# Inventory Management SQL Project

## Project Overview

This SQL project is designed to manage the inventory of medicines and supplies in an urgent clinic. It includes tracking expiration dates, low-stock alerts, order history, and an automatic reordering system.

## Database Schema

### Suppliers Table

CREATE TABLE Suppliers (  
 supplier\_id INT PRIMARY KEY AUTO\_INCREMENT,  
 name VARCHAR(255) NOT NULL,  
 contact\_info TEXT  
 );

### Medicines Table

CREATE TABLE Medicines (  
 medicine\_id INT PRIMARY KEY AUTO\_INCREMENT,  
 name VARCHAR(255) NOT NULL,  
 category VARCHAR(100),  
 dosage VARCHAR(50),  
 expiration\_date DATE,  
 supplier\_id INT,  
 stock\_quantity INT DEFAULT 0,  
 reorder\_level INT DEFAULT 0,  
 FOREIGN KEY (supplier\_id) REFERENCES Suppliers(supplier\_id) ON DELETE SET NULL  
 );

### Supplies Table

CREATE TABLE Supplies (  
 supply\_id INT PRIMARY KEY AUTO\_INCREMENT,  
 name VARCHAR(255) NOT NULL,  
 category VARCHAR(100),  
 supplier\_id INT,  
 stock\_quantity INT DEFAULT 0,  
 reorder\_level INT DEFAULT 0,  
 FOREIGN KEY (supplier\_id) REFERENCES Suppliers(supplier\_id) ON DELETE SET NULL  
 );

### Transactions Table

CREATE TABLE Transactions (  
 transaction\_id INT PRIMARY KEY AUTO\_INCREMENT,  
 item\_id INT NOT NULL,  
 item\_type ENUM('medicine', 'supply') NOT NULL,  
 quantity INT NOT NULL,  
 transaction\_type ENUM('in', 'out') NOT NULL,  
 transaction\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
 );

### Orders Table

CREATE TABLE Orders (  
 order\_id INT PRIMARY KEY AUTO\_INCREMENT,  
 supplier\_id INT,  
 order\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 status ENUM('pending', 'completed', 'cancelled') DEFAULT 'pending',  
 FOREIGN KEY (supplier\_id) REFERENCES Suppliers(supplier\_id) ON DELETE SET NULL  
 );

### Auto\_Orders Table (For Automatic Reordering)

CREATE TABLE Auto\_Orders (  
 auto\_order\_id INT PRIMARY KEY AUTO\_INCREMENT,  
 item\_id INT NOT NULL,  
 item\_type ENUM('medicine', 'supply') NOT NULL,  
 quantity INT NOT NULL,  
 order\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 supplier\_id INT,  
 status ENUM('pending', 'ordered') DEFAULT 'pending',  
 FOREIGN KEY (supplier\_id) REFERENCES Suppliers(supplier\_id) ON DELETE SET NULL  
 );

## Triggers & Automation

### Trigger for Automatic Reordering (Medicines)

DELIMITER //  
 CREATE TRIGGER auto\_reorder AFTER UPDATE ON Medicines  
 FOR EACH ROW  
 BEGIN  
 IF NEW.stock\_quantity < NEW.reorder\_level THEN  
 INSERT INTO Auto\_Orders (item\_id, item\_type, quantity, supplier\_id, status)  
 VALUES (NEW.medicine\_id, 'medicine', (NEW.reorder\_level - NEW.stock\_quantity) \* 2, NEW.supplier\_id, 'pending');  
 END IF;  
 END;  
 //  
 DELIMITER ;

### Trigger for Automatic Reordering (Supplies)

DELIMITER //  
 CREATE TRIGGER auto\_reorder\_supplies AFTER UPDATE ON Supplies  
 FOR EACH ROW  
 BEGIN  
 IF NEW.stock\_quantity < NEW.reorder\_level THEN  
 INSERT INTO Auto\_Orders (item\_id, item\_type, quantity, supplier\_id, status)  
 VALUES (NEW.supply\_id, 'supply', (NEW.reorder\_level - NEW.stock\_quantity) \* 2, NEW.supplier\_id, 'pending');  
 END IF;  
 END;  
 //  
 DELIMITER ;

### Scheduled Event for Processing Orders

DELIMITER //  
 CREATE EVENT process\_auto\_orders  
 ON SCHEDULE EVERY 1 DAY  
 DO  
 BEGIN  
 UPDATE Auto\_Orders  
 SET status = 'ordered'  
 WHERE status = 'pending';  
 END;  
 //  
 DELIMITER ;

## Conclusion

This SQL project provides a structured approach to managing clinic inventory. It ensures:  
- Efficient stock tracking with automatic reordering.  
- Proper supplier management.  
- Logging of all transactions and orders.  
  
This implementation can be expanded further with reporting features and user role management if needed.