ADDIS ABABA UNIVERSITY

College of Natural and Computational Science (CNCS)

Department of Computer Science

Fundamental of Database Project

Pharmacy Management System

Section 1

Name	**ID NO.**
1. Adonias Abiyot	UGR/0796/15
2. Biniam Daniel	UGR/2308/16
3. Tsegaye Shewamare	UGR/2048/16
4. Ezra Ayalew	UGR/9077/16
5. Natan Tewodros	UGR/6982/16
6. Mesfin Tibebu	UGR/2459/16
7. Samuel Kinfe (leader)	UGR/2027/16
8. Sawel Yohannes	UGR/2969/16

Submitted to: Instructor Fikirte G

Submission date: Wednesday, June 10, 2025

Contents

2. Introduction

Pharmacies play a crucial role in healthcare, ensuring the availability of medications for patients. However, many pharmacies still rely on manual inventory management, which can lead to issues like running out of stock or having expired medications on shelves. To address these challenges, we aim to develop a Pharmacy Management System (PMS) that will help manage pharmacy inventory more efficiently. The PMS stores all pharmacy data in a SQL server, letting staff manage inventory details easily. It tracks inventory continuously, alerts staff about low stock or expiring medications, and offers quick access to medication information.

3. Profile of the Pharmacy

Hemen Pharmacy is located at 4 Kilo around post office. It was established to provide a wide range of medications to the local community. The pharmacy operates daily and has good and collaborative employees who ensure the availability and proper management of medical products. Supported by various healthcare institutions and organizations, it is known for its reliable service and contribution to public health. The pharmacy aims to create a safe and efficient environment for both staff and customers by using modern inventory management technologies. After discussing with the managers, we have found solutions to improve the pharmacy and enhance its operations.

4. Mission of the Pharmacy

The mission of Hemen Pharmacy is to provide high-quality medications and excellent customer service to the community. The pharmacy aims to make sure medications are always available and to prevent the use of expired products. By offering reliable and accessible pharmaceutical care, the pharmacy supports the health and well-being of its customers. The pharmacy also aims to educate customers about their medications and promote safe and effective use.

5. Vision of the Pharmacy

The vision of Hemen Pharmacy is to become a key part of the community, known for its excellent service, reliability, and commitment to health. The pharmacy strives to meet the changing needs of the community by being a leader in pharmacy services and care. It aims to be a place where community members can engage, learn, and grow. Through innovation, dedication, and hard work, Hemen Pharmacy seeks to provide the tools and opportunities needed to achieve the best health outcomes for everyone.

6. Project Goals and Results

Our project aims to help pharmacies manage their inventory more efficiently. We observed that many pharmacies face significant issues due to manual inventory management, such as running out of stock, having expired medications on shelves, difficulty in tracking inventory, and time consuming manual processes.

Our Pharmacy Management System (PMS) will solve these problems by providing:

1. Real-time Inventory Tracking:
- ✓ Current stock level information.

2. Automated Alerts:
- ✓ Low stock warnings.
- ✓ Expiry notifications.

3. Easy Information Access:

- ✓ Quick access to medication details.
- ✓ Information on expiration dates and suppliers.

7. Entities and Attributes

1. Medication

Attributes:

- MedicationID (PK)
- Name
- Category
- Form
- ReorderLevel
- ExpirationDate
- SupplierID (FK)

Attribute Types:

Primary Key: MedicationID

Atomic: Name, Category, Form, ReorderLevel, ExpirationDate

Foreign Key: SupplierID

2. Supplier

Attributes:

- SupplierID (PK)
- CompanyName
- Phone
- Email
- Address_City
- Address_State
- Address_Country

Attribute Types:

Primary Key: SupplierID

Atomic: Phone, Email

Composite: Address (City, State, Country)

3. Inventory

Attributes:

- Inventory_ID (PK)

- Medication_ID (FK)
- QuantityInStock
- LastUpdated
- Location_ShelfNo
- Location_Room
- EmployeeID(FK)

Attribute Types:

Primary Key: Inventory_ID

Atomic: QuantityInStock, LastUpdated

Foreign Key: MedicationID,EmployeeID

Composite: Location (ShelfNo, Room)

4. PurchaseOrder

Attributes:

- PurchaseOrderID (PK)
- OrderDate
- Status
- ExpectedDeliveryDate o
- TotalAmount
- SupplierID (FK)
- EmployeeID(FK)

Attribute Types:

Primary Key: PurchaseOrderID

Atomic: OrderDate, Status,ExpectedDeliveryDate,TotalAmount,

Foreign Key: SupplierID,EmployeeID

Derived: TotalAmount

5. PurchaseOrderDetail

Attributes:

- PurchaseOrderID (part of composite PK, FK)
- MedicationID (part of composite PK, FK)
- QuantityOrdered o
- UnitPrice
- TotalPrice

Attribute Types:

Composite Primary Key: (PurchaseOrderID, MedicationID)

Atomic: QuantityOrdered, UnitPrice, TotalPrice

Foreign Keys: PurchaseOrderID, MedicationID

Derived: TotalPrice

6. Alert

Attributes:

- AlertID (PK)
- MedicationID (FK)
- AlertType
- AlertDate
- Resolved

Attribute Types:

Primary Key: AlertID

Atomic: AlertType, AlertDate, Resolved

Foreign Key: MedicationID

7. **Employee**

Attributes:

- EmployeeID (PK)
- Name_FName
- Name_LName
- Role
- ContactNumber
- Email

Attribute Types:

Primary Key: EmployeeID

Atomic: Role, ContactNumber, Email

Composite: Name (FName, LName)

From the entities above, the PurchaseOrderDetail entity is a **weak entity** because it cannot be uniquely identified without the relationship it has with the PurchaseOrder and Medication entities. Each PurchaseOrderDetail is associated with one purchase order and one medication, and the combination of PurchaseOrderID and MedicationID is what makes each PurchaseOrderDetail unique.

8. Relationships between entities

Supplier to Medication

Name: Supplies

A supplier can supply multiple medications, but each medication is supplied by one supplier**.**

Keys:

Medication: MedicationID (PK), SupplierID (FK)

Supplier: SupplierID (PK)

Medication to Inventory

Name: TrackedBy

An inventory record tracks the stock of a medication, and each medication can have multiple inventory records.

Keys:

Inventory: InventoryID (PK), MedicationID (FK)

Medication: MedicationID (PK)

Supplier to PurchaseOrder

Name: PlacesOrder

A supplier can have multiple purchase orders, but each purchase order is placed with one supplier.

Keys:

PurchaseOrder: PurchaseOrderID (PK), SupplierID (FK)

Supplier: SupplierID (PK)

PurchaseOrderDetail to PurchaseOrder

Name: Contains

A purchase order contains multiple purchase order details, each belonging to one purchase order. This is an identifying relationship, meaning that the existence of a purchase order detail is dependent on the corresponding purchase order.

Keys:

PurchaseOrder: PurchaseOrderID (PK)

PurchaseOrderDetail: PurchaseOrderID (part of composite PK, FK), MedicationID (part of composite PK, FK)

<u>Medication to PurchaseOrderDetail</u>

Name: Includes

A medication can appear in multiple purchase order details, but each purchase

Order detail includes one medication.
Keys:

PurchaseOrderDetail: PurchaseOrderID (part of composite PK, FK), MedicationID (part of composite PK, FK)
Medication: MedicationID(PK)
<u>Medication to Alert</u>
Name: HasAlert
A medication can have multiple alerts, and each alert is specific to one medication.

Keys:

Alert: AlertID (PK), MedicationID (FK)

Medication: MedicationID (PK)

<u>Employee to Inventory</u>

Name: ManagesInventory

An employee can manage multiple inventory records, but each inventory record is managed by one employee.

Keys:

Inventory: InventoryID (PK), EmployeeID (FK)

Employee: EmployeeID (PK)

<u>Employee to PurchaseOrder</u>

Name: ProcessesOrder

An employee can process multiple purchase orders, but each purchase order is processed by one employee.

Keys:

PurchaseOrder: PurchaseOrderID (PK), EmployeeID (FK)

Employee: EmployeeID (PK)

9. Constraints

A. Cardinality ratio and participation constraints

Supplies: One medicine can only have one supplier, but one supplier must supply and can supply many medicines.

Cardinality:

One-to-Many (1: N) from Supplier to Medicine.

One-to-One (1:1) from Medicine to Supplier.

Participation:

Total participation from Supplier (each supplier must supply at least one medicine).

Total participation from Medicine (each medicine must have exactly one supplier).

Place Order: One purchase order can have only one supplier, but one supplier can have no orders or multiple purchase orders.

Cardinality:

One-to-Many (1: N) from Supplier to Purchase Order.

One-to-One (1:1) from Purchase Order to Supplier.

Participation:

Partial participation from Supplier (a supplier can have no orders or many orders).

Total participation from Purchase Order (each order must be associated with exactly one supplier).

Has Alert: One medication can have zero or multiple alerts, but one alert is associated with one medication.

Cardinality:

One-to-Many (1: N) from Medication to Alert.

One-to-One (1:1) from Alert to Medication.

Participation:

Partial participation from Medication (a medication can have no alerts or many alert).

Total participation from Alert (each alert must be associated with exactly one medication).

Includes: One medication can have zero or multiple purchase order details, but one purchase order detail is associated with one medication.

Cardinality:

One-to-Many (1: N) from Medication to Purchase Order Detail.

One-to-One (1:1) from Purchase Order Detail to Medication.

Participation:

-Partial participation from Medication (a medication can have no purchase order details or many purchase order detail).

-Total participation from Purchase Order Detail (each purchase order detail must be associated with one medication).

TrackedBy: One inventory can only contain one type of medicine or no medicine, but one medicine can be stored in multiple inventories (every medicine must have inventory).

Cardinality:

One-to-Many (1: N) from Medicine to Inventory.

One-to-One (1:1) from Inventory to Medicine.

Participation:

Total participation from Medicine (every medicine must be stored in at least one inventory).

Partial participation from Inventory (an inventory can be empty or contain one type of medicine).

Contains (Identifying Relationship): Each purchase order contains multiple purchase order details. Each purchase order detail belongs to one purchase order.

Cardinality:

One-to-Many (1: N) from Purchase Order to Purchase Order Detail.

One-to-One (1:1) from Purchase Order Detail to Purchase Order.

Participation:

Total participation from Purchase Order Detail (each detail must be part of exact one purchase order).

Partial participation from Purchase Order (each order my contain zero or many detail).

ProcessesOrder: One employee must order and can order multiple purchases. One purchase order is ordered only by one employee.

Cardinality:

One-to-Many (1: N) from Employee to Purchase Order.

One-to-One (1:1) from Purchase Order to Employee.

Participation:

Total participation from Employee (every employee must process at least one order).

Total participation from Purchase Order (each purchase order must be processed by one employee).

Manages Inventory: One employee must manage and can manage multiple inventories. One inventory is managed by only one employee.

Cardinality:

-One-to-Many (1: N) from Employee to Inventory.

One-to-One (1:1) from Inventory to Employee.

Participation:

Total participation from Employee (each employee must manage at least one inventory).

Total participation from Inventory (each inventory must be managed by one employee).

B. General relational constraints

Primary keys, foreign keys, and name entries cannot be NULL.

Primary key IDs will have a character value of less than 12 characters.

Quantity-related attributes (QuantityInStock, ReorderLevel, QuantityOrdered) must be non-negative.

Date-related attributes (ExpirationDate, OrderDate, ExpectedDeliveryDate, LastUpdated, AlertDate) should be valid dates.

On Update, all foreign keys will also cascade with the changes made.

Normalization

First Normal Form (1NF):

Our database fulfills the requirements of the First Normal Form. Each table in the database has a unique primary key that ensures the uniqueness of each row. The attributes within each table are atomic, meaning that each attribute contains only indivisible values. Additionally, any composite attributes have been decomposed into separate columns. For example, in the Supplier table, attributes such as Address, City, State, and Country are stored in individual columns rather than as a single composite attribute.

Second Normal Form (2NF):

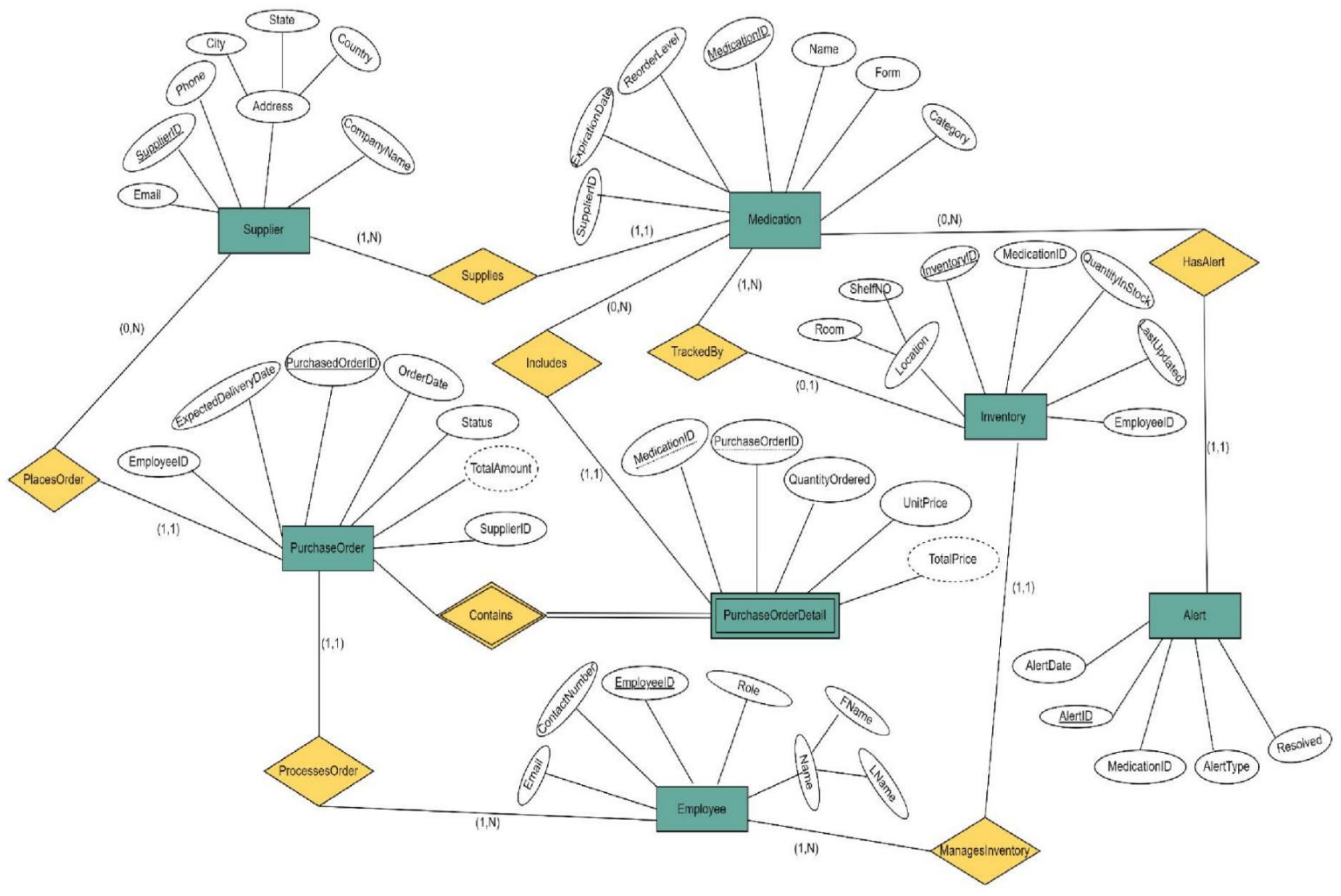
The Second Normal Form is satisfied in our database by ensuring that each non-primary attribute is fully dependent on the primary key of the relation. In each table, every attribute is functionally dependent on the primary key and not on a subset of the primary key. For instance, in the PurchaseOrderDetail table, the combination of PurchaseOrderID and MedicationID serves as the composite primary key, and attributes like QuantityOrdered and UnitPrice are fully dependent on this composite key. This eliminates partial dependencies.

Third Normal Form (3NF):

Our database design adheres to the Third Normal Form by eliminating transitive dependencies. This means that non-primary key attributes are dependent only on the primary key and not on any other non-primary key attributes. For example, in the Medication table, attributes such as Name, Category, Form, ReorderLevel, and ExpirationDate depend solely on the primary key

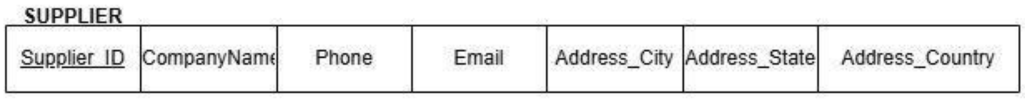
MedicationID. In the Supplier table, attributes like CompanyName, ContactPerson, Phone, Email, Address, City, State, and Country are all dependent directly on the SupplierID primary key and not on any other non-key attributes.

10. ER DIAGRAM



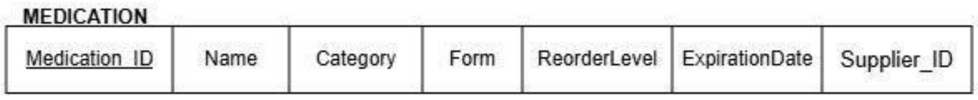
11. Logical design of the database

1. Converting ER diagram into relational tables



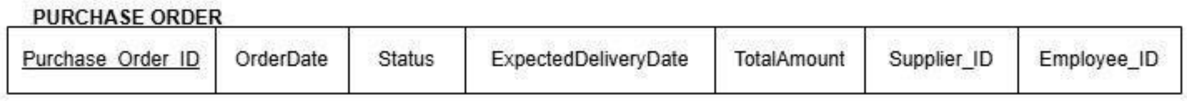
SUPPLIER

Supplier_ID	CompanyName	Phone	Email	Address_City	Address_State	Address_Country



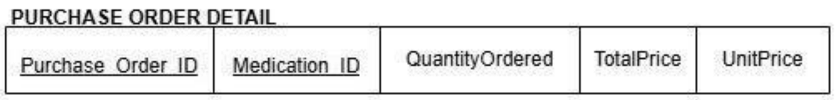
MEDICATION

Medication_ID	Name	Category	Form	ReorderLevel	ExpirationDate	Supplier_ID



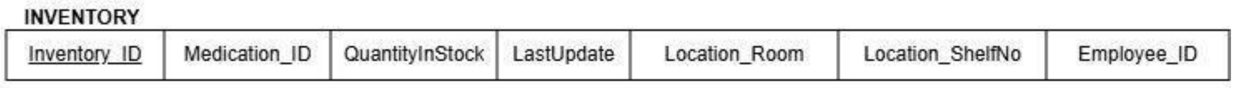
PURCHASE ORDER

Purchase_Order_ID	OrderDate	Status	ExpectedDeliveryDate	TotalAmount	Supplier_ID	Employee_ID



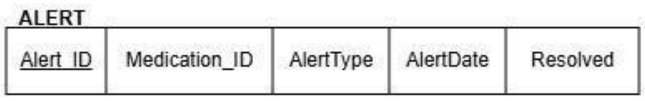
PURCHASE ORDER DETAIL

Purchase_Order_ID	Medication_ID	QuantityOrdered	TotalPrice	UnitPrice



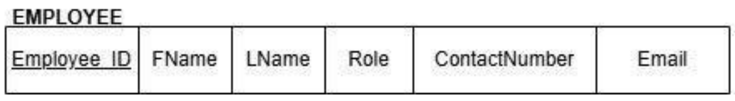
INVENTORY

Inventory_ID	Medication_ID	QuantityInStock	LastUpdate	Location_Room	Location_ShelfNo	Employee_ID



ALERT

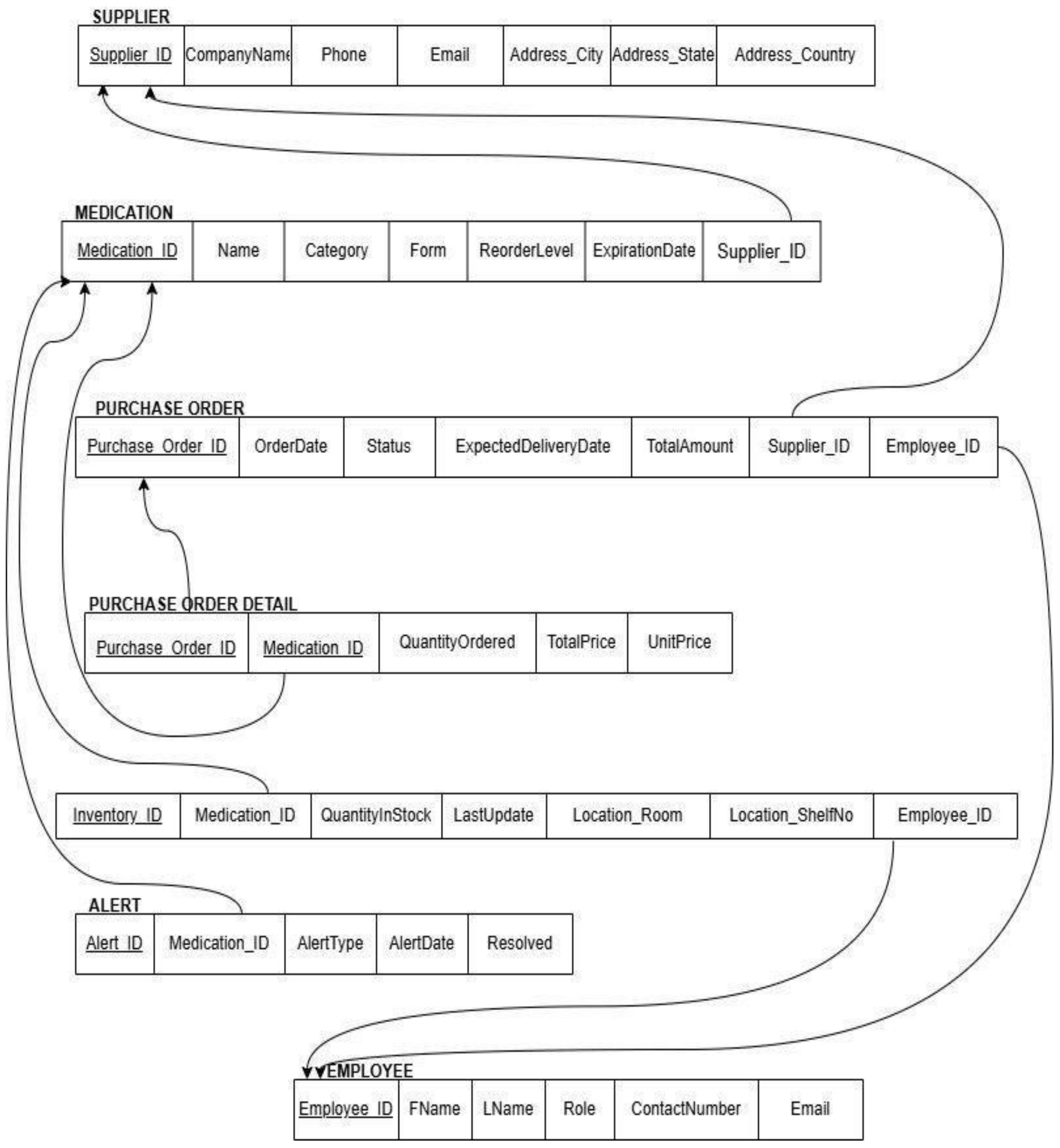
Alert_ID	Medication_ID	AlertType	AlertDate	Resolved



EMPLOYEE

Employee_ID	FName	LName	Role	ContactNumber	Email

2. Mapping


SUPPLIER
Supplier_ID | CompanyName | Phone | Email | Address_City | Address_State | Address_Country
MEDICATION
Medication_ID | Name | Category | Form | ReorderLevel | ExpirationDate | Supplier_ID
PURCHASE ORDER
Purchase_Order_ID | OrderDate | Status | ExpectedDeliveryDate | TotalAmount | Supplier_ID | Employee_ID
PURCHASE ORDER DETAIL
Purchase_Order_ID | Medication_ID | QuantityOrdered | TotalPrice | UnitPrice
Inventory_ID | Medication_ID | QuantityInStock | LastUpdate | Location_Room | Location_ShelfNo | Employee_ID
ALERT
Alert_ID | Medication_ID | AlertType | AlertDate | Resolved
EMPLOYEE
Employee_ID | FName | LName | Role | ContactNumber | Email

Purchaseorder table

Purchase order Id	OrderDate	Status	ExpectedDelliveryDate	TotalAmount	Supplier_ID	EmployeeID
1	2024-06-01	Completed	2024-06-08	25.00	1	1
2	2024-06-02	Completed	2024-06-09	7.50	2	2
3	2024-06-03	Pending	2024-06-10	25.00	3	1
4	2024-06-04	Pending	2024-06-11	25.00	4	2
5	2024-06-05	Pending	2024-06-12	30.00	5	1

Puchaseorderdetail table

Purchase order id	Medication ID	Quantity Ordered	UnitPrice	Total Price
1	1	10	2.50	25.00
2	2	15	0.50	7.50
3	3	5	5.00	25.00
4	4	20	1.25	25.00
5	5	10	3.00	30.00

Supplier table

Supplier id	Company Name	Phone	Email	Address_City	Address_State	Address_Country
1	Pharma Supplier PLC	+251911234567	1	Addis Ababa'	Addis Ababa'	Ethiopia
2	Health Products Inc.	+251922345678	2	Gondar	Amhara	Ethiopia
3	Med Supply Co.	+251933456789	3	Hawassa	SNNPR	Ethiopia
4	Dire Pharma Ltd.	+251944567890	4	Dire Dawa	Dire Dawa	Ethiopia
5	Alemu Pharmaceuticals	+251955678901	5	Jimma	Oromia	Ethiopia

Medication table

Medication Id	Name	Category	Form	ReorderLevel	ExpirationDate	Supplier_Id
1	Paracetamol	Analgesic	Tablet	20	2024-06-30	1
2	Ibuprofen	Antiinflammatory	Tablet	15	2025-01-15	2
3	Amoxicillin	Antibiotic	Capsule	10	2024-12-01	3
4	Ciprofloxacin	Antibiotic	Tablet	10	2023-11-25	4
5	Doxycycline	Antibiotic	Capsule	5	2025-02-20	5

Inventory table

Inventory id	Medication id	lastUpdate	Location_Room	Location_No	Employee id
1	1	2024-06-01	A1	1	4
2	2	2024-06-01	A2	2	5
3	3	2024-06-01	B1	3	4
4	4	2024-06-01	B2	4	5
5	5	2024-06-01	C1	5	5

Employee table

Employee id	F Name	L Name	Role	ContactNumber	Email
1	Abebe	Kebede	Pharmacist	+251987654321	abebe.kebede@pharmacy.et
2	Biniam	Tsegaye	Pharmacy assistant	+251976543210	biniam.tsegaye@pharmacy.et
3	Chaltu	Bekele	pharmacist	+251965432109	chaltu.bekele@pharmacy.et
4	Dawit	Asrat	Inventory Manager	+251954321098	dawit.asrat@pharmacy.et
5	Eden	Mulugeta	Inventory Manager	+251943210987	eden.mulugeta@pharmacy.et

12. DATABASE IMPLEMENTATION AND QUERYING USING SQL

Implementation

Creating our schema and relations:

drop database if exists pharmacymanagementsystem; create database pharmacymanagementsystem;

Create table Supplier (

SupplierID INT PRIMARY KEY NOT NULL,

CompanyName VARCHAR(100) NOT NULL,

ContactPerson VARCHAR(100) NOT NULL,

Phone VARCHAR(20) NOT NULL,

Email VARCHAR(100) NOT NULL UNIQUE,

Address VARCHAR(200) NOT NULL,

City VARCHAR(50) NOT NULL,

State VARCHAR(50) NOT NULL,

Country VARCHAR(50) NOT NULL

);

Create table Medication (MedicationID INT PRIMARY KEY NOT NULL, Name VARCHAR(100) NOT NULL, Category VARCHAR(50) NOT NULL, Form VARCHAR(50) NOT NULL, ReorderLevel INT NOT NULL CHECK (ReorderLevel >= 0), ExpirationDate DATE NOT NULL, SupplierID INT NOT NULL, FOREIGN KEY (SupplierID) REFERENCES Supplier(SupplierID));

Create table Inventory (

InventoryID INT PRIMARY KEY NOT NULL,

MedicationID INT NOT NULL,

QuantityInStock INT NOT NULL CHECK (QuantityInStock >= 0),

LastUpdated DATE NOT NULL,

Room VARCHAR(50) NOT NULL,

Shelf_no INT NOT NULL CHECK (Shelf_no > 0),

FOREIGN KEY (MedicationID) REFERENCES Medication(MedicationID));

Create table PurchaseOrder (PurchaseOrderID INTAUTO_INCREMENT PRIMARY KEY, OrderDate DATE NOT NULL,

Status VARCHAR(50) NOT NULL CHECK (Status IN ('Pending', 'Completed', 'Cancelled')), ExpectedDeliveryDate DATE NOT NULL, SupplierID INT NOT NULL, Total_Amount

DECIMAL (10, 2) DEFAULT 0, FOREIGN KEY (SupplierID) REFERENCES Supplier(SupplierID));

Create table PurchaseOrderDetail (

PurchaseOrderID INT NOT NULL,

MedicationID INT NOT NULL,

QuantityOrdered INT NOT NULL CHECK (QuantityOrdered > 0),

UnitPrice DECIMAL(10, 2) NOT NULL CHECK (UnitPrice >= 0),

Total_Price DECIMAL(10, 2) GENERATED ALWAYS AS (QuantityOrdered * UnitPrice) STORED,

PRIMARY KEY (PurchaseOrderID, MedicationID),

FOREIGN KEY (PurchaseOrderID) REFERENCES PurchaseOrder(PurchaseOrderID),

FOREIGN KEY (MedicationID) REFERENCES Medication(MedicationID)

);

Create table Alert (AlertID INTAUTO_INCREMENT PRIMARY KEY, MedicationID INT NOT NULL, AlertType VARCHAR(50) NOT NULL, AlertDate DATE NOT NULL, Resolved BOOLEAN NOT NULL, FOREIGN KEY (MedicationID) REFERENCES Medication(MedicationID));

Create table Employee (

EmployeeID INT PRIMARY KEY NOT NULL,

F_name VARCHAR(50) NOT NULL,

L_name VARCHAR(50) NOT NULL,

Role VARCHAR(50) NOT NULL,

ContactNumber VARCHAR(20) NOT NULL,

Email VARCHAR(100) NOT NULL UNIQUE

);

Populating our relations:

INSERT INTO Supplier (SupplierID, CompanyName, ContactPerson, Phone, Email, Address, City, State, Country) VALUES (1, 'Pharma Supplier PLC', 'Abebe Bekele', '+251911234567', 'info@pharmasupplier.com', 'Bole Road', 'Addis Ababa', 'Addis Ababa', 'Ethiopia'), (2, 'Health Products Inc.', 'Mulu Tadesse', '+251922345678', 'contact@healthproducts.com', 'Piassa', 'Gondar', 'Amhara', 'Ethiopia'), (3, 'Med Supply Co.', 'Tigist Kebede', '+251933456789', 'sales@medsupply.com', 'Main Street', 'Hawassa', 'SNNPR', 'Ethiopia'), (4, 'Dire Pharma Ltd.', 'Kassahun Mengistu', '+251944567890', 'support@direpharma.com', 'Haramaya Avenue', 'Dire

Dawa', 'Dire Dawa', 'Ethiopia'), (5, 'Alemu Pharmaceuticals', 'Sisay Alemu', '+251955678901', 'info@alemupharma.com', 'Jimma Road', 'Jimma', 'Oromia', 'Ethiopia');

INSERT INTO Medication (MedicationID, Name, Category, Form, ReorderLevel, ExpirationDate, SupplierID)

VALUES

(1, 'Paracetamol', 'Analgesic', 'Tablet', 20, '2024-06-30', 1),

(2, 'Ibuprofen', 'Anti-inflammatory', 'Tablet', 15, '2025-01-15', 2),

(3, 'Amoxicillin', 'Antibiotic', 'Capsule', 10, '2024-12-01', 3),

(4, 'Ciprofloxacin', 'Antibiotic', 'Tablet', 10, '2023-11-25', 4),

(5, 'Doxycycline', 'Antibiotic', 'Capsule', 5, '2025-02-20', 5),

(6, 'Metformin', 'Antidiabetic', 'Tablet', 15, '2024-10-10', 1),

(7, 'Aspirin', 'Analgesic', 'Tablet', 20, '2025-08-15', 2),

(8, 'Lisinopril', 'Antihypertensive', 'Tablet', 10, '2024-07-20', 3),

(9, 'Atorvastatin', 'Lipid-lowering', 'Tablet', 10, '2025-05-25', 4),

(10, 'Omeprazole', 'Antacid', 'Capsule', 5, '2024-09-30', 5);

INSERT INTO Inventory (InventoryID, MedicationID, QuantityInStock, LastUpdated, Room, Shelf_no)

VALUES

(1, 1, 10, '2024-06-01', 'A1', 1),

(2, 2, 25, '2024-06-01', 'A2', 2),

(3, 3, 5, '2024-06-01', 'B1', 3),

(4, 4, 15, '2024-06-01', 'B2', 4),

(5, 5, 2, '2024-06-01', 'C1', 5),

(6, 6, 20, '2024-06-01', 'C2', 6),

(7, 7, 8, '2024-06-01', 'D1', 7),

(8, 8, 12, '2024-06-01', 'D2', 8),

(9, 9, 30, '2024-06-01', 'E1', 9),

(10, 10, 40, '2024-06-01', 'E2', 10);

INSERT INTO PurchaseOrder (PurchaseOrderID, OrderDate, Status, ExpectedDeliveryDate, SupplierID)

VALUES

```
(1, '2024-06-01', 'Completed', '2024-06-08', 1),
(2, '2024-06-02', 'Completed', '2024-06-09', 2),
(3, '2024-06-03', 'Pending', '2024-06-10', 3),
(4, '2024-06-04', 'Pending', '2024-06-11', 4),
(5, '2024-06-05', 'Pending', '2024-06-12', 5);
INSERT INTO PurchaseOrderDetail (PurchaseOrderID, MedicationID, QuantityOrdered,
UnitPrice)
VALUES
(1, 1, 10, 2.50),
(1, 2, 15, 0.50),
(2, 3, 5, 5.00),
(2, 4, 20, 1.25),
(3, 5, 10, 3.00),
(3, 6, 15, 0.75),
(4, 7, 20, 1.50),
(4, 8, 25, 2.00),
(5, 9, 30, 0.25),
(5, 10, 40, 1.00);
UPDATE PurchaseOrder po
JOIN (
SELECT PurchaseOrderID, SUM(Total_Price) AS TotalAmount
FROM PurchaseOrderDetail
GROUP BY PurchaseOrderID
) pod ON po.PurchaseOrderID = pod.PurchaseOrderID
SET po.Total_Amount = pod.TotalAmount
WHERE po.PurchaseOrderID IN (SELECT DISTINCT PurchaseOrderID FROM
PurchaseOrderDetail);
INSERT INTO Alert (MedicationID, AlertType, AlertDate, Resolved)
SELECT
M.MedicationID,
'Low Stock' AS AlertType,
CURRENT_DATE AS AlertDate,
FALSE AS Resolved
FROM
Medication M
JOIN
Inventory I ON M.MedicationID = I.MedicationID
WHERE
I.QuantityInStock <= M.ReorderLevel;
-- step 2: Insert Expired Medication Alerts
INSERT INTO Alert (MedicationID, AlertType, AlertDate, Resolved)
SELECT
MedicationID,
'Expired' AS AlertType,
CURRENT_DATE AS AlertDate,
FALSE AS Resolved
FROM
Medication
WHERE
ExpirationDate < CURRENT_DATE;
INSERT INTO Employee (EmployeeID, F_name, L_name, Role, ContactNumber, Email)
VALUES
(1, 'Abebe', 'Kebede', 'Pharmacist', '+251987654321', 'abebe.kebede@pharmacy.et'),
(2, 'Biniam', 'Tsegaye', 'Inventory Manager', '+251976543210',
'biniam.tsegaye@pharmacy.et'),
(3, 'Chaltu', 'Bekele', 'Order Processor', '+251965432109', 'chaltu.bekele@pharmacy.et'),
(4, 'Dawit', 'Asrat', 'Pharmacy Technician', '+251954321098', 'dawit.asrat@pharmacy.et'), (5,
'Eden', 'Mulugeta', 'Pharmacist', '+251943210987', 'eden.mulugeta@pharmacy.et');
```

Queries

Query 1: List all Medications with their Supplier Information

```
SELECT
m.MedicationID,
m.Name AS MedicationName,
m.Category,
s.CompanyName AS SupplierName,
s.ContactPerson,
s.Phone,
s.Email,
s.City,
s.State,
s.Country
FROM
Medication m
JOIN
Supplier s ON m.SupplierID = s.SupplierID;
```

Query 2: Find Expired Medications

```
SELECT
MedicationID,
Name,
ExpirationDate
FROM
Medication
WHERE
ExpirationDate < CURDATE();
```

Query 3: Find Medications that need to be Reordered

```
SELECT
m.MedicationID,
m.Name FROM
Medication m
JOIN
```

```
Inventory i ON m.MedicationID = i.MedicationID
WHERE
i.QuantityInStock <= m.ReorderLevel;
```

Query 4: Get all Active Alerts for Medications

```
SELECT
a.AlertID,
m.Name AS MedicationName,
a.AlertType,
a.AlertDate,
a.Resolved
FROM
Alert a
JOIN
Medication m ON a.MedicationID = m.MedicationID
WHERE
a.Resolved = FALSE;
```

Query 5:

Step 1: Insert new purchase orders for medications below reorder level

```
INSERT INTO PurchaseOrder (OrderDate, Status, ExpectedDeliveryDate, SupplierID)
SELECT CURRENT_DATE, 'Pending', DATE_ADD(CURRENT_DATE, INTERVAL 7 DAY), m.SupplierID
FROM Medication m
JOIN Inventory i ON m.MedicationID = i.MedicationID
WHERE i.QuantityInStock <= m.ReorderLevel;
```

Step 2: Insert new purchase order details with previous unit price

```
INSERT INTO PurchaseOrderDetail (PurchaseOrderID, MedicationID, QuantityOrdered, UnitPrice)
SELECT
po.PurchaseOrderID, m.MedicationID,
(m.ReorderLevel * 2) AS QuantityOrdered, -- Example quantity to order
```

```
(SELECT pod.UnitPrice
FROM PurchaseOrderDetail pod
WHERE pod.MedicationID = m.MedicationID
ORDER BY pod.PurchaseOrderID DESC
LIMIT 1) AS UnitPrice -- Fetch the most recent UnitPrice
FROM Medication m
JOIN Inventory ON m.MedicationID = i.MedicationID
JOIN PurchaseOrder po ON po.SupplierID = m.SupplierID
WHERE i.QuantityInStock <= m.ReorderLevel
AND po.OrderDate = CURRENT_DATE;
```

Step 3: Update Total_Amount in PurchaseOrder table with a WHERE clause

```
UPDATE PurchaseOrder po
JOIN (
SELECT PurchaseOrderID, SUM(Total_Price) AS TotalAmount
FROM PurchaseOrderDetail
GROUP BY PurchaseOrderID
) pod ON po.PurchaseOrderID = pod.PurchaseOrderID
SET po.Total_Amount = pod.TotalAmount
WHERE po.PurchaseOrderID IN (SELECT DISTINCT PurchaseOrderID FROM
PurchaseOrderDetail);
```

Step 4: Update alerts for medications that are low on stock and have had purchase orders generated

```
SET SQL_SAFE_UPDATES = 0;
WITH cte AS (
SELECT a.AlertID
FROM Alert a
JOIN Medication m ON a.MedicationID = m.MedicationID JOIN Inventory i ON
m.MedicationID = i.MedicationID WHERE i.QuantityInStock <= m.ReorderLevel
AND a.AlertType = 'Low Stock'
)
UPDATE Alert
```

```
SET Resolved = TRUE
WHERE AlertID IN (SELECT AlertID FROM cte);
SET SQL_SAFE_UPDATES = 1;
```

Query 6: List all Purchase Orders with their Details

```
SELECT
po.PurchaseOrderID,
m.Name AS MedicationName,
s.CompanyName AS SupplierName,
pod.QuantityOrdered, pod.UnitPrice,
(pod.QuantityOrdered * pod.UnitPrice) AS TotalPrice, po.OrderDate,
po.ExpectedDeliveryDate, po.Status
FROM
PurchaseOrder po
JOIN
Supplier s ON po.SupplierID = s.SupplierID
JOIN
PurchaseOrderDetail pod ON po.PurchaseOrderID = pod.PurchaseOrderID
JOIN
Medication m ON pod.MedicationID = m.MedicationID;
```

Query 7: Get Inventory Details for a Specific Medication

```
SELECT
i.InventoryID,
m.Name AS MedicationName,
i.QuantityInStock,
i.LastUpdated,
i.Room,
i.Shelf_no
FROM
Inventory i
JOIN
```

```
Medication m ON i.MedicationID = m.MedicationID
WHERE
m.Name = 'Paracetamol';
```

Query 8: List All Employees with their Contact Information

```
SELECT
EmployeeID,
F_name AS FirstName,
L_name AS LastName,
ContactNumber,
Email
FROM
Employee;
```

Query 9: Total Stock Value of Each Medication

```
SELECT
m.MedicationID,
m.Name AS MedicationName,
SUM (i.QuantityInStock * pod.UnitPrice) AS TotalStockValue
FROM
Medication m
JOIN
Inventory i ON m.MedicationID = i.MedicationID
JOIN
PurchaseOrderDetail pod ON m.MedicationID = pod.MedicationID
GROUP BY
m.MedicationID,
m.Name;
```

Query 11: List Suppliers in a Specific City

```
SELECT
SupplierID,
CompanyName,
ContactPerson,
Phone,
Email,
City,
State,
Country
FROM
Supplier
WHERE
City = 'Addis Ababa';
```

13. Conclusion

In conclusion, the Pharmacy Management System Database aims to help pharmacies manage their inventory more efficiently. Many pharmacies face issues with stock outs and expired medications due to manual inventory tracking. This new database system will address these problems.

The key goals of the system are:

- ✓ Tracking inventory in real-time
- ✓ Alerting staff when stock is low or medications are expiring soon
- ✓ Providing easy access to information about medications, like expiration dates and suppliers

By implementing this database, pharmacies will be able to improve their inventory management. They can avoid running out of medications or having expired products on the shelves. The system will make it simpler for staff to find important details about the medications they stock.