



# Medical Supply Management System

Asif Akbar

Roll:2007106

CSE 3110: Database Systems Laboratory

# Introduction



## Project Overview

Medical supply management system database efficiently organizes the management of medical supplies crucial for healthcare providers, featuring tables like Suppliers, Supplies, Buyers, Orders and Usage. It tracks supplier details, supply specifics such as type and stock levels, and buyer information, primarily hospitals or clinics. The Orders table manages the logistics of medical supply orders, including quantities and delivery schedules, while the Usage table records the consumption of supplies, which aids in inventory and usage tracking. This system enhances supply chain operations by ensuring adequate stock levels, facilitating financial oversight, and improving communication between suppliers and buyers. Overall, it supports healthcare facilities in maintaining uninterrupted service and efficient care delivery by managing critical medical supplies effectively.



## Importance of database management in medical field

Database management plays a crucial role in medical supply management by ensuring operational efficiency and regulatory compliance. The structured tables like 'Suppliers', 'Supplies', 'Orders', and 'Usage' facilitate accurate tracking of inventory, supply sourcing, and usage patterns, which are essential for maintaining adequate stock levels and preventing shortages. This system aids in optimizing purchasing strategies and financial management, thereby enhancing patient care by ensuring that essential medical supplies are always available. Overall, effective database management supports strategic decision-making and improves the quality of healthcare services.



## Objectives

- ❖ To design a relational database schema for managing medical supply related data.
- ❖ To maintain comprehensive records of suppliers, including their contact details, supply capabilities, and performance, which helps in managing procurement processes more effectively and ensures the reliability of supply chains.
- ❖ To streamline the process of ordering and receiving supplies, including monitoring the status of each order, managing delivery timelines, and ensuring that orders are fulfilled according to the healthcare facility's needs.
- ❖ To document the usage of supplies within healthcare facilities, providing valuable data that can be used for future planning, budgeting, and reporting. This helps in understanding consumption patterns and can aid in more accurate forecasting and procurement planning.
- ❖ To generate reports that help in decision-making and strategic planning for healthcare facility management.
- ❖ To implement the SQL queries for effective data retrieval, update and management.

These objectives collectively aim to enhance the efficiency of medical supply management, reduce wasteful expenditures, and improve the overall quality of patient care by ensuring that necessary medical supplies are always available and effectively utilized.



## Database Schema

The medical supply management system uses multiple tables to efficiently store information that are useful to capture all necessary details. Here is a detailed overview of each table in my database schema:

### 1. **Suppliers:**

- **Supplier\_ID:** A unique identifier for each supplier (Primary Key).
- **Name :** The name of the supplier.
- **Contact\_Info :** Contact details of the supplier, typically a phone number or email.

- **Address** : Physical address of the supplier.

## 2. **Supplies:**

- **Supply\_ID**: A unique identifier for each supply item (Primary Key).
- **Name** : The name of the supply item.
- **Type** : The category or type of the supply.
- **Manufacturer** : The company that manufactures the supply.
- **Supplier\_ID** : A foreign key linking each item to its supplier.
- **Cost** : The cost per unit of the supply.
- **Quantity\_In\_Stock**: Current stock quantity of the item.

## 3. **Buyers:**

- **Buyer\_ID** : A unique identifier for each buyer (Primary Key).
- **Name** : The name of the buyer.
- **Contact\_Number** : Contact phone number of the buyer.
- **Email** : Email address of the buyer.

## 4. **Orders:**

- **Order\_ID**: A unique identifier for each order (Primary Key).
- **Supply\_ID**: A foreign key linking to the supply being ordered.
- **Buyer\_ID**: A foreign key linking to the buyer who placed the order.
- **Quantity\_Ordered**: The number of units ordered.
- **Order\_Date**: The date when the order was placed.
- **Expected\_Delivery\_Date**: The expected date for the order to be delivered.
- **Status**: The status of the order, such as 'active', 'inactive', or 'pending'.

## 5. **Usage:**

- **Usage\_ID**: A unique identifier for each usage entry (Primary Key).
- **Supply\_ID**: A foreign key that links to the supply being used.
- **Quantity\_Used**: The amount of the supply that was used.
- **Usage\_Date**: The date when the supply was used.
- **Used\_By**: A foreign key linking to the buyer who used the supply.

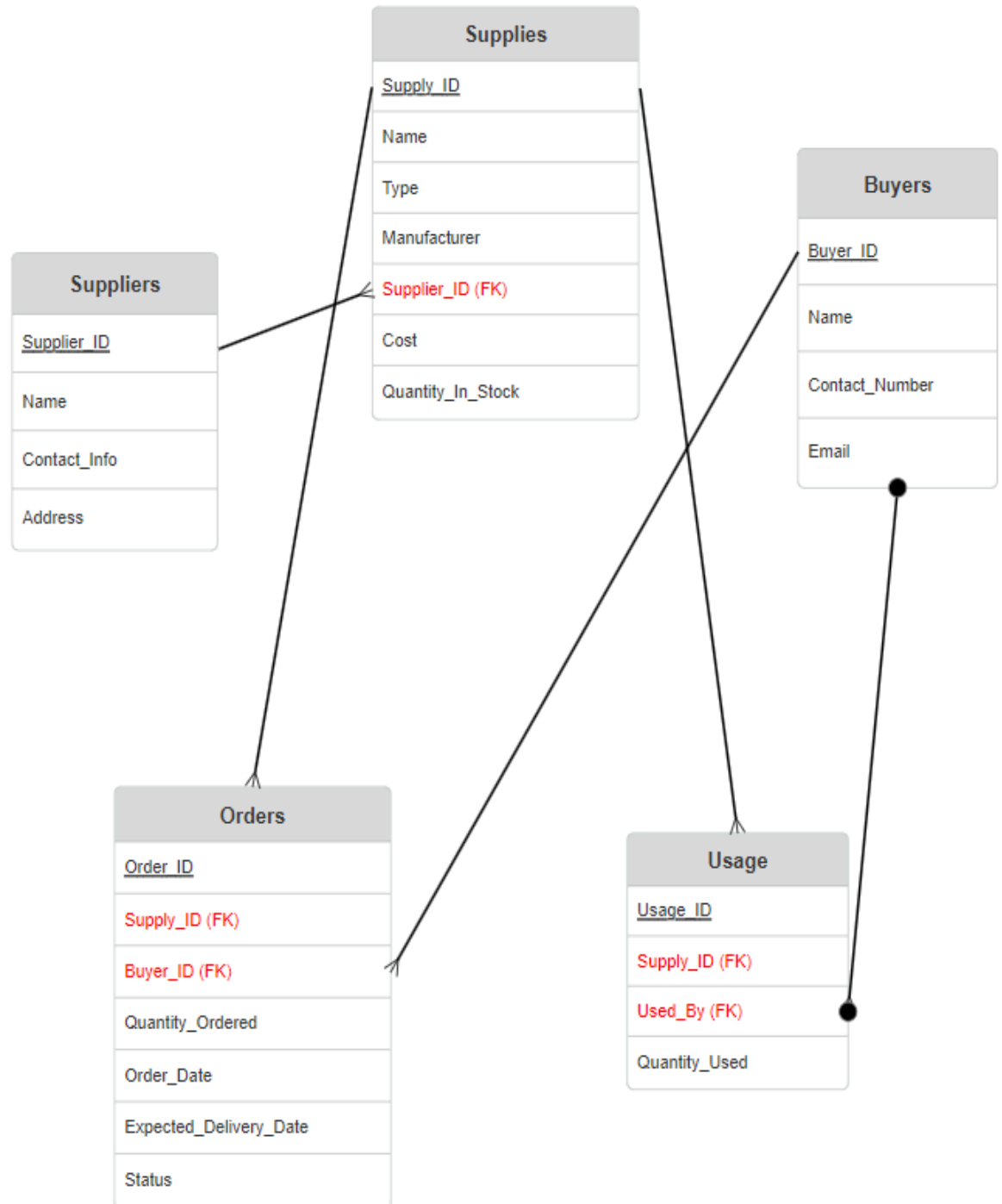
Each table is designed to capture all necessary details related to the management of medical supplies, from procurement to usage. This allows for a comprehensive view of the supply chain, helps manage inventory, and ensures efficient operation within healthcare facilities.

## Table Relationships

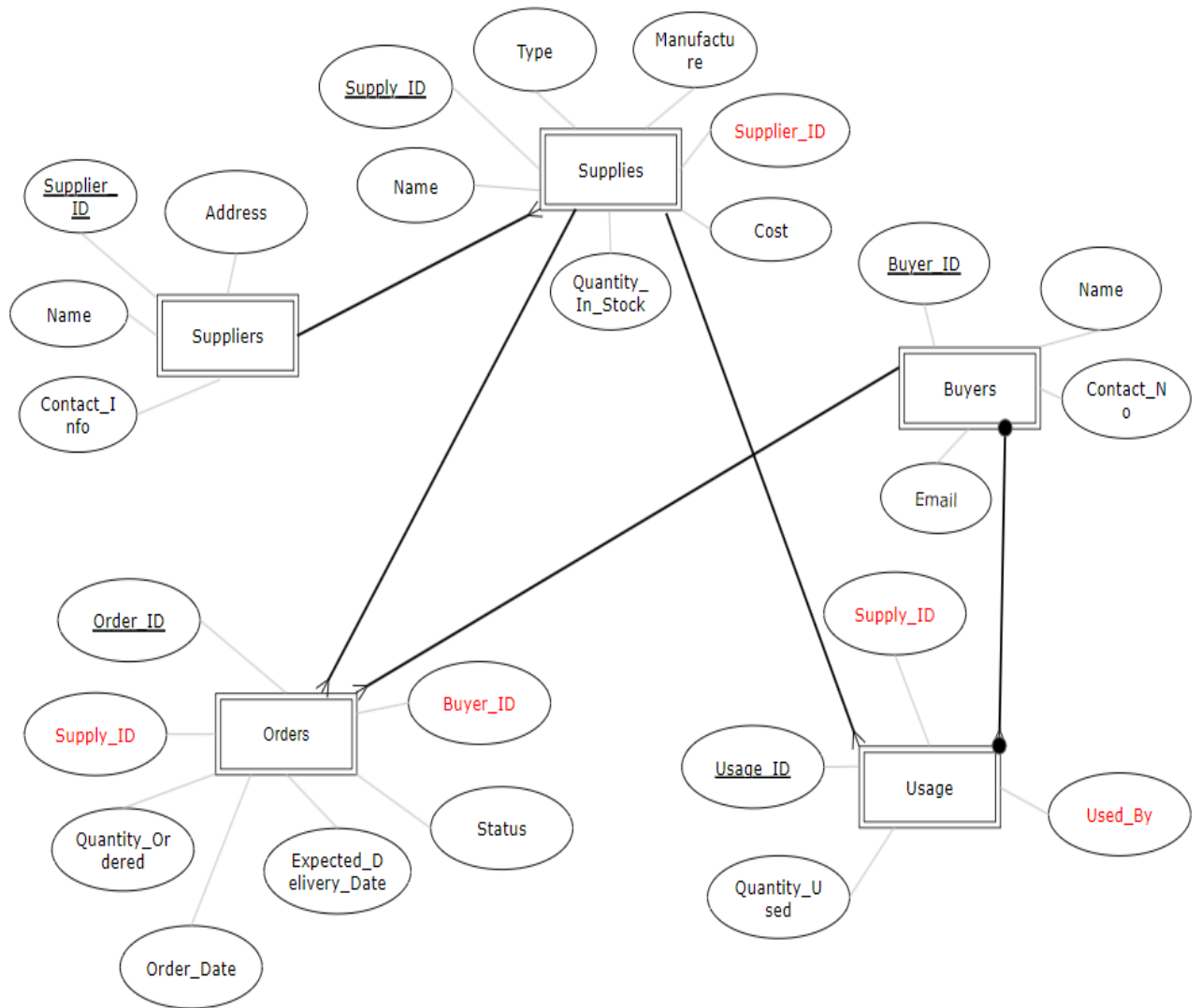
```
1  -- Creating the Suppliers table
2  CREATE TABLE Suppliers (
3      Supplier_ID INT PRIMARY KEY,
4      Name VARCHAR(255),
5      Contact_Info VARCHAR(255),
6      Address VARCHAR(255)
7  );
8
9  -- Creating the Supplies table
10 CREATE TABLE Supplies (
11     Supply_ID INT PRIMARY KEY,
12     Name VARCHAR(255),
13     Type VARCHAR(100),
14     Manufacturer VARCHAR(255),
15     Supplier_ID INT,
16     Cost DECIMAL(10,2),
17     Quantity_In_Stock INT,
18     FOREIGN KEY (Supplier_ID) REFERENCES Suppliers(Supplier_ID)
19 );
20
21 -- Creating the Buyers table
22 CREATE TABLE Buyers (
23     Buyer_ID INT PRIMARY KEY,
24     Name VARCHAR(255),
25     Contact_Number VARCHAR(15),
26     Email VARCHAR(100)
27 );
28
```

```
28 );
29 -- Creating the Orders table
30 CREATE TABLE Orders (
31     Order_ID INT PRIMARY KEY,
32     Supply_ID INT,
33     Buyer_ID INT,
34     Quantity_Ordered INT,
35     Order_Date DATE,
36     Expected_Delivery_Date DATE,
37     Status VARCHAR(20) CHECK (Status IN ('active', 'inactive', 'pending')),
38     FOREIGN KEY (Supply_ID) REFERENCES Supplies(Supply_ID),
39     FOREIGN KEY (Buyer_ID) REFERENCES Buyers(Buyer_ID)
40 );
41
42 -- Creating the Usage table
43 CREATE TABLE Usage (
44     Usage_ID INT PRIMARY KEY,
45     Supply_ID INT,
46     Quantity_Used INT,
47     Usage_Date DATE,
48     Used_By INT,
49     FOREIGN KEY (Supply_ID) REFERENCES Supplies(Supply_ID),
50     FOREIGN KEY (Used_By) REFERENCES Buyers(Buyer_ID)
51 );
```

## Schema Diagram



## Entity Relationship Diagram (ERD)



## SQL Queries Implementation

We can design several sql queries to extract information from the database. Here are some examples below:

i. Retrieving Buyer Information:

```
SELECT Name FROM Buyers WHERE Buyer_ID IN (SELECT Buyer_ID FROM Orders WHERE Supply_ID IN ( SELECT Supply_ID FROM Supplies WHERE Supplier_ID = (SELECT Supplier_ID FROM Suppliers WHERE Name = 'MedEquip Inc.') ) );
```

The SQL query outlined is designed to retrieve the names of buyers who have purchased supplies from a supplier named "MedEquip Inc." This is achieved by using a series of nested subqueries that connect multiple tables: Buyers, Orders, Supplies and Suppliers.

ii. Retrieving Supplies Information:

```
SELECT Name FROM Supplies WHERE Supply_ID IN (SELECT Supply_ID FROM Orders WHERE Buyer_ID = (SELECT Buyer_ID FROM Buyers WHERE Name = 'John Doe'));
```

The SQL query outlined is designed to retrieve the names of supplies that have been purchased by a specific buyer, in this case, 'John Doe'

iii. Retrieving Orders Information :

```
SELECT Orders.Quantity_Ordered, Orders.Order_Date, Orders.Expected_Delivery_Date, Orders.Status FROM Orders WHERE Supply_ID IN (SELECT Supply_ID FROM Supplies WHERE Name = 'N95 Respirator Mask');
```

The SQL query outlined retrieves specific details of all orders related to a supply named "N95 Respirator Mask".

iv. Retrieving Usage Information :

```
SELECT Usage.Quantity_Used FROM Usage WHERE Supply_ID IN (SELECT Supply_ID FROM Supplies WHERE Name = 'N95 Respirator Mask');
```

The SQL query outlined here specifically retrieves the quantities of the "N95 Respirator Mask" that have been used, as recorded in the Usage table of my database.



- v. Calculate the maximum quantity In Stock in the supplies table:

```
WITH Max_Quantity(qty) AS ( SELECT MAX(Quantity_In_Stock) FROM Supplies  
SELECT Supplies.* FROM Supplies, Max_Quantity WHERE  
Supplies.Quantity_In_Stock = Max_Quantity.qty;
```

- vi. Retrieve all suppliers who offer supplies at a cost that is greater than some costs of supplies manufactured by 'HealthGear'.

```
SELECT * FROM Suppliers WHERE Supplier_ID IN ( SELECT Supplier_ID FROM  
Supplies WHERE Cost > SOME (SELECT Cost FROM Supplies WHERE  
Manufacturer = 'HealthGear'));
```

- vii. Retrieve all supplies where the supplier is based in 'MedCity, MA' and there exists an order with the status 'active' for those supplies.

```
SELECT * FROM Supplies WHERE Supplier_ID IN ( SELECT Supplier_ID FROM Suppliers  
WHERE Address LIKE '%MedCity, MA%') AND EXISTS ( SELECT * FROM Orders  
WHERE Supply_ID = Supplies.Supply_ID AND Status = 'active');
```

## PL/SQL

- viii. Insert a new buyer into the Buyers table using PL/SQL. Assume I want to add a buyer with the ID 5, name "Alice Wonderland", contact number "555-0500", and email [alice.wonderland@example.com](mailto:alice.wonderland@example.com)

```
SET SERVEROUTPUT ON; DECLARE v_buyer_id Buyers.Buyer_ID%TYPE := 5;  
v_name Buyers.Name%TYPE := 'Alice Wonderland'; v_contact_number  
Buyers.Contact_Number%TYPE := '555-0500'; v_email Buyers.Email%TYPE :=  
'alice.wonderland@example.com'; BEGIN INSERT INTO Buyers (Buyer_ID, Name,  
Contact_Number, Email) VALUES (v_buyer_id, v_name, v_contact_number,  
v_email); DBMS_OUTPUT.PUT_LINE('New buyer inserted: ' || v_name);
```

- ix. Use a PL/SQL block with a cursor to iterate through all records in the Suppliers table. For each supplier, output the supplier's ID, name, contact information, and address, along with the number of rows processed up to that point.

```

SET SERVEROUTPUT ON; DECLARE cursor suppliers_cursor IS SELECT * FROM
Suppliers; supplier_row Suppliers%ROWTYPE; BEGIN OPEN suppliers_cursor;
FETCH suppliers_cursor INTO supplier_row; WHILE suppliers_cursor%FOUND
LOOP DBMS_OUTPUT.PUT_LINE('Supplier_ID: ' || supplier_row.Supplier_ID || '
Name: ' || supplier_row.Name || ' Contact_Info: ' || supplier_row.Contact_Info || '
Address: ' || supplier_row.Address); DBMS_OUTPUT.PUT_LINE('Row count: ' ||
suppliers_cursor%ROWCOUNT); FETCH suppliers_cursor INTO supplier_row;
END LOOP; CLOSE suppliers_cursor; END; /

```

- x. **Procedures** : Write a PL/SQL procedure that accepts a supply ID as input and returns the name of the supply and the total quantity ordered for that supply. The procedure should also increment the input supply ID by 1 and output this new value along with the supply name and total quantity ordered.

```

CREATE OR REPLACE PROCEDURE GetSupplyOrderDetails(var1 IN NUMBER,
var2 OUT VARCHAR2, var3 IN OUT NUMBER) AS t_show CHAR(30);
total_quantity_ordered NUMBER; BEGIN t_show := 'From procedure: '; SELECT
Name INTO var2 FROM Supplies WHERE Supply_ID = var1; SELECT
SUM(Quantity_Ordered) INTO total_quantity_ordered FROM Orders WHERE
Supply_ID = var1; var3 := var1 + 1; DBMS_OUTPUT.PUT_LINE(t_show || 'Supply
name: ' || var2 || ', Total quantity ordered: ' || total_quantity_ordered || ', Next
supply ID: ' || var3); END; / SET SERVEROUTPUT ON; DECLARE supply_id
Supplies.Supply_ID%TYPE := 1; supply_name Supplies.Name%TYPE; next_id
NUMBER; BEGIN GetSupplyOrderDetails(supply_id, supply_name, next_id); END;
/

```

- xi. **Functions** : Create a PL/SQL function that takes a supplier ID as an input and returns the name of the supplier associated with that ID.

```

CREATE OR REPLACE FUNCTION GetSupplierName(var1 IN INT) RETURN
VARCHAR2 AS supplier_name Supplies.Name%TYPE; BEGIN SELECT Name INTO
supplier_name FROM Suppliers WHERE Supplier_ID = var1; RETURN supplier_name;
END; / SET SERVEROUTPUT ON; DECLARE value VARCHAR2(255); BEGIN value :=
GetSupplierName(1); DBMS_OUTPUT.PUT_LINE('Supplier Name: ' || value); END; /

```

- xii. **Triggers** : Create a PL/SQL trigger that, before deleting an order from the Orders table, also deletes any usage records from the Usage table that are linked to the supply being ordered.

```
SET SERVEROUTPUT ON; CREATE OR REPLACE TRIGGER BeforeOrderDelete
BEFORE DELETE ON Orders REFERENCING OLD AS old FOR EACH ROW
BEGIN DELETE FROM Usage WHERE Supply_ID = :old.Supply_ID;
DBMS_OUTPUT.PUT_LINE('Deleted usage records for supply ID: ' ||
:old.Supply_ID); END; / DELETE FROM Orders WHERE Order_ID = 101;
```

## Who are the users of Medical Supply Management Database?

The users or targeted customers of a medical supply management system database typically encompass a range of stakeholders in the healthcare sector who require efficient management of medical supplies. Here are the key user groups:

1. **Hospitals and Clinics:** Medical facilities are primary users, relying on such systems to maintain adequate stock of medical supplies, manage procurement processes, and ensure that they have the necessary items on hand for patient care.
2. **Pharmacy Chains :** Large pharmacy chains use these systems to manage inventory across multiple locations, keeping track of sales data, stock levels, and supply orders to avoid shortages and overstocking.
3. **Medical Supply Distributors and Suppliers :** Distributors and suppliers of medical equipment and supplies utilize these databases to manage orders from healthcare providers, track shipments, and maintain records of inventory levels and customer information.
4. **Healthcare Administrators :** Administrators in healthcare facilities, including hospitals and outpatient centers, use these systems to oversee supply chain operations, budgeting, and financial planning related to medical supplies.
5. **Government and Non-Profit Health Organizations :** Governmental health departments and non-profit organizations that manage multiple health initiatives can use these systems to efficiently allocate resources, manage donations, and distribute supplies where they are needed most.

6. **Emergency Medical Services (EMS):** EMS providers rely on these systems to ensure that emergency vehicles and kits are always stocked with necessary medical supplies for urgent care.

These users rely on the medical supply management system to ensure a seamless, efficient, and cost-effective supply chain that can support timely and effective healthcare delivery. The system helps manage the complexities of sourcing, storing, and utilizing medical supplies within the bounds of regulatory compliance and safety standards.

## What are the benefits of using Medical Supply Management System?

Using a medical supply management system offers several significant benefits in real-world scenarios, particularly in enhancing the efficiency and effectiveness of healthcare services. Here are the key benefits:

1. **Improved Inventory Management:** These systems provide precise tracking and management of medical supplies inventory levels. This helps prevent both shortages and excess stock, ensuring that the right supplies are available when needed without tying up capital in unused stock.
2. **Enhanced Operational Efficiency:** Automation of ordering, stocking, and distribution processes reduces manual workloads and minimizes human errors. This leads to smoother operations and allows healthcare staff to focus more on patient care rather than administrative tasks.
3. **Cost Reduction:** By optimizing inventory levels and reducing wastage (e.g., from expired products), medical supply management systems help reduce costs. They also support better budgeting and financial management by providing detailed and accurate data on supply usage and costs.
4. **Compliance and Safety:** These systems help healthcare organizations comply with regulations related to medical supplies, such as storage conditions and expiry management. They ensure that supplies used are safe and effective, maintaining necessary documentation for audits and inspections.

5. **Data-Driven Decisions:** The systems gather and analyze data on supply usage trends, which can inform future purchasing decisions and help forecast demand more accurately. This leads to more informed decision-making at all levels of the organization.

6. **Increased Accountability:** With detailed tracking of where and how supplies are used, these systems enhance accountability among staff. This is crucial in scenarios where precise documentation of supply usage is necessary, such as in surgical procedures or controlled medication dispensation.

7. **Enhanced Supplier Management:** Effective management of supplier relationships is facilitated through systematic tracking of supplier performance, order accuracy, and timely delivery. This can help in negotiating better terms and improving service levels.

8. **Scalability:** As healthcare facilities grow, medical supply management systems can scale to accommodate increased needs, adding new users, more supplies, and additional facilities without a drop in performance or efficiency.

In summary, medical supply management systems significantly contribute to streamlined operations, improved care delivery, and reduced operational costs in the healthcare industry. These systems are crucial for maintaining the balance between cost efficiency, compliance, safety, and high-quality patient care.



## Conclusion

In conclusion, the Medical Supply Management System (MSMS) is an innovative solution that substantially enhances the efficiency and effectiveness of medical supply chains. This system, meticulously designed to manage a complex array of data concerning suppliers, supplies, buyers, and usage, plays a pivotal role in ensuring that healthcare providers maintain an optimal inventory level, thus preventing shortages that could impact patient care.

The implementation of the MSMS allows for streamlined procurement processes, efficient inventory tracking, and improved financial oversight. These capabilities not only reduce wasteful expenditures but also foster a proactive approach to

managing supply demands, thereby enabling healthcare facilities to deliver uninterrupted and efficient care.

Moreover, the adaptability of this database system to integrate complex SQL queries and PL/SQL procedures further empowers healthcare administrators to make data-driven decisions, which are crucial for strategic planning and operational management. The system's robust structure and detailed record-keeping enhance compliance with regulatory standards, ensure safety in supply usage, and improve overall supply chain transparency.

Ultimately, the Medical Supply Management System stands as a testament to the critical role of advanced database management in the medical field. It not only supports the logistical aspects of medical supply but also contributes significantly to the broader goal of enhancing patient care and operational efficiency in healthcare settings. This project has demonstrated potential benefits that extend beyond simple management tasks, influencing broader strategic objectives that can lead to more informed and effective healthcare practices.