

**NATIONAL INSTITUE OF TECHNOLOGY, DURGAPUR**

PROJECT :- PHARMACY INVENTORY MANAGEMENT SYSTEM

**TEAM MEMBERS**

|  |  |  |
| --- | --- | --- |
| **NAME** | **ROLL NO.** | **REG. NO.** |
| RAKESH MAHATO | 23CS8056 | 23U10221 |
| SOORYANSH SINGH | 23CS8057 | 23U10227 |
| MODADUGU NAGA VENKATESH | 23CS8058 | 23U10229 |
| MD. SAAD RAHMAN | 23CS8059 | 23U10232 |
| BAIKANI SHIVA | 23CS8060 | 23U10237 |

**ACKNOWLEDGMENT**

We, the project team, would like to express our heartfelt gratitude to our project guide and all faculty members for their continuous support, encouragement, and valuable feedback throughout the development of this project. Their guidance has played a crucial role in the successful completion of our Pharmacy Inventory Management System.

We also extend our thanks to our classmates, friends, and family for their constant support. Working together as a team of five members has been a highly collaborative and enriching experience. Each member contributed significantly to the design, development, and documentation of the system.

**ABSTRACT**

The Pharmacy Inventory Management System is a database-driven project designed to simplify and optimize the inventory and sales management of a pharmacy. With growing demands and customer expectations, traditional manual tracking methods prove inefficient and prone to error. This project presents a structured solution built using MySQL, focusing on accurate medicine tracking, supplier management, and insightful sales data analytics. It ensures that expired or low-stock medicines are flagged in time, revenue is easily computed, and supplier relations are optimized through meaningful data-driven decisions.

**INTRODUCTION**

In the rapidly evolving landscape of healthcare, pharmacies play a critical role in ensuring the timely delivery of medications to patients. Efficient inventory management is essential for minimizing losses, reducing operational delays, and ensuring high-quality customer service. Poor inventory practices often result in out-of-stock situations, expired medications, and inefficient sales tracking.

This project aims to address these challenges by developing a **Pharmacy Inventory Management System** using **MySQL**. The system is designed to streamline the management of medicines, suppliers, and sales, offering real-time insights into stock levels, expiration alerts, supplier performance, and sales data. With a robust database schema and structured queries, the system helps pharmacies operate more effectively, reduce waste, and improve business intelligence.

**PROBLEM STATEMENT**

A digital Pharmacy Inventory Management System is needed to automate the process, reduce errors, and provide a centralized database to store and manage data. The system should enable quick access to medicine stock status, supplier information, and sales transactions, allowing pharmacy staff to make informed decisions in real-time.

To support these functionalities, the system is structured around three core tables:

1. **Medicines**: Stores detailed information about each medicine, including pricing, stock levels, and expiry dates.
2. **Suppliers**: Maintains contact and address details for medicine suppliers, ensuring traceability and efficient procurement.
3. **Sales**: Records each sale transaction with relevant details like quantity sold, date of sale, and revenue generated.

In addition to maintaining data integrity, the system supports several critical queries that address practical business questions:

* Identify medicines that will expire soon, helping in timely clearance or returns.
* Calculate monthly sales to evaluate business performance.
* Detect low-stock medicines to prompt reordering.
* Determine the most popular medicines based on sales data.
* Assess which suppliers contribute the most to sales revenue.
* Monitor supplier inventory contributions to optimize partnerships.
* Highlight medicines that have never been sold, pointing to possible overstock or irrelevant inventory.
* Count the number of active suppliers for administrative insights.
* Compute the average revenue generated per day, helping in forecasting and business planning.

**OBJECTIVE OF THE PROJECT**

1. To design and develop a reliable and efficient inventory management system tailored for pharmacies.
2. To ensure real-time tracking of medicine stock, availability, pricing, and expiry dates for timely replenishment and waste reduction.
3. To streamline the management of supplier information, including contact details and medicine procurement history.
4. To maintain a comprehensive sales record that includes transaction dates, quantities sold, and revenue generated.
5. To enable fast and accurate execution of essential SQL queries for monitoring business performance, generating insights, and supporting decision-making processes.
6. To improve overall operational efficiency and accuracy in inventory control, reducing the chances of stock-outs or overstocking.

**SYSTEM DESIGN**

The Entity-Relationship Diagram (ERD) represents the logical structure of the pharmacy inventory system, illustrating the entities, their attributes, and the relationships among them.

* **Entities:**
  + **Medicines:** Represents the inventory of medicines available in the pharmacy. It holds attributes such as MedicineID (primary key), Name, Price, Stock, and ExpiryDate.
  + **Suppliers:** Represents the suppliers who provide medicines to the pharmacy. Attributes include SupplierID (primary key), Name, Contact, and Address.
  + **Sales:** Represents sales transactions. It includes SaleID (primary key), MedicineID (foreign key), QuantitySold, SaleDate, and TotalPrice.
  + **MedicineSupplier**: A many-to-many relationship table showing which suppliers supply which medicines and in what quantity.
  + **Customers**: Keeps track of customer names, the medicine they purchase, and quantity.
* **Relationship Types Used:**
* **One-to-Many between Medicines and Sales**: A single medicine can appear in multiple sales records, indicating repeated purchases of the same item over time. This relationship allows tracking sales volume and identifying popular or frequently sold medicines.
* **One-to-Many between Medicines and Customers**: Each customer can purchase one or more types of medicine, but each record in the Customers table is tied to a specific medicine purchase. This design helps in understanding customer purchasing behaviour and medicine preference.
* **Many-to-Many between Medicines and Suppliers (via MedicineSupplier)**: Medicines can be supplied by multiple suppliers, and each supplier can offer multiple medicines. This many-to-many relationship is implemented using the MedicineSupplier junction table. It allows tracking supplier contributions, sourcing options, and managing procurement effectively.

These relationship types together create a robust structure for understanding inventory flow, supplier dependencies, customer interactions, and sales performance, which are crucial for efficient pharmacy management.

**SCHEMA DESIGN**

The schema design outlines the structure of the database using three primary tables:

* **Medicines Table:**
  + MedicineID INT, Primary Key, Auto-Incremented
  + Name VARCHAR(100)
  + Price DECIMAL(10,2)
  + Stock INT
  + ExpiryDate DATE
* **Suppliers Table:**
  + SupplierID INT, Primary Key, Auto-Incremented
  + Name VARCHAR(100)
  + Contact VARCHAR(20)
  + Address TEXT
* **Sales Table:**
  + SaleID INT, Primary Key, Auto-Incremented
  + MedicineID INT, Foreign Key
  + QuantitySold INT
  + SaleDate DATE
  + TotalPrice DECIMAL(10,2)
* **MedicineSupplier Table:**
  + MedicineID INT, Foreign Key
  + SupplierID INT, Foreign Key
  + Quantity INT
  + Composite Primary Key (MedicineID, SupplierID)
* **Customers Table:**
  + CustomerName VARCHAR(100), NOT NULL
  + MedicineID INT, Foreign Key
  + Quantity INT

**CREATING TABLES**

create table Medicines(

MedicineID int primary key auto\_increment,

Name varchar(100) not null,

Price decimal(10,2) not null,

Stock int not null,

ExpiryDate date not null

);

create table Suppliers(

SupplierID int primary key auto\_increment,

Name varchar(100) not null,

Contact varchar(20),

Address varchar(255)

);

create table Sales(

SaleID int primary key auto\_increment,

MedicineId int not null,

QuantitySold int not null,

SaleDate date not null,

TotalPrice decimal(10,2) not null,

foreign key (MedicineID) references Medicines(MedicineID) on delete cascade

);

--extra tables

create table medicinesupplier (

MedicineID int,

SupplierID int,

Quantity int,

primary key (MedicineID, SupplierID),

foreign key (MedicineID) references Medicines(MedicineID),

foreign key (SupplierID) references Suppliers(SupplierID)

);

create table customers (

customername varchar(100) not null,

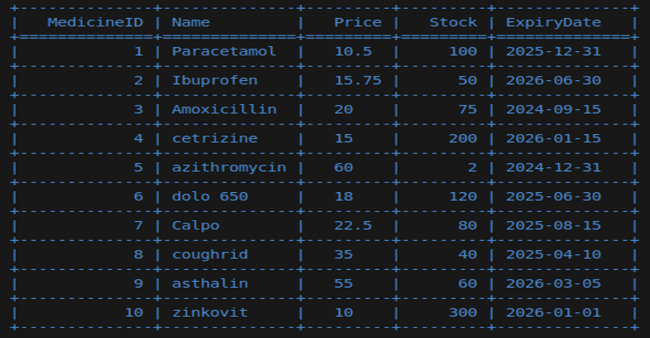
MedicineID int,

Quantity int,

foreign key (MedicineID) references Medicines(MedicineID)

);

**INSERT DATA INTO TABLE**

insert into Medicines(Name,Price,Stock,ExpiryDate)values

('Paracetamol',10.50,100,'2025-12-31'),

('Ibuprofen',15.75,50,'2026-06-30'),

('Amoxicillin',20.00,75,'2024-09-15'),

('cetrizine',15.00,200,'2026-01-15'),

('azithromycin',60.00,2,'2024-12-31'),

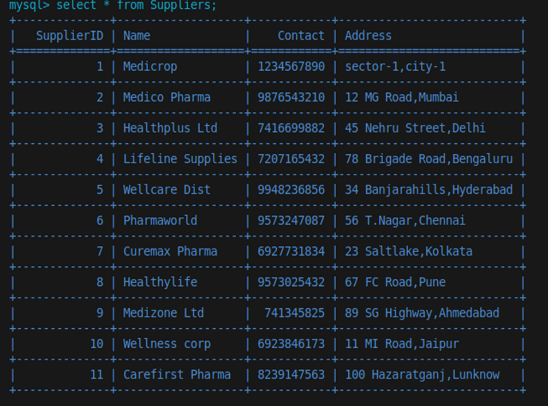
('dolo 650',18.00,120,'2025-06-30'),

('Calpo',22.50,80,'2025-08-15'),

('coughrid',35.00,40,'2025-04-10'),

('asthalin',55.00,60,'2026-03-05'),

('zinkovit',10.00,300,'2026-01-01');

insert into Suppliers(Name,Contact,Address) values

('Medicrop','1234567890','sector-1,city-1'),

('Medico Pharma','9876543210','12 MG Road,Mumbai'),

('Healthplus Ltd','7416699882','45 Nehru Street,Delhi'),

('Lifeline Supplies','7207165432','78 Brigade Road,Bengaluru'),

('Wellcare Dist','9948236856','34 Banjarahills,Hyderabad'),

('Pharmaworld','9573247087','56 T.Nagar,Chennai'),

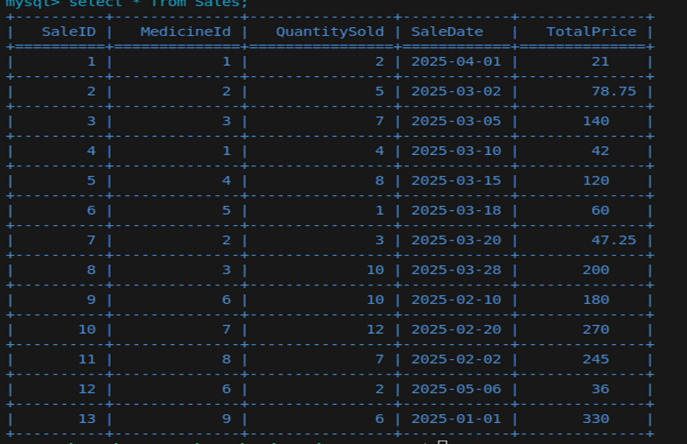
('Curemax Pharma','6927731834','23 Saltlake,Kolkata'),

('Healthylife','9573025432','67 FC Road,Pune'),

('Medizone Ltd','741345825','89 SG Highway,Ahmedabad'),

('Wellness corp','6923846173','11 MI Road,Jaipur'),

('Carefirst Pharma','8239147563','100 Hazaratganj,Lunknow');

insert into Sales(MedicineId,QuantitySold,SaleDate,Totalprice) values

(1,2,'2025-04-01',21.00);

(2,5,'2025-03-02',78.75),

(3,7,'2025-03-05',140),

(1,4,'2025-03-10',42),

(4,8,'2025-03-15',120),

(5,1,'2025-03-18',60),

(2,3,'2025-03-20',47.25),

(3,10,'2025-03-28',200),

(6,10,'2025-02-10',180),

(7,12,'2025-02-20',270),

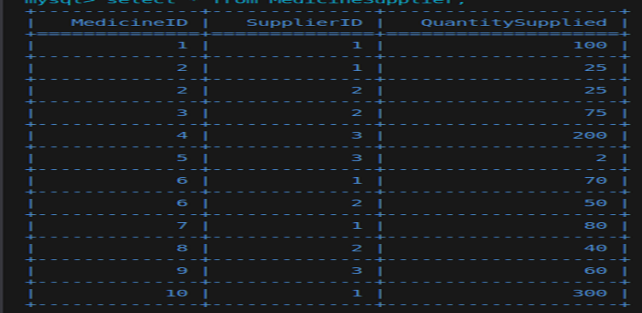
(8,7,'2025-02-02',245),

(6,2,'2025-05-06',36),

(9,6,'2025-01-01',330);

insert into medicinesupplier (MedicineID, SupplierID) value(1, 1,100),

(2, 1,25),

(2, 2,25),

(3, 5),

(4, 3,200),

(5, 3,2),

(6, 1,70),

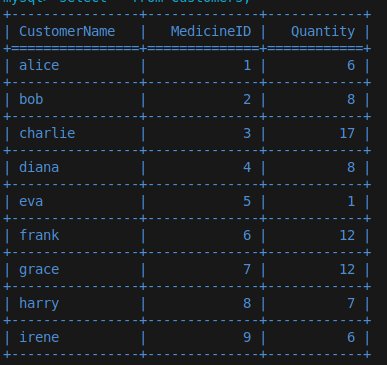
(6, 2,50),

(7, 1,80),

(8, 2,40),

(9, 3,60),

(10, 1,300);

insert into customers (customername, MedicineID, Quantity) values

('alice', 1, 6),

('bob', 2, 8),

('charlie', 3, 17),

('diana', 4, 8),

('eva', 5, 1),

('frank', 6, 12),

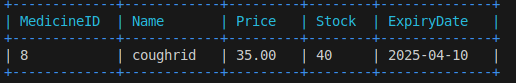
('grace', 7, 12),

('harry', 8, 7),

('irene', 9, 6);

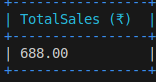
Queries

--Query 1

select \*

from Medicines

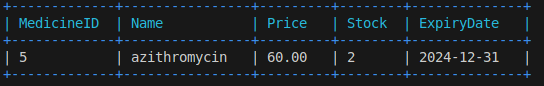
where expirydate between curdate() and date\_add(curdate(), interval 30 day);

--Query 2

select sum(totalprice) as totalsales

from sales

where month(saledate)=3 and year(saledate)=2025;

--Query 3

select \*

from Medicines

where Stock < 10;

--Query 4

select m.name, sum(s.QuantitySold) as totalquantitysold

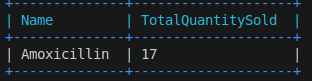
from Sales s

join Medicines m on s.Medicineid = m.Medicineid

group by s.Medicineid

order by totalquantitysold desc

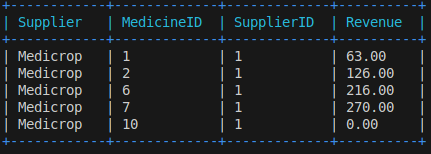
limit 1;



--Query 5

select

supp.Name as Supplierame,

 ms.MedicineID,

ms.SupplierID,

sum(s.TotalPrice) as totalrevenue

from Sales s

join Medicines m on s.MedicineID = m.MedicineID

join medicinesupplier ms on m.MedicineID = ms.MedicineID

join Suppliers supp on ms.SupplierID = supp.SupplierID

where supp.SupplierID = 1

group by supp.Name, ms.MedicineID, ms.SupplierID;

--Query 6

SELECT s.SupplierID, s.Name, SUM(ms.Quantity) AS TotalMedicinesSupplied

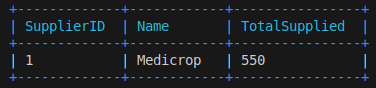
FROM medicinesupplier ms

JOIN Suppliers s ON ms.SupplierID = s.SupplierID

GROUP BY s.SupplierID, s.Name

ORDER BY TotalMedicinesSupplied DESC

LIMIT 1;



--Query 7

select

c.name as customername,

m.name as name,

c.Quantity,

m.Price,

(c.Quantity \* m.Price) as totalcost

from customers c

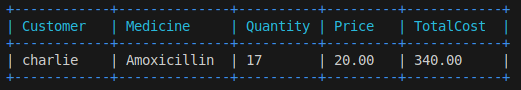
join Medicines m on c.MedicineID = m.MedicineID

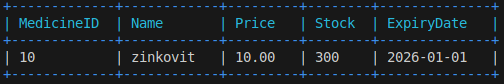
where (c.Quantity \* m.Price) = (

select max(c2.Quantity \* m2.Price)

from customers c2

join Medicines m2 on c2.MedicineID = m2.MedicineID

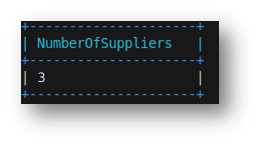
);

--Query 8

select \*

from Medicines

where Medicineid not in (select distinct Medicineid from Sales);



--Query 9

select count(distinct SupplierID) as numberOfSupp

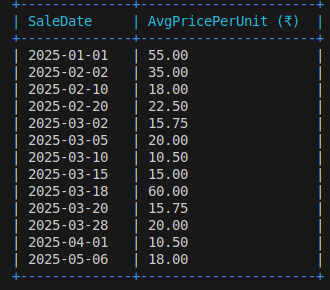
from medicinesupplier;

--Query 10

select SaleDate, avg(TotalPrice / QuantitySold) as avgprice

from Sales

group by SaleDate;



**Conclusion and Suggestions**

The Pharmacy Inventory Management System provides a scalable and structured approach to handling pharmaceutical stock and sales. It ensures better planning, reduces losses from expiry, and promotes smarter inventory decisions through robust queries.

**Suggestions for Improvement:**

* Develop a front-end interface for user-friendly operations.
* Integrate barcode scanning for fast medicine lookup.
* Include notifications for low-stock or upcoming expiry.
* Add customer and billing modules.
* Use analytics dashboards for visual insights and reporting.

GitHub Repo Link:

[GitHub - Naga-Venkatesh/Pharmacy-Inventory\_Management\_System: Mini project for DBMS Group No 12](https://github.com/Naga-Venkatesh/Pharmacy-Inventory_Management_System)

**Bibliography**

1. Silberschatz, A., Korth, H. F., & Sudarshan, S. (2020). *Database System Concepts* (7th ed.). McGraw-Hill Education.
2. MySQL Documentation – https://dev.mysql.com/doc/
3. W3Schools SQL Tutorial – https://www.w3schools.com/sql/
4. TutorialsPoint MySQL – https://www.tutorialspoint.com/mysql/index.htm
5. GeeksforGeeks DBMS Guide – https://www.geeksforgeeks.org/dbms/