: Pharmacists will be able to record and track prescriptions, including details such as medication dosage, refill status, and patient information. This feature will facilitate accurate dispensing of medications and help in monitoring patient adherence to prescribed treatments.

3. **Patient Information Management**: The PMS will provide a secure database for storing and managing patient information, including medical history, allergies, and insurance details. Pharmacists will be able to retrieve patient records quickly, enabling personalized care and medication counseling.

4. **Sales Monitoring**: The system will track sales transactions, including prescription medications, over-the-counter products, and other pharmaceutical items. Comprehensive reporting tools will be integrated to analyze sales trends, monitor profitability, and identify opportunities for cost-saving measures.

Requirements

During research phase, we arrived at following requirements based on the pharmacy flow:

Customer

When a customer arrives in the pharmacy, we identify them based on their SSN(Serial number). If they are a new customer, they are asked for their name, date of birth, phone number, gender and address. The address and date of birth are required to be recorded for drug control purposes under The National Health Policy, 2017.

Insurance

As of 2021, the Pradhan Mantri Jan Arogya Yojana (PMJAY), which is a government-sponsored health insurance scheme that provides free healthcare coverage for up to Rs. 5 lakhs per family per year for around 500 million people, has been rolled out across the country. According to the government, the scheme has already benefited millions of people who were previously uninsured or underinsured.

If a customer has health insurance, we store the insurance ID (unique for each customer), company name, start date, end date and Co-Insurance. Co-Insurance is a percentage amount that an insurance company pays for a medicinal purchase (Managing your healthcare costs, n.d.). Given the customer SSN and insurance ID, the system should be able to automatically calculate the amount paid by the insurance company and customer. Overall, while the percentage of Indians who have health insurance coverage is not known exactly, it is clear that there is a significant need for expanded health insurance coverage in the country, particularly for vulnerable and underserved populations.

Employee

An employee has the same details as a customer but they are also given a company ID, that is unique for them. An employee has to have one of the following roles:

1. Pharmacist
2. Intern (can work in the pharmacy part time)
3. Cashier

Apart from cashier, all other roles require a license from the State's Medical Board as they directly deal with mixing and preparation of drugs.

Prescription

Most of the drugs in the pharmacy can only be sold with a prescription. A prescription contains the customer's SSN, the prescribing Doctor’s ID (required by law) and when the prescription was prescribed.

Each prescription contains a number of prescribed drugs with drug name and quantity. By law, a pharmacy cannot sell more than prescribed quantity or anything that is not listed on prescription. The prescription is required to be stored under the Narcotic Drugs and Psychotropic Substances (NDPS) Act, 1985.

Order

An order is created from the prescription. This data has to be stored separately because customer may buy less medicine than prescription specifies.

Each order has a unique Order ID that is automatically assigned by the system. Each order can have multiple drugs, each with their ordered quantity and price. We also record the batch number of the drug. This data can be requested by the government under the Narcotic Drugs and Psychotropic Substances (NDPS) Act, 1985 and has to be stored.

Bill

Once an order has been completed, a bill is generated by the system. This bill is handed over to the customer and contains order information, insurance information as well as breakdown of amount paid. The breakdown should be automatically calculated by the system based on insurance, customer and medicine data.

Medicine(Inventory)

The classification of drugs into different schedules is an important aspect of drug regulation in India, as it helps to ensure that drugs are used safely and appropriately, and that drugs with a higher potential for abuse and dependence are subject to stricter regulations.

Drugs are divided into “over the counter”, “restricted” and “prescription only”. While not needed by law everywhere, it is beneficial to store an up to date inventory for record keeping as well as knowing when we run out of stock.

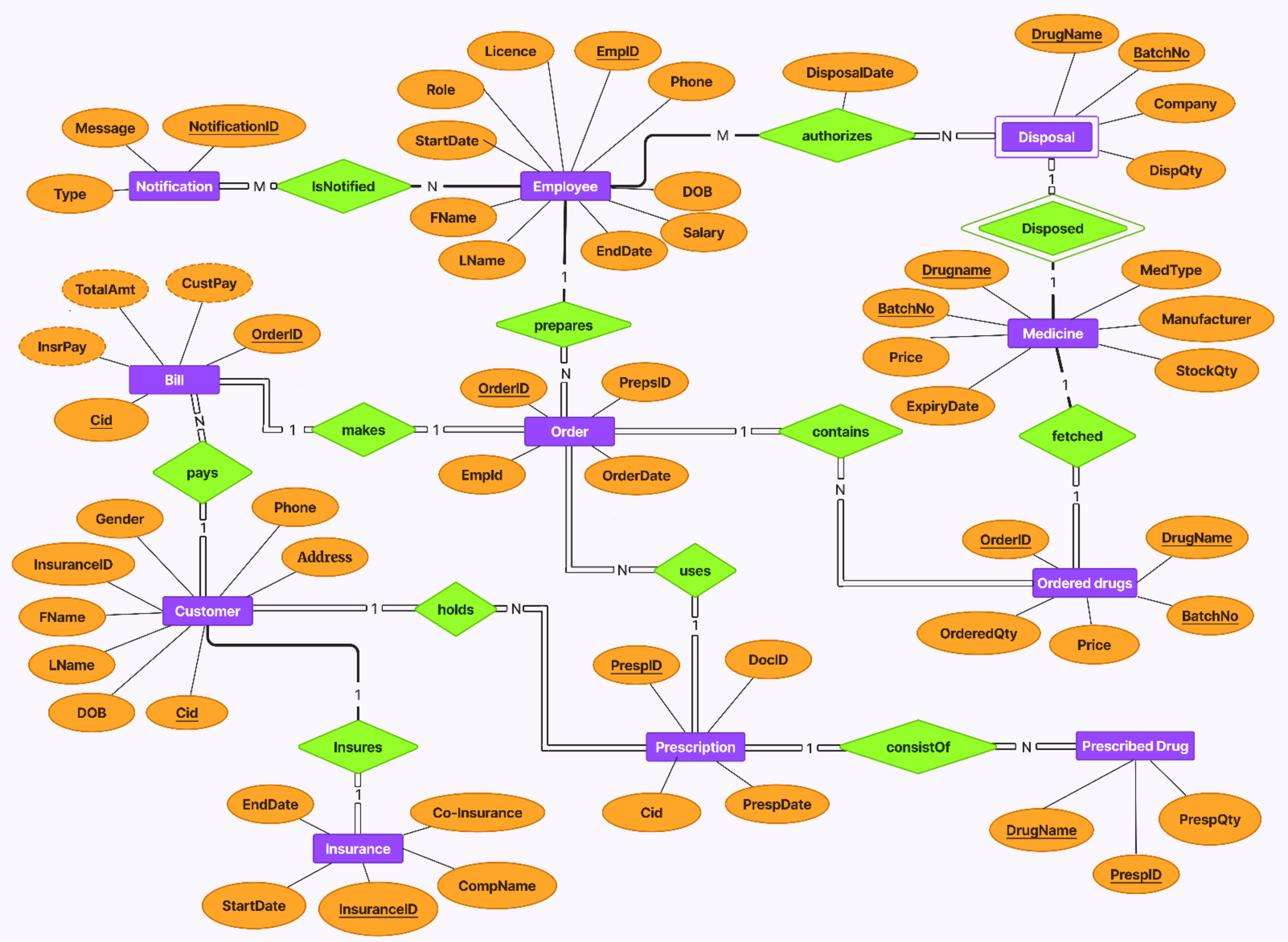
Notifications

The system should be able to generate notifications based on the following four events:

1. Stock for a medicine is low (less than 100 tablets)
2. Some medicine will expire in next 60 days
3. Drugs are marked for disposal
4. Drugs are successfully disposed

ER Modelling, Relations and Normalization

ER Diagram



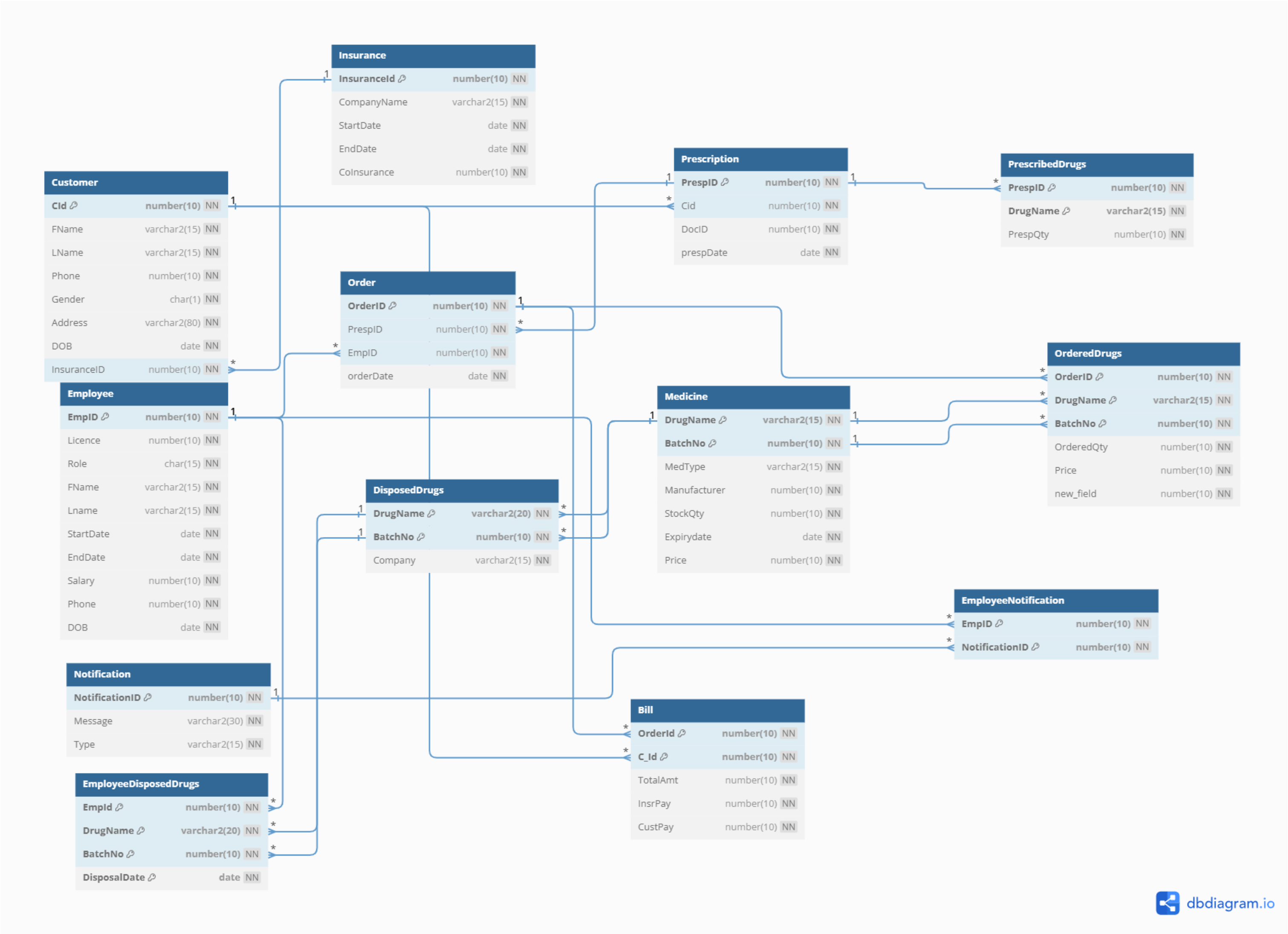
ER to RELATION

Entity 1	Name of the Relationship	Entity 2	Cardinality
Customer	Insures	Insurance	1:1
Customer	Holds	Prescription	1:N
Customer	Pays	Bill	1:N
Prescription	ConsistOf	Prescription Drug	1:N
Order	Makes	Bill	1:1
Order	Uses	Prescription	N:1
Order	Contains	Ordered Drugs	1:N

Ordered Drugs	Fetched	Medicine	1:1
Disposal	Dispose	Medicine	1:1
Employee	Authorizes	Disposal	M:N
Employee	Prepares	Order	1:N
Employee	isNotified	Notification	N:M

Relationship in the system	**Type of Binary Relationship**
1. A single customer can have multiple prescriptions. 2. A prescription consists of multiple drugs. 3. A prescription can generate multiple orders. 4. A single order can contain multiple drugs. 5. A customer can make multiple purchases. 6. One employee can prepare multiple orders. 7. A specific order can only be prepared by one employee.	one to many
1. One order can generate only one bill. 2. Batch number is assumed to be unique among manufacturers	One to one
1. One employee can receive multiple notifications and one notification can be sent to multiple employees. 2. Multiple employees can dispose same drug and one employee can dispose multiple drugs.	Many to many

SCHEMA



Insurance		
InsuranceId	**number(10)**	NN
CompanyName	varchar2(15)	NN
StartDate	date	NN
EndDate	date	NN
CoInsurance	number(10)	NN

Prescription		
PrespID	**number(10)**	NN
Cid	number(10)	NN
DocID	number(10)	NN
prespDate	date	NN

PrescribedDrugs		
PrespID	**number(10)**	NN
DrugName	**varchar2(15)**	NN
PrespQty	number(10)	NN

Customer		
CId	**number(10)**	NN
FName	varchar2(15)	NN
LName	varchar2(15)	NN
Phone	number(10)	NN
Gender	char(1)	NN
Address	varchar2(80)	NN
DOB	date	NN
InsuranceID	number(10)	NN

Order		
OrderID	**number(10)**	NN
PrespID	number(10)	NN
EmpID	number(10)	NN
orderDate	date	NN

OrderedDrugs		
OrderID	**number(10)**	NN
DrugName	**varchar2(15)**	NN
BatchNo	**number(10)**	NN
OrderedQty	number(10)	NN
Price	number(10)	NN
new_field	number(10)	NN

Employee		
EmpID	**number(10)**	NN
Licence	number(10)	NN
Role	char(15)	NN
FName	varchar2(15)	NN
Lname	varchar2(15)	NN
StartDate	date	NN
EndDate	date	NN
Salary	number(10)	NN
Phone	number(10)	NN
DOB	date	NN

Medicine		
DrugName	**varchar2(15)**	NN
BatchNo	**number(10)**	NN
MedType	varchar2(15)	NN
Manufacturer	number(10)	NN
StockQty	number(10)	NN
Expirydate	date	NN
Price	number(10)	NN

DisposedDrugs		
DrugName	**varchar2(20)**	NN
BatchNo	**number(10)**	NN
Company	varchar2(15)	NN

EmployeeNotification		
EmpID	**number(10)**	NN
NotificationID	**number(10)**	NN

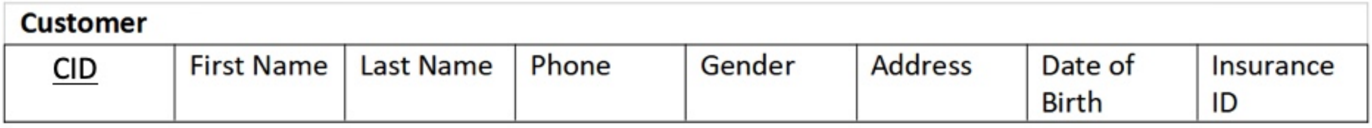
Notification		
NotificationID	**number(10)**	NN
Message	varchar2(30)	NN
Type	varchar2(15)	NN

Bill		
OrderId	**number(10)**	NN
C_Id	**number(10)**	NN
TotalAmt	number(10)	NN
InsrPay	number(10)	NN
CustPay	number(10)	NN

EmployeeDisposedDrugs		
EmpId	**number(10)**	NN
DrugName	**varchar2(20)**	NN
BatchNo	**number(10)**	NN
DisposalDate	**date**	NN

Relations

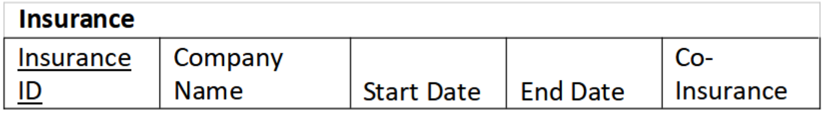
The final relations are listed below:



Customer

CID	First Name	Last Name	Phone	Gender	Address	Date of Birth	Insurance ID

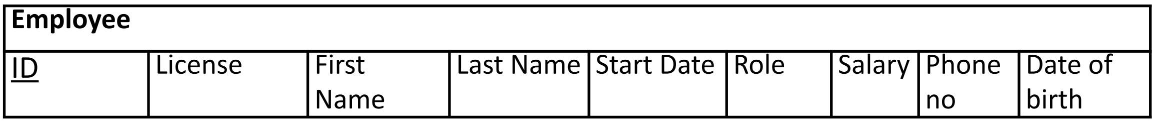
Primary Key: CID
Foreign Key: Customer(Insurance ID) →Insurance(Insurance ID)



Insurance

Insurance ID	Company Name	Start Date	End Date	Co-Insurance

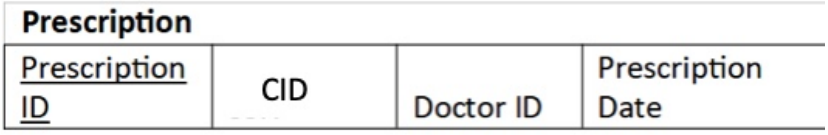
Primary Key: Insurance ID



Employee

ID	License	First Name	Last Name	Start Date	Role	Salary	Phone no	Date of birth

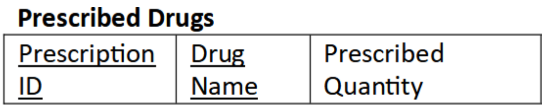
Primary Key: ID



Prescription

Prescription ID	CID	Doctor ID	Prescription Date

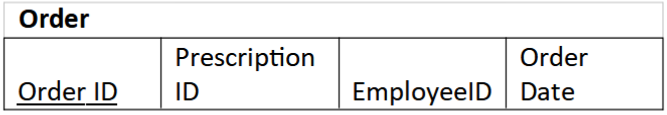
Primary Key: Prescription ID
Foreign Key: Prescription(CID)→Customer(CID)



Prescribed Drugs

Prescription ID	Drug Name	Prescribed Quantity

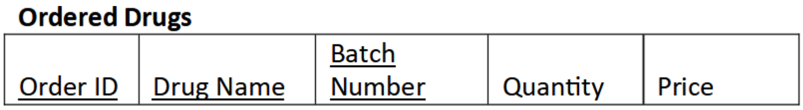
Primary Key: Prescription ID, Drug Name
Foreign Key: Prescribed Drugs(Prescription ID)→Prescription(Prescription ID)



Order

Order ID	Prescription ID	EmployeeID	Order Date

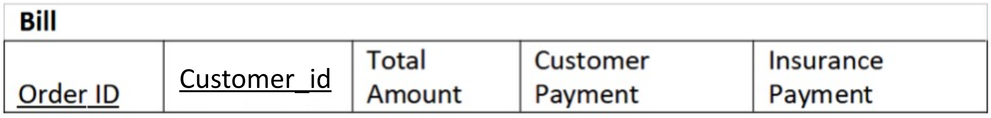
Primary Key: Order ID
Foreign Key: Order(Prescription ID)→Prescription(Prescription ID), Order(Employee ID)→ Employee(ID)



Ordered Drugs

Order ID	Drug Name	Batch Number	Quantity	Price

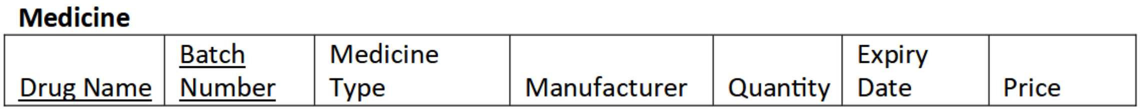
Primary Key: Order ID, Drug Name, Batch Number
Foreign Key: Ordered Drugs(Order ID)→Order(Order ID),
Ordered Drugs(Drug Name, Batch Number)→Medicine(Drug Name, Batch Number)



Bill

Order ID	Customer_id	Total Amount	Customer Payment	Insurance Payment

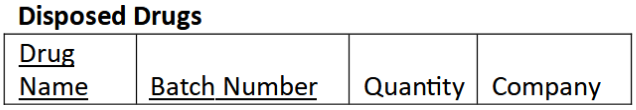
Primary Key: Order ID, Customer_ID
Foreign Key: Bill(Order ID)→Order(Order ID), Bill(Customer_ID)→Customer(CID)



Medicine

Drug Name	Batch Number	Medicine Type	Manufacturer	Quantity	Expiry Date	Price

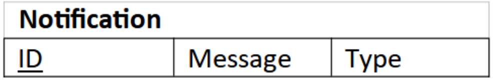
Primary Key: Drug Name, Batch Number



Disposed Drugs

Drug Name	Batch Number	Quantity	Company

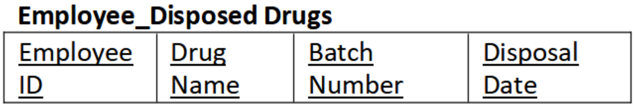
Primary Key: Drug Name, Batch Number
Foreign Key: Disposed Drugs(Drug Name, Batch Number)→Medicine(Drug Name, Batch Number)



Notification

ID	Message	Type

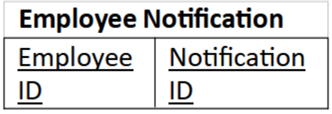
Primary Key: ID



Employee_Disposed Drugs

Employee ID	Drug Name	Batch Number	Disposal Date

Primary Key: Employee ID, Drug Name, Batch Number, Disposal Date
Foreign Key: Employee_Disposed Drugs(Employee ID)→Employee (Employee ID), Employee_Disposed Drugs(Drug Name, Batch Number)→Disposed Drugs(Drug Name, Batch Number)



Employee Notification

Employee ID	Notification ID

Primary: Employee ID, Notification ID
Foreign Key: Employee Notification(Employee ID)→Employee(ID), Employee Notification(Notification ID)→Notification(Notification ID)

Normalization

The following dependencies exist in our schema:

1. Insurance(Insurance ID, Company Name, Start Date, End Date, Co-Insurance)
   Insurance ID→Company Name, Start Date, End Date, Co-Insurance

2. Customer(Cid, First Name, Last Name, Phone, Gender, Address, Date of Birth, Insurance ID)
   Cid→First Name, Last Name, Phone, Gender, Address, Date of Birth, Insurance ID

3. Prescription(Prescription ID, Cid, Doctor ID, Prescribed Date)
   Prescription ID→Cid, Doctor ID, Prescribed Date

4. Prescribed_Drugs(Prescription ID, Drug Name, Prescribed Quantity)
   Prescription ID, Drug Name→Prescribed Quantity

5. Order(Order ID, Prescription ID, Employee ID, Order Date)
   Order ID→Prescription ID, Employee ID, Order Date

6. Ordered Drugs(Order ID, Dug Name, Batch Number, Ordered Quantity, Price)
   Order ID, Dug Name, Batch Number→Ordered Quantity, Price

7. Bill(Order ID, C_id, Total Amount, Customer Payment, Insurance Payment) Order ID, C_id→Total Amount, Customer Payment, Insurance Payment

8. Employee(Employee ID, First Name, Last Name, Start Date, End Date, Role, Salary, Phone Number, Date of Birth)
   Employee ID→ First Name, Last Name, Start Date, End Date, Role, Salary, Phone Number, Date of Birth

9. Employee_Notification(Employee ID, Notification ID)
   All Keys.

10. Notification(Notification ID, Type, Message)
    Notification ID→Type, Message

11. Employee_Disposed_Drugs(Employee ID, Drug Name, Batch Number, Disposal Date)
    All Keys.

12. Disposed Drugs(Drug Name, Batch Number, Quantity, Company)
    Drug Name, Batch Number→Quantity, Company

13. Medicine(Drug Name, Batch Number, Medicine Type, Manufacturer, Stock Quantity, Expiry Date, Price)
    Drug Name, Batch Number→Medicine Type, Manufacturer, Stock Quantity, Expiry Date, Price

    None of the above dependencies violate 3NF rules, so above relations are in 3NF.

PL SQL COMMANDS

Table Creation

SQL commands for creating the tables in our database:

```
CREATE TABLE Insurance ( insurance_id NUMBER(10) NOT NULL, company_name
CHAR(255) NOT NULL, start_date DATE NOT NULL, end_date DATE NOT NULL,
co_insurance NUMBER(4) NOT NULL, PRIMARY KEY (insurance_id) );
```

```
CREATE INDEX "Insurance_Company Name" ON Insurance (company_name);
```

```
CREATE TABLE CUSTOMER (
SSN NUMBER(10) NOT NULL, first_name CHAR(255) NOT NULL, last_name CHAR(255)
NOT NULL, phone NUMBER(10) NOT NULL UNIQUE, gender CHAR(1) NOT NULL, address
CHAR(1000) NOT NULL, date_of_birth DATE NOT NULL, insurance_id NUMBER(10) NOT
NULL UNIQUE, PRIMARY KEY (CID), CONSTRAINT insures FOREIGN KEY (insurance_id)
REFERENCES Insurance(insurance_id) ON DELETE SET NULL);
```

```
CREATE TABLE Prescription (
prescription_id NUMBER(10) NOT NULL, CID NUMBER(10) NOT NULL, doctor_id
NUMBER(10) NOT NULL, prescribed_date DATE NOT NULL, PRIMARY KEY (prescription_id),
CONSTRAINT holds FOREIGN KEY (SSN) REFERENCES Customer (SSN));
```

```
CREATE TABLE "PRESCRIBED_DRUGS" (
prescription_id NUMBER(10) NOT NULL, drug_name CHAR(255) NOT NULL,
prescribed_quantity NUMBER(10) NOT NULL, refill_limit NUMBER(10) NOT NULL, PRIMARY
KEY (prescription_id,drug_name), CONSTRAINT "consists of" FOREIGN KEY (prescription_id)
REFERENCES Prescription (prescription_id) ON DELETE CASCADE);
```

```
CREATE TABLE Employee (ID NUMBER(5) NOT NULL, CID NUMBER(10) NOT NULL
UNIQUE, License NUMBER(10) 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 NUMBER(4) NOT NULL, phone_number NUMBER(10) NOT NULL, date_of_birth
DATE NOT NULL, PRIMARY KEY (ID));
```

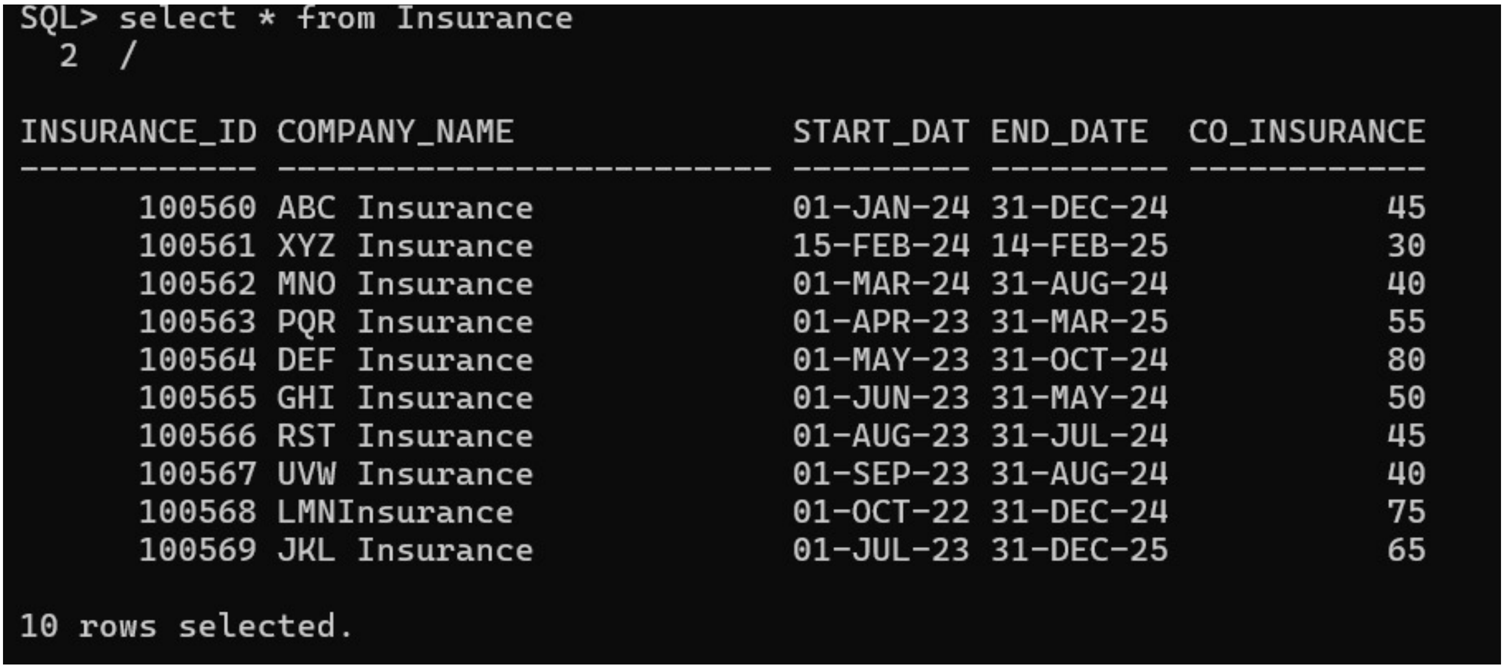
```
CREATE TABLE "Order" (
order_id NUMBER(10) NOT NULL, prescription_id NUMBER(10) NOT NULL, EmployeeID
NUMBER(5) NOT NULL, order_date DATE NOT NULL, PRIMARY KEY (order_id),
CONSTRAINT uses FOREIGN KEY (prescription_id) REFERENCES Prescription
(prescription_id), CONSTRAINT prepares FOREIGN KEY (EmployeeID) REFERENCES
Employee (ID));
```

```
CREATE TABLE Medicine (
drug_name CHAR(255) NOT NULL, batch_number NUMBER(10) NOT NULL, MedicineType
CHAR(255) NOT NULL, Manufacturer CHAR(255) NOT NULL, stock_quantity NUMBER(10)
NOT NULL, expiry_date DATE NOT NULL, Price NUMBER(4) NOT NULL, PRIMARY KEY
(drug_name,batch_number) );
```

```
CREATE TABLE "ORDERED_DRUGS" ( order_id NUMBER(10) NOT NULL, drug_name
CHAR(255) NOT NULL, batch_number NUMBER(10) NOT NULL, ordered_quantity
NUMBER(10) NOT NULL, Price NUMBER(2) NOT NULL, PRIMARY KEY (order_id,
drug_name,batch_number), CONSTRAINT "contains" FOREIGN KEY (order_id)
REFERENCES "Order" (order_id) ON DELETE CASCADE, CONSTRAINT "Fulfilled From"
FOREIGN KEY (drug_name, batch_number) REFERENCES Medicine(drug_name,
batch_number));

CREATE TABLE Notification (
ID NUMBER(10) NOT NULL, Message CHAR(255) NOT NULL, Type CHAR(255) NOT NULL,
PRIMARY KEY (ID));

CREATE TABLE Employee_Notification (
EmployeeID NUMBER(5) NOT NULL, NotificationID NUMBER(10) NOT NULL, PRIMARY KEY
(EmployeeID,NotificationID), CONSTRAINT FKEmployee_N849182 FOREIGN KEY
(EmployeeID) REFERENCES Employee (ID) ON DELETE CASCADE, CONSTRAINT
FKEmployee_N664471 FOREIGN KEY (NotificationID) REFERENCES Notification (ID) ON
DELETE CASCADE);

CREATE TABLE "Disposed Drugs" ( drug_name CHAR(255) NOT NULL, batch_number
NUMBER(10) NOT NULL, Quantity NUMBER(10) NOT NULL, Company CHAR(255) NOT
NULL, PRIMARY KEY (drug_name,batch_number), CONSTRAINT disposed FOREIGN KEY
(drug_name, batch_number) REFERENCES Medicine (drug_name,batch_number));

CREATE TABLE "EMPLOYEE_DISPOSED_DRUGS" ( EmployeeID NUMBER(5) NOT NULL,
drug_name CHAR(255) NOT NULL, batch_number NUMBER(10) NOT NULL, disposal_date
DATE NOT NULL, PRIMARY KEY (EmployeeID, drug_name, batch_number, disposal_date),
CONSTRAINT FKEmployee_D470142 FOREIGN KEY (EmployeeID) REFERENCES Employee
(ID), CONSTRAINT FKEmployee_D990025 FOREIGN KEY (drug_name, batch_number)
REFERENCES "Disposed Drugs" (drug_name, batch_number));

CREATE TABLE Bill (
order_id NUMBER(10) NOT NULL, Customer_ID NUMBER(10) NOT NULL, total_amount
NUMBER(4) NOT NULL, customer_payment NUMBER(4) NOT NULL, insurance_payment
NUMBER(4) NOT NULL, PRIMARY KEY (order_id, Customer_ID), CONSTRAINT makes
FOREIGN KEY (order_id) REFERENCES "Order" (order_id), CONSTRAINT pays FOREIGN
KEY (Customer_ID) REFERENCES Customer (CID) );
```

TABLES DATA

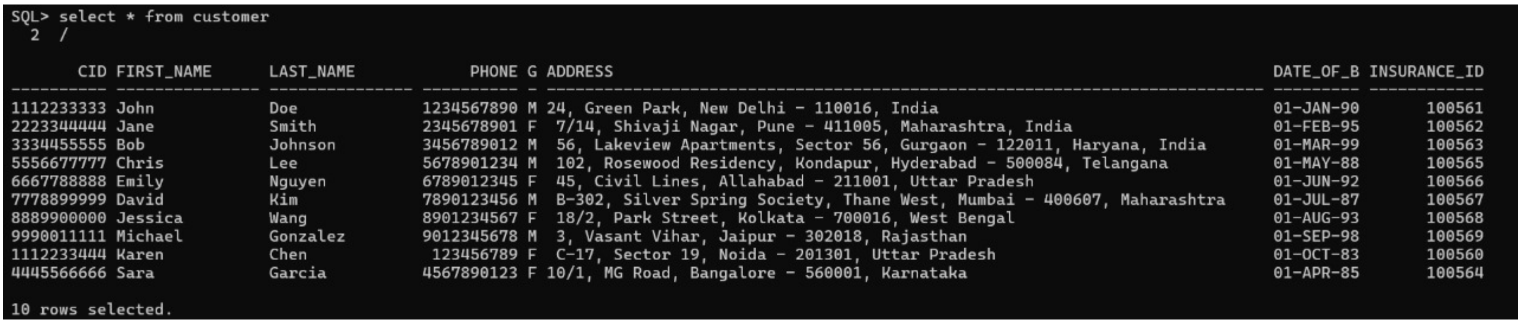
1. **Insurance Table Values:**



```
SQL> select * from Insurance
  2  /

INSURANCE_ID COMPANY_NAME              START_DAT END_DATE  CO_INSURANCE
------------ ------------------------- --------- --------- ------------
      100560 ABC Insurance             01-JAN-24 31-DEC-24           45
      100561 XYZ Insurance             15-FEB-24 14-FEB-25           30
      100562 MNO Insurance             01-MAR-24 31-AUG-24           40
      100563 PQR Insurance             01-APR-23 31-MAR-25           55
      100564 DEF Insurance             01-MAY-23 31-OCT-24           80
      100565 GHI Insurance             01-JUN-23 31-MAY-24           50
      100566 RST Insurance             01-AUG-23 31-JUL-24           45
      100567 UVW Insurance             01-SEP-23 31-AUG-24           40
      100568 LMNInsurance              01-OCT-22 31-DEC-24           75
      100569 JKL Insurance             01-JUL-23 31-DEC-25           65

10 rows selected.
```

Caption

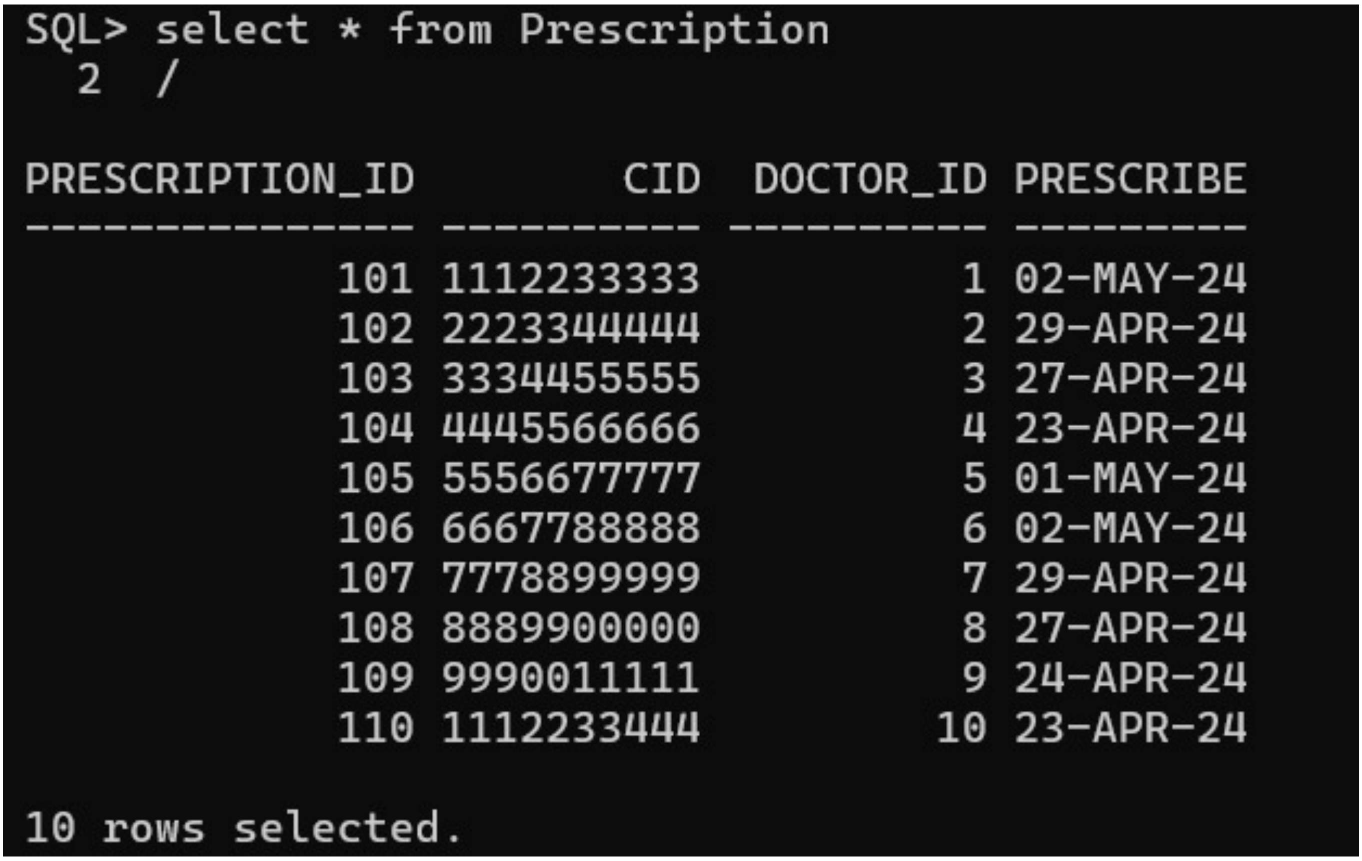
2. **Customer Table Values:**



```
SQL> select * from customer
  2  /

       CID FIRST_NAME      LAST_NAME            PHONE G ADDRESS                                                                     DATE_OF_B INSURANCE_ID
---------- --------------- --------------- ---------- - --------------------------------------------------------------------------- --------- ------------
1112233333 John            Doe             1234567890 M 24, Green Park, New Delhi - 110016, India                                   01-JAN-90       100561
2223344444 Jane            Smith           2345678901 F  7/14, Shivaji Nagar, Pune - 411005, Maharashtra, India                     01-FEB-95       100562
3334455555 Bob             Johnson         3456789012 M  56, Lakeview Apartments, Sector 56, Gurgaon - 122011, Haryana, India       01-MAR-99       100563
5556677777 Chris           Lee             5678901234 M  102, Rosewood Residency, Kondapur, Hyderabad - 500084, Telangana           01-MAY-88       100565
6667788888 Emily           Nguyen          6789012345 F  45, Civil Lines, Allahabad - 211001, Uttar Pradesh                         01-JUN-92       100566
7778899999 David           Kim             7890123456 M  B-302, Silver Spring Society, Thane West, Mumbai - 400607, Maharashtra     01-JUL-87       100567
8889900000 Jessica         Wang            8901234567 F  18/2, Park Street, Kolkata - 700016, West Bengal                           01-AUG-93       100568
9990011111 Michael         Gonzalez        9012345678 M  3, Vasant Vihar, Jaipur - 302018, Rajasthan                                01-SEP-98       100569
1112233444 Karen           Chen             123456789 F  C-17, Sector 19, Noida - 201301, Uttar Pradesh                             01-OCT-83       100560
4445566666 Sara            Garcia          4567890123 F 10/1, MG Road, Bangalore - 560001, Karnataka                                01-APR-85       100564

10 rows selected.
```

Caption

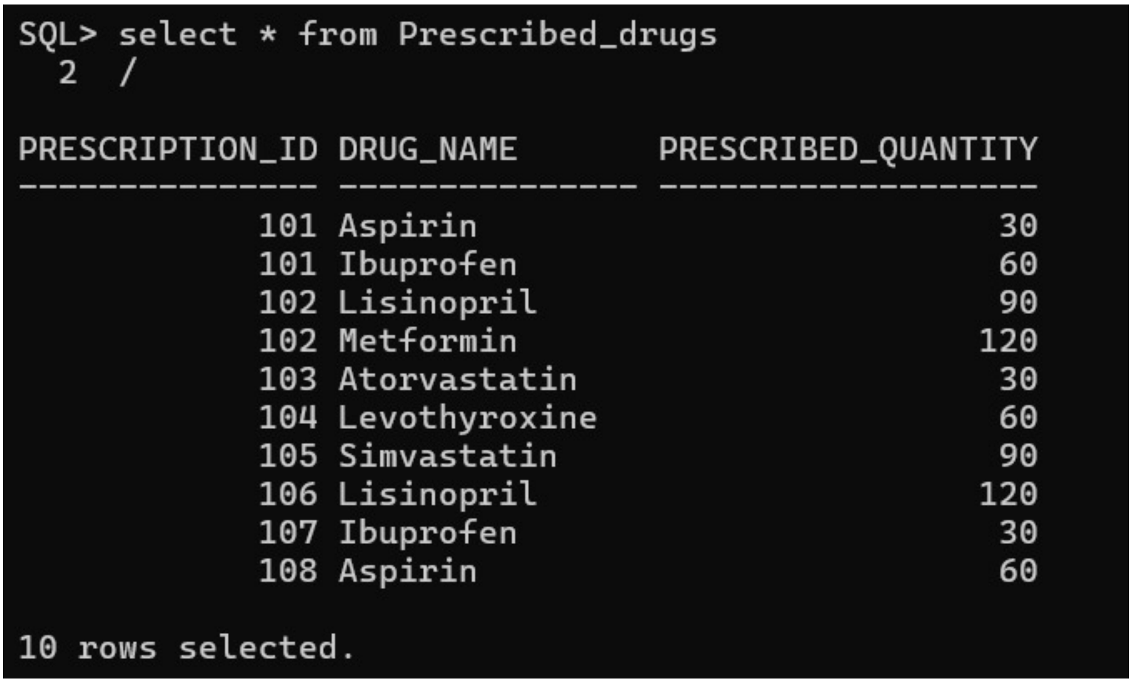
3. **Prescription Table Values:**



```
SQL> select * from Prescription
  2  /

PRESCRIPTION_ID        CID  DOCTOR_ID PRESCRIBE
--------------- ---------- ---------- ---------
            101 1112233333          1 02-MAY-24
            102 2223344444          2 29-APR-24
            103 3334455555          3 27-APR-24
            104 4445566666          4 23-APR-24
            105 5556677777          5 01-MAY-24
            106 6667788888          6 02-MAY-24
            107 7778899999          7 29-APR-24
            108 8889900000          8 27-APR-24
            109 9990011111          9 24-APR-24
            110 1112233444         10 23-APR-24

10 rows selected.
```

Caption

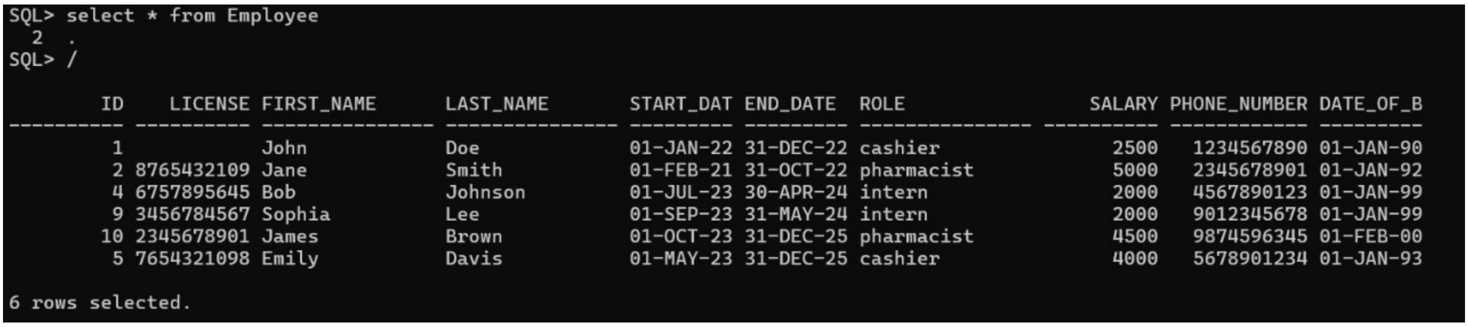
4. **Prescribed Drugs Table Values:**



```
SQL> select * from Prescribed_drugs
  2  /

PRESCRIPTION_ID DRUG_NAME       PRESCRIBED_QUANTITY
--------------- --------------- -------------------
            101 Aspirin                          30
            101 Ibuprofen                        60
            102 Lisinopril                       90
            102 Metformin                       120
            103 Atorvastatin                     30
            104 Levothyroxine                    60
            105 Simvastatin                      90
            106 Lisinopril                      120
            107 Ibuprofen                        30
            108 Aspirin                          60

10 rows selected.
```

Caption

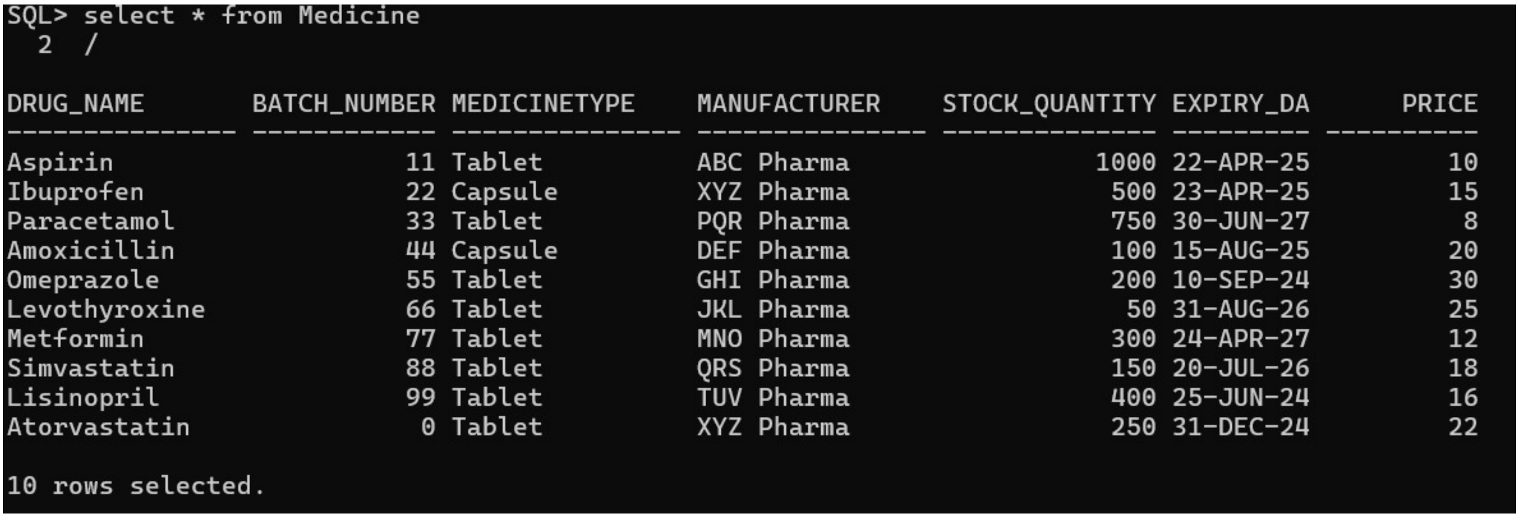
5. **Employee Table Values:**



```
SQL> select * from Employee
  2  .
SQL> /

        ID    LICENSE FIRST_NAME      LAST_NAME       START_DAT END_DATE  ROLE                SALARY PHONE_NUMBER DATE_OF_B
---------- ---------- --------------- --------------- --------- --------- --------------- ---------- ------------ ---------
         1            John            Doe             01-JAN-22 31-DEC-22 cashier               2500   1234567890 01-JAN-90
         2 8765432109 Jane            Smith           01-FEB-21 31-OCT-22 pharmacist            5000   2345678901 01-JAN-92
         4 6757895645 Bob             Johnson         01-JUL-23 30-APR-24 intern                2000   4567890123 01-JAN-99
         9 3456784567 Sophia          Lee             01-SEP-23 31-MAY-24 intern                2000   9012345678 01-JAN-99
        10 2345678901 James           Brown           01-OCT-23 31-DEC-25 pharmacist            4500   9874596345 01-FEB-00
         5 7654321098 Emily           Davis           01-MAY-23 31-DEC-25 cashier               4000   5678901234 01-JAN-93

6 rows selected.
```

Caption

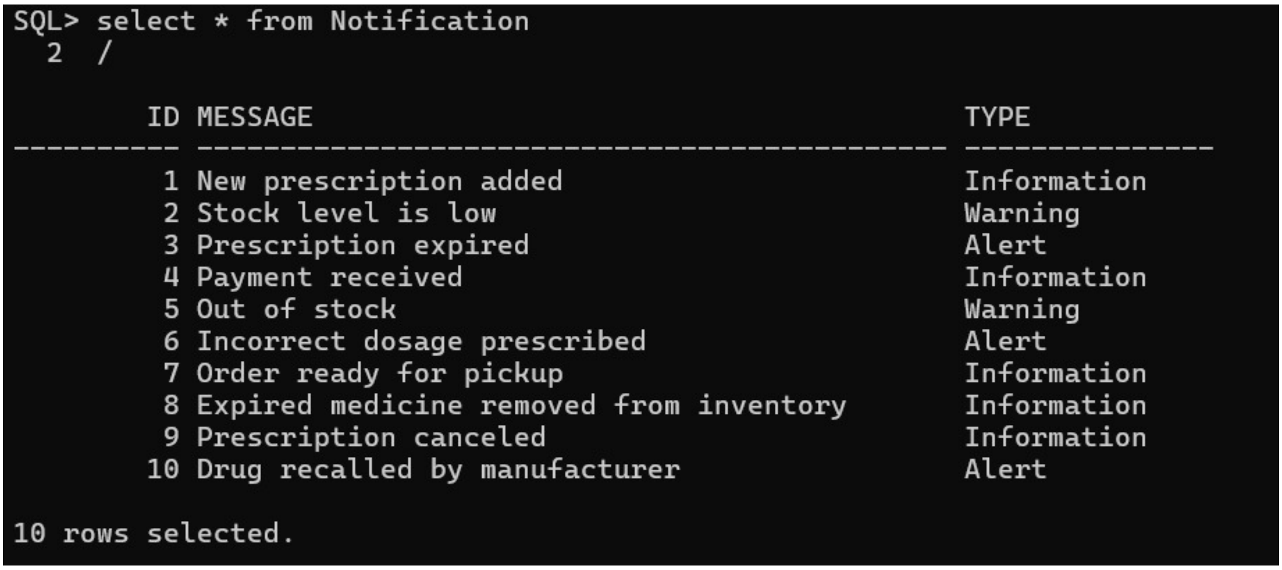
6. Medicine Table Values:



```
SQL> select * from Medicine
  2  /

DRUG_NAME       BATCH_NUMBER MEDICINETYPE    MANUFACTURER    STOCK_QUANTITY EXPIRY_DA      PRICE
--------------- ------------ --------------- --------------- -------------- --------- ----------
Aspirin                   11 Tablet          ABC Pharma                1000 22-APR-25         10
Ibuprofen                 22 Capsule         XYZ Pharma                 500 23-APR-25         15
Paracetamol               33 Tablet          PQR Pharma                 750 30-JUN-27          8
Amoxicillin               44 Capsule         DEF Pharma                 100 15-AUG-25         20
Omeprazole                55 Tablet          GHI Pharma                 200 10-SEP-24         30
Levothyroxine             66 Tablet          JKL Pharma                  50 31-AUG-26         25
Metformin                 77 Tablet          MNO Pharma                 300 24-APR-27         12
Simvastatin               88 Tablet          QRS Pharma                 150 20-JUL-26         18
Lisinopril                99 Tablet          TUV Pharma                 400 25-JUN-24         16
Atorvastatin               0 Tablet          XYZ Pharma                 250 31-DEC-24         22

10 rows selected.
```

Caption

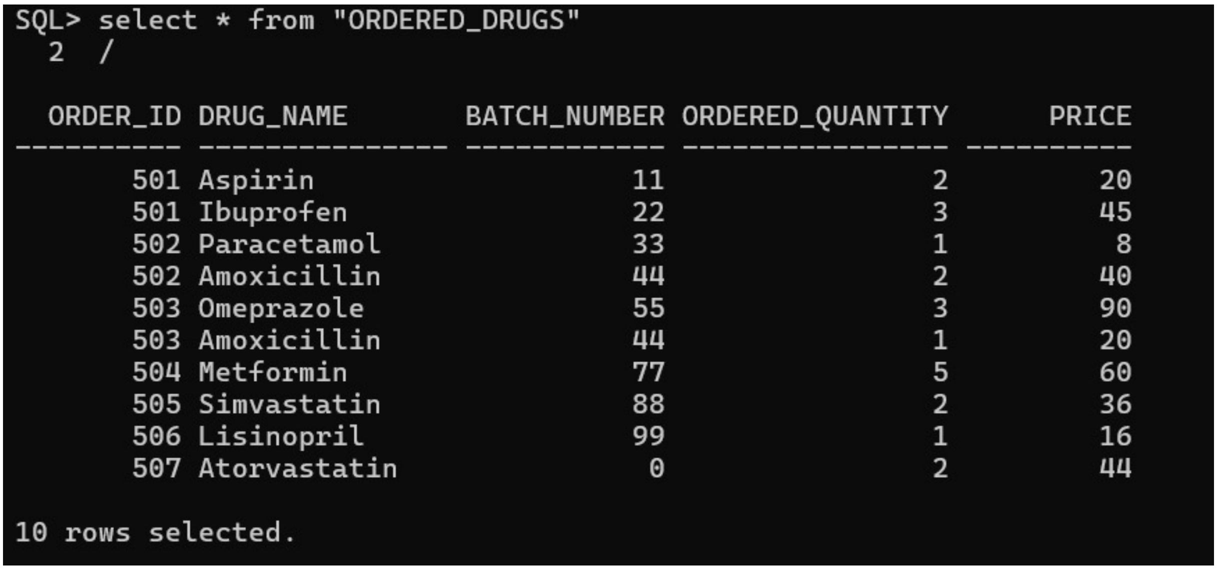
7. Notification Table Values:



```
SQL> select * from Notification
  2  /

        ID MESSAGE                                            TYPE
---------- -------------------------------------------------- ---------------
         1 New prescription added                             Information
         2 Stock level is low                                 Warning
         3 Prescription expired                               Alert
         4 Payment received                                   Information
         5 Out of stock                                       Warning
         6 Incorrect dosage prescribed                        Alert
         7 Order ready for pickup                             Information
         8 Expired medicine removed from inventory            Information
         9 Prescription canceled                              Information
        10 Drug recalled by manufacturer                      Alert

10 rows selected.
```

Caption

8. Ordered drugs:

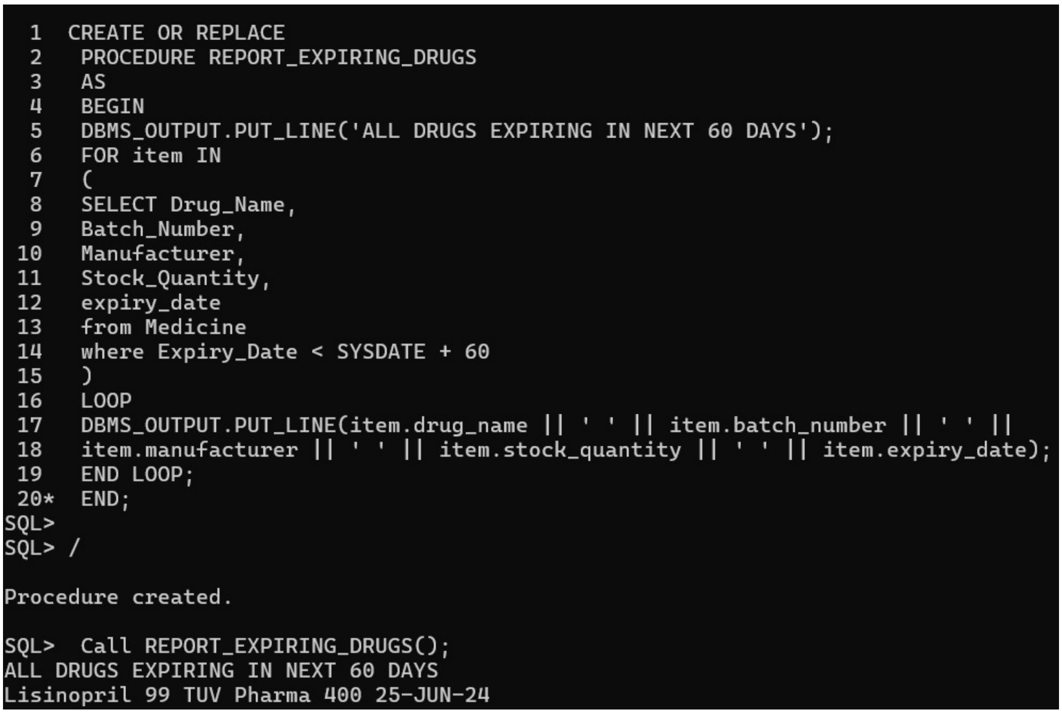


```
SQL> select * from "ORDERED_DRUGS"
  2  /

  ORDER_ID DRUG_NAME       BATCH_NUMBER ORDERED_QUANTITY      PRICE
---------- --------------- ------------ ---------------- ----------
       501 Aspirin                   11                2         20
       501 Ibuprofen                 22                3         45
       502 Paracetamol               33                1          8
       502 Amoxicillin               44                2         40
       503 Omeprazole                55                3         90
       503 Amoxicillin               44                1         20
       504 Metformin                 77                5         60
       505 Simvastatin               88                2         36
       506 Lisinopril                99                1         16
       507 Atorvastatin               0                2         44

10 rows selected.
```

Caption

PROCEDURES

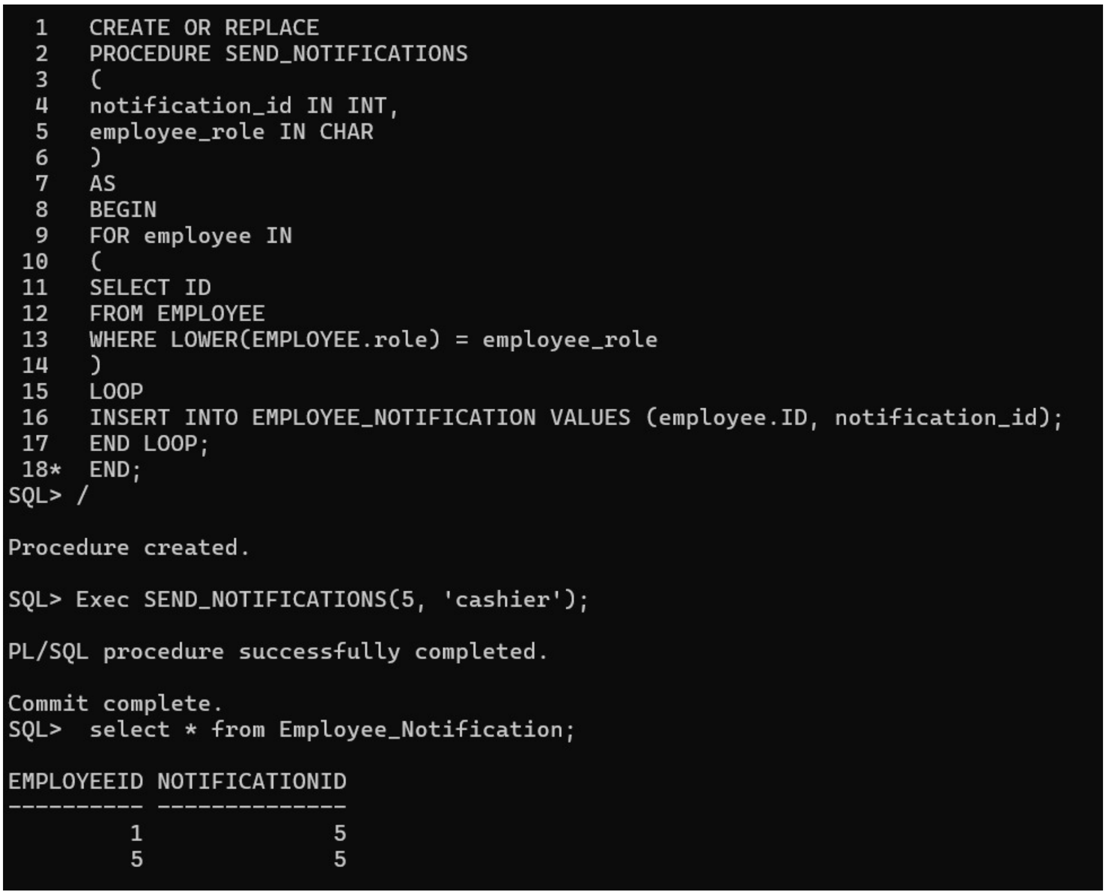
1. Report Expiring Drug



```
CREATE OR REPLACE
PROCEDURE REPORT_EXPIRING_DRUGS
AS
BEGIN
DBMS_OUTPUT.PUT_LINE('ALL DRUGS EXPIRING IN NEXT 60 DAYS');
FOR item IN
(
SELECT Drug_Name,
Batch_Number,
Manufacturer,
Stock_Quantity,
expiry_date
from Medicine
where Expiry_Date < SYSDATE + 60
)
LOOP
DBMS_OUTPUT.PUT_LINE(item.drug_name || ' ' || item.batch_number || ' ' ||
item.manufacturer || ' ' || item.stock_quantity || ' ' || item.expiry_date);
END LOOP;
END;
SQL>
SQL> /

Procedure created.

SQL>  Call REPORT_EXPIRING_DRUGS();
ALL DRUGS EXPIRING IN NEXT 60 DAYS
Lisinopril 99 TUV Pharma 400 25-JUN-24
```

Caption

2. Send Notifications

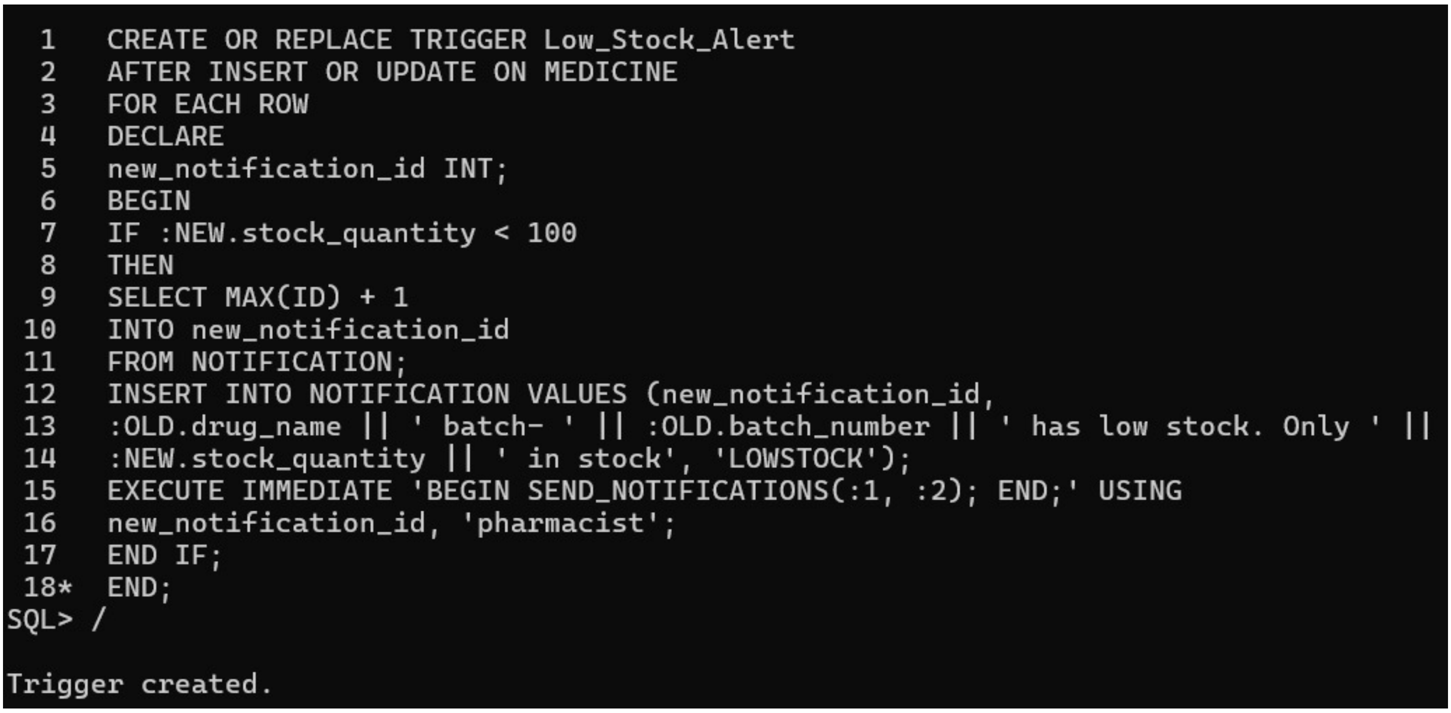


```
CREATE OR REPLACE
PROCEDURE SEND_NOTIFICATIONS
(
notification_id IN INT,
employee_role IN CHAR
)
AS
BEGIN
FOR employee IN
(
SELECT ID
FROM EMPLOYEE
WHERE LOWER(EMPLOYEE.role) = employee_role
)
LOOP
INSERT INTO EMPLOYEE_NOTIFICATION VALUES (employee.ID, notification_id);
END LOOP;
END;
SQL> /

Procedure created.

SQL> Exec SEND_NOTIFICATIONS(5, 'cashier');

PL/SQL procedure successfully completed.

Commit complete.
SQL>  select * from Employee_Notification;

EMPLOYEEID NOTIFICATIONID
---------- --------------
         1              5
         5              5
```

Caption

TRIGGERS

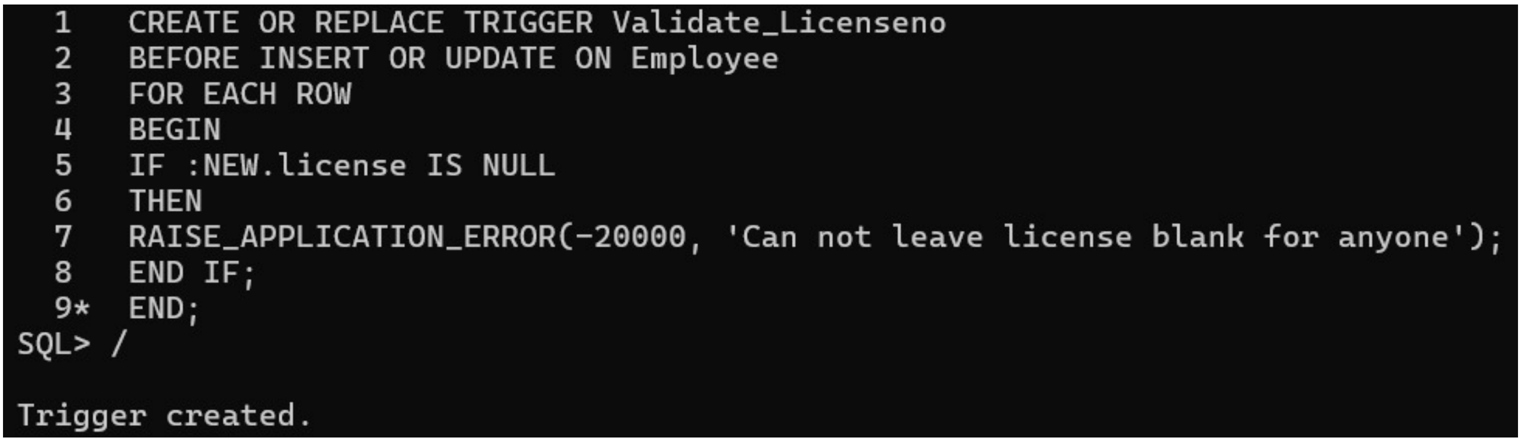
1. Low Stock Alert



```
CREATE OR REPLACE TRIGGER Low_Stock_Alert
AFTER INSERT OR UPDATE ON MEDICINE
FOR EACH ROW
DECLARE
new_notification_id INT;
BEGIN
IF :NEW.stock_quantity < 100
THEN
SELECT MAX(ID) + 1
INTO new_notification_id
FROM NOTIFICATION;
INSERT INTO NOTIFICATION VALUES (new_notification_id,
:OLD.drug_name || ' batch- ' || :OLD.batch_number || ' has low stock. Only ' ||
:NEW.stock_quantity || ' in stock', 'LOWSTOCK');
EXECUTE IMMEDIATE 'BEGIN SEND_NOTIFICATIONS(:1, :2); END;' USING
new_notification_id, 'pharmacist';
END IF;
END;
SQL> /

Trigger created.
```

Caption

2. Validate license

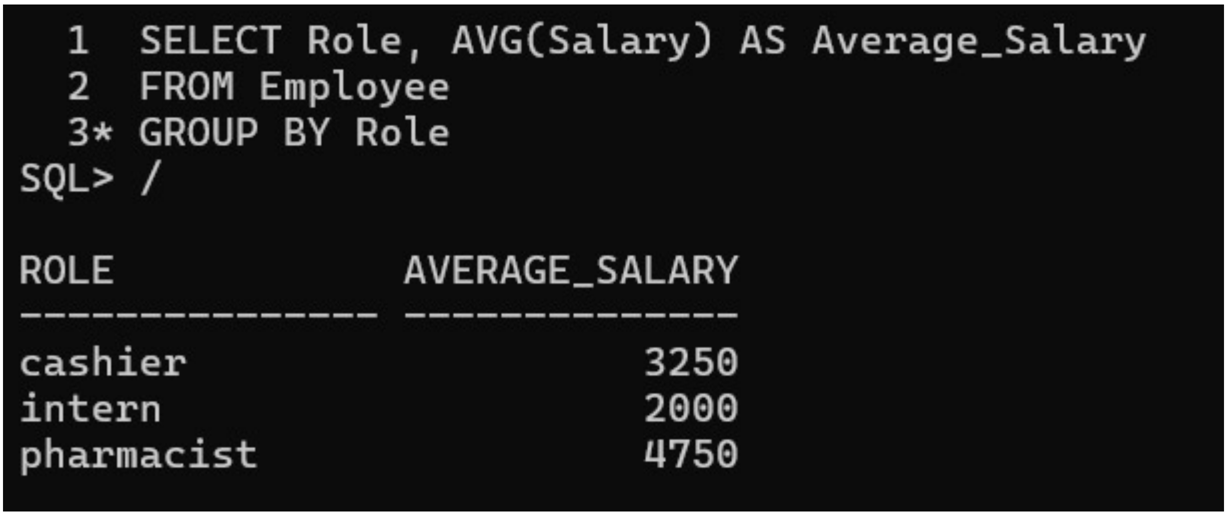


```
CREATE OR REPLACE TRIGGER Validate_Licenseno
BEFORE INSERT OR UPDATE ON Employee
FOR EACH ROW
BEGIN
IF :NEW.license IS NULL
THEN
RAISE_APPLICATION_ERROR(-20000, 'Can not leave license blank for anyone');
END IF;
END;
SQL> /

Trigger created.
```

Caption

QUERIES

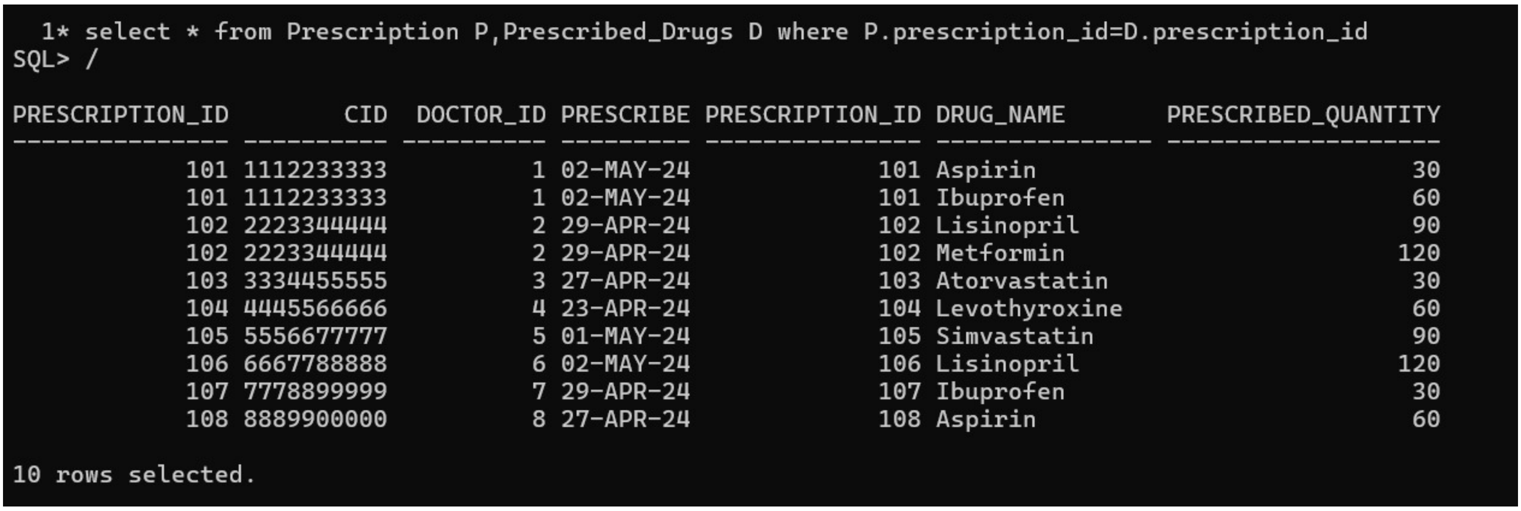
1. Find the average salary of employees grouped by their roles:



```
  1  SELECT Role, AVG(Salary) AS Average_Salary
  2  FROM Employee
  3* GROUP BY Role
SQL> /

ROLE                AVERAGE_SALARY
------------------- --------------
cashier                       3250
intern                        2000
pharmacist                    4750
```

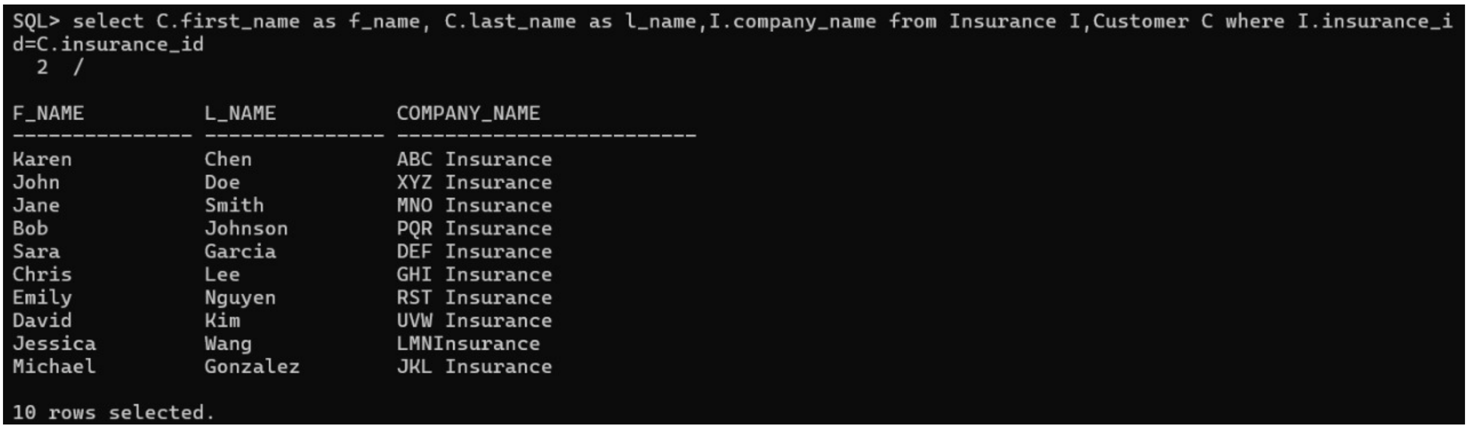
Caption

2. Find the total number of prescriptions and the total quantity of drugs prescribed for each customer:



```
  1* select * from Prescription P,Prescribed_Drugs D where P.prescription_id=D.prescription_id
SQL> /

PRESCRIPTION_ID        CID  DOCTOR_ID PRESCRIBE PRESCRIPTION_ID DRUG_NAME       PRESCRIBED_QUANTITY
--------------- ---------- ---------- --------- --------------- --------------- -------------------
            101 1112233333          1 02-MAY-24             101 Aspirin                          30
            101 1112233333          1 02-MAY-24             101 Ibuprofen                        60
            102 2223344444          2 29-APR-24             102 Lisinopril                       90
            102 2223344444          2 29-APR-24             102 Metformin                       120
            103 3334455555          3 27-APR-24             103 Atorvastatin                     30
            104 4445566666          4 23-APR-24             104 Levothyroxine                    60
            105 5556677777          5 01-MAY-24             105 Simvastatin                      90
            106 6667788888          6 02-MAY-24             106 Lisinopril                      120
            107 7778899999          7 29-APR-24             107 Ibuprofen                        30
            108 8889900000          8 27-APR-24             108 Aspirin                          60

10 rows selected.
```

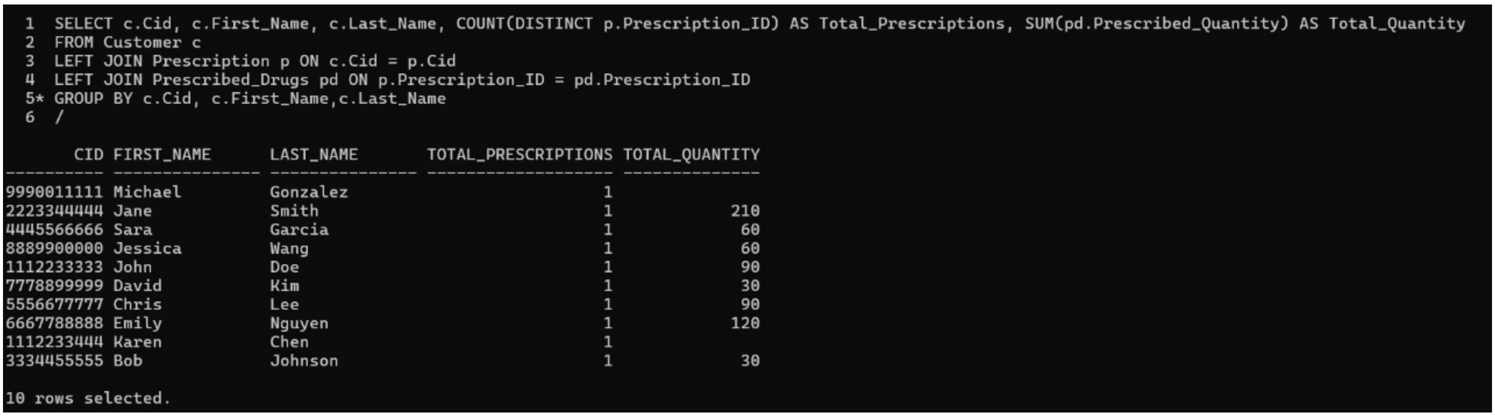
3. To display name and company name of the customers:



```
SQL> select C.first_name as f_name, C.last_name as l_name,I.company_name from Insurance I,Customer C where I.insurance_i
d=C.insurance_id
  /

F_NAME          L_NAME          COMPANY_NAME
--------------- --------------- -------------------------
Karen           Chen            ABC Insurance
John            Doe             XYZ Insurance
Jane            Smith           MNO Insurance
Bob             Johnson         PQR Insurance
Sara            Garcia          DEF Insurance
Chris           Lee             GHI Insurance
Emily           Nguyen          RST Insurance
David           Kim             UVW Insurance
Jessica         Wang            LMNInsurance
Michael         Gonzalez        JKL Insurance

10 rows selected.
```

Caption

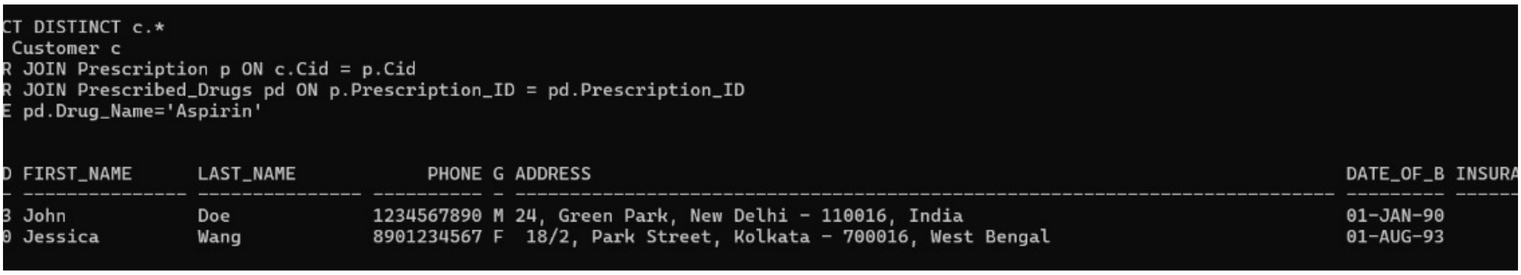
4. To display the prescription details of customers:



```
SELECT c.Cid, c.First_Name, c.Last_Name, COUNT(DISTINCT p.Prescription_ID) AS Total_Prescriptions, SUM(pd.Prescribed_Quantity) AS Total_Quantity
FROM Customer c
LEFT JOIN Prescription p ON c.Cid = p.Cid
LEFT JOIN Prescribed_Drugs pd ON p.Prescription_ID = pd.Prescription_ID
GROUP BY c.Cid, c.First_Name,c.Last_Name
/

       CID FIRST_NAME      LAST_NAME       TOTAL_PRESCRIPTIONS TOTAL_QUANTITY
---------- --------------- --------------- ------------------- --------------
9990011111 Michael         Gonzalez                          1
2223344444 Jane            Smith                             1            210
4445566666 Sara            Garcia                            1             60
8889900000 Jessica         Wang                              1             60
1112233333 John            Doe                               1             90
7778899999 David           Kim                               1             30
5556677777 Chris           Lee                               1             90
6667788888 Emily           Nguyen                            1            120
1112233444 Karen           Chen                              1
3334455555 Bob             Johnson                           1             30

10 rows selected.
```

Caption

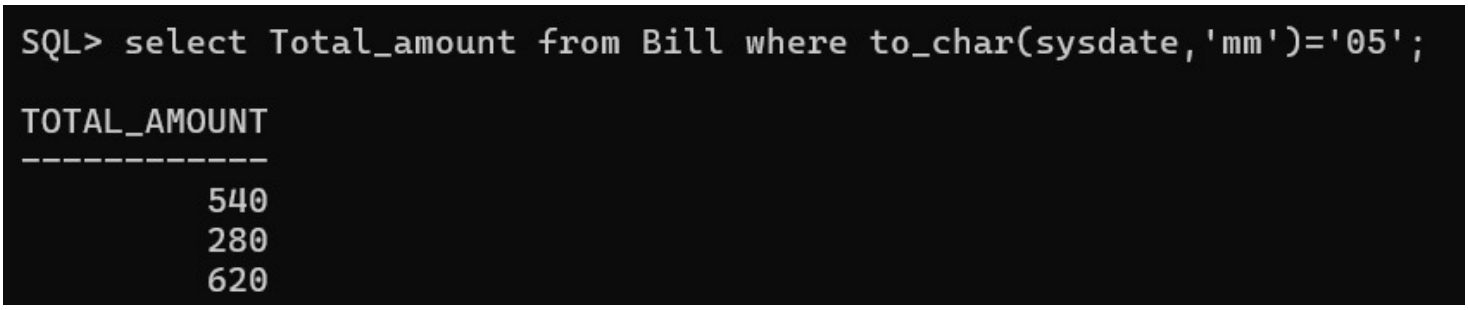
5. Find customers who have prescriptions for a specific drug (e.g., Drug Name = ‘Aspirin'):



```
CT DISTINCT c.*
 Customer c
R JOIN Prescription p ON c.Cid = p.Cid
R JOIN Prescribed_Drugs pd ON p.Prescription_ID = pd.Prescription_ID
E pd.Drug_Name='Aspirin'

D FIRST_NAME      LAST_NAME            PHONE G ADDRESS                                                                          DATE_OF_B INSURA
- --------------- --------------- ---------- - -------------------------------------------------------------------------------- --------- ------
3 John            Doe             1234567890 M 24, Green Park, New Delhi - 110016, India                                        01-JAN-90
0 Jessica         Wang            8901234567 F  18/2, Park Street, Kolkata - 700016, West Bengal                                01-AUG-93
```

Caption

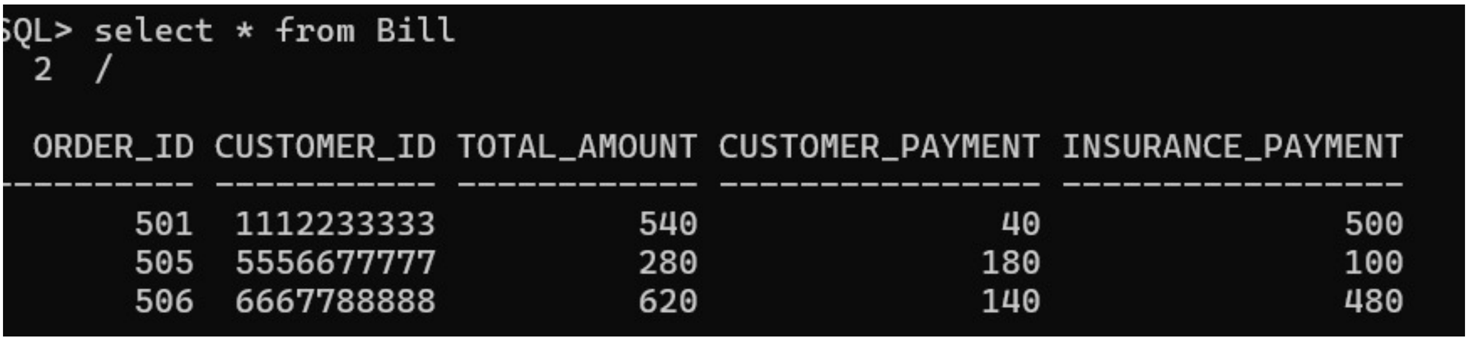
6. To find the total sales in the month of May:



```
SQL> select Total_amount from Bill where to_char(sysdate,'mm')='05';

TOTAL_AMOUNT
------------
         540
         280
         620
```

Caption

7. To show all the records of bill table:



```
SQL> select * from Bill
  2  /

  ORDER_ID CUSTOMER_ID TOTAL_AMOUNT CUSTOMER_PAYMENT INSURANCE_PAYMENT
---------- ----------- ------------ ---------------- -----------------
       501  1112233333          540               40               500
       505  5556677777          280              180               100
       506  6667788888          620              140               480
```

Caption

CONCLUSION

In conclusion, the development of a pharmacy management system using a database management system has proven to be an effective solution for improving the efficiency and accuracy of pharmacy operations. The system provides a user-friendly interface for managing patient information, prescription orders, inventory, and sales records.

The database management system used in this project allows for seamless data integration and real-time updates, ensuring that the pharmacy has access to up-to-date information. The system also provides various reports, such as inventory levels, sales reports, and prescription history, which aid in decision-making and inventory management.

Overall, the pharmacy management system developed in this project has the potential to revolutionize the way pharmacies operate, streamlining processes and improving the quality of care for patients. Further enhancements and optimizations could be made to the system in the future, but the current version serves as an effective tool for managing pharmacy operations.