

MEDICAL SUPPLY MANAGEMENT SYSTEM

A PROJECT REPORT

Submitted by

PRAGADEESWARAN K [Reg No: RA2211047010135]

PRASHETH P [Reg No: RA2211047010138]

LOGAPRIYAN R [Reg No: RA2211047010144]

Under the Guidance of

BEAULAH JEYAVATHANA

Assistant Professor, Department of Computational Intelligence

in partial fulfillment of the requirements for the degree of

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE



DEPARTMENT OF COMPUTATIONAL INTELLIGENCE

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR– 603 203

NOV 2024



SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR-603 203

BONAFIDE CERTIFICATE

Register no RA2211047010135, RA2211047010138, RA2211047010144 Certified to be
the bonafide work done **PRAGADEESWARAN K, PRASHETH P, LOGAPRIYAN R**
of III year/V sem B.Tech Degree Course in the Project Course – **21CSC205P Database
Management Systems** in **SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**,
Kattankulathur for the academic year 2024-2025.


Faculty in Charge

Dr. BEAULAH JEYAVATHANA. R
Assistant Professor
Department of Computational Intelligence
SRMIST -KTR


HEAD OF THE DEPARTMENT

Dr. R. Annie Uthra
Head of Department
Department of Computational Intelligence
SRMIST - KTR

ABSTRACT

Efficient medical supply management is crucial for ensuring the availability of essential medical resources, thereby supporting quality healthcare services. This paper introduces a Medical Supply Management (MSM) system designed to streamline inventory management, order processing, and user-specific functionalities.

The MSM system features separate login portals for suppliers and customers, providing secure, role-specific access. Suppliers can manage inventory, update product details, and process orders, while customers can place orders, track order status, and receive notifications on product availability. This dual-access setup enhances transparency and communication.

Key features include real-time inventory tracking, automated stock replenishment alerts, and detailed reporting capabilities. These functions help maintain optimal stock levels, minimize shortages and overstock situations, and enhance overall operational efficiency. The system also supports product categorization, batch tracking, and expiry date monitoring to ensure the availability of safe medical supplies.

By leveraging advanced database management and robust security protocols, the MSM system ensures data integrity, confidentiality, and ease of use. Streamlining supply chain processes and improving inventory control, the MSM system aims to enhance the efficiency of medical supplies management, reduce administrative overheads, and contribute to improved healthcare delivery.

Problem Statement

The healthcare industry faces significant challenges in managing medical supplies efficiently. Inadequate inventory management, delayed order processing, and poor communication between suppliers and customers often result in stock shortages, overstock situations, and compromised patient care. Traditional methods of managing medical supplies are often labour-intensive, error-prone, and lack real-time tracking capabilities, leading to inefficiencies and increased operational costs.

The absence of a streamlined system to manage the procurement, inventory, and distribution of medical supplies exacerbates these issues. Suppliers struggle to maintain accurate inventory records and timely order fulfilment, while customers face difficulties in tracking orders and ensuring the availability of essential medical supplies. These challenges not only impact the operational efficiency of medical supply chains but also hinder the ability of healthcare providers to deliver timely and effective patient care.

Address these issues, there is a need for a comprehensive Medical Supply Management (MSM) system that integrates advanced inventory management, real-time tracking, and automated order processing. Such a system should provide secure, role-specific access for suppliers and customers, enhance communication and transparency, and ensure the consistent availability of medical supplies.

TABLE OF CONTENTS

Chapter No	Chapter Name	Page No
1.	Problem understanding, Identification of Entity and Relationships, Construction of DB using ER Model for the project	6-12
2.	Design of Relational Schemas, Creation of Database Tables for the project.	13-22
3.	Complex queries based on the concepts of constraints, sets, joins, views, Triggers and Cursors.	23-42
4.	Analyzing the pitfalls, identifying the dependencies, and applying normalizations	43-51
5.	Implementation of concurrency control and recovery mechanisms	52-55
6.	Code for the project	56-89
7.	Result and Discussion (Screen shots of the implementation with front end.	90-98
8.	Attach the Real Time project certificate / Online course certificate	99-100

1. Problem understanding, Identification of Entity and Relationships, Construction of DB using ER Model for the project

Problem Understanding:

The healthcare sector faces critical challenges in managing medical supplies, primarily due to ineffective and outdated inventory management practices. The core issues include:

1. Inefficient Inventory Management:

- **Stock Shortages and Overstocking:** The absence of precise inventory tracking leads to frequent stock-outs, risking patient care, or overstocking, which ties up resources and increases holding costs.
- **Manual Processes:** Traditional manual methods are labor-intensive and prone to human errors, causing inaccuracies in inventory records and delaying critical decisions.

2. Delayed Order Processing:

- **Time-Consuming Ordering:** The reliance on manual or semi-automated order processing slows down the procurement and restocking process, often resulting in delayed deliveries and disruptions in the supply chain.
- **Supplier-Customer Communication Gaps:** Poor communication between suppliers and customers contributes to delays in order approvals, fulfillment, and response times, which can lead to further inefficiencies.

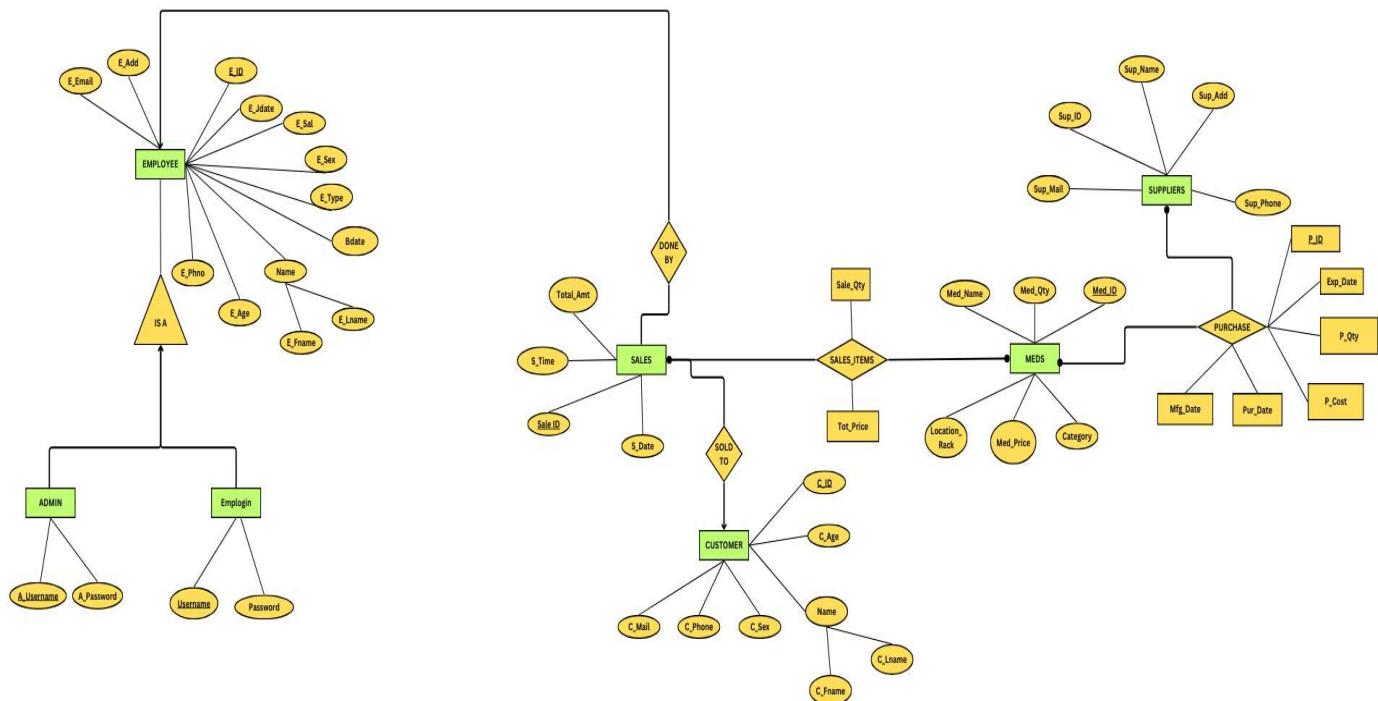
3. Lack of Real-Time Tracking and Transparency:

- **Limited Visibility:** Current systems often lack real-time tracking, making it difficult for stakeholders to monitor inventory status and movement accurately.
- **Transparency Issues:** Without real-time data, suppliers struggle to update inventory records dynamically, while customers lack visibility into the status of their orders, impacting planning and availability.

4. High Operational Costs:

- **Resource-Intensive Management:** Manual processes and insufficient technology integration lead to high operational costs due to inefficiencies, inaccuracies, and additional labor requirements.

ER DIAGRAM FOR MEDICAL SUPPLY MANAGEMENT SYSTEM



ER EXPLANATION

I. EMPLOYEE ENTITY:

Description: Manages employee records within the medical supply management system, managing personal information, employment details, and authentication for various roles.

Functionality:

- Stores detailed employee information such as name, address, phone number, age, sex, job type, salary, and employment dates.
- Differentiates roles within the employee category, specifically Admin and Pharmacist, each with unique login credentials.

Functional Requirements:

- **Employee Registration:**
 - Collect and store employee details such as name, address, phone number, age, sex, job type, salary, and employment dates.
- **Role-specific Login:**
 - Allow Admin and Pharmacist to log in using their respective usernames and passwords.
- **Profile Management:**
 - Enable employees to update their personal and professional details.
- **Role Assignment:**
 - Differentiate access and functionality based on whether the employee is an Admin or a Pharmacist.

Non-Functional Requirements:

- **Security:**
 - Encrypt employee passwords and personal information to ensure data security.
- **Usability:**
 - Provide an intuitive interface for employee registration and profile management.
- **Performance:**
 - Manage concurrent access by multiple employees without performance degradation.
- **Scalability:**
 - Support the addition of new employee roles and an increasing number of employees records.

II. SUPPLIERS ENTITY:

Description: Manages supplier information within the medical supply management system, ensuring accurate supplier details and efficient supply chain management.

Functionality:

- Stores supplier details such as name, address, phone number, email, and unique supplier ID.
- Facilitates the management of supplier information to streamline the purchasing process.

Functional Requirements:

- **Supplier Registration:**
 - Collect and store supplier details including name, address, phone number, and email.
- **Supplier Management:**
 - Allow the updating of supplier information and maintaining an accurate database.
- **Purchase Tracking:**
 - Record purchases made from suppliers to maintain a history of transactions.

Non-Functional Requirements:

- **Security:**
 - Protect supplier data with appropriate encryption techniques.
- **Usability:**
 - Design a user-friendly interface for managing supplier information.
- **Performance:**
 - Ensure quick retrieval and updating of supplier details.
- **Scalability:**
 - Accommodate a growing number of suppliers without affecting system performance.

III. MEDS ENTITY:

Description: Manages medication records within the medical supply management system, handling details of each medicine such as name, quantity, price, category, and storage location.

Functionality:

- Stores detailed information on each medication including name, quantity, price, category, and storage location.
- Tracks inventory levels and provides alerts for stock replenishment.

Functional Requirements:

- **Medication Registration:**
 - Collect and store medication details including name, quantity, price, category, and storage location.
- **Inventory Management:**
 - Track and update inventory levels, providing alerts when stock is low.
- **Price Management:**
 - Allow updating of medication prices, as necessary.

Non-Functional Requirements:

- **Usability:**
 - Ensure an intuitive interface for managing medication information.
- **Performance:**
 - Manage high volumes of inventory updates efficiently.
- **Reliability:**
 - Maintain accurate records to ensure inventory integrity.
- **Scalability:**
 - Support an expanding inventory as the range of medications grows.

IV. SALES ENTITY:

Description: Manages sales transactions within the medical supply management system, recording details of each sale and linking it to the relevant customer and employee.

Functionality:

- Records details of sales transactions including sale ID, date, time, total amount, and the employee who managed the sale.
- Tracks the connection between sales transactions and customer purchases.

Functional Requirements:

- **Sales Recording:**
 - Record details of each sales transaction, including date, time, and total amount.
- **Employee Tracking:**
 - Link sales transactions to the employee who processed the sale.
- **Customer Linkage:**
 - Associate sales transactions with customer purchases.

Non-Functional Requirements:

- **Security:**
 - Ensure the confidentiality of sales data.
- **Performance:**
 - Optimize sales transaction processing to manage high transaction volumes efficiently.
- **Reliability:**
 - Maintain accurate sales records for audit and analysis purposes.
- **Scalability:**
 - Support a growing number of sales transactions as business volume increases.

V. CUSTOMER ENTITY:

Description: Manages customer information within the medical supply management system, handling details such as name, age, gender, contact information, and purchase history.

Functionality:

- Stores customer details including name, age, gender, contact information, and unique customer ID.
- Facilitates management of customer profiles and tracking of purchase history.

Functional Requirements:

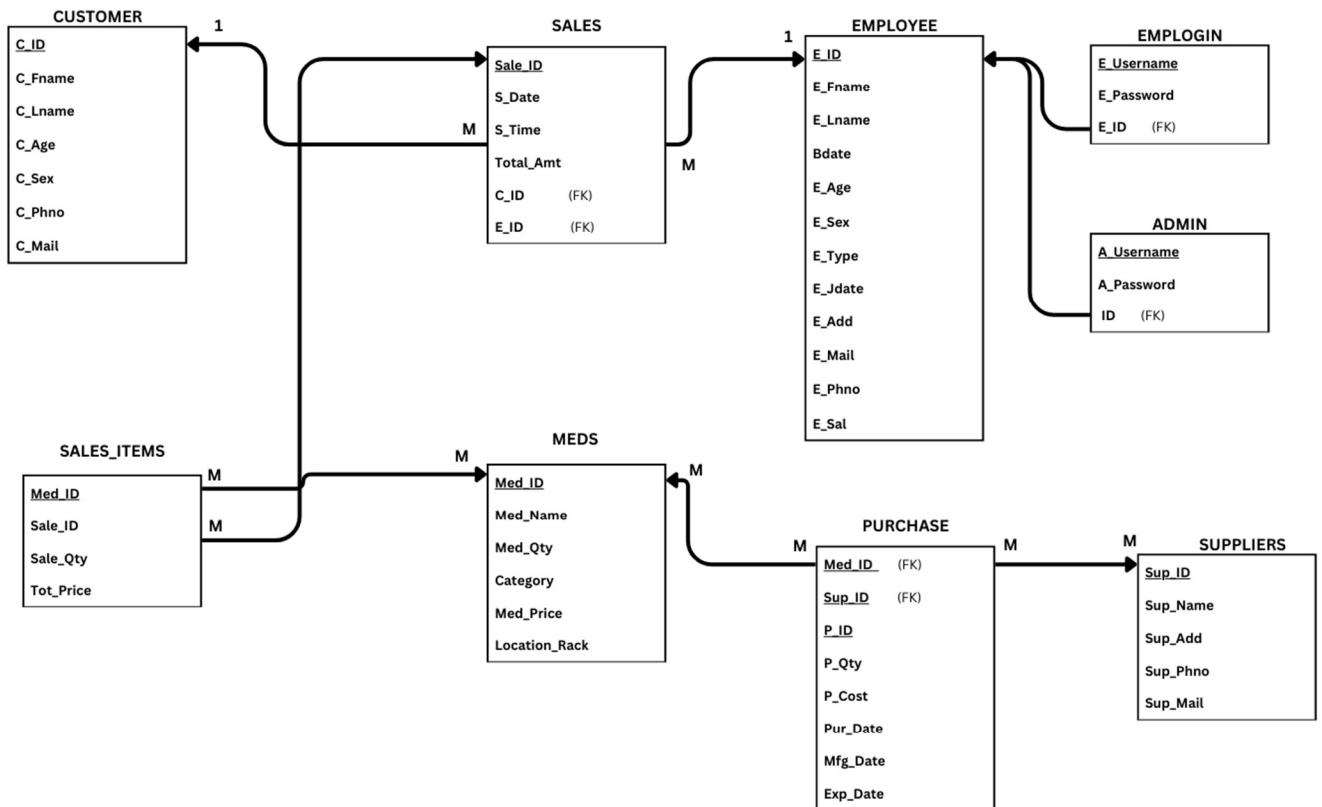
- **Customer Registration:**
 - Collect and store customer details including name, age, gender, and contact information.
- **Profile Management:**
 - Allow customers to update their personal information.
- **Purchase Tracking:**
 - Record and track customer purchase history.

Non-Functional Requirements:

- **Security:**
 - Protect customer data with encryption and secure access controls.
- **Usability:**
 - Design an easy-to-navigate interface for managing customer information.
- **Performance:**
 - Ensure fast and efficient access to customer profiles and purchase history.
- **Scalability:**
 - Accommodate an increasing number of customer records and transactions.

2. Design of Relational Schemas, Creation of Database Tables for the project.

SCHEMA DIAGRAM FOR MEDICAL SUPPLY MANAGEMENT SYSTEM:



SCHEMA TABLE REPRESENTATION:

ADMIN TABLE

ID	A_USERNAME	A_PASSWORD

ADMIN (ID, A_USERNAME, A_PASSWORD)

CUSTOMER TABLE

C_ID	C_FNAME	C_LNAME	C_AGE	C_SEX	C_PHNO	C_MAIL

CUSTOMER (C_ID, C_FNAME, C_LNAME, C_AGE, C_SEX, C_PHNO, C_MAIL)

EMPLOGIN TABLE

E_ID	E_USERNAME	E_PASSWORD

EMPMLOGIN (E_ID, E_USERNAME, E_PASS)

MEDS TABLE

MED_ID	MED_NAME	MED_QTY	CATEGORY	PRICE	LOCATION_RACK

MEDS (MED_ID, MED_NAME, MED_QTY, CATEGORY, MED_PRICE, LOCATION_RACK)

PURCHASE TABLE

P_ID	SUP_ID	MED_ID	P_QTY	P_COST	PUR_DATE	MFG_DATE	EXP_DATE

PURCHASE (P_ID, SUP_ID, MED_ID, P_QTY, P_COST, PUR_DATE, MFG_DATE, EXP_DATE)

SALES TABLE

<u>SALE_ID</u>	<u>C_ID</u>	<u>S_DATE</u>	<u>S_TIME</u>	<u>TOTAL_AMT</u>	<u>E_ID</u>

SALES (SALE_ID, C_ID, S_DATE, S_TIME, TOTAL_AMT, E_ID)

SALES_ITEM TABLE

<u>SALE_ID</u>	<u>MED_ID</u>	<u>SALE_QTY</u>	<u>TOT_PRICE</u>

SALES_ITEMS (SALE_ID, MED_ID, SALE_QTY, TOT_PRICE)

SUPPLIERS TABLE

<u>SUP_ID</u>	<u>SUP_NAME</u>	<u>SUP_ADD</u>	<u>SUP_PHNO</u>	<u>SUP_MAIL</u>

SUPPLIERS (SUP_ID, SUP_NAME, SUP_ADD, SUP_PHNO, SUP_MAIL)

EMPLOYEE TABLE

<u>E_ID</u>	<u>E_FNAME</u>	<u>E_LNAME</u>	<u>BDATE</u>	<u>E_AGE</u>	<u>E_SEX</u>	<u>E_TYPE</u>	<u>E_JDATE</u>	<u>E_SAL</u>	<u>E_PHNO</u>	<u>E_MAIL</u>	<u>E_ADD</u>

EMPLOYEE (E_ID, E_FNAME, E_LNAME, BDATE, E_AGE, E_SEX, E_TYPE, E_JDATE, E_SALE, E_PHNO, E_MAIL, E_ADD)

DATABASE CREATION

- **Create Database Statement:**

```
CREATE DATABASE medical;
USE medical;
```

- **Creating Tables:**

```
CREATE TABLE `admin` (
`ID` decimal(7,0) NOT NULL,
`A_USERNAME` varchar(50) NOT NULL,
`A_PASSWORD` varchar(50) NOT NULL
);
```

```
CREATE TABLE `customer` (
`C_ID` decimal(6,0) NOT NULL,
`C_FNAME` varchar(30) NOT NULL,
`C_LNAME` varchar(30) DEFAULT NULL,
`C_AGE` int(11) NOT NULL,
`C_SEX` varchar(6) NOT NULL,
`C_PHNO` decimal(10,0) NOT NULL,
`C_MAIL` varchar(40) DEFAULT NULL
);
```

```
CREATE TABLE `emplogin` (
`E_ID` decimal(7,0) NOT NULL,
`E_USERNAME` varchar(20) NOT NULL,
`E_PASS` varchar(30) NOT NULL
);
```

```
CREATE TABLE `employee` (
`E_ID` decimal(7,0) NOT NULL,
`E_FNAME` varchar(30) NOT NULL,
`E_LNAME` varchar(30) DEFAULT NULL,
`BDATE` date NOT NULL,
`E_AGE` int(11) NOT NULL,
`E_SEX` varchar(6) NOT NULL,
`E_TYPE` varchar(20) NOT NULL,
`E_JDATE` date NOT NULL,
`E_SAL` decimal(8,2) NOT NULL,
`E_PHNO` decimal(10,0) NOT NULL,
`E_MAIL` varchar(40) DEFAULT NULL,
`E_ADD` varchar(40) DEFAULT NULL
);
```

```
CREATE TABLE `meds` (
`MED_ID` decimal(6,0) NOT NULL,
`MED_NAME` varchar(50) NOT NULL,
`MED_QTY` int(11) NOT NULL,
`CATEGORY` varchar(20) DEFAULT NULL,
`MED_PRICE` decimal(6,2) NOT NULL,
`LOCATION_RACK` varchar(30) DEFAULT NULL
);
```

```

CREATE TABLE `purchase` (
  `P_ID` decimal(4,0) NOT NULL,
  `SUP_ID` decimal(3,0) NOT NULL,
  `MED_ID` decimal(6,0) NOT NULL,
  `P_QTY` int(11) NOT NULL,
  `P_COST` decimal(8,2) NOT NULL,
  `PUR_DATE` date NOT NULL,
  `MFG_DATE` date NOT NULL,
  `EXP_DATE` date NOT NULL
);

CREATE TABLE `sales` (
  `SALE_ID` int(11) NOT NULL,
  `C_ID` decimal(6,0) NOT NULL,
  `S_DATE` date DEFAULT NULL,
  `S_TIME` time DEFAULT NULL,
  `TOTAL_AMT` decimal(8,2) DEFAULT NULL,
  `E_ID` decimal(7,0) NOT NULL
);

CREATE TABLE `sales_items` (
  `SALE_ID` int(11) NOT NULL,
  `MED_ID` decimal(6,0) NOT NULL,
  `SALE_QTY` int(11) NOT NULL,
  `TOT_PRICE` decimal(8,2) NOT NULL
);

CREATE TABLE `suppliers` (
  `SUP_ID` decimal(3,0) NOT NULL,
  `SUP_NAME` varchar(25) NOT NULL,
  `SUP_ADD` varchar(30) NOT NULL,
  `SUP_PHNO` decimal(10,0) NOT NULL,
  `SUP_MAIL` varchar(40) NOT NULL
);

```

INSERT STATEMENTS:

```
INSERT INTO `admin` (`ID`, `A_USERNAME`, `A_PASSWORD`) VALUES
('1', 'admin', 'password');
```

```
INSERT INTO `customer` (`C_ID`, `C_FNAME`, `C_LNAME`, `C_AGE`, `C_SEX`, `C_PHNO`, `C_MAIL`)
VALUES
('987101', 'Safia', 'Malik', 22, 'Female', '9632587415', 'safia@gmail.com'),
('987102', 'Varun', 'Ilango', 24, 'Male', '9987565423', 'varun@gmail.com'),
('987103', 'Suja', 'Suresh', 45, 'Female', '7896541236', 'suja@hotmail.com'),
('987104', 'Agatha', 'Elizabeth', 30, 'Female', '7845129635', 'agatha@gmail.com'),
('987105', 'Zayed', 'Shah', 40, 'Male', '6789541235', 'zshah@hotmail.com'),
('987106', 'Vijay', 'Kumar', 60, 'Male', '8996574123', 'vijayk@yahoo.com'),
('987107', 'Meera', 'Das', 35, 'Female', '7845963259', 'meera@gmail.com');
```

```
INSERT INTO `emplogin` (`E_ID`, `E_USERNAME`, `E_PASS`) VALUES
('4567005', 'amaya', 'pass1'),
('4567002', 'anita', 'pass2'),
('4567010', 'daniel', 'pass3'),
('4567003', 'harish', 'pass4'),
('4567009', 'shayla', 'pass5'),
('4567006', 'shoaib', 'pass6'),
('4567001', 'varshini', 'pass7');
```

```

INSERT INTO `employee` ('E_ID', 'E_FNAME', 'E_LNAME', 'BDATE', 'E AGE', 'E_SEX', 'E_TYPE',
'E_JDATE', 'E_SAL', 'E_PHNO', 'E_MAIL', 'E_ADD') VALUES
('1', 'Admin', '-', '1989-05-24', 30, 'Female', 'Admin', '2009-06-24', '95000.00', '9874563219',
'admin@pharmacia.com', 'Chennai'),
('4567001', 'Varshini', 'Elangovan', '1995-10-05', 25, 'Female', 'Pharmacist', '2017-11-12', '25000.00',
'9967845123', 'evarsh@hotmail.com', 'Thiruvanmiyur'),
('4567002', 'Anita', 'Shree', '2000-10-03', 20, 'Female', 'Pharmacist', '2012-10-06', '45000.00', '8546123566',
'anita@gmail.com', 'Adyar'),
('4567003', 'Harish', 'Raja', '1998-02-01', 22, 'Male', 'Pharmacist', '2019-07-06', '21000.00', '7854123694',
'harishraja@live.com', 'T.Nagar'),
('4567005', 'Amaya', 'Singh', '1992-01-02', 28, 'Female', 'Pharmacist', '2017-05-16', '32000.00', '7894532165',
'amaya@gmail.com', 'Kottivakkam'),
('4567006', 'Shoaib', 'Ahmed', '1999-12-11', 20, 'Male', 'Pharmacist', '2018-09-05', '28000.00', '7896541234',
'shoaib@hotmail.com', 'Porur'),
('4567009', 'Shayla', 'Hussain', '1980-02-28', 40, 'Female', 'Manager', '2010-05-06', '80000.00', '7854123695',
'shaylah@gmail.com', 'Adyar'),
('4567010', 'Daniel', 'James', '1993-04-05', 27, 'Male', 'Pharmacist', '2016-01-05', '30000.00', '7896541235',
'daniels@gmail.com', 'Kodambakkam');

```

```

INSERT INTO `meds` ('MED_ID', 'MED_NAME', 'MED_QTY', 'CATEGORY', 'MED_PRICE',
'LOCATION_RACK') VALUES
('123001', 'Dolo 650 MG', 625, 'Tablet', '1.00', 'rack 5'),
('123002', 'Panadol Cold & Flu', 90, 'Tablet', '2.50', 'rack 6'),
('123003', 'Livogen', 25, 'Capsule', '5.00', 'rack 3'),
('123004', 'Gelusil', 440, 'Tablet', '1.25', 'rack 4'),
('123005', 'Cyclopam', 120, 'Tablet', '6.00', 'rack 2'),
('123006', 'Benadryl 200 ML', 35, 'Syrup', '50.00', 'rack 10'),
('123007', 'Lopamide', 15, 'Capsule', '5.00', 'rack 7'),
('123008', 'Vitamic C', 90, 'Tablet', '3.00', 'rack 8'),
('123009', 'Omeprazole', 35, 'Capsule', '4.00', 'rack 3'),
('123010', 'Concur 5 MG', 600, 'Tablet', '3.50', 'rack 9'),
('123011', 'Augmentin 250 ML', 115, 'Syrup', '80.00', 'rack 7');

```

```

INSERT INTO `purchase` ('P_ID', 'SUP_ID', 'MED_ID', 'P_QTY', 'P_COST', 'PUR_DATE', 'MFG_DATE',
'EXP_DATE') VALUES
('1001', '136', '123010', 200, '1500.50', '2020-03-01', '2019-05-05', '2021-05-10'),
('1002', '123', '123002', 1000, '3000.00', '2020-02-01', '2018-06-01', '2020-12-05'),
('1003', '145', '123006', 20, '800.00', '2020-04-22', '2017-02-05', '2020-07-01'),
('1004', '156', '123004', 250, '1000.00', '2020-04-02', '2020-05-06', '2023-05-06'),
('1005', '123', '123005', 200, '1200.00', '2020-02-01', '2019-08-02', '2021-04-01'),
('1006', '162', '123010', 500, '1500.00', '2019-04-22', '2018-01-01', '2020-05-02'),
('1007', '123', '123001', 500, '450.00', '2020-01-02', '2019-01-05', '2022-03-06');

```

```

INSERT INTO `sales` ('SALE_ID', 'C_ID', 'S_DATE', 'S_TIME', 'TOTAL_AMT', 'E_ID') VALUES
(1, '987101', '2020-04-15', '13:23:03', '180.00', '4567009'),
(2, '987106', '2020-04-21', '20:19:31', '585.00', '1'),
(3, '987103', '2020-04-15', '11:23:53', '120.00', '4567010'),
(4, '987104', '2020-04-14', '18:20:00', '955.00', '4567006'),
(5, '987103', '2020-04-21', '15:24:43', '45.00', '1'),
(6, '987102', '2020-03-11', '10:24:43', '140.00', '4567001'),
(7, '987105', '2020-04-24', '00:25:54', '350.00', '1'),
(8, '987104', '2020-04-24', '00:47:47', '35.00', '4567001'),
(12, '987103', '2020-04-24', '19:33:16', '60.00', '1'),
(13, '987104', '2020-04-24', '21:30:43', '700.00', '1');

```

```
INSERT INTO `sales_items` ('SALE_ID', 'MED_ID', 'QTY SOLD', 'TOTAL PRICE') VALUES  
(1, '123001', 1, '100.00'),  
(1, '123005', 2, '80.00'),  
(1, '123003', 1, '0.00'),  
(2, '123008', 5, '120.00'),  
(2, '123010', 20, '120.00'),  
(2, '123006', 1, '60.00'),  
(3, '123001', 2, '20.00'),  
(3, '123003', 1, '0.00'),  
(4, '123001', 1, '100.00'),  
(4, '123005', 2, '80.00'),  
(5, '123005', 1, '6.00'),  
(6, '123001', 1, '20.00'),  
(6, '123003', 1, '0.00'),  
(7, '123004', 2, '20.00'),  
(8, '123008', 3, '15.00');
```

SHOW TABLES:

```
mysql> show tables;  
+-----+  
| Tables_in_medical |  
+-----+  
| admin           |  
| customer        |  
| emplogin        |  
| employee        |  
| meds            |  
| purchase         |  
| sales           |  
| sales_items     |  
| suppliers        |  
+-----+  
9 rows in set (0.00 sec)
```

DESC ADMIN:

```
mysql> desc admin;  
+-----+-----+-----+-----+-----+-----+  
| Field    | Type     | Null | Key  | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| ID       | decimal(7,0) | NO   | MUL  | NULL    |       |  
| A_USERNAME | varchar(50) | NO   | PRI   | NULL    |       |  
| A_PASSWORD | varchar(50) | NO   |       | NULL    |       |  
+-----+-----+-----+-----+-----+-----+  
3 rows in set (0.00 sec)
```

DESC CUSTOMER:

```
mysql> desc customer;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| C_ID  | decimal(6,0) | NO   | PRI  | NULL    |       |
| C_FNAME | varchar(30) | NO   |       | NULL    |       |
| C_LNAME | varchar(30) | YES  |       | NULL    |       |
| C_AGE  | int     | NO   |       | NULL    |       |
| C_SEX   | varchar(6)  | NO   |       | NULL    |       |
| C_PHNO  | decimal(10,0) | NO   | UNI  | NULL    |       |
| C_MAIL  | varchar(40) | YES  | UNI  | NULL    |       |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

DESC EMPLOGIN:

```
mysql> desc emplogin;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| E_ID  | decimal(7,0) | NO   | MUL  | NULL    |       |
| E_USERNAME | varchar(20) | NO   | PRI  | NULL    |       |
| E_PASS  | varchar(30) | NO   |       | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

DESC EMPLOYEE:

```
mysql> desc employee;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| E_ID  | decimal(7,0) | NO   | PRI  | NULL    |       |
| E_FNAME | varchar(30) | NO   |       | NULL    |       |
| E_LNAME | varchar(30) | YES  |       | NULL    |       |
| BDATE  | date    | NO   |       | NULL    |       |
| E_AGE  | int     | NO   |       | NULL    |       |
| E_SEX   | varchar(6)  | NO   |       | NULL    |       |
| E_TYPE  | varchar(20) | NO   |       | NULL    |       |
| E_JDATE | date    | NO   |       | NULL    |       |
| E_SAL   | decimal(8,2) | NO   |       | NULL    |       |
| E_PHNO  | decimal(10,0) | NO   |       | NULL    |       |
| E_MAIL  | varchar(40) | YES  |       | NULL    |       |
| E_ADD   | varchar(40) | YES  |       | NULL    |       |
+-----+-----+-----+-----+-----+-----+
12 rows in set (0.00 sec)
```

DESC MEDS:

```
mysql> desc meds;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| MED_ID | decimal(6,0) | NO | PRI | NULL |       |
| MED_NAME | varchar(50) | NO |     | NULL |       |
| MED_QTY | int | NO |     | NULL |       |
| CATEGORY | varchar(20) | YES |     | NULL |       |
| MED_PRICE | decimal(6,2) | NO |     | NULL |       |
| LOCATION_RACK | varchar(30) | YES |     | NULL |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

DESC PURCHASE:

```
mysql> desc purchase;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| P_ID | decimal(4,0) | NO | PRI | NULL |       |
| SUP_ID | decimal(3,0) | NO | MUL | NULL |       |
| MED_ID | decimal(6,0) | NO | PRI | NULL |       |
| P_QTY | int | NO |     | NULL |       |
| P_COST | decimal(8,2) | NO |     | NULL |       |
| PUR_DATE | date | NO |     | NULL |       |
| MFG_DATE | date | NO |     | NULL |       |
| EXP_DATE | date | NO |     | NULL |       |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

DESC SALES:

```
mysql> desc sales;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| SALE_ID | int | NO | PRI | NULL | auto_increment |
| C_ID | decimal(6,0) | NO | MUL | NULL |       |
| S_DATE | date | YES |     | NULL |       |
| S_TIME | time | YES |     | NULL |       |
| TOTAL_AMT | decimal(8,2) | YES |     | NULL |       |
| E_ID | decimal(7,0) | NO | MUL | NULL |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

DESC SALES_ITEMS:

```
mysql> desc sales_items;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| SALE_ID | int    | NO   | PRI   | NULL    |       |
| MED_ID  | decimal(6,0) | NO   | PRI   | NULL    |       |
| SALE_QTY | int    | NO   |       | NULL    |       |
| TOT_PRICE | decimal(8,2) | NO   |       | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

DESC SUPPLIERS:

```
mysql> desc suppliers;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| SUP_ID | decimal(3,0) | NO   | PRI   | NULL    |       |
| SUP_NAME | varchar(25) | NO   |       | NULL    |       |
| SUP_ADD | varchar(30) | NO   |       | NULL    |       |
| SUP_PHNO | decimal(10,0) | NO   |       | NULL    |       |
| SUP_MAIL | varchar(40) | NO   |       | NULL    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

3. Complex queries based on the concepts of constraints, sets, joins, views, Triggers and Cursors.

- **Constraints:**

```
ALTER TABLE `admin`  
ADD PRIMARY KEY (`A_USERNAME`),  
ADD UNIQUE KEY `USERNAME` (`A_USERNAME`),  
ADD KEY `ID` (`ID`);
```

```
ALTER TABLE `customer`  
ADD PRIMARY KEY (`C_ID`),  
ADD UNIQUE KEY `C_PHNO` (`C_PHNO`),  
ADD UNIQUE KEY `C_MAIL` (`C_MAIL`);
```

```
ALTER TABLE `emplogin`  
ADD PRIMARY KEY (`E_USERNAME`),  
ADD KEY `E_ID` (`E_ID`);
```

```
ALTER TABLE `employee`  
ADD PRIMARY KEY (`E_ID`);
```

```
ALTER TABLE `meds`  
ADD PRIMARY KEY (`MED_ID`);
```

```
ALTER TABLE `purchase`  
ADD PRIMARY KEY (`P_ID`,`MED_ID`),  
ADD KEY `SUP_ID` (`SUP_ID`),  
ADD KEY `MED_ID` (`MED_ID`);
```

```
ALTER TABLE `sales`  
ADD PRIMARY KEY (`SALE_ID`),  
ADD KEY `C_ID` (`C_ID`),  
ADD KEY `E_ID` (`E_ID`);
```

```
ALTER TABLE `sales_items`  
ADD PRIMARY KEY (`SALE_ID`,`MED_ID`),  
ADD KEY `MED_ID` (`MED_ID`);
```

```
ALTER TABLE `suppliers`  
ADD PRIMARY KEY (`SUP_ID`);
```

```
ALTER TABLE `sales`  
MODIFY `SALE_ID` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=21;
```

```
ALTER TABLE `admin`  
ADD CONSTRAINT `admin_fk_1` FOREIGN KEY (`ID`) REFERENCES `employee` (`E_ID`);
```

```
ALTER TABLE `emplogin`  
ADD CONSTRAINT `emplogin_fk_1` FOREIGN KEY (`E_ID`) REFERENCES `employee` (`E_ID`);
```

```
ALTER TABLE `purchase`  
ADD CONSTRAINT `purchase_fk_1` FOREIGN KEY (`SUP_ID`) REFERENCES `suppliers` (`SUP_ID`),  
ADD CONSTRAINT `purchase_fk_2` FOREIGN KEY (`MED_ID`) REFERENCES `meds` (`MED_ID`);
```

```

ALTER TABLE `sales`
    ADD CONSTRAINT `sales_fk_1` FOREIGN KEY (`C_ID`) REFERENCES `customer` (`C_ID`),
    ADD CONSTRAINT `sales_fk_2` FOREIGN KEY (`E_ID`) REFERENCES `employee` (`E_ID`);

ALTER TABLE `sales_items`
    ADD CONSTRAINT `sales_items_fk_1` FOREIGN KEY (`SALE_ID`) REFERENCES `sales` (`SALE_ID`),
    ADD CONSTRAINT `sales_items_fk_2` FOREIGN KEY (`MED_ID`) REFERENCES `meds` (`MED_ID`);

```

- **Procedures:**

```
DELIMITER $$
```

```

CREATE DEFINER='root'@'localhost' PROCEDURE `EXPIRY` () NO SQL
BEGIN
    SELECT p_id, sup_id, med_id, p_qty, p_cost, pur_date, mfg_date, exp_date
    FROM purchase
    WHERE exp_date BETWEEN CURDATE() AND DATE_SUB(CURDATE(), INTERVAL -6 MONTH);
END$$

```

```

CREATE DEFINER='root'@'localhost' PROCEDURE `SEARCH_INVENTORY` (IN `search`
VARCHAR(255)) NO SQL
BEGIN
    DECLARE mid DECIMAL(6);
    DECLARE mname VARCHAR(50);
    DECLARE mqty INT;
    DECLARE mcategory VARCHAR(20);
    DECLARE mprice DECIMAL(6,2);
    DECLARE location VARCHAR(30);
    DECLARE exit_loop BOOLEAN DEFAULT FALSE;
    DECLARE MED_CURSOR CURSOR FOR SELECT MED_ID, MED_NAME, MED_QTY, CATEGORY,
MED_PRICE, LOCATION_RACK FROM MEDS;
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET exit_loop=TRUE;
    CREATE TEMPORARY TABLE IF NOT EXISTS T1 (medid DECIMAL(6), medname VARCHAR(50),
medqty INT, medcategory VARCHAR(20), medprice DECIMAL(6,2), medlocation VARCHAR(30));
    OPEN MED_CURSOR;
    med_loop: LOOP
        FETCH FROM MED_CURSOR INTO mid, mname, mqty, mcategory, mprice, location;
        IF exit_loop THEN
            LEAVE med_loop;
        END IF;

        IF(CONCAT(mid, mname, mcategory, location) LIKE CONCAT('%', search, '%')) THEN
            INSERT INTO T1(medid, medname, medqty, medcategory, medprice, medlocation)
            VALUES(mid, mname, mqty, mcategory, mprice, location);
        END IF;
    END LOOP med_loop;
    CLOSE MED_CURSOR;
    SELECT medid, medname, medqty, medcategory, medprice, medlocation FROM T1;
END$$

```

```

CREATE DEFINER='root'@'localhost' PROCEDURE `STOCK` () NO SQL
BEGIN
    SELECT med_id, med_name, med_qty, category, med_price, location_rack
    FROM meds
    WHERE med_qty <= 50;
END$$

```

```

CREATE DEFINER='root'@'localhost' PROCEDURE `TOTAL_AMT` (IN `ID` INT, OUT `AMT`
DECIMAL(8,2)) NO SQL
BEGIN

```

```

UPDATE SALES
    SET S_DATE = SYSDATE(), S_TIME = CURRENT_TIMESTAMP(), TOTAL_AMT = (SELECT
        SUM(TOT_PRICE) FROM SALES_ITEMS WHERE SALES_ITEMS.SALE_ID = ID)
        WHERE SALES.SALE_ID = ID;
        SELECT TOTAL_AMT INTO AMT
        FROM SALES
        WHERE SALE_ID = ID;
END$$

DELIMITER ;

```

- **Functions:**

```
DELIMITER $$
```

```

CREATE DEFINER='root'@'localhost' FUNCTION `P_AMT` ('start' DATE, `end` DATE) RETURNS
DECIMAL(8,2) NO SQL
    DETERMINISTIC
BEGIN
    DECLARE PAMT DECIMAL(8,2) DEFAULT 0.0;
    SELECT SUM(P_COST) INTO PAMT
    FROM PURCHASE
    WHERE PUR_DATE >= start AND PUR_DATE <= end;
    RETURN PAMT;
END$$

```

```

CREATE DEFINER='root'@'localhost' FUNCTION `S_AMT` ('start' DATE, `end` DATE) RETURNS
DECIMAL(8,2) NO SQL
BEGIN
    DECLARE SAMT DECIMAL(8,2) DEFAULT 0.0;
    SELECT SUM(TOTAL_AMT) INTO SAMT
    FROM SALES
    WHERE S_DATE >= start AND S_DATE <= end;
    RETURN SAMT;
END$$

```

```
DELIMITER ;
```

- **Triggers:**

```
DELIMITER $$
```

```

CREATE TRIGGER `QTYDELETE` AFTER DELETE ON `purchase` FOR EACH ROW
BEGIN
    UPDATE meds
    SET MED_QTY = MED_QTY - old.P_QTY
    WHERE meds.MED_ID = old.MED_ID;
END$$

```

```

CREATE TRIGGER `QTYINSERT` AFTER INSERT ON `purchase` FOR EACH ROW
BEGIN
    UPDATE meds
    SET MED_QTY = MED_QTY + new.P_QTY
    WHERE meds.MED_ID = new.MED_ID;
END$$

```

```

CREATE TRIGGER `QTYUPDATE` AFTER UPDATE ON `purchase` FOR EACH ROW
BEGIN
    UPDATE meds
    SET MED_QTY = MED_QTY - old.P_QTY
    WHERE meds.MED_ID = new.MED_ID;
    UPDATE meds
    SET MED_QTY = MED_QTY + new.P_QTY
    WHERE meds.MED_ID = new.MED_ID;
END$$

```

```

CREATE TRIGGER `SALE_ID_DELETE` BEFORE DELETE ON `sales` FOR EACH ROW
BEGIN
    DELETE FROM sales_items
    WHERE sales_items.SALE_ID = old.SALE_ID;
END$$

```

DELIMITER ;

UPDATE STATEMENTS:

- Update total amount for a specific sale:

```

MariaDB [medical]> UPDATE SALES
    -> SET S_DATE = CURDATE(), S_TIME = CURRENT_TIMESTAMP(), TOTAL_AMT = (
    ->     SELECT SUM(TOT_PRICE)
    ->     FROM SALES_ITEMS
    ->     WHERE SALE_ID = 1
    -> )
    -> WHERE SALE_ID = 1;
Query OK, 1 row affected, 1 warning (0.010 sec)
Rows matched: 1  Changed: 1  Warnings: 1

```

- Update medication quantity after deleting a purchase:

```

MariaDB [medical]> UPDATE meds
    -> SET MED_QTY = MED_QTY - 20
    -> WHERE MED_ID = 123001;
Query OK, 1 row affected (0.004 sec)
Rows matched: 1  Changed: 1  Warnings: 0

```

- Update medication quantity after inserting a purchase:

```

MariaDB [medical]> UPDATE meds
    -> SET MED_QTY = MED_QTY + 20
    -> WHERE MED_ID = 123001;
Query OK, 1 row affected (0.002 sec)
Rows matched: 1  Changed: 1  Warnings: 0

```

- Delete sales items before deleting a sale:

```

MariaDB [medical]> DELETE FROM sales_items
    -> WHERE SALE_ID = 1;
Query OK, 2 rows affected (0.009 sec)

```

JOIN STATEMENTS:

- Inner Join to get details of purchases along with supplier information:

```
MariaDB [medical]> SELECT p.p_id, p.sup_id, s.sup_name, p.p_qty, p.p_cost  
-> FROM purchase p  
-> INNER JOIN suppliers s ON p.sup_id = s.sup_id;  
+-----+-----+-----+-----+  
| p_id | sup_id | sup_name | p_qty | p_cost |  
+-----+-----+-----+-----+  
| 1001 | 136   | ABC PharmaSupply | 200   | 1500.50 |  
| 1002 | 123   | XYZ Pharmaceuticals | 1000  | 3000.00 |  
| 1003 | 145   | Daily Pharma Ltd | 20    | 800.00  |  
| 1004 | 156   | MedAll             | 250   | 1000.00 |  
| 1005 | 123   | XYZ Pharmaceuticals | 200   | 1200.00 |  
| 1006 | 162   | MedHead Pharmaceuticals | 500  | 1500.00 |  
| 1007 | 123   | XYZ Pharmaceuticals | 500   | 450.00  |  
+-----+-----+-----+-----+  
7 rows in set (0.001 sec)
```

- Using both left and right join

```
MariaDB [medical]> SELECT s.sup_name, p.p_id, p.p_qty  
-> FROM suppliers s  
-> LEFT JOIN purchase p ON s.sup_id = p.sup_id  
->  
-> UNION  
->  
-> SELECT s.sup_name, p.p_id, p.p_qty  
-> FROM suppliers s  
-> RIGHT JOIN purchase p ON s.sup_id = p.sup_id;  
+-----+-----+-----+  
| sup_name | p_id | p_qty |  
+-----+-----+-----+  
| XYZ Pharmaceuticals | 1002 | 1000 |  
| XYZ Pharmaceuticals | 1005 | 200  |  
| XYZ Pharmaceuticals | 1007 | 500  |  
| ABC PharmaSupply   | 1001 | 200  |  
| Daily Pharma Ltd  | 1003 | 20   |  
| MedAll             | 1004 | 250  |  
| MedHead Pharmaceuticals | 1006 | 500  |  
+-----+-----+-----+
```

- Self Join to find medications with the same category:

```

MariaDB [medical]> SELECT m1.med_name AS Med1, m2.med_name AS Med2
    -> FROM meds m1
    -> INNER JOIN meds m2 ON m1.category = m2.category AND m1.med_id <> m2.med_id;
+-----+-----+
| Med1 | Med2 |
+-----+-----+
| Panadol Cold & Flu | Dolo 650 MG
| Gelusil | Dolo 650 MG
| Cyclopam | Dolo 650 MG
| Vitamic C | Dolo 650 MG
| Concur 5 MG | Dolo 650 MG
| Dolo 650 MG | Panadol Cold & Flu
| Gelusil | Panadol Cold & Flu
| Cyclopam | Panadol Cold & Flu
| Vitamic C | Panadol Cold & Flu
| Concur 5 MG | Panadol Cold & Flu
| Lopamide | Livogen
| Omeprazole | Livogen
| Dolo 650 MG | Gelusil
| Panadol Cold & Flu | Gelusil
| Cyclopam | Gelusil
| Vitamic C | Gelusil
| Concur 5 MG | Gelusil
| Dolo 650 MG | Cyclopam
| Panadol Cold & Flu | Cyclopam
| Gelusil | Cyclopam
| Vitamic C | Cyclopam
| Concur 5 MG | Cyclopam
| Augmentin 250 ML | Benadryl 200 ML
| Livogen | Lopamide
| Omeprazole | Lopamide
| Dolo 650 MG | Vitamic C
| Panadol Cold & Flu | Vitamic C
| Gelusil | Vitamic C
| Cyclopam | Vitamic C
| Concur 5 MG | Vitamic C
| Livogen | Omeprazole
| Lopamide | Omeprazole
| Dolo 650 MG | Concur 5 MG
| Panadol Cold & Flu | Concur 5 MG
| Gelusil | Concur 5 MG
| Cyclopam | Concur 5 MG
| Vitamic C | Concur 5 MG
| Benadryl 200 ML | Augmentin 250 ML
+-----+-----+

```

38 rows in set (0.000 sec)

ORDER BY Statements:

- Order medications by name in ascending order:

```
MariaDB [medical]> SELECT med_id, med_name, med_price
-> FROM meds
-> ORDER BY med_name ASC;
+-----+-----+-----+
| med_id | med_name           | med_price |
+-----+-----+-----+
| 123011 | Augmentin 250 ML   | 80.00    |
| 123006 | Benadryl 200 ML    | 50.00    |
| 123010 | Concur 5 MG        | 3.50     |
| 123005 | Cyclopam            | 6.00     |
| 123001 | Dolo 650 MG         | 1.00     |
| 123004 | Gelusil              | 1.25     |
| 123003 | Livogen               | 5.00     |
| 123007 | Lopamide              | 5.00     |
| 123009 | Omeprazole            | 4.00     |
| 123002 | Panadol Cold & Flu  | 2.50     |
| 123008 | Vitamic C             | 3.00     |
+-----+-----+-----+
11 rows in set (0.000 sec)
```

- Order sales by date in descending order:

```
MariaDB [medical]> SELECT sale_id, s_date, total_amt
-> FROM sales
-> ORDER BY s_date DESC;
+-----+-----+-----+
| sale_id | s_date      | total_amt |
+-----+-----+-----+
| 1       | 2024-10-02  | 180.00    |
| 18      | 2020-12-04  | 160.00    |
| 17      | 2020-12-04  | 57.50     |
| 16      | 2020-12-04  | 30.00     |
| 15      | 2020-12-04  | 420.00    |
| 20      | 2020-12-04  | 150.00    |
| 7       | 2020-04-24  | 350.00    |
| 12      | 2020-04-24  | 60.00     |
| 13      | 2020-04-24  | 62.50     |
| 8       | 2020-04-24  | 35.00     |
| 5       | 2020-04-21  | 45.00     |
| 2       | 2020-04-21  | 585.00    |
| 3       | 2020-04-15  | 120.00    |
| 4       | 2020-04-14  | 955.00    |
| 6       | 2020-03-11  | 140.00    |
+-----+-----+-----+
```

- Order suppliers by name, then by ID:

```
MariaDB [medical]> SELECT sup_id, sup_name
-> FROM suppliers
-> ORDER BY sup_name ASC, sup_id ASC;
+-----+-----+
| sup_id | sup_name          |
+-----+-----+
| 136   | ABC PharmaSupply
| 145   | Daily Pharma Ltd
| 156   | MedAll
| 162   | MedHead Pharmaceuticals
| 123   | XYZ Pharmaceuticals
+-----+
5 rows in set (0.000 sec)
```

GROUP BY Statements

- Group by supplier and count total purchases per supplier:

```
MariaDB [medical]> SELECT sup_id, COUNT(*) AS total_purchases
-> FROM purchase
-> GROUP BY sup_id;
+-----+-----+
| sup_id | total_purchases |
+-----+-----+
| 123    |            3 |
| 136    |            1 |
| 145    |            1 |
| 156    |            1 |
| 162    |            1 |
+-----+
5 rows in set (0.001 sec)
```

- Group by medication category and get the average price:

```
MariaDB [medical]> SELECT category, AVG(med_price) AS avg_price
-> FROM meds
-> GROUP BY category;
+-----+-----+
| category | avg_price |
+-----+-----+
| Capsule  | 4.666667 |
| Syrup    | 65.000000 |
| Tablet   | 2.875000 |
+-----+
3 rows in set (0.003 sec)
```

- Group by sale date and sum the total amount for each date:

```
MariaDB [medical]> SELECT s_date, SUM(total_amt) AS total_sales
-> FROM sales
-> GROUP BY s_date;
+-----+-----+
| s_date      | total_sales |
+-----+-----+
| 2020-03-11  |    140.00   |
| 2020-04-14  |    955.00   |
| 2020-04-15  |    120.00   |
| 2020-04-21  |    630.00   |
| 2020-04-24  |    507.50   |
| 2020-12-04  |    817.50   |
| 2024-10-02  |    180.00   |
+-----+-----+
7 rows in set (0.000 sec)
```

- Group by medication and find the maximum quantity sold:

```
MariaDB [medical]> SELECT med_id, MAX(sale_qty) AS max_quantity_sold
-> FROM sales_items
-> GROUP BY med_id;
+-----+-----+
| med_id | max_quantity_sold |
+-----+-----+
| 123001 |          100 |
| 123002 |           25 |
| 123003 |           75 |
| 123004 |           10 |
| 123005 |           60 |
| 123006 |            3 |
| 123007 |            5 |
| 123008 |           40 |
| 123009 |           10 |
| 123010 |          250 |
| 123011 |            2 |
+-----+-----+
11 rows in set (0.001 sec)
```

- Group by medication name and get the total quantity available:

```
MariaDB [medical]> SELECT med_name, SUM(med_qty) AS total_available
-> FROM meds
-> GROUP BY med_name;
+-----+-----+
| med_name        | total_available |
+-----+-----+
| Augmentin 250 ML |          117 |
| Benadryl 200 ML  |           35 |
| Concur 5 MG     |          600 |
| Cyclopam         |          120 |
| Dolo 650 MG      |          645 |
| Gelusil          |          440 |
| Livogen          |           25 |
| Lopamide          |           15 |
| Omeprazole        |           35 |
| Panadol Cold & Flu |          90 |
| Vitamin C         |           90 |
+-----+-----+
11 rows in set (0.000 sec)
```

Queries

1. Customer with the highest total sales amount:

```
MariaDB [medical]> SELECT C_ID, C_FNAME, C_LNAME, (SELECT SUM(TOTAL_AMT) FROM sales WHERE customer.C_ID = sales.C_ID) AS  
total_sales  
    -> FROM customer  
    -> ORDER BY total_sales DESC  
    -> LIMIT 1;  
+-----+-----+-----+-----+  
| C_ID | C_FNAME | C_LNAME | total_sales |  
+-----+-----+-----+-----+  
| 987104 | Agatha | Elizabeth |      1052.50 |  
+-----+-----+-----+-----+  
1 row in set (0.036 sec)
```

2. Medicine details for the most expensive purchase:

```
MariaDB [medical]> SELECT MED_ID, MED_NAME, MED_PRICE  
    -> FROM meds  
    -> WHERE MED_ID = (SELECT MED_ID FROM purchase ORDER BY P_COST DESC LIMIT 1);  
+-----+-----+-----+  
| MED_ID | MED_NAME           | MED_PRICE |  
+-----+-----+-----+  
| 123002 | Panadol Cold & Flu |      2.50 |  
+-----+-----+-----+  
1 row in set (0.008 sec)
```

3. Employee details for the sale with the highest total amount:

```
MariaDB [medical]> SELECT E_ID, E_FNAME, E_LNAME, E_SAL  
    -> FROM employee  
    -> WHERE E_ID = (SELECT E_ID FROM sales ORDER BY TOTAL_AMT DESC LIMIT 1);  
+-----+-----+-----+-----+  
| E_ID   | E_FNAME | E_LNAME | E_SAL   |  
+-----+-----+-----+-----+  
| 4567006 | Shoaib  | Ahmed   | 28000.00 |  
+-----+-----+-----+-----+  
1 row in set (0.004 sec)
```

4. Supplier ID with the most purchases:

```
MariaDB [medical]> SELECT SUP_ID, (SELECT COUNT(*) FROM purchase WHERE SUP_ID = supplier.SUP_ID) AS purchase_count  
    -> FROM purchase supplier  
    -> ORDER BY purchase_count DESC  
    -> LIMIT 1;  
+-----+-----+  
| SUP_ID | purchase_count |  
+-----+-----+  
| 123   |          3 |  
+-----+-----+  
1 row in set (0.002 sec)
```

5. Average purchase cost of medicines for each supplier:

```
MariaDB [medical]> SELECT SUP_ID, (SELECT AVG(P_COST) FROM purchase WHERE SUP_ID = supplier.SUP_ID) AS avg_purchase_cost  
    -> FROM purchase supplier;  
+-----+-----+  
| SUP_ID | avg_purchase_cost |  
+-----+-----+  
| 123   |     1550.000000 |  
| 123   |     1550.000000 |  
| 123   |     1550.000000 |  
| 136   |     1500.500000 |  
| 145   |      800.000000 |  
| 156   |     1000.000000 |  
| 162   |     1500.000000 |  
+-----+-----+  
7 rows in set (0.001 sec)
```

6. Details of medicine with the highest stock quantity:

```
SELECT MED_ID, MED_NAME, MED_QTY
  -> FROM meds]
  -> WHERE MED_QTY = (SELECT MAX(MED_QTY) FROM meds);
+-----+-----+-----+
| MED_ID | MED_NAME      | MED_QTY |
+-----+-----+-----+
| 123001 | Dolo 650 MG |     645 |
+-----+-----+-----+
1 row in set (0.000 sec)
```

7. Customer who made the earliest purchase:

```
MariaDB [medical]> SELECT C_ID, C_FNAME, C_LNAME
  -> FROM customer
  -> WHERE C_ID = (SELECT C_ID FROM sales ORDER BY S_DATE ASC LIMIT 1);
+-----+-----+-----+
| C_ID   | C_FNAME | C_LNAME |
+-----+-----+-----+
| 987102 | Varun   | Ilango  |
+-----+-----+-----+
1 row in set (0.000 sec)
```

8. Employees who joined after 2015:

```
MariaDB [medical]> SELECT E_ID, E_FNAME, E_LNAME, E_JDATE
  -> FROM employee
  -> WHERE E_JDATE > '2015-01-01';
+-----+-----+-----+-----+
| E_ID   | E_FNAME | E_LNAME | E_JDATE |
+-----+-----+-----+-----+
| 4567001 | Varshini | Elangovan | 2017-11-12 |
| 4567003 | Harish   | Raja       | 2019-07-06 |
| 4567005 | Amaya    | Singh      | 2017-05-16 |
| 4567006 | Shoaib   | Ahmed      | 2018-09-05 |
| 4567010 | Daniel   | James      | 2016-01-05 |
+-----+-----+-----+-----+
5 rows in set (0.000 sec)
```

9. Total quantity of each medicine sold:

```
MariaDB [medical]> SELECT MED_ID, MED_NAME, (SELECT SUM(SALE_QTY) FROM sales_items WHERE meds.MED_ID = sales_items.MED_ID) AS total_sold_qty
  -> FROM meds;
+-----+-----+-----+
| MED_ID | MED_NAME      | total_sold_qty |
+-----+-----+-----+
| 123001 | Dolo 650 MG |      155 |
| 123002 | Panadol Cold & Flu |      35 |
| 123003 | Livogen        |     125 |
| 123004 | Gelusil         |      10 |
| 123005 | Cyclopam        |     140 |
| 123006 | Benadryl 200 ML |       5 |
| 123007 | Lopamide         |       5 |
| 123008 | Vitanic C       |      50 |
| 123009 | Omeprazole       |      15 |
| 123010 | Concur 5 MG    |     250 |
| 123011 | Augmentin 250 ML |       3 |
+-----+-----+-----+
11 rows in set (0.007 sec)
```

10. Employee with the most sales transactions:

```
MariaDB [medical]> SELECT E_ID, (SELECT COUNT(*) FROM sales WHERE employee.E_ID = sales.E_ID) AS total_sales
  -> FROM employee
  -> ORDER BY total_sales DESC
  -> LIMIT 1;
+-----+-----+
| E_ID | total_sales |
+-----+-----+
| 1    |      7 |
+-----+-----+
1 row in set (0.001 sec)
```

11. Medicine with the earliest expiry date:

```
MariaDB [medical]> SELECT MED_ID, MED_NAME, (SELECT MIN(EXP_DATE) FROM purchase WHERE meds.MED_ID = purchase.MED_ID) AS earliest_expiry  
    -> FROM meds;  
+-----+-----+-----+  
| MED_ID | MED_NAME | earliest_expiry |  
+-----+-----+-----+  
| 123001 | Dolo 650 MG | 2022-03-06 |  
| 123002 | Panadol Cold & Flu | 2020-12-05 |  
| 123003 | Livogen | NULL |  
| 123004 | Gelusil | 2023-05-06 |  
| 123005 | Cyclopam | 2021-04-01 |  
| 123006 | Benadryl 200 ML | 2020-07-01 |  
| 123007 | Lopamide | NULL |  
| 123008 | Vitaminic C | NULL |  
| 123009 | Omeprazole | NULL |  
| 123010 | Concur 5 MG | 2020-05-02 |  
| 123011 | Augmentin 250 ML | NULL |  
+-----+-----+-----+
```

12. Average age of customers:

```
MariaDB [medical]> SELECT AVG(C_AGE) AS avg_customer_age  
    -> FROM customer;  
+-----+  
| avg_customer_age |  
+-----+  
|      36.5714 |  
+-----+  
1 row in set (0.000 sec)
```

13. Employee with the highest salary:

```
MariaDB [medical]> SELECT E_ID, E_FNAME, E_LNAME, E_SAL  
    -> FROM employee  
    -> WHERE E_SAL = (SELECT MAX(E_SAL) FROM employee);  
+-----+-----+-----+  
| E_ID | E_FNAME | E_LNAME | E_SAL |  
+-----+-----+-----+  
|     1 | Admin    | -        | 95000.00 |  
+-----+-----+-----+
```

14. Details of purchases made in 2020:

```
MariaDB [medical]> SELECT P_ID, SUP_ID, MED_ID, P_QTY, P_COST, PUR_DATE  
    -> FROM purchase  
    -> WHERE YEAR(PUR_DATE) = 2020;  
+-----+-----+-----+-----+-----+-----+  
| P_ID | SUP_ID | MED_ID | P_QTY | P_COST | PUR_DATE |  
+-----+-----+-----+-----+-----+-----+  
| 1001 | 136    | 123010 | 200   | 1500.50 | 2020-03-01 |  
| 1002 | 123    | 123002 | 1000  | 3000.00 | 2020-02-01 |  
| 1003 | 145    | 123006 | 20    | 800.00  | 2020-04-22 |  
| 1004 | 156    | 123004 | 250   | 1000.00 | 2020-04-02 |  
| 1005 | 123    | 123005 | 200   | 1200.00 | 2020-02-01 |  
| 1007 | 123    | 123001 | 500   | 450.00  | 2020-01-02 |  
+-----+-----+-----+-----+-----+-----+  
6 rows in set (0.000 sec)
```

15. Customer who made the latest purchase:

```
MariaDB [medical]> SELECT C_ID, C_FNAME, C_LNAME  
    -> FROM customer  
    -> WHERE C_ID = (SELECT C_ID FROM sales ORDER BY S_DATE DESC LIMIT 1);  
+-----+-----+-----+  
| C_ID | C_FNAME | C_LNAME |  
+-----+-----+-----+  
| 987101 | Safia    | Malik   |  
+-----+-----+-----+  
1 row in set (0.000 sec)
```

16. Total sales amount for each employee:

```
MariaDB [medical]> SELECT E_ID, (SELECT SUM(TOTAL_AMT) FROM sales WHERE employee.E_ID = sales.E_ID) AS total_sales_amount
-> FROM employee;
+-----+-----+
| E_ID | total_sales_amount |
+-----+-----+
| 1 | 1547.50 |
| 4567001 | 547.50 |
| 4567002 | NULL |
| 4567003 | NULL |
| 4567005 | NULL |
| 4567006 | 955.00 |
| 4567009 | 180.00 |
| 4567010 | 120.00 |
+-----+
8 rows in set (0.000 sec)
```

17. Details of medicines in 'Tablet' category:

```
MariaDB [medical]> SELECT MED_ID, MED_NAME, MED_QTY, MED_PRICE
-> FROM meds
-> WHERE CATEGORY = 'Tablet';
+-----+-----+-----+
| MED_ID | MED_NAME | MED_QTY | MED_PRICE |
+-----+-----+-----+
| 123001 | Dolo 650 MG | 645 | 1.00 |
| 123002 | Panadol Cold & Flu | 90 | 2.50 |
| 123004 | Gelusil | 440 | 1.25 |
| 123005 | Cyclopam | 120 | 6.00 |
| 123008 | Vitamic C | 90 | 3.00 |
| 123010 | Concur 5 MG | 600 | 3.50 |
+-----+-----+-----+
6 rows in set (0.000 sec)
```

18. Customer with the most purchases:

```
MariaDB [medical]> SELECT C_ID, (SELECT COUNT(*) FROM sales WHERE customer.C_ID = sales.C_ID) AS total_purchases
-> FROM customer
-> ORDER BY total_purchases DESC
-> LIMIT 1;
+-----+
| C_ID | total_purchases |
+-----+
| 987103 | 5 |
+-----+
1 row in set (0.000 sec)
```

19. Supplier details for a given medicine ID:

```
MariaDB [medical]> SELECT SUP_ID, (SELECT MED_NAME FROM meds WHERE meds.MED_ID = purchase.MED_ID) AS medicine_name
-> FROM purchase
-> WHERE MED_ID = 123010;
+-----+
| SUP_ID | medicine_name |
+-----+
| 136 | Concur 5 MG |
| 162 | Concur 5 MG |
+-----+
2 rows in set (0.001 sec)
```

20. Total number of medicines in each category:

```
MariaDB [medical]> SELECT CATEGORY, (SELECT COUNT(*) FROM meds WHERE meds.CATEGORY = category) AS total_medicines
-> FROM meds category;
+-----+-----+
| CATEGORY | total_medicines |
+-----+-----+
| Tablet | 11 |
| Tablet | 11 |
| Capsule | 11 |
| Tablet | 11 |
| Tablet | 11 |
| Syrup | 11 |
| Capsule | 11 |
| Tablet | 11 |
| Capsule | 11 |
| Tablet | 11 |
| Syrup | 11 |
+-----+
11 rows in set (0.000 sec)
```

21. Employee with the most recent joining date:

```
MariaDB [medical]> SELECT E_ID, E_FNAME, E_LNAME, E_JDATE
    -> FROM employee
    -> WHERE E_JDATE = (SELECT MAX(E_JDATE) FROM employee);
+-----+-----+-----+-----+
| E_ID | E_FNAME | E_LNAME | E_JDATE |
+-----+-----+-----+-----+
| 4567003 | Harish | Raja | 2019-07-06 |
+-----+-----+-----+-----+
1 row in set (0.000 sec)
```

22. Total amount spent by each customer:

```
MariaDB [medical]> SELECT C_ID, (SELECT SUM(TOTAL_AMT) FROM sales WHERE customer.C_ID = sales.C_ID) AS total_spent
    -> FROM customer;
+-----+-----+
| C_ID | total_spent |
+-----+-----+
| 987105 | 510.00 |
| 987104 | 1052.50 |
| 987107 | 420.00 |
| 987103 | 432.50 |
| 987106 | 615.00 |
| 987101 | 180.00 |
| 987102 | 140.00 |
+-----+-----+
7 rows in set (0.002 sec)
```

23. Details of the medicine with the highest price:

```
MariaDB [medical]> SELECT MED_ID, MED_NAME, MED_PRICE
    -> FROM meds
    -> WHERE MED_PRICE = (SELECT MAX(MED_PRICE) FROM meds);
+-----+-----+-----+
| MED_ID | MED_NAME | MED_PRICE |
+-----+-----+-----+
| 123011 | Augmentin 250 ML | 80.00 |
+-----+-----+-----+
1 row in set (0.000 sec)
```

24. Customer who spent the most in a single purchase:

```
MariaDB [medical]> SELECT C_ID, C_FNAME, C_LNAME
    -> FROM customer
    -> WHERE C_ID = (SELECT C_ID FROM sales ORDER BY TOTAL_AMT DESC LIMIT 1);
+-----+-----+-----+
| C_ID | C_FNAME | C_LNAME |
+-----+-----+-----+
| 987104 | Agatha | Elizabeth |
+-----+-----+-----+
1 row in set (0.000 sec)
```

25. Total quantity of a specific medicine sold:

```
MariaDB [medical]> SELECT MED_ID, MED_NAME, (SELECT SUM(SALE_QTY) FROM sales_items WHERE meds.MED_ID = sales_items.MED_ID) AS total_qty_sold
    -> FROM meds
    -> WHERE MED_ID = 123001;
+-----+-----+-----+
| MED_ID | MED_NAME | total_qty_sold |
+-----+-----+-----+
| 123001 | Dolo 650 MG | 155 |
+-----+-----+-----+
1 row in set (0.000 sec)
```

26. Suppliers who supplied a specific medicine:

```
MariaDB [medical]> SELECT SUP_ID
    -> FROM purchase
    -> WHERE MED_ID = 123010;
+-----+
| SUP_ID |
+-----+
|   136 |
|   162 |
+-----+
2 rows in set (0.000 sec)
```

27. Details of all managers:

```
MariaDB [medical]> SELECT E_ID, E_FNAME, E_LNAME, E_TYPE, E_SAL
    -> FROM employee
    -> WHERE E_TYPE = 'Manager';
+-----+-----+-----+-----+
| E_ID | E_FNAME | E_LNAME | E_TYPE | E_SAL |
+-----+-----+-----+-----+
| 4567009 | Shayla | Hussain | Manager | 80000.00 |
+-----+-----+-----+-----+
1 row in set (0.000 sec)
```

28. Total purchase cost for each supplier:

```
MariaDB [medical]> SELECT SUP_ID, (SELECT SUM(P_COST) FROM purchase WHERE purchase.SUP_ID = supplier.SUP_ID) AS total_purchase_cost
    -> FROM purchase supplier;
+-----+-----+
| SUP_ID | total_purchase_cost |
+-----+-----+
| 123 | 4650.00 |
| 123 | 4650.00 |
| 123 | 4650.00 |
| 136 | 1500.50 |
| 145 | 800.00 |
| 156 | 1000.00 |
| 162 | 1500.00 |
+-----+-----+
7 rows in set (0.000 sec)
```

29. Details of the first sale made:

```
MariaDB [medical]> SELECT SALE_ID, C_ID, S_DATE, S_TIME, TOTAL_AMT
    -> FROM sales
    -> ORDER BY S_DATE ASC, S_TIME ASC
    -> LIMIT 1;
+-----+-----+-----+-----+
| SALE_ID | C_ID | S_DATE | S_TIME | TOTAL_AMT |
+-----+-----+-----+-----+
|       6 | 987102 | 2020-03-11 | 10:24:43 | 140.00 |
+-----+-----+-----+-----+
1 row in set (0.000 sec)
```

30. Employees who are pharmacists:

```
MariaDB [medical]> SELECT E_ID, E_FNAME, E_LNAME, E_SAL
-> FROM employee
-> WHERE E_TYPE = 'Pharmacist';
+-----+-----+-----+
| E_ID | E_FNAME | E_LNAME | E_SAL |
+-----+-----+-----+
| 4567001 | Varshini | Elangovan | 25000.00 |
| 4567002 | Anita | Shree | 45000.00 |
| 4567003 | Harish | Raja | 21000.00 |
| 4567005 | Amaya | Singh | 32000.00 |
| 4567006 | Shoaib | Ahmed | 28000.00 |
| 4567010 | Daniel | James | 30000.00 |
+-----+-----+-----+
6 rows in set (0.000 sec)
```

31. Total sales amount for each customer:

```
MariaDB [medical]> SELECT C_ID, (SELECT SUM(TOTAL_AMT) FROM sales WHERE customer.C_ID = sales.C_ID) AS total_sales
-> FROM customer;
+-----+-----+
| C_ID | total_sales |
+-----+-----+
| 987105 | 510.00 |
| 987104 | 1052.50 |
| 987107 | 420.00 |
| 987103 | 432.50 |
| 987106 | 615.00 |
| 987101 | 180.00 |
| 987102 | 140.00 |
+-----+-----+
7 rows in set (0.000 sec)
```

32. Details of all employees with salary greater than 30000:

```
MariaDB [medical]> SELECT E_ID, E_FNAME, E_LNAME, E_SAL
-> FROM employee
-> WHERE E_SAL > 30000;
+-----+-----+-----+
| E_ID | E_FNAME | E_LNAME | E_SAL |
+-----+-----+-----+
| 1 | Admin | - | 95000.00 |
| 4567002 | Anita | Shree | 45000.00 |
| 4567005 | Amaya | Singh | 32000.00 |
| 4567009 | Shayla | Hussain | 80000.00 |
+-----+-----+-----+
4 rows in set (0.000 sec)
```

33. Total quantity of medicines purchased for each supplier:

```
MariaDB [medical]> SELECT SUP_ID, (SELECT SUM(P_QTY) FROM purchase WHERE purchase.SUP_ID = supplier.SUP_ID) AS total_qty_purchased
-> FROM purchase supplier;
+-----+-----+
| SUP_ID | total_qty_purchased |
+-----+-----+
| 123 | 1700 |
| 123 | 1700 |
| 123 | 1700 |
| 136 | 200 |
| 145 | 20 |
| 156 | 250 |
| 162 | 500 |
+-----+-----+
7 rows in set (0.000 sec)
```

34. Customer who spent the least in a single purchase:

```
MariaDB [medical]> SELECT C_ID, C_FNAME, C_LNAME
    -> FROM customer
    -> WHERE C_ID = (SELECT C_ID FROM sales ORDER BY TOTAL_AMT ASC LIMIT 1);
+-----+-----+-----+
| C_ID | C_FNAME | C_LNAME |
+-----+-----+-----+
| 987106 | Vijay   | Kumar   |
+-----+-----+-----+
1 row in set (0.000 sec)
```

35. Details of the medicine with the lowest stock quantity:

```
MariaDB [medical]> SELECT MED_ID, MED_NAME, MED_QTY
    -> FROM meds
    -> WHERE MED_QTY = (SELECT MIN(MED_QTY) FROM meds);
+-----+-----+-----+
| MED_ID | MED_NAME | MED_QTY |
+-----+-----+-----+
| 123007 | Lopamide |      15 |
+-----+-----+-----+
1 row in set (0.001 sec)
```

36. Total number of purchases for each customer:

```
MariaDB [medical]> SELECT C_ID, (SELECT COUNT(*) FROM sales WHERE customer.C_ID = sales.C_ID) AS total_purchases
    -> FROM customer;
+-----+-----+
| C_ID | total_purchases |
+-----+-----+
| 987105 |          2 |
| 987104 |          3 |
| 987107 |          1 |
| 987103 |          5 |
| 987106 |          2 |
| 987101 |          1 |
| 987102 |          1 |
+-----+-----+
7 rows in set (0.000 sec)
```

37. Details of medicines in a specific rack location:

```
MariaDB [medical]> SELECT MED_ID, MED_NAME, MED_QTY, MED_PRICE
    -> FROM meds
    -> WHERE LOCATION_RACK = 'rack 5';
+-----+-----+-----+-----+
| MED_ID | MED_NAME     | MED_QTY | MED_PRICE |
+-----+-----+-----+-----+
| 123001 | Dolo 650 MG |     645 |      1.00 |
+-----+-----+-----+-----+
1 row in set (0.000 sec)
```

38. Suppliers who supplied medicines in 2020:

```
MariaDB [medical]> SELECT DISTINCT SUP_ID
    -> FROM purchase
    -> WHERE YEAR(PUR_DATE) = 2020;
+-----+
| SUP_ID |
+-----+
| 123   |
| 136   |
| 145   |
| 156   |
+-----+
4 rows in set (0.000 sec)
```

39. Employee with the least salary:

```
MariaDB [medical]> SELECT E_ID, E_FNAME, E_LNAME, E_SAL
    -> FROM employee
    -> WHERE E_SAL = (SELECT MIN(E_SAL) FROM employee);
+-----+-----+-----+
| E_ID | E_FNAME | E_LNAME | E_SAL |
+-----+-----+-----+
| 4567003 | Harish | Raja | 21000.00 |
+-----+-----+-----+
1 row in set (0.000 sec)
```

40. Total sales amount for a specific employee:

```
MariaDB [medical]> SELECT E_ID, E_FNAME, E_LNAME, (SELECT SUM(TOTAL_AMT) FROM sales WHERE employee.E_ID = sales.E_ID) AS total_sales
    -> FROM employee
    -> WHERE E_ID = 1;
+-----+-----+-----+
| E_ID | E_FNAME | E_LNAME | total_sales |
+-----+-----+-----+
| 1 | Admin | - | 1547.50 |
+-----+-----+-----+
1 row in set (0.000 sec)
```

41. Details of medicines purchased by a specific supplier:

```
MariaDB [medical]> SELECT MED_ID, MED_NAME, (SELECT SUM(P_QTY) FROM purchase WHERE meds.MED_ID = purchase.MED_ID AND SUP_ID = 201) AS total_qty_purchased
    -> FROM meds;
+-----+-----+-----+
| MED_ID | MED_NAME | total_qty_purchased |
+-----+-----+-----+
| 123001 | Dolo 650 MG | NULL |
| 123002 | Panadol Cold & Flu | NULL |
| 123003 | Livogen | NULL |
| 123004 | Gelusil | NULL |
| 123005 | Cyclopam | NULL |
| 123006 | Benadryl 200 ML | NULL |
| 123007 | Lopamide | NULL |
| 123008 | Vitamin C | NULL |
| 123009 | Omeprazole | NULL |
| 123010 | Concur 5 MG | NULL |
| 123011 | Augmentin 250 ML | NULL |
+-----+-----+-----+
11 rows in set (0.001 sec)
```

42. Employee who joined recently:

```
MariaDB [medical]> SELECT E_ID, E_FNAME, E_LNAME, E_JDATE
    -> FROM employee
    -> WHERE E_JDATE = (SELECT MAX(E_JDATE) FROM employee);
+-----+-----+-----+
| E_ID | E_FNAME | E_LNAME | E_JDATE |
+-----+-----+-----+
| 4567003 | Harish | Raja | 2019-07-06 |
+-----+-----+-----+
1 row in set (0.000 sec)
```

43. Total quantity of each category of medicine:

```
MariaDB [medical]> SELECT CATEGORY, (SELECT SUM(MED_QTY) FROM meds WHERE meds.CATEGORY = category) AS total_qty
    -> FROM meds category;
+-----+-----+
| CATEGORY | total_qty |
+-----+-----+
| Tablet | 2212 |
| Tablet | 2212 |
| Capsule | 2212 |
| Tablet | 2212 |
| Tablet | 2212 |
| Syrup | 2212 |
| Capsule | 2212 |
| Tablet | 2212 |
| Capsule | 2212 |
| Tablet | 2212 |
| Syrup | 2212 |
+-----+-----+
```

44. Total amount spent by a specific customer:

```
MariaDB [medical]> SELECT C_ID, C_FNAME, C_LNAME, (SELECT SUM(TOTAL_AMT) FROM sales WHERE customer.C_ID = sales.C_ID) AS total_spent
-> FROM customer
-> WHERE C_ID = 987101;
+-----+-----+-----+
| C_ID | C_FNAME | C_LNAME | total_spent |
+-----+-----+-----+
| 987101 | Safia | Malik | 180.00 |
+-----+-----+-----+
1 row in set (0.001 sec)
```

45. Suppliers who supplied medicines with expiry date in 2021:

```
MariaDB [medical]> SELECT DISTINCT SUP_ID
-> FROM purchase
-> WHERE YEAR(EXP_DATE) = 2021;
+-----+
| SUP_ID |
+-----+
| 123 |
| 136 |
+-----+
2 rows in set (0.000 sec)
```

46. Total quantity of a specific medicine purchased:

```
MariaDB [medical]> SELECT MED_ID, MED_NAME, (SELECT SUM(P_QTY) FROM purchase WHERE meds.MED_ID = purchase.MED_ID) AS total_qty_purchase
d
-> FROM meds
-> WHERE MED_ID = 123010;
+-----+-----+-----+
| MED_ID | MED_NAME | total_qty_purchased |
+-----+-----+-----+
| 123010 | Concur 5 MG | 700 |
+-----+-----+-----+
1 row in set (0.001 sec)
```

47. Details of all employees with age greater than 30:

```
MariaDB [medical]> SELECT E_ID, E_FNAME, E_LNAME, E_AGE
-> FROM employee
-> WHERE E_AGE > 30;
+-----+-----+-----+-----+
| E_ID | E_FNAME | E_LNAME | E_AGE |
+-----+-----+-----+-----+
| 4567009 | Shayla | Hussain | 40 |
+-----+-----+-----+-----+
1 row in set (0.000 sec)
```

48. Total sales amount for each year:

```
MariaDB [medical]> SELECT YEAR(S_DATE) AS year, (SELECT SUM(TOTAL_AMT) FROM sales WHERE YEAR(sales.S_DATE) = year) AS total_sales_amount
t
-> FROM sales
-> GROUP BY YEAR(S_DATE);
+-----+
| year | total_sales_amount |
+-----+
| 2020 | 3170.00 |
| 2024 | 180.00 |
+-----+
2 rows in set (0.001 sec)
```

49. Employee who made the most sales transactions in 2020:

```
MariaDB [medical]> SELECT E_ID, (SELECT COUNT(*) FROM sales WHERE employee.E_ID = sales.E_ID AND YEAR(S_DATE) = 2020) AS total_sales
-> FROM employee
-> ORDER BY total_sales DESC
-> LIMIT 1;
+-----+
| E_ID | total_sales |
+-----+
| 1 | 7 |
+-----+
1 row in set (0.001 sec)
```

50. Total purchase cost for each medicine category:

```
MariaDB [medical]> SELECT category.CATEGORY,
->           (SELECT SUM(purchase.P_COST)
->             FROM purchase
->             JOIN meds ON meds.MED_ID = purchase.MED_ID
->             WHERE meds.CATEGORY = category.CATEGORY) AS total_purchase_cost
->   FROM meds category;
+-----+-----+
| CATEGORY | total_purchase_cost |
+-----+-----+
| Tablet   |      8650.50 |
| Tablet   |      8650.50 |
| Capsule  |        NULL |
| Tablet   |      8650.50 |
| Tablet   |      8650.50 |
| Syrup    |      800.00 |
| Capsule  |        NULL |
| Tablet   |      8650.50 |
| Capsule  |        NULL |
| Tablet   |      8650.50 |
| Syrup    |      800.00 |
+-----+
11 rows in set (0.001 sec)
```

4. Analyzing the pitfalls, identifying the dependencies, and applying normalizations

Non-Normalized customer Table

C_ID	C_FNAME	C_LNAME	C AGE	C SEX	C PHNO	C MAIL
987101	Safia	Malik	22	Female	9632587415, 1234567890	safia@gmail.com, safia@xyz.com
987102	Varun	Ilango	24	Male	9987565423	varun@gmail.com
987103	Suja	Suresh	45	Female	7896541236	suja@hotmail.com
987104	Agatha	Elizabeth	30	Female	7845129635	agatha@gmail.com
987105	Zayed	Shah	40	Male	6789541235	zshah@hotmail.com
987106	Vijay	Kumar	60	Male	8996574123	vijayk@yahoo.com
987107	Meera	Das	35	Female	7845963259	meera@gmail.com

Normalize to 1NF

Condition: Each cell must contain atomic values. We will separate the multiple values into different rows.

1NF Table

C_ID	C_FNAME	C_LNAME	C AGE	C SEX	C PHNO	C MAIL
987101	Safia	Malik	22	Female	9632587415	safia@gmail.com
987101	Safia	Malik	22	Female	1234567890	safia@xyz.com
987102	Varun	Ilango	24	Male	9987565423	varun@gmail.com
987103	Suja	Suresh	45	Female	7896541236	suja@hotmail.com
987104	Agatha	Elizabeth	30	Female	7845129635	agatha@gmail.com
987105	Zayed	Shah	40	Male	6789541235	zshah@hotmail.com
987106	Vijay	Kumar	60	Male	8996574123	vijayk@yahoo.com
987107	Meera	Das	35	Female	7845963259	meera@gmail.com

2NF Table

C_ID	C_FNAME	C_LNAME	C AGE	C SEX	C PHNO	C MAIL
987101	Safia	Malik	22	Female	9632587415	safia@gmail.com
987101	Safia	Malik	22	Female	1234567890	safia@xyz.com
987102	Varun	Ilango	24	Male	9987565423	varun@gmail.com
987103	Suja	Suresh	45	Female	7896541236	suja@hotmail.com
987104	Agatha	Elizabeth	30	Female	7845129635	agatha@gmail.com
987105	Zayed	Shah	40	Male	6789541235	zshah@hotmail.com
987106	Vijay	Kumar	60	Male	8996574123	vijayk@yahoo.com
987107	Meera	Das	35	Female	7845963259	meera@gmail.com

In this table version:

- C_ID remains the primary key.
- All attributes are fully functionally dependent on C_ID.

Normalize to 3NF

the Customer table is already in 3NF as there are no transitive dependencies.

Normalize to BCNF

Tables is in BCNF since the primary key (C_ID) uniquely identifies all other attributes in their respective tables.

Normalize to 4NF

Both tables are in 4NF as there are no multi-valued dependencies.

Non-Normalized Employee Table

E_ID	E_FNAME	E_LNAME	BDATE	E_AGE	E_SEX	E_TYPE	E_JDATE	E_SAL	E_PHNO	E_MAIL	E_ADD
1	Admin	-	1989-05-24	30	Female	Admin	2009-06-24	95000.00	9874563219, 9123456789	admin@pharmacia.com, admin_alt@xyz.com	Chennai
4567001	Varshini	Elangovan	1995-10-05	25	Female	Pharmacist	2017-11-12	25000.00	9967845123	evarsh@hotmail.com	Thiruvanmiyur
4567002	Anita	Shree	2000-10-03	20	Female	Pharmacist	2012-10-06	45000.00	8546123566	anita@gmail.com	Adyar
4567003	Harish	Raja	1998-02-01	22	Male	Pharmacist	2019-07-06	21000.00	7854123694	harishraja@live.com	T.Nagar
4567005	Amaya	Singh	1992-01-02	28	Female	Pharmacist	2017-05-16	32000.00	7894532165	amaya@gmail.com	Kottivakkam
4567006	Shoaib	Ahmed	1999-12-11	20	Male	Pharmacist	2018-09-05	28000.00	7896541234	shoaib@hotmail.com	Porur
4567009	Shayla	Hussain	1980-02-28	40	Female	Manager	2010-05-06	80000.00	7854123695	shaylah@gmail.com	Adyar
4567010	Daniel	James	1993-04-05	27	Male	Pharmacist	2016-01-05	30000.00	7896541235	daniels@gmail.com	Kodambakkam

Normalize to 1NF

Condition: Each cell must contain atomic values. We will separate multiple phone numbers and email addresses into individual rows.

1NF Table

E_ID	E_FNAME	E_LNAME	BDATE	E_AGE	E_SEX	E_TYPE	E_JDATE	E_SAL	E_PHN_O	E_MAIL	E_ADD
1	Admin	-	1989-05-24	30	Female	Admin	2009-06-24	95000.00	9874563219	admin@pharmacia.com	Chennai
1	Admin	-	1989-05-24	30	Female	Admin	2009-06-24	95000.00	9123456789	admin_alt@xyz.com	Chennai
4567001	Varshini	Elangovan	1995-10-05	25	Female	Pharmacist	2017-11-12	25000.00	9967845123	evarsh@hotmail.com	Thiruvanmiyur
4567002	Anita	Shree	2000-10-03	20	Female	Pharmacist	2012-10-06	45000.00	8546123566	anita@gmail.com	Adyar
4567003	Harish	Raja	1998-02-01	22	Male	Pharmacist	2019-07-06	21000.00	7854123694	harishraja@live.com	T.Nagar
4567005	Amaya	Singh	1992-01-02	28	Female	Pharmacist	2017-05-16	32000.00	7894532165	amaya@gmail.com	Kottivakkam
4567006	Shoaib	Ahmed	1999-12-11	20	Male	Pharmacist	2018-09-05	28000.00	7896541234	shoaib@hotmail.com	Porur
4567009	Shayla	Hussain	1980-02-28	40	Female	Manager	2010-05-06	80000.00	7854123695	shaylah@gmail.com	Adyar
4567010	Daniel	James	1993-04-05	27	Male	Pharmacist	2016-01-05	30000.00	7896541235	daniels@gmail.com	Kodambakkam

Normalize to 2NF

In 2NF, we ensure that all non-key attributes are fully dependent on the primary key.

2NF Table (Single Table)

E_ID	E_FNAME	E_LNAME	BDATE	E_AGE	E_SEX	E_TYP	E_JDA	E_SA	E_PHNO	E_MAIL	E_ADD
1	Admin	-	1989-05-24	30	Female	Admin	2009-06-24	95000.00	9874563219	admin@pharmacy.com	Chennai
1	Admin	-	1989-05-24	30	Female	Admin	2009-06-24	95000.00	9123456789	admin_alt@xyz.com	Chennai
4567001	Varshini	Elangovan	1995-10-05	25	Female	Pharmacist	2017-11-12	25000.00	9967845123	evarsh@hotmail.com	Thiruvanmiyur
4567002	Anita	Shree	2000-10-03	20	Female	Pharmacist	2012-10-06	45000.00	8546123566	anita@gmail.com	Adyar
4567003	Harish	Raja	1998-02-01	22	Male	Pharmacist	2019-07-06	21000.00	7854123694	harishraja@gmail.com	T.Nagar
4567005	Amaya	Singh	1992-01-02	28	Female	Pharmacist	2017-05-16	32000.00	7894532165	amaya@gmail.com	Kottivakkam
4567006	Shoaib	Ahmed	1999-12-11	20	Male	Pharmacist	2018-09-05	28000.00	7896541234	shoaib@hotmail.com	Porur
4567009	Shayla	Hussain	1980-02-28	40	Female	Manager	2010-05-06	80000.00	7854123695	shaylah@gmail.com	Adyar
4567010	Daniel	James	1993-04-05	27	Male	Pharmacist	2016-01-05	30000.00	7896541235	daniels@gmail.com	Kodambakkam

3NF

Since there's no transitive dependency (based on your preference to keep E_AGE), the table remains in 3NF as is, given that each non-key attribute is functionally dependent on the primary key (E_ID).

BCNF

For BCNF, we check that each functional dependency has a super key on the left-hand side. Since all dependencies are based on the primary key E_ID, this table is also in BCNF without any changes.

4NF

To satisfy 4NF, there should be no multi-valued dependencies unless they are dependent on the primary key. As there are no multi-valued dependencies in this table, it also meets 4NF criteria.

Non-normalized Purchase Table:

P_ID	SUP_ID	SUP_NAME	MED_ID	MED_NAME	CATEGORY	P_QTY	P_COST	PUR_DATE	MFG_DATE	EXP_DATE
1001	136	Rama Traders	123010	Concur 5 MG	Tablet	200	1500.50	2020-03-01	2019-05-05	2021-05-10
1002	123	ABC Pharma	123002	Panadol Cold & Flu	Tablet	1000	3000.00	2020-02-01	2018-06-01	2020-12-05
1003	145	Medix	123006	Benadryl 200 ML	Syrup	20	800.00	2020-04-22	2017-02-05	2020-07-01
1004	156	Pharma LTD	123004	Gelusil	Tablet	250	1000.00	2020-04-02	2020-05-06	2023-05-06
1005	123	ABC Pharma	123005	Cyclopam	Tablet	200	1200.00	2020-02-01	2019-08-02	2021-04-01
1006	162	Cureline	123010	Concur 5 MG	Tablet	500	1500.00	2019-04-22	2018-01-01	2020-05-02
1007	123	ABC Pharma	123001	Dolo 650 MG	Tablet	500	450.00	2020-01-02	2019-01-05	2022-03-06

1NF

To bring this table to **1NF**, we make sure each field contains atomic values. In this case, each attribute already holds atomic values, so it satisfies 1NF.

2NF

To achieve **2NF**, we need to eliminate partial dependencies. The primary key here would likely be a composite key of (P_ID, MED_ID), and to eliminate partial dependencies, we separate out data based on dependencies:

1. **Purchase Table (with P_ID as primary key):**

P_ID	SUP_ID	MED_ID	P_QTY	P_COST	PUR_DATE
1001	136	123010	200	1500.50	2020-03-01
1002	123	123002	1000	3000.00	2020-02-01
1003	145	123006	20	800.00	2020-04-22
1004	156	123004	250	1000.00	2020-04-02
1005	123	123005	200	1200.00	2020-02-01
1006	162	123010	500	1500.00	2019-04-22
1007	123	123001	500	450.00	2020-01-02

2. **Supplier Table (with SUP_ID as primary key):**

SUP_ID	SUP_NAME	SUP_ADD	SUP_PHNO	SUP_MAIL
123	ABC Pharma	Chennai	7854123698	abcpharma@gmail.com
136	Rama Traders	Mumbai	9847563214	ramatrd@gmail.com
145	Medix	Bangalore	7896541235	medix@outlook.com
156	Pharma LTD	Hyderabad	7896541236	pharma_ltd@gmail.com
162	Cureline	Delhi	7854123697	cureline@gmail.com

Medication Table (with MED_ID as primary key):

MED_ID	MED_NAME	MED_QTY	CATEGORY	MED_PRICE	LOCATION_RACK
123001	Dolo 650 MG	625	Tablet	1.00	rack 5
123002	Panadol Cold & Flu	90	Tablet	2.50	rack 6
123003	Livogen	25	Capsule	5.00	rack 3
123004	Gelusil	440	Tablet	1.25	rack 4
123005	Cyclopam	120	Tablet	6.00	rack 2
123006	Benadryl 200 ML	35	Syrup	50.00	rack 10
123007	Lopamide	15	Capsule	5.00	rack 7
123008	Vitamic C	90	Tablet	3.00	rack 8
123009	Omeprazole	35	Capsule	4.00	rack 3
123010	Concur 5 MG	600	Tablet	3.50	rack 9
123011	Augmentin 250 ML	115	Syrup	80.00	rack 7

3NF

In **3NF**, we need to remove transitive dependencies. Since there are no transitive dependencies here (i.e., no attribute is indirectly dependent on the primary key), the tables are already in **3NF**.

BCNF

For **BCNF**, we ensure that every determinant is a candidate key. All tables meet BCNF as each functional dependency has a super key on the left side.

4NF

Since there are no multi-valued dependencies, the tables meet **4NF** requirements.

Table: admin

ID	A_USERNAME	A_PASSWORD
1	admin	password

Table: admin**1NF (First Normal Form):**

- All columns contain atomic (indivisible) values.
- No repeating groups.

2NF (Second Normal Form):

- Table is already in 1NF.
- There is only one candidate key (A_USERNAME), and all other attributes (ID, A_PASSWORD) are fully functionally dependent on it.
- Hence, it satisfies 2NF.

3NF (Third Normal Form):

- Already in 2NF.
- No transitive dependencies exist; all non-key attributes (A_PASSWORD) are directly dependent on the primary key (A_USERNAME).

BCNF (Boyce-Codd Normal Form):

- No overlapping candidate keys.
- All functional dependencies are such that the left-hand side is a superkey.
- The table satisfies BCNF.

emlogin Table

E_ID	E_USERNAME	E_PASS
4567005	amaya	pass1
4567002	anita	pass2
4567010	daniel	pass3
4567003	harish	pass4
4567009	shayla	pass5
4567006	shoaib	pass6
4567001	varshini	pass7

Table: emlogin**1NF:**

- All columns contain atomic values.
- No repeating groups.

2NF:

- Table is in 1NF.
- The primary key (E_USERNAME) uniquely identifies each row.
- All other attributes (E_ID, E_PASS) are fully functionally dependent on the primary key.
- Satisfies 2NF.

3NF:

- Already in 2NF.
- No transitive dependencies exist.
- All non-key attributes are directly dependent on the primary key (E_USERNAME).

BCNF:

- Satisfies BCNF conditions with no overlapping candidate keys or non-trivial functional dependencies.

sales Table

SALE_ID	C_ID	S_DATE	S_TIME	TOTAL_AMT	E_ID
1	987101	2020-04-15	13:23:03	180.00	4567009
2	987106	2020-04-21	20:19:31	585.00	1
3	987103	2020-04-15	11:23:53	120.00	4567010
4	987104	2020-04-14	18:20:00	955.00	4567006
5	987102	2020-04-14	14:16:34	350.00	4567002
6	987101	2020-04-14	11:10:56	105.00	4567001
7	987105	2020-04-21	12:14:22	725.00	4567003

Table: sales**1NF:**

- All columns contain atomic values.
- No repeating groups.

2NF:

- Table is in 1NF.
- The primary key (SALE_ID) uniquely identifies each row.
- All non-key attributes (C_ID, S_DATE, S_TIME, TOTAL_AMT, E_ID) depend on the entire primary key.
- Satisfies 2NF.

3NF:

- Already in 2NF.
- No transitive dependencies exist.
- All non-key attributes are directly dependent on the primary key (SALE_ID).

BCNF:

- The table satisfies BCNF as there are no overlapping candidate keys or non-trivial functional dependencies.

sales_items Table

SALE_ID	MED_ID	SALE_QTY	TOT_PRICE
1	123001	100	100.00
2	123010	200	600.00
3	123005	10	60.00
4	123006	5	250.00
5	123001	5	25.00
6	123002	50	125.00
7	123010	100	300.00

Table: sales_items

1NF:

- All columns contain atomic values.
- No repeating groups.

2NF:

- Table is in 1NF.
- The composite primary key (SALE_ID, MED_ID) uniquely identifies each row.
- All non-key attributes (SALE_QTY, TOT_PRICE) depend on the entire primary key.
- Satisfies 2NF.

3NF:

- Already in 2NF.
- No transitive dependencies exist.
- All non-key attributes are directly dependent on the primary key (SALE_ID, MED_ID).

BCNF:

- The table satisfies BCNF as there are no overlapping candidate keys or non-trivial functional dependencies.

suppliers Table

SUP_ID	SUP_NAME	SUP_ADD	SUP_PHNO	SUP_MAIL
123	ABC Pharma	Chennai	7854123698	abcpharma@gmail.com
136	Rama Traders	Mumbai	9847563214	ramatrd@gmail.com
145	Medix	Bangalore	7896541235	medix@outlook.com
156	Pharma LTD	Hyderabad	7896541236	pharma_ltd@gmail.com
162	Cureline	Delhi	7854123697	cureline@gmail.com

Table: suppliers**1NF:**

- All columns contain atomic values.
- No repeating groups.

2NF:

- Table is in 1NF.
- The primary key (SUP_ID) uniquely identifies each row.
- All non-key attributes (SUP_NAME, SUP_ADD, SUP_PHNO, SUP_MAIL) depend on the entire primary key.
- Satisfies 2NF.

3NF:

- Already in 2NF.
- No transitive dependencies exist.
- All non-key attributes are directly dependent on the primary key (SUP_ID).

BCNF:

- The table satisfies BCNF as there are no overlapping candidate keys or non-trivial functional dependencies.

6. Implementation of concurrency control and recovery mechanisms

1. Concurrency Control Mechanisms

Locking Mechanisms & Transaction Isolation:

-- Procedure for updating total amount with concurrency control

DELIMITER //

CREATE PROCEDURE UpdateTotalAmount(IN sale_id INT)

BEGIN

DECLARE EXIT HANDLER FOR SQLEXCEPTION

BEGIN

-- Rollback in case of any error

ROLLBACK;

END;

-- Start transaction

START TRANSACTION;

-- Lock the sales and sales_items tables

LOCK TABLES sales WRITE, sales_items WRITE;

-- Calculate total amount for the sale

UPDATE sales

SET total_amt = (SELECT SUM(price) FROM sales_items WHERE sale_id = sale_id)

WHERE id = sale_id;

-- Commit the transaction

COMMIT;

-- Unlock tables

UNLOCK TABLES;

END//

DELIMITER ;

2. Recovery Mechanisms

Audit Log Table

The following audit_log table will track all changes (INSERT, UPDATE, DELETE) in the sales, purchase, and meds tables, including information about the type of operation, table affected, and a timestamp.

-- Audit Log Table

```
CREATE TABLE audit_log (
    log_id INT AUTO_INCREMENT PRIMARY KEY,
    table_name VARCHAR(50),
    operation_type ENUM('INSERT', 'UPDATE', 'DELETE'),
    operation_details TEXT,
    timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Triggers to Populate Audit Log

Here are triggers that record changes in the audit_log table for each operation on the sales, purchase, and meds tables.

-- Trigger for INSERT on sales table

```
DELIMITER //
CREATE TRIGGER sales_insert_audit AFTER INSERT ON sales
FOR EACH ROW
BEGIN
    INSERT INTO audit_log (table_name, operation_type, operation_details)
    VALUES ('sales', 'INSERT', CONCAT('Inserted record with ID: ', NEW.id));
END//
DELIMITER ;
```

-- Trigger for UPDATE on sales table

```
DELIMITER //
CREATE TRIGGER sales_update_audit AFTER UPDATE ON sales
FOR EACH ROW
BEGIN
    INSERT INTO audit_log (table_name, operation_type, operation_details)
```

```

VALUES ('sales', 'UPDATE', CONCAT('Updated record with ID: ', OLD.id));

END//


DELIMITER ;

-- Trigger for DELETE on sales table

DELIMITER //

CREATE TRIGGER sales_delete_audit AFTER DELETE ON sales
FOR EACH ROW
BEGIN

INSERT INTO audit_log (table_name, operation_type, operation_details)
VALUES ('sales', 'DELETE', CONCAT('Deleted record with ID: ', OLD.id));

END//


DELIMITER ;

```

Error Handling in TOTAL_AMT Procedure

To handle errors in the TOTAL_AMT procedure, I've included an error handler that logs errors and performs a rollback if any issue occurs during execution.

```

-- Procedure for calculating total amount with error handling

DELIMITER //

CREATE PROCEDURE TOTAL_AMT(IN sale_id INT)
BEGIN

DECLARE EXIT HANDLER FOR SQLEXCEPTION
BEGIN

-- Log the error in the audit log

INSERT INTO audit_log (table_name, operation_type, operation_details)
VALUES ('sales', 'ERROR', CONCAT('Error calculating total amount for sale ID: ', sale_id));

-- Rollback transaction

ROLLBACK;

END;

```

```
-- Start transaction

START TRANSACTION;

-- Lock tables for calculation

LOCK TABLES sales WRITE, sales_items WRITE;

-- Calculate and update the total amount

UPDATE sales

SET total_amt = (SELECT SUM(price) FROM sales_items WHERE sale_id = sale_id)

WHERE id = sale_id;

-- Commit the transaction

COMMIT;

-- Unlock tables

UNLOCK TABLES;

END//  
DELIMITER ;
```

7. Code for the project

adminmainpage.php:

```
<!DOCTYPE html>

<html>
<head>
<link rel="stylesheet" type="text/css" href="nav2.css">
<title>
Admin Dashboard
</title>
</head>
<body>
<div class="sidenav">
<h2 style="font-family:Arial; color:white; text-align:center;"> Medical Supply Management System </h2>
<p style="margin-top:-20px;color:white;line-height:1;font-size:12px;text-align:center">DBMS Project</p>
<a href="adminmainpage.php">Dashboard</a>
<button class="dropdown-btn">Inventory
<i class="down"></i>
</button>
<div class="dropdown-container">
<a href="inventory-add.php">Add New Medicine</a>
<a href="inventory-view.php">Manage Inventory</a>
</div>
<button class="dropdown-btn">Suppliers
<i class="down"></i>
</button>
<div class="dropdown-container">
<a href="supplier-add.php">Add New Supplier</a>
<a href="supplier-view.php">Manage Suppliers</a>
</div>
<button class="dropdown-btn">Stock Purchase
<i class="down"></i>
</button>
<div class="dropdown-container">
<a href="purchase-add.php">Add New Purchase</a>
<a href="purchase-view.php">Manage Purchases</a>
```

```

</div>

    <button class="dropdown-btn">Employees
        <i class="down"></i>
    </button>

    <div class="dropdown-container">
        <a href="employee-add.php">Add New Employee</a>
        <a href="employee-view.php">Manage Employees</a>
    </div>

    <button class="dropdown-btn">Customers
        <i class="down"></i>
    </button>

    <div class="dropdown-container">
        <a href="customer-add.php">Add New Customer</a>
        <a href="customer-view.php">Manage Customers</a>
    </div>

    <a href="sales-view.php">View Sales Invoice Details</a>
    <a href="salesitems-view.php">View Sold Products Details</a>
    <a href="pos1.php">Add New Sale</a>

    <button class="dropdown-btn">Reports
        <i class="down"></i>
    </button>

    <div class="dropdown-container">
        <a href="stockreport.php">Medicines - Low Stock</a>
        <a href="expiryreport.php">Medicines - Soon to Expire</a>
        <a href="salesreport.php">Transactions Reports</a>
    </div>
</div>

<div class="topnav">
    <a href="logout.php">Logout(Logged in as Admin)</a>
</div>

<center>
    <div class="head">
        <h2> ADMIN DASHBOARD </h2>
    </div>
</center>

<a href="pos1.php" title="Add New Sale">

```

```


</a>

<a href="inventory-view.php" title="View Inventory">

</a>

<a href="employee-view.php" title="View Employees">

</a>
<br>

<a href="salesreport.php" title="View Transactions">

</a>

<a href="stockreport.php" title="Low Stock Alert">

</a>
</body>
<script>
var dropdown = document.getElementsByClassName("dropdown-btn");
var i;
for (i = 0; i < dropdown.length; i++) {
  dropdown[i].addEventListener("click", function() {
    this.classList.toggle("active");
    var dropdownContent = this.nextElementSibling;
    if (dropdownContent.style.display === "block") {
      dropdownContent.style.display = "none";
    } else {
      dropdownContent.style.display = "block";
    }
  });
}
</script>
</html>

```

customer-add.php:

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="stylesheet" type="text/css" href="nav2.css">
<link rel="stylesheet" type="text/css" href="form4.css">
<title>
Customers
</title>
</head>
<body>
<div class="sidenav">
<h2 style="font-family:Arial; color:white; text-align:center;"> Medical Supply Management System </h2>
<p style="margin-top:-20px;color:white;line-height:1;font-size:12px;text-align:center">DBMS Project</p>
<a href="adminmainpage.php">Dashboard</a>
<button class="dropdown-btn">Inventory
<i class="down"></i>
</button>
<div class="dropdown-container">
<a href="inventory-add.php">Add New Medicine</a>
<a href="inventory-view.php">Manage Inventory</a>
</div>
<button class="dropdown-btn">Suppliers
<i class="down"></i>
</button>
<div class="dropdown-container">
<a href="supplier-add.php">Add New Supplier</a>
<a href="supplier-view.php">Manage Suppliers</a>
</div>
<button class="dropdown-btn">Stock Purchase
<i class="down"></i>
</button>
<div class="dropdown-container">
<a href="purchase-add.php">Add New Purchase</a>
<a href="purchase-view.php">Manage Purchases</a>
```

```

</div>

<button class="dropdown-btn">Employees
    <i class="down"></i>
</button>

<div class="dropdown-container">
    <a href="employee-add.php">Add New Employee</a>
    <a href="employee-view.php">Manage Employees</a>
</div>

<button class="dropdown-btn">Customers
    <i class="down"></i>
</button>

<div class="dropdown-container">
    <a href="customer-add.php">Add New Customer</a>
    <a href="customer-view.php">Manage Customers</a>
</div>

<a href="sales-view.php">View Sales Invoice Details</a>
<a href="salesitems-view.php">View Sold Products Details</a>
<a href="pos1.php">Add New Sale</a>

<button class="dropdown-btn">Reports
    <i class="down"></i>
</button>

<div class="dropdown-container">
    <a href="stockreport.php">Medicines - Low Stock</a>
    <a href="expiryreport.php">Medicines - Soon to Expire</a>
    <a href="salesreport.php">Transactions Reports</a>
</div>

</div>

<div class="topnav">
    <a href="logout.php">Logout</a>
</div>

<center>
    <div class="head">
        <h2> ADD CUSTOMER DETAILS</h2>
    </div>
</center>

<br><br><br><br><br><br><br><br>

<div class="one">

```

```

<div class="row">
    <form action="=$_SERVER['PHP_SELF']?" method="post">
        <div class="column">
            <p>
                <label for="cid">Customer ID:</label><br>
                <input type="number" name="cid">
            </p>
            <p>
                <label for="cname">First Name:</label><br>
                <input type="text" name="cname">
            </p>
            <p>
                <label for="clname">Last Name:</label><br>
                <input type="text" name="clname">
            </p>
            <p>
                <label for="age">Age:</label><br>
                <input type="number" name="age">
            </p>
            <p>
                <label for="sex">Sex: </label><br>
                <select id="sex" name="sex">
                    <option value="selected">Select</option>
                    <option>Female</option>
                    <option>Male</option>
                    <option>Others</option>
                </select>
            </p>
        </div>
        <div class="column">
            <p>
                <label for="phno">Phone Number:</label><br>
                <input type="number" name="phno">
            </p>

```

```

<p>
    <label for="emid">Email ID:</label><br>
    <input type="text" name="emid">
</p>
</div>

<input type="submit" name="add" value="Add Customer">
</form>
<br>
<?php
    include "config.php";
    if(isset($_POST['add']))
    {
        $id = mysqli_real_escape_string($conn, $_REQUEST['cid']);
        $fname = mysqli_real_escape_string($conn, $_REQUEST['cfname']);
        $lname = mysqli_real_escape_string($conn, $_REQUEST['clname']);
        $age = mysqli_real_escape_string($conn, $_REQUEST['age']);
        $sex = mysqli_real_escape_string($conn, $_REQUEST['sex']);
        $phno = mysqli_real_escape_string($conn, $_REQUEST['phno']);
        $mail = mysqli_real_escape_string($conn, $_REQUEST['emid']);

        $sql = "INSERT INTO customer VALUES ($id, '$fname', '$lname', $age, '$sex', '$phno', '$mail')";

        if(mysqli_query($conn, $sql)){
            echo "<p style='font-size:8;'>Customer successfully added!</p>";
        } else{
            echo "<p style='font-size:8; color:red;'>Error! Check details.</p>";
        }
    }
    $conn->close();
?>
</div>
</div>
</body>
<script>
    var dropdown = document.getElementsByClassName("dropdown-btn");
    var i;

```

```

for (i = 0; i < dropdown.length; i++) {
    dropdown[i].addEventListener("click", function() {
        this.classList.toggle("active");
        var dropdownContent = this.nextElementSibling;
        if (dropdownContent.style.display === "block") {
            dropdownContent.style.display = "none";
        } else {
            dropdownContent.style.display = "block";
        }
    });
}
</script>
</html>

```

customer-delete.php:

```

<?php
    include "config.php";
    $sql="DELETE FROM customer where c_id='$_GET[id]'";
    if ($conn->query($sql))
        header("location:customer-view.php");
    else
        echo "error";
?>

```

customer-update.php:

```

<!DOCTYPE html>
<html>

<head>
<link rel="stylesheet" type="text/css" href="nav2.css">
<link rel="stylesheet" type="text/css" href="form4.css">
<title>
    Customers
</title>
</head>

<body>

```

```
<div class="sidenav">

<h2 style="font-family:Arial; color:white; text-align:center;"> Medical Supply Management System </h2>
<p style="margin-top:-20px;color:white;line-height:1;font-size:12px;text-align:center">DBMS Project</p>

<a href="adminmainpage.php">Dashboard</a>

<button class="dropdown-btn">Inventory

<i class="down"></i>

</button>

<div class="dropdown-container">

    <a href="inventory-add.php">Add New Medicine</a>

    <a href="inventory-view.php">Manage Inventory</a>

</div>

<button class="dropdown-btn">Suppliers

<i class="down"></i>

</button>

<div class="dropdown-container">

    <a href="supplier-add.php">Add New Supplier</a>

    <a href="supplier-view.php">Manage Suppliers</a>

</div>

<button class="dropdown-btn">Stock Purchase

<i class="down"></i>

</button>

<div class="dropdown-container">

    <a href="purchase-add.php">Add New Purchase</a>

    <a href="purchase-view.php">Manage Purchases</a>

</div>

<button class="dropdown-btn">Employees

<i class="down"></i>

</button>

<div class="dropdown-container">

    <a href="employee-add.php">Add New Employee</a>

    <a href="employee-view.php">Manage Employees</a>

</div>

<button class="dropdown-btn">Customers

<i class="down"></i>

</button>

<div class="dropdown-container">

    <a href="customer-add.php">Add New Customer</a>
```

```

<a href="customer-view.php">Manage Customers</a>
</div>

<a href="sales-view.php">View Sales Invoice Details</a>
<a href="salesitems-view.php">View Sold Products Details</a>
<a href="pos1.php">Add New Sale</a>

<button class="dropdown-btn">Reports
<i class="down"></i>
</button>

<div class="dropdown-container">
<a href="stockreport.php">Medicines - Low Stock</a>
<a href="expiryreport.php">Medicines - Soon to Expire</a>
<a href="salesreport.php">Transactions Reports</a>
</div>
</div>

<div class="topnav">
<a href="logout.php">Logout</a>
</div>
<center>
<div class="head">
<h2> UPDATE CUSTOMER DETAILS</h2>
</div>
</center>
<div class="one">
<div class="row">
<?php
include "config.php";
if(isset($_GET['id']))
{
$id=$_GET['id'];
$qry1="SELECT * FROM customer WHERE c_id='$id'";
$result = $conn->query($qry1);
$row = $result -> fetch_row();
}
if( isset($_POST['update']))
{

```

```

$id = $_POST['cid'];

$fname = $_POST['cfname'];

$lname = $_POST['clname'];

$age = $_POST['age'];

$sex = $_POST['sex'];

$phno = $_POST['phno'];

$mail = $_POST['emid'];

$sql="UPDATE customer SET c_fname='".$fname',c_lname='".$lname',c_age='".$age',c_sex='".$sex',c_phno='".$phno',c_mail='".$mail' where
c_id='".$id"';

if ($conn->query($sql))

header("location:customer-view.php");

else

echo "<p style='font-size:8; color:red;'>Error! Unable to update.</p>";

}

?>

<form action="<?=$_SERVER['PHP_SELF']?>" method="post">

<div class="column">

<p>

<label for="cid">Customer ID:</label><br>

<input type="number" name="cid" value="<?php echo $row[0]; ?>" readonly>

</p>

<p>

<label for="cfname">First Name:</label><br>

<input type="text" name="cfname" value="<?php echo $row[1]; ?>">

</p>

<p>

<label for="clname">Last Name:</label><br>

<input type="text" name="clname" value="<?php echo $row[2]; ?>">

</p>

<p>

<label for="age">Age:</label><br>

<input type="number" name="age" value="<?php echo $row[3]; ?>">

</p>

<p>

<label for="sex">Sex:</label><br>

<input type="text" name="sex" value="<?php echo $row[4]; ?>">

</p>

```

```

</div>

<div class="column">
    <p>
        <label for="phno">Phone Number:</label><br>
        <input type="number" name="phno" value=<?php echo $row[5]; ?>>
    </p>
    <p>
        <label for="emid">Email ID:</label><br>
        <input type="text" name="emid" value=<?php echo $row[6]; ?>>
    </p>
</div>
<input type="submit" name="update" value="Update">
</form>
</div>
</div>
</body>
<script>
    var dropdown = document.getElementsByClassName("dropdown-btn");
    var i;
    for (i = 0; i < dropdown.length; i++) {
        dropdown[i].addEventListener("click", function() {
            this.classList.toggle("active");
            var dropdownContent = this.nextElementSibling;
            if (dropdownContent.style.display === "block") {
                dropdownContent.style.display = "none";
            } else {
                dropdownContent.style.display = "block";
            }
        });
    }
</script>
</html>

```

customer-view.php:

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css" href="nav2.css">
<link rel="stylesheet" type="text/css" href="table1.css">
<title>
Customers
</title>
</head>
<body>
<div class="sidenav">
<h2 style="font-family:Arial; color:white; text-align:center;"> Medical Supply Management System </h2>
<p style="margin-top:-20px;color:white;line-height:1;font-size:12px;text-align:center">DBMS Project</p>
<a href="adminmainpage.php">Dashboard</a>
<button class="dropdown-btn">Inventory
<i class="down"></i>
</button>
<div class="dropdown-container">
<a href="inventory-add.php">Add New Medicine</a>
<a href="inventory-view.php">Manage Inventory</a>
</div>
<button class="dropdown-btn">Suppliers
<i class="down"></i>
</button>
<div class="dropdown-container">
<a href="supplier-add.php">Add New Supplier</a>
<a href="supplier-view.php">Manage Suppliers</a>
</div>
<button class="dropdown-btn">Stock Purchase
<i class="down"></i>
</button>
<div class="dropdown-container">
<a href="purchase-add.php">Add New Purchase</a>
<a href="purchase-view.php">Manage Purchases</a>
```

```

</div>

<button class="dropdown-btn">Employees
<i class="down"></i>
</button>

<div class="dropdown-container">
    <a href="employee-add.php">Add New Employee</a>
    <a href="employee-view.php">Manage Employees</a>
</div>

<button class="dropdown-btn">Customers
<i class="down"></i>
</button>

<div class="dropdown-container">
    <a href="customer-add.php">Add New Customer</a>
    <a href="customer-view.php">Manage Customers</a>
</div>

<a href="sales-view.php">View Sales Invoice Details</a>
<a href="salesitems-view.php">View Sold Products Details</a>
<a href="pos1.php">Add New Sale</a>

<button class="dropdown-btn">Reports
<i class="down"></i>
</button>

<div class="dropdown-container">
    <a href="stockreport.php">Medicines - Low Stock</a>
    <a href="expiryreport.php">Medicines - Soon to Expire</a>
    <a href="salesreport.php">Transactions Reports</a>
</div>

</div>

<div class="topnav">
    <a href="logout.php">Logout</a>
</div>

<center>
    <div class="head">
        <h2> CUSTOMER LIST</h2>
    </div>
</center>

<table align="right" id="table1" style="margin-right:100px;">
    <tr>

```

```

<th>Customer ID</th>
<th>First Name</th>

<th>Last Name</th>
<th>Age</th>
<th>Sex</th>
<th>Phone Number</th>
<th>Email Address</th>
<th>Action</th>

</tr>
<?php
    include "config.php";
    $sql = "SELECT c_id,c_fname,c_lname,c_age,c_sex,c_phno,c_mail FROM customer";
    $result = $conn->query($sql);
    if ($result->num_rows > 0) {
        while($row = $result->fetch_assoc()) {
            echo "<tr>";
            echo "<td>" . $row["c_id"] . "</td>";
            echo "<td>" . $row["c_fname"] . "</td>";
            echo "<td>" . $row["c_lname"] . "</td>";
            echo "<td>" . $row["c_age"] . "</td>";
            echo "<td>" . $row["c_sex"] . "</td>";
            echo "<td>" . $row["c_phno"] . "</td>";
            echo "<td>" . $row["c_mail"] . "</td>";
            echo "<td align=center>";
            echo "<a class='button1 edit-btn' href=customer-update.php?id=".$row['c_id']."'>Edit</a>";
            echo "<a class='button1 del-btn' href=customer-delete.php?id=".$row['c_id']."'>Delete</a>";
            echo "</td>";
            echo "</tr>";
        }
        echo "</table>";
    }
    $conn->close();
?>
</body>
<script>
    var dropdown = document.getElementsByClassName("dropdown-btn");

```

```

var i;

for (i = 0; i < dropdown.length; i++) {
    dropdown[i].addEventListener("click", function() {

        this.classList.toggle("active");

        var dropdownContent = this.nextElementSibling;
        if (dropdownContent.style.display === "block") {
            dropdownContent.style.display = "none";
        } else {
            dropdownContent.style.display = "block";
        }
    });
}

};

</script>

</html>

```

employee-add.php:

```

<!DOCTYPE html>

<html>
<head>

<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="stylesheet" type="text/css" href="nav2.css">
<link rel="stylesheet" type="text/css" href="form4.css">

<title>
Employees
</title>

</head>
<body>

<div class="sidenav">

<h2 style="font-family:Arial; color:white; text-align:center;">Medical Supply Management System </h2>
<p style="margin-top:-20px;color:white;line-height:1;font-size:12px;text-align:center">DBMS Project</p>
<a href="adminmainpage.php">Dashboard</a>
<button class="dropdown-btn">Inventory
<i class="down"></i>
</button>
<div class="dropdown-container">
<a href="inventory-add.php">Add New Medicine</a>

```

```
<a href="inventory-view.php">Manage Inventory</a>
</div>

<button class="dropdown-btn">Suppliers
<i class="down"></i>
</button>

<div class="dropdown-container">
<a href="supplier-add.php">Add New Supplier</a>
<a href="supplier-view.php">Manage Suppliers</a>
</div>

<button class="dropdown-btn">Stock Purchase
<i class="down"></i>
</button>

<div class="dropdown-container">
<a href="purchase-add.php">Add New Purchase</a>
<a href="purchase-view.php">Manage Purchases</a>
</div>

<button class="dropdown-btn">Employees
<i class="down"></i>
</button>

<div class="dropdown-container">
<a href="employee-add.php">Add New Employee</a>
<a href="employee-view.php">Manage Employees</a>
</div>

<button class="dropdown-btn">Customers
<i class="down"></i>
</button>

<div class="dropdown-container">
<a href="customer-add.php">Add New Customer</a>
<a href="customer-view.php">Manage Customers</a>
</div>

<a href="sales-view.php">View Sales Invoice Details</a>
<a href="salesitems-view.php">View Sold Products Details</a>
<a href="pos1.php">Add New Sale</a>
<button class="dropdown-btn">Reports
<i class="down"></i>
</button>

<div class="dropdown-container">
```

```

<a href="stockreport.php">Medicines - Low Stock</a>
    <a href="expiryreport.php">Medicines - Soon to Expire</a>
    <a href="salesreport.php">Transactions Reports</a>
</div>

</div>
<div class="topnav">
    <a href="logout.php">Logout</a>
</div>
<center>
<div class="head">
<h2> ADD EMPLOYEE DETAILS</h2>
</div>
</center>
<br><br><br><br><br><br><br>
<div class="one row">
<?php
    include "config.php";
    if(isset($_POST['add']))
    {
        $id = mysqli_real_escape_string($conn, $_REQUEST['eid']);
        $fname = mysqli_real_escape_string($conn, $_REQUEST['efname']);
        $lname = mysqli_real_escape_string($conn, $_REQUEST['elname']);
        $bdate = mysqli_real_escape_string($conn, $_REQUEST['ebdate']);
        $age = mysqli_real_escape_string($conn, $_REQUEST['eage']);
        $sex = mysqli_real_escape_string($conn, $_REQUEST['esex']);
        $etype = mysqli_real_escape_string($conn, $_REQUEST['etype']);
        $jdate = mysqli_real_escape_string($conn, $_REQUEST['ejdate']);
        $sal = mysqli_real_escape_string($conn, $_REQUEST['esal']);
        $phno = mysqli_real_escape_string($conn, $_REQUEST['ephno']);
        $mail = mysqli_real_escape_string($conn, $_REQUEST['e_mail']);
        $add = mysqli_real_escape_string($conn, $_REQUEST['eadd']);
        $sql = "INSERT INTO employee VALUES ($id, '$fname','$lname','$bdate','$age','$sex','$etype','$jdate','$sal','$phno', '$mail','$add')";
        if(mysqli_query($conn, $sql)){
            echo "<p style='font-size:8;'>Employee successfully added!</p>";
        } else{
            echo "<p style='font-size:8; color:red;'>Error! Check details.</p>";
        }
    }

```

```

}

}

$conn->close();

?>

<form action=<?=$_SERVER['PHP_SELF']?>" method="post">

<div class="column">

<p>
    <label for="eid">Employee ID:</label><br>
    <input type="number" name="eid">
</p>

<p>
    <label for="efname">First Name:</label><br>
    <input type="text" name="efname">
</p>

<p>
    <label for="elname">Last Name:</label><br>
    <input type="text" name="elname">
</p>

<p>
    <label for="ebdate">Date of Birth:</label><br>
    <input type="date" name="ebdate">
</p>

<p>
    <label for="eage">Age:</label><br>
    <input type="number" name="eage">
</p>

<p>
    <label for="esex">Sex:</label><br>
    <select id="esex" name="esex">
        <option value="selected">Select</option>
        <option>Female</option>
        <option>Male</option>
        <option>Others</option>
    </select>
</p>

```

```

</div>

<div class="column">
    <p>
        <label for="etype">Employee Type:</label><br>
        <select id="etype" name="etype">
            <option value="selected">Select</option>
            <option>Pharmacist</option>
            <option>Manager</option>
        </select>
    </p>
    <p>
        <label for="ejdate">Date of Joining:</label><br>
        <input type="date" name="ejdate">
    </p>
    <p>
        <label for="esal">Salary:</label><br>
        <input type="number" step="0.01" name="esal">
    </p>
    <p>
        <label for="ephno">Phone Number:</label><br>
        <input type="number" name="ephno">
    </p>

    <p>
        <label for="e_mail">Email ID:</label><br>
        <input type="text" name="e_mail">
    </p>
    <p>
        <label for="eadd">Address:</label><br>
        <input type="text" name="eadd">
    </p>
</div>

<input type="submit" name="add" value="Add Employee">
</form>
<br>
</div>

```

```

</body>

<script>

var dropdown = document.getElementsByClassName("dropdown-btn");

var i;

for (i = 0; i < dropdown.length; i++) {

dropdown[i].addEventListener("click", function() {

this.classList.toggle("active");

var dropdownContent = this.nextElementSibling;

if (dropdownContent.style.display === "block") {

dropdownContent.style.display = "none";

} else {

dropdownContent.style.display = "block";

}

});

}

}

</script>

</html>

```

employee-delete.php:

```

<?php

include "config.php";

$sql="DELETE FROM employee where e_id='$_GET[id]';

if ($conn->query($sql))

header("location:employee-view.php");

else

echo "error";

?>

```

employee-update.php:

```

<?php

include "config.php";

if(isset($_GET['id']))

{

$id=$_GET['id'];

$qry1="SELECT * FROM employee WHERE e_id='$id'";
```

```

$result = $conn->query($qry1);

$row = $result -> fetch_row();

}

?>

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" type="text/css" href="nav2.css">

<link rel="stylesheet" type="text/css" href="form4.css">

<title>

Employees

</title>

</head>

<body>

<div class="sidenav">

<h2 style="font-family:Arial; color:white; text-align:center;"> Medical Supply Management System</h2>

<p style="margin-top:-20px;color:white;line-height:1;font-size:12px;text-align:center">DBMS Project</p>

<a href="adminmainpage.php">Dashboard</a>

<button class="dropdown-btn">Inventory

<i class="down"></i>

</button>

<div class="dropdown-container">

<a href="inventory-add.php">Add New Medicine</a>

<a href="inventory-view.php">Manage Inventory</a>

</div>

<button class="dropdown-btn">Suppliers

<i class="down"></i>

</button>

<div class="dropdown-container">

<a href="supplier-add.php">Add New Supplier</a>

<a href="supplier-view.php">Manage Suppliers</a>

</div>

<button class="dropdown-btn">Stock Purchase

<i class="down"></i>

</button>

```

```

<div class="dropdown-container">
    <a href="purchase-add.php">Add New Purchase</a>
    <a href="purchase-view.php">Manage Purchases</a>
</div>

<button class="dropdown-btn">Employees
    <i class="down"></i>
</button>
<div class="dropdown-container">
    <a href="employee-add.php">Add New Employee</a>
    <a href="employee-view.php">Manage Employees</a>
</div>

<button class="dropdown-btn">Customers
    <i class="down"></i>
</button>
<div class="dropdown-container">
    <a href="customer-add.php">Add New Customer</a>
    <a href="customer-view.php">Manage Customers</a>
</div>

<a href="sales-view.php">View Sales Invoice Details</a>
<a href="salesitems-view.php">View Sold Products Details</a>
<a href="pos1.php">Add New Sale</a>
<button class="dropdown-btn">Reports
    <i class="down"></i>
</button>
<div class="dropdown-container">
    <a href="stockreport.php">Medicines - Low Stock</a>
    <a href="expiryreport.php">Medicines - Soon to Expire</a>
    <a href="salesreport.php">Transactions Reports</a>
</div>
</div>

<div class="topnav">
    <a href="logout.php">Logout</a>
</div>
<center>
    <div class="head">

```

```

<h2> UPDATE EMPLOYEE DETAILS</h2>
</div>
</center>

<div class="one">

<div class="row">

<?php
if( isset($_POST['update']))


{
    $id = mysqli_real_escape_string($conn, $_REQUEST['eid']);

    $fname = mysqli_real_escape_string($conn, $_REQUEST['efname']);
    $lname = mysqli_real_escape_string($conn, $_REQUEST['elname']);
    $bdate = mysqli_real_escape_string($conn, $_REQUEST['ebdate']);

    $age = mysqli_real_escape_string($conn, $_REQUEST['eage']);
    $sex = mysqli_real_escape_string($conn, $_REQUEST['esex']);
    $etyp = mysqli_real_escape_string($conn, $_REQUEST['etype']);
    $jdate = mysqli_real_escape_string($conn, $_REQUEST['ejdate']);
    $sal = mysqli_real_escape_string($conn, $_REQUEST['esal']);
    $phno = mysqli_real_escape_string($conn, $_REQUEST['ephno']);
    $mail = mysqli_real_escape_string($conn, $_REQUEST['e_mail']);
    $add = mysqli_real_escape_string($conn, $_REQUEST['eadd']);

    $sql="UPDATE employee
        SET e_fname='".$fname."',e_lname='".$lname."',bdate='".$bdate."',e_age='".$age."',e_sex='".$sex',
        e_type='".$etyp."',e_jdate='".$jdate."',e_sal='".$sal."',e_phno='".$phno."',e_mail='".$mail."',e_add='".$add"' where e_id='".$id."'";
    if ($conn->query($sql))
        header("location:employee-view.php");
    else
        echo "<p style='font-size:8; color:red;'>Error! Unable to update.</p>";
}
?>

<form action=<?=$_SERVER['PHP_SELF']?>" method="post">
    <div class="column">
        <p>
            <label for="eid">Employee ID:</label><br>

```

```

<input type="number" name="eid" value="<?php echo $row[0]; ?>" readonly>
</p>
<p>
    <label for="efname">First Name:</label><br>

    <input type="text" name="efname" value="<?php echo $row[1]; ?>">
</p>
<p>
    <label for="elname">Last Name:</label><br>
    <input type="text" name="elname" value="<?php echo $row[2]; ?>">
</p>

<p>
    <label for="ebdate">Date of Birth:</label><br>
    <input type="date" name="ebdate" value="<?php echo $row[3]; ?>">
</p>
<p>
    <label for="eage">Age:</label><br>
    <input type="number" name="eage" value="<?php echo $row[4]; ?>">
</p>
<p>
    <label for="esex">Sex:</label><br>
    <input type="text" name="esex" value="<?php echo $row[5]; ?>">
</p>
</div>
<div class="column">
<p>
    <label for="etype">Employee Type:</label><br>
    <input type="text" name="etype" value="<?php echo $row[6]; ?>">
</p>
<p>
    <label for="ejdate">Date of Joining:</label><br>
    <input type="date" name="ejdate" value="<?php echo $row[7]; ?>">
</p>
<p>
    <label for="esal">Salary:</label><br>
    <input type="number" step="0.01" name="esal" value="<?php echo $row[8]; ?>">

```

```

</p>

<p>
    <label for="ephno">Phone Number:</label><br>
    <input type="number" name="ephno" value="<?php echo $row[9]; ?>">
</p>

<p>

<label for="e_mail">Email ID:</label><br>
    <input type="text" name="e_mail" value="<?php echo $row[10]; ?>">
</p>

<p>

<label for="eadd">Address:</label><br>
    <input type="text" name="eadd" value="<?php echo $row[11]; ?>">
</p>

</div>

<input type="submit" name="update" value="Update">
</form>
</div>
</div>

</body>

<script>

var dropdown = document.getElementsByClassName("dropdown-btn");
var i;

for (i = 0; i < dropdown.length; i++) {
    dropdown[i].addEventListener("click", function() {
        this.classList.toggle("active");
        var dropdownContent = this.nextElementSibling;
        if (dropdownContent.style.display === "block") {

```

```

dropdownContent.style.display = "none";

} else {
    dropdownContent.style.display = "block";
}
});

}

}

</script>
</html>

```

employee-view.php:

```

<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="stylesheet" type="text/css" href="nav2.css">
<link rel="stylesheet" type="text/css" href="table1.css">
<title>
Employees
</title>
</head>
<body>

<div class="sidenav">
<h2 style="font-family:Arial; color:white; text-align:center;"> Medical Supply Management System </h2>
<p style="margin-top:-20px;color:white;line-height:1;font-size:12px;text-align:center">DBMS Project</p>
<a href="adminmainpage.php">Dashboard</a>
<button class="dropdown-btn">Inventory
<i class="down"></i>
</button>
<div class="dropdown-container">
<a href="inventory-add.php">Add New Medicine</a>
<a href="inventory-view.php">Manage Inventory</a>
</div>
<button class="dropdown-btn">Suppliers
<i class="down"></i>
</button>
<div class="dropdown-container">
<a href="supplier-add.php">Add New Supplier</a>
<a href="supplier-view.php">Manage Suppliers</a>

```

```

</div>

<button class="dropdown-btn">Stock Purchase
<i class="down"></i>
</button>

<div class="dropdown-container">
    <a href="purchase-add.php">Add New Purchase</a>
    <a href="purchase-view.php">Manage Purchases</a>
</div>

<button class="dropdown-btn">Employees
<i class="down"></i>
</button>

<div class="dropdown-container">
    <a href="employee-add.php">Add New Employee</a>
    <a href="employee-view.php">Manage Employees</a>
</div>

<button class="dropdown-btn">Customers
<i class="down"></i>
</button>

<div class="dropdown-container">
    <a href="customer-add.php">Add New Customer</a>
    <a href="customer-view.php">Manage Customers</a>
</div>

<a href="sales-view.php">View Sales Invoice Details</a>
<a href="salesitems-view.php">View Sold Products Details</a>
<a href="pos1.php">Add New Sale</a>

<button class="dropdown-btn">Reports
<i class="down"></i>
</button>

<div class="dropdown-container">
    <a href="stockreport.php">Medicines - Low Stock</a>
    <a href="expiryreport.php">Medicines - Soon to Expire</a>
    <a href="salesreport.php">Transactions Reports</a>
</div>

</div>
</div>

<div class="topnav">
    <a href="logout.php">Logout</a>
</div>

```

```

<center>
<div class="head">
<h2> EMPLOYEE LIST</h2>
</div>
</center>

<table align="right" id="table1" style="margin-right:20px;">
<tr>
<th>Employee ID</th>

<th>First Name</th>
<th>Last Name</th>
<th>Date of Birth</th>
<th>Age</th>
<th>Sex</th>
<th>Employee Type</th>
<th>Date of Joining</th>
<th>Salary</th>
<th>Phone Number</th>
<th>Email Address</th>
<th>Home Address</th>
<th>Action</th>
</tr>
<?php
include "config.php";
$sql = "SELECT e_id, e_fname, e_lname, bdate, e_age, e_sex, e_type, e_jdate, e_sal, e_phno, e_mail, e_add FROM employee where
e_id<>1";
$result = $conn->query($sql);
if ($result->num_rows > 0) {
while($row = $result->fetch_assoc()) {
echo "<tr>";
echo "<td>" . $row["e_id"] . "</td>";
echo "<td>" . $row["e_fname"] . "</td>";
echo "<td>" . $row["e_lname"] . "</td>";
echo "<td>" . $row["bdate"] . "</td>";
echo "<td>" . $row["e_age"] . "</td>";
echo "<td>" . $row["e_sex"] . "</td>";
}
}

```

```

echo "<td>" . $row["e_type"]. "</td>";

echo "<td>" . $row["e_jdate"]. "</td>";

echo "<td>" . $row["e_sal"]. "</td>";

echo "<td>" . $row["e_phno"]. "</td>";

echo "<td>" . $row["e_mail"]. "</td>";

echo "<td>" . $row["e_add"]. "</td>";

echo "<td align=center>";

echo "<a class='button1 edit-btn' href=employee-update.php?id=". $row['e_id']."'>Edit</a>";

echo "<a onclick='return confirm('Are you sure to delete?');' class='button1 del-btn' href=employee-  
delete.php?id=". $row['e_id']."'>Delete</a>";

echo "</td>";

echo "</tr>";

}

echo "</table>";

}

$conn->close();

?>

</body>

<script>

var dropdown = document.getElementsByClassName("dropdown-btn");

var i;

for (i = 0; i < dropdown.length; i++) {

dropdown[i].addEventListener("click", function() {

this.classList.toggle("active");

var dropdownContent = this.nextElementSibling;

if (dropdownContent.style.display === "block") {

dropdownContent.style.display = "none";

} else {

dropdownContent.style.display = "block";

}

});

}

</script>

</html>

```

expiryreport.php:

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="stylesheet" type="text/css" href="nav2.css">
<link rel="stylesheet" type="text/css" href="table1.css">
<title>
Reports
</title>
</head>

<body>
<div class="sidenav">
<h2 style="font-family:Arial; color:white; text-align:center;"> Medical Supply Management System </h2>
<p style="margin-top:-20px;color:white;line-height:1;font-size:12px;text-align:center">DBMS Project</p>
<a href="adminmainpage.php">Dashboard</a>
<button class="dropdown-btn">Inventory
<i class="down"></i>
</button>
<div class="dropdown-container">
<a href="inventory-add.php">Add New Medicine</a>
<a href="inventory-view.php">Manage Inventory</a>
</div>
<button class="dropdown-btn">Suppliers
<i class="down"></i>
</button>
<div class="dropdown-container">
<a href="supplier-add.php">Add New Supplier</a>
<a href="supplier-view.php">Manage Suppliers</a>
</div>
<button class="dropdown-btn">Stock Purchase
<i class="down"></i>
</button>
<div class="dropdown-container">
<a href="purchase-add.php">Add New Purchase</a>
<a href="purchase-view.php">Manage Purchases</a>
```

```

</div>

<button class="dropdown-btn">Employees
    <i class="down"></i>
</button>

<div class="dropdown-container">
    <a href="employee-add.php">Add New Employee</a>
    <a href="employee-view.php">Manage Employees</a>
</div>

<button class="dropdown-btn">Customers
    <i class="down"></i>
</button>

<div class="dropdown-container">
    <a href="customer-add.php">Add New Customer</a>
    <a href="customer-view.php">Manage Customers</a>
</div>

<a href="sales-view.php">View Sales Invoice Details</a>
<a href="salesitems-view.php">View Sold Products Details</a>
<a href="pos1.php">Add New Sale</a>

<button class="dropdown-btn">Reports
    <i class="down"></i>
</button>

<div class="dropdown-container">
    <a href="stockreport.php">Medicines - Low Stock</a>
    <a href="expiryreport.php">Medicines - Soon to Expire</a>
    <a href="salesreport.php">Transactions Reports</a>
</div>

</div>

<div class="topnav">
    <a href="logout.php">Logout</a>
</div>

<center>
    <div class="head">

        <h2> STOCK EXPIRING WITHIN 6 MONTHS</h2>
        </div>
</center>

```

```

<table align="right" id="table1" style="margin-right:100px;">
<tr>
<th>Purchase ID</th>
<th>Supplier ID</th>
<th>Medicine ID</th>
<th>Quantity</th>
<th>Cost of Purchase</th>
<th>Date of Purchase</th>
<th>Manufacturing Date</th>
<th>Expiry Date</th>
</tr>
<?php
include "config.php";
$result=mysqli_query($conn,"CALL `EXPIRY`()");

if ($result->num_rows > 0) {
while($row = $result->fetch_assoc()) {
echo "<tr>";
echo "<td>" . $row["p_id"]. "</td>";
echo "<td>" . $row["sup_id"]. "</td>";
echo "<td>" . $row["med_id"]. "</td>";
echo "<td>" . $row["p_qty"]. "</td>";
echo "<td>" . $row["p_cost"]. "</td>";
echo "<td>" . $row["pur_date"]. "</td>";
echo "<td>" . $row["mfg_date"] . "</td>";
echo "<td style='color:red;'>" . $row["exp_date"]. "</td>";
echo "</tr>";
}
echo "</table>";
}
$conn->close();
?>
</body>
<script>

var dropdown = document.getElementsByClassName("dropdown-btn");
var i;


```

```
for (i = 0; i < dropdown.length; i++) {  
    dropdown[i].addEventListener("click", function() {  
        this.classList.toggle("active");  
        var dropdownContent = this.nextElementSibling;  
        if (dropdownContent.style.display === "block") {  
            dropdownContent.style.display = "none";  
        } else {  
            dropdownContent.style.display = "block";  
        }  
    });  
}  
</script>  
</html>
```

config.php:

```
<?php  
$conn = mysqli_connect("localhost", "root", "", "medical");  
if ($conn->connect_error) {  
    die("Connection failed: " . $conn->connect_error);  
}  
?>
```

8. Result and Discussion (Screen shots of the implementation with front end.)

8.1. Admin Login:



Fig 8.1 Displays the Login Page of Admin for Accessing MSM

8.2. Pharmacist Login:



Fig 8.2 Displays the Login Page of Pharmacist for Accessing MSM

8.3. Admin Dashboard:



Fig 8.3 Displays the Admission Dashboard

8.4. Add Sale:

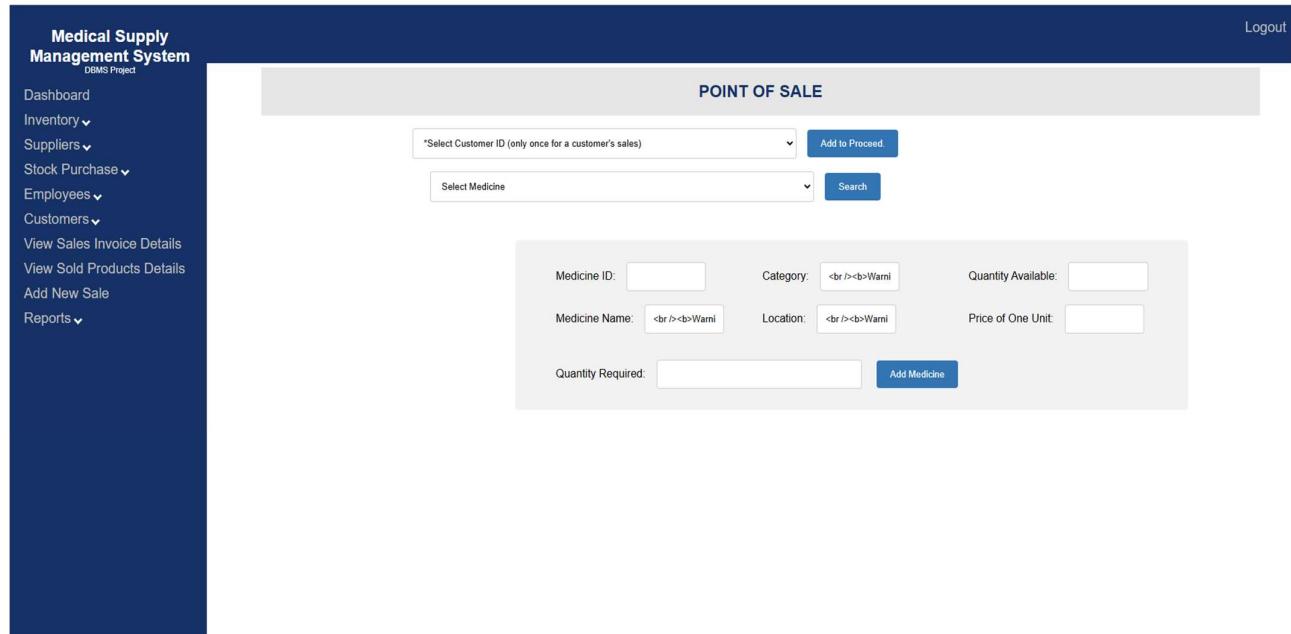


Fig 8.4 Displays the POS Page for Making a Sales on Medicine

8.5. Add New Medicine:

The screenshot shows the 'Add New Medicine Details' page. The left sidebar lists various management functions: Dashboard, Inventory, Suppliers, Stock Purchase, Employees, Customers, View Sales Invoice Details, View Sold Products Details, Add New Sale, and Reports. The main content area is titled 'ADD MEDICINE DETAILS'. It includes input fields for Medicine ID, Price, Medicine Name, Location, Quantity, Category (with a dropdown menu showing 'Tablet'), and an 'Add Medicine' button.

Fig 8.5 Displays the Add New Medicine Page for adding a new medicine to the database

8.6. Medicine Inventory:

The screenshot shows the 'Medicine Inventory' page. The left sidebar lists various management functions. The main content area is titled 'MEDICINE INVENTORY' and displays a table of medicine details. The table columns are: Medicine ID, Medicine Name, Quantity Available, Category, Price, Location in Store, and Action (with 'Edit' and 'Delete' buttons). The table rows list ten different medicines with their respective details.

Medicine ID	Medicine Name	Quantity Available	Category	Price	Location in Store	Action
123001	Dolo 650 MG	653	Tablet	1.00	rack 5	<button>Edit</button> <button>Delete</button>
123002	Panadol Cold & Flu	86	Tablet	2.50	rack 6	<button>Edit</button> <button>Delete</button>
123003	Livogen	25	Capsule	5.00	rack 3	<button>Edit</button> <button>Delete</button>
123004	Gelusil	440	Tablet	1.25	rack 4	<button>Edit</button> <button>Delete</button>
123005	Cyclopam	120	Tablet	6.00	rack 2	<button>Edit</button> <button>Delete</button>
123006	Benadryl 200 ML	35	Syrup	50.00	rack 10	<button>Edit</button> <button>Delete</button>
123007	Lopamide	15	Capsule	5.00	rack 7	<button>Edit</button> <button>Delete</button>
123008	Vitamic C	90	Tablet	3.00	rack 8	<button>Edit</button> <button>Delete</button>
123009	Omeprazole	35	Capsule	4.00	rack 3	<button>Edit</button> <button>Delete</button>
123010	Concur 5 MG	600	Tablet	3.50	rack 9	<button>Edit</button> <button>Delete</button>

Fig 8.6 Displays the Medicine Inventory which can be edited or deleted from database

8.7. Add Supplier Details:

The screenshot shows the 'Add Supplier Details' page. At the top left is the system title 'Medical Supply Management System' with a subtitle 'DBMS Project'. On the top right is a 'Logout' link. The left side features a dark sidebar with various navigation options: Dashboard, Inventory, Suppliers, Stock Purchase, Employees, Customers, View Sales Invoice Details, View Sold Products Details, Add New Sale, and Reports. The main content area is titled 'ADD SUPPLIER DETAILS'. It contains four input fields: 'Supplier ID' (with a placeholder box), 'Supplier Company Name' (with a placeholder box), 'Address' (with a placeholder box), and two text boxes for 'Phone Number' and 'Email Address'. A blue rectangular button labeled 'Add Supplier' is positioned at the bottom right of the input area.

Fig 8.7 Displays the Add Supplier details page for adding the supplier details.

8.8. Suppliers List:

The screenshot shows the 'SUPPLIERS LIST' page. At the top left is the system title 'Medical Supply Management System' with a subtitle 'DBMS Project'. On the top right is a 'Logout' link. The left side features a dark sidebar with various navigation options: Dashboard, Inventory, Suppliers, Stock Purchase, Employees, Customers, View Sales Invoice Details, View Sold Products Details, Add New Sale, and Reports. The main content area is titled 'SUPPLIERS LIST'. It displays a table with six columns: 'Supplier ID', 'Company Name', 'Address', 'Phone Number', 'Email Address', and 'Action'. The table contains five rows of data, each with a unique Supplier ID and company name, along with their respective address, phone number, and email address. To the right of each row is a blue 'Edit' button and a red 'Delete' button.

Supplier ID	Company Name	Address	Phone Number	Email Address	Action
123	XYZ Pharmaceuticals	Chennai, Tamil Nadu	8745632145	xyz@xyzpharma.com	<button>Edit</button> <button>Delete</button>
136	ABC PharmaSupply	Trichy	7894561235	abc@pharmsupp.com	<button>Edit</button> <button>Delete</button>
145	Daily Pharma Ltd	Hyderabad	7854699321	daily@dpharma.com	<button>Edit</button> <button>Delete</button>
156	MedAll	Chennai	9874585236	mainid@medall.com	<button>Edit</button> <button>Delete</button>
162	MedHead Pharmaceuticals	Trichy	7894561335	abc@pharmsupp.com	<button>Edit</button> <button>Delete</button>

Fig 8.8 Displays the Suppliers List

8.9. Add Purchase Details:

The screenshot shows the 'ADD PURCHASE DETAILS' form. It includes fields for Purchase ID, Purchase Cost, Supplier ID, Date of Purchase, Medicine ID, Manufacturing Date, Purchase Quantity, Expiry Date, and an 'Add Purchase' button.

Purchase ID:	Purchase Cost:
<input type="text"/>	<input type="text"/>

Supplier ID:	Date of Purchase:
<input type="text"/>	<input type="text"/>

Medicine ID:	Manufacturing Date:
<input type="text"/>	<input type="text"/>

Purchase Quantity:	Expiry Date:
<input type="text"/>	<input type="text"/>

Fig 8.9 Displays the Add Purchase details page for adding the purchase details.

8.10. Stock Purchase List:

The screenshot shows the 'STOCK PURCHASE LIST' page with a table of purchase details and edit/delete buttons.

Purchase ID	Supplier ID	Medicine ID	Medicine Name	Quantity	Cost of Purchase	Date of Purchase	Manufacturing Date	Expiry Date	Action
1001	136	123010	Concur 5 MG	200	1500.50	2020-03-01	2019-05-05	2021-05-10	Edit Delete
1002	123	123002	Panadol Cold & Flu	1000	3000.00	2020-02-01	2018-06-01	2020-12-05	Edit Delete
1003	145	123006	Benadryl 200 ML	20	800.00	2020-04-22	2017-02-05	2020-07-01	Edit Delete
1004	156	123004	Gelusil	250	1000.00	2020-04-02	2020-05-06	2023-05-06	Edit Delete
1005	123	123005	Cyclopam	200	1200.00	2020-02-01	2019-08-02	2021-04-01	Edit Delete
1006	162	123010	Concur 5 MG	500	1500.00	2019-04-22	2018-01-01	2020-05-02	Edit Delete
1007	123	123001	Dolo 650 MG	500	450.00	2020-01-02	2019-01-05	2022-03-06	Edit Delete

Fig 8.10 Displays the Stock Purchase List.

8.11. Add Employee Details:

ADD EMPLOYEE DETAILS

Employee ID:	Employee Type:
<input type="text"/>	<input type="text"/>
First Name:	Date of Joining:
<input type="text"/>	<input type="text"/>
Last Name:	Salary:
<input type="text"/>	<input type="text"/>
Date of Birth:	Phone Number:
<input type="text"/>	<input type="text"/>
Age:	Email ID:
<input type="text"/>	<input type="text"/>
Sex:	Address:
<input type="text"/>	<input type="text"/>
Add Employee	

Fig 8.11 Displays the Add Employee details page for adding the employee details.

8.12. Employee Details:

EMPLOYEE LIST

Employee ID	First Name	Last Name	Date of Birth	Age	Sex	Employee Type	Date of Joining	Salary	Phone Number	Email Address	Home Address	Action
4567001	Varshini	Elangovan	1995-10-05	25	Female	Pharmacist	2017-11-12	25000.00	9967845123	evash@hotmail.com	Thiruvanmiyur	Edit Delete
4567002	Anita	Shree	2000-10-03	20	Female	Pharmacist	2012-10-06	45000.00	8546123566	anita@gmail.com	Adyar	Edit Delete
4567003	Harish	Raja	1998-02-01	22	Male	Pharmacist	2019-07-06	21000.00	7854123694	harishraja@live.com	T Nagar	Edit Delete
4567005	Amaya	Singh	1992-01-02	28	Female	Pharmacist	2017-05-16	32000.00	7894532165	amaya@gmail.com	Kottivakkam	Edit Delete
4567006	Shoaib	Ahmed	1999-12-11	20	Male	Pharmacist	2018-09-05	28000.00	7896541234	shoib@hotmail.com	Porur	Edit Delete
4567009	Shayla	Hussain	1980-02-28	40	Female	Manager	2010-05-06	80000.00	7854123695	shaylah@gmail.com	Adyar	Edit Delete
4567010	Daniel	James	1993-04-05	27	Male	Pharmacist	2016-01-05	30000.00	7896541235	daniels@gmail.com	Kodambakkam	Edit Delete

Fig 8.12 Displays the Employee Details in List which can be edited or removed from database.

8.13. Add Customer Details:

The screenshot shows the 'Add Customer Details' page. The left sidebar lists various system functions: Dashboard, Inventory, Suppliers, Stock Purchase, Employees, Customers, View Sales Invoice Details, View Sold Products Details, Add New Sale, and Reports. The main content area is titled 'ADD CUSTOMER DETAILS'. It contains several input fields: 'Customer ID' (text input), 'Phone Number' (text input), 'First Name' (text input), 'Email ID' (text input), 'Last Name' (text input), 'Age' (text input), and 'Sex' (dropdown menu with 'Select' option). A blue 'Add Customer' button is located at the bottom right of the form.

Fig 8.13 Displays the Add Customer details page for adding the customer details.

8.14. Customer List:

The screenshot shows the 'CUSTOMER LIST' page. The left sidebar lists various system functions: Dashboard, Inventory, Suppliers, Stock Purchase, Employees, Customers, View Sales Invoice Details, View Sold Products Details, Add New Sale, and Reports. The main content area is titled 'CUSTOMER LIST' and features a table with the following data:

Customer ID	First Name	Last Name	Age	Sex	Phone Number	Email Address	Action
987101	Safia	Malik	22	Female	9632587415	safia@gmail.com	<button>Edit</button> <button>Delete</button>
987102	Varun	Ilango	24	Male	9887565423	varun@gmail.com	<button>Edit</button> <button>Delete</button>
987103	Suja	Suresh	45	Female	7896541236	suja@hotmail.com	<button>Edit</button> <button>Delete</button>
987104	Agatha	Elizabeth	30	Female	7845129635	agatha@gmail.com	<button>Edit</button> <button>Delete</button>
987105	Zayed	Shah	40	Male	6789541235	zshah@hotmail.com	<button>Edit</button> <button>Delete</button>
987106	Vijay	Kumar	60	Male	8996574123	vijayk@yahoo.com	<button>Edit</button> <button>Delete</button>
987107	Meera	Das	35	Female	7845963259	meera@gmail.com	<button>Edit</button> <button>Delete</button>

Fig 8.14 Displays the Customer Details in list which can be edited or removed from database.

8.15. Sales Invoice Details:

The screenshot shows a dark blue header with the title "Medical Supply Management System" and a "Logout" link. On the left, a sidebar lists navigation options: Dashboard, Inventory, Suppliers, Stock Purchase, Employees, Customers, View Sales Invoice Details, View Sold Products Details, Add New Sale, and Reports. The main content area is titled "SALES INVOICE DETAILS" and contains a table with 21 rows of sales data.

Sale ID	Customer ID	Date and Time	Sale Amount	Employee ID
1	987101	2024-10-02 00:16:00	180.00	4567009
2	987106	2020-04-21 20:19:31	585.00	1
3	987103	2020-04-15 11:23:53	120.00	4567010
4	987104	2020-04-14 18:20:00	955.00	4567006
5	987103	2020-04-21 15:24:43	45.00	1
6	987102	2020-03-11 10:24:43	140.00	4567001
7	987105	2020-04-24 00:25:54	350.00	1
8	987104	2020-04-24 00:47:47	35.00	4567001
12	987103	2020-04-24 19:33:16	60.00	1
13	987104	2020-04-24 21:15:56	62.50	4567001
15	987107	2020-12-04 18:39:46	420.00	1
16	987106	2020-12-04 18:52:21	30.00	1
17	987103	2020-12-04 19:35:56	57.50	1
18	987105	2020-12-04 19:36:56	160.00	4567001
20	987103	2020-12-04 22:53:18	150.00	4567001
21	987105	2024-11-09 20:49:57	5.00	1

Fig 8.15 Displays the Sales Invoice Details in Table.

8.16. List of Products Sold:

The screenshot shows a dark blue header with the title "Medical Supply Management System" and a "Logout" link. On the left, a sidebar lists navigation options: Dashboard, Inventory, Suppliers, Stock Purchase, Employees, Customers, View Sales Invoice Details, View Sold Products Details, Add New Sale, and Reports. The main content area is titled "LIST OF PRODUCTS SOLD" and contains a table with 21 rows of product sale data.

Sale ID	Medicine ID	Medicine Name	Quantity Sold	Total Price
2	123003	Livogen	75	225.00
2	123005	Cyclopam	60	360.00
3	123008	Vitamin C	40	120.00
4	123010	Concur 5 MG	250	875.00
4	123011	Augmentin 250 ML	1	80.00
5	123001	Dolo 650 MG	45	45.00
6	123006	Benadryl 200 ML	2	100.00
6	123009	Omeprazole	10	40.00
7	123001	Dolo 650 MG	100	100.00
7	123003	Livogen	50	250.00
8	123001	Dolo 650 MG	10	10.00
8	123002	Paracet Cold & Flu	10	25.00
12	123005	Cyclopam	10	60.00
13	123002	Paracet Cold & Flu	25	62.50
15	123005	Cyclopam	45	270.00
15	123006	Benadryl 200 ML	3	150.00
16	123008	Vitamin C	10	30.00
17	123004	Gelusil	10	12.50
17	123007	Lopamide	5	25.00
17	123009	Omeprazole	5	20.00
18	123011	Augmentin 250 ML	2	160.00
20	123002	Paracet Cold & Flu	2	5.00
20	123005	Cyclopam	25	150.00
21	123002	Paracet Cold & Flu	2	5.00

Fig 8.16 Displays the List of Products Sold.

8.17. Medicals Low on Stock (Less than 50):

MEDICINES LOW ON STOCK(LESS THAN 50)

Medicine ID	Medicine Name	Quantity Available	Category	Price
123003	Livogen	25	Capsule	5.00
123006	Benadryl 200 ML	35	Syrup	50.00
123007	Lopamide	15	Capsule	5.00
123009	Omeprazole	35	Capsule	4.00

Fig 8.17 Displays the Medicines which are low on stock(less than 50).

8.18. Stock Expiring within 6 Months:

STOCK EXPIRING WITHIN 6 MONTHS

Purchase ID	Supplier ID	Medicine ID	Quantity	Cost of Purchase	Date of Purchase	Manufacturing Date	Expiry Date

Fig 8.18 Displays the Stock of Medicine which are expiring within 6 months.

8.19. Transaction Reports:

TRANSACTION REPORTS

Start Date:	dd-mm-yyyy	End Date:	dd-mm-yyyy
View Records			

Purchase ID	Supplier ID	Medicine ID	Quantity	Date of Purchase	Cost of Purchase(in Rs)
1001	136	123010	200	2020-03-01	1500.50
1003	145	123006	20	2020-04-22	800.00
1004	156	123004	250	2020-04-02	1000.00
Total					Rs 3300.50

Sale ID	Customer ID	Employee ID	Date	Sale Amount(in Rs)
2	987106	1	2020-04-21	585.00
3	987103	4567010	2020-04-15	120.00
4	987104	4567006	2020-04-14	955.00
5	987103	1	2020-04-21	45.00
6	987102	4567001	2020-03-11	140.00
7	987105	1	2020-04-24	350.00
8	987104	4567001	2020-04-24	35.00
12	987103	1	2020-04-24	60.00
13	987104	4567001	2020-04-24	62.50
Total				Rs 2352.50

Fig 8.19 Displays the Transaction Reports of Medicines which are sold.

Online course certificate



Elite NPTEL ONLINE CERTIFICATION

(Funded by the MoE, Govt. of India)



This certificate is awarded to
PRAGADEESWARAN
for successfully completing the course



Data Base Management System

with a consolidated score of **61** %

Online Assignments	18.75/25	Proctored Exam	42/75
--------------------	----------	----------------	-------

Total number of candidates certified in this course: **7134**

Jul-Sep 2024
(8 week course)

Prof. Haimanti Banerji
Coordinator, NPTEL
IIT Kharagpur



Indian Institute of Technology Kharagpur



Roll No: NPTEL24CS75S133105294

To verify the certificate



No. of credits recommended: 2 or 3

CERTIFICATE OF EXCELLENCE

THIS CERTIFICATE IS AWARDED TO

SCALER
Topics

PRASHETH P (RA2211047010138)

In recognition of the completion of the tutorial: DBMS Course - Master the Fundamentals and Advanced Concepts

Following are the learning items, which are covered in this tutorial

74 Video Tutorials 16 Modules 16 Challenges

14 October 2024

Anshuman Singh

Co-founder SCALER



CERTIFICATE OF EXCELLENCE

THIS CERTIFICATE IS AWARDED TO

SCALER
Topics

Logapriyan R

In recognition of the completion of the tutorial: **DBMS Course - Master the Fundamentals and Advanced Concepts**

Following are the learning items, which are covered in this tutorial

▶ 74 Video Tutorials ⏱ 16 Modules ⚙ 16 Challenges

17 September 2024



Anshuman Singh

Co-founder **SCALER** 

