```sql
CREATE DATABASE Inventory;

USE Inventory;

-- Creating the Suppliers Table
CREATE TABLE Suppliers (
    supplier_id INT PRIMARY KEY,
    name VARCHAR(100),
    contact_name VARCHAR(100),
    phone VARCHAR(20),
    email VARCHAR(100)
);

-- Inserting the values into Suppliers table
INSERT INTO Suppliers (supplier_id, name, contact_name, phone,
email) VALUES
(1, 'Tech Distributors Inc', 'Alice Thompson', '123-456-7890',
'alice@techdist.com'),
(2, 'GadgetSupply Co.', 'Brian Matthews', '234-567-8901',
'brian@gadgetsupply.com'),
(3, 'Office Plus', 'Charles Lee', '345-678-9012',
'charles@officeplus.com'),
(4, 'CableTech', 'Dana White', '456-789-0123',
'dana@cabletech.com'),
(5, 'SmartPeripherals Ltd', 'Emily Carter', '567-890-1234',
'emily@smartperipherals.com'),
(6, 'NeoTech Solutions', 'Frank Zhang', '678-901-2345',
'frank@neotech.com'),
(7, 'Global Components', 'Grace Kim', '789-012-3456',
'grace@globalcomponents.com');

-- Getting all the records from Suppliers table
SELECT * from Suppliers;

-- Creating the Products Table
CREATE TABLE Products (
    product_id INT PRIMARY KEY,
    name VARCHAR(100),
    description TEXT,
    price DECIMAL(10,2),
    quantity INT,
    supplier_id INT,
    FOREIGN KEY (supplier_id) REFERENCES Suppliers(supplier_id)
);

-- Inserting the values into Products table
INSERT INTO Products (product_id, name, description, price,
quantity, supplier_id) VALUES
(1, 'LED Monitor 24\"', 'Full HD LED Monitor', 120.00, 25, 1),
(2, 'Wireless Mouse', 'Ergonomic mouse', 15.00, 100, 2),
(3, 'Mechanical Keyboard', 'RGB backlit keyboard', 50.00, 40, 2),
(4, 'Laptop Backpack', 'Waterproof laptop bag', 30.00, 60, 3),
```

```
(5, 'USB-C Hub', '5-in-1 USB-C Hub', 25.00, 75, 4),
(6, 'HDMI Cable', '2m 4K HDMI cable', 10.00, 150, 4),
(7, 'External Hard Drive', '1TB USB 3.0', 65.00, 35, 1),
(8, 'Wireless Keyboard', 'Slim wireless keyboard', 22.00, 80, 2),
(9, 'Laptop Stand', 'Adjustable aluminum stand', 18.00, 50, 3),
(10, 'Webcam', '1080p USB webcam', 35.00, 45, 1),
(11, 'Bluetooth Speaker', 'Portable speaker', 40.00, 55, 5),
(12, 'Power Bank', '10000mAh fast-charging', 28.00, 70, 6),
(13, 'Wireless Earbuds', 'Bluetooth earbuds', 60.00, 90, 5),
(14, 'Desk Lamp', 'LED with USB port', 20.00, 45, 3),
(15, 'Smartphone Stand', 'Adjustable desk holder', 12.00, 100, 4),
(16, 'Cooling Pad', 'Laptop cooling fan', 22.00, 60, 1),
(17, 'Flash Drive 64GB', 'USB 3.0 drive', 9.00, 200, 7),
(18, 'Portable SSD', '512GB high-speed SSD', 110.00, 25, 6),
(19, 'Gaming Headset', 'Surround sound headset', 55.00, 30, 5),
(20, 'Monitor Mount', 'Dual screen arm mount', 48.00, 20, 7);

-- Getting all the records from Products table
SELECT * from Products;


-- Creating the Inventory_Transactions Table
CREATE TABLE Inventory_Transactions (
    transaction_id INT PRIMARY KEY,
    product_id INT,
    transaction_type VARCHAR(10), -- 'purchase' or 'sale'
    quantity INT,
    transaction_date DATE,
    FOREIGN KEY (product_id) REFERENCES Products(product_id)
);

-- Inserting the values into Inventory_Transactions table
INSERT INTO Inventory_Transactions (transaction_id, product_id,
transaction_type, quantity, transaction_date) VALUES
(1, 1, 'purchase', 30, '2025-05-01'),
(2, 2, 'purchase', 120, '2025-05-02'),
(3, 3, 'sale', 10, '2025-05-03'),
(4, 4, 'purchase', 50, '2025-05-04'),
(5, 5, 'sale', 15, '2025-05-05'),
(6, 6, 'purchase', 100, '2025-05-06'),
(7, 1, 'sale', 5, '2025-05-07'),
(8, 7, 'purchase', 20, '2025-05-07'),
(9, 8, 'sale', 20, '2025-05-08'),
(10, 9, 'purchase', 50, '2025-05-08'),
(11, 10, 'sale', 5, '2025-05-09'),
(12, 3, 'sale', 5, '2025-05-10'),
(13, 6, 'sale', 20, '2025-05-11'),
(14, 2, 'sale', 40, '2025-05-12'),
(15, 5, 'purchase', 40, '2025-05-13'),
(16, 4, 'sale', 20, '2025-05-14'),
(17, 7, 'sale', 10, '2025-05-15'),
(18, 9, 'sale', 10, '2025-05-15'),
```

```
(19, 10, 'purchase', 30, '2025-05-16'),
(20, 1, 'purchase', 10, '2025-05-17'),
(21, 11, 'purchase', 60, '2025-05-17'),
(22, 12, 'purchase', 80, '2025-05-17'),
(23, 13, 'sale', 30, '2025-05-17'),
(24, 14, 'purchase', 50, '2025-05-17'),
(25, 15, 'sale', 25, '2025-05-17');

-- Getting all the records from Inventory_Transactions table
SELECT * from Inventory_Transactions;

-- Add a new product in the Products table
INSERT INTO Products (product_id, name, description, price,
quantity, supplier_id)
VALUES (21, 'Ergonomic Chair', 'Adjustable office chair', 150.00,
10, 3);

-- Add a new supplier in the Suppliers table
INSERT INTO Suppliers (supplier_id, name, contact_name, phone,
email)
VALUES (8, 'WorkGear Supplies', 'Helen Troy', '890-123-4567',
'helen@workgear.com');

-- Updating the price of a product in the Products Table
UPDATE Products
SET price = 130.00
WHERE product_id = 1;

SELECT * FROM Products;

-- Updating the supplier contact info in the Suppliers Table
UPDATE Suppliers
SET phone = '999-888-7777'
WHERE supplier_id = 2;

SELECT * FROM Suppliers;

-- Retrieving the Product Information with Supplier Information
SELECT p.product_id, p.name AS product_name, p.price, p.quantity,
s.name AS supplier_name
FROM Products AS p
JOIN Suppliers AS s ON p.supplier_id = s.supplier_id;

-- Retrieving the Inventory Transactions with Product Name
SELECT t.transaction_id, p.name AS product_name,
t.transaction_type, t.quantity, t.transaction_date
FROM Inventory_Transactions t
JOIN Products p ON t.product_id = p.product_id;

-- Filtering the Products supplied by 'CableTech'
SELECT p.name, p.price
FROM Products p
```

```sql
JOIN Suppliers s ON p.supplier_id = s.supplier_id
WHERE s.name = 'CableTech';

-- Sorting the Products by Price in the Descending order
SELECT name, price, quantity
FROM Products
ORDER BY price DESC;

-- Filtering the Transactions by Date Range
SELECT *
FROM Inventory_Transactions
WHERE transaction_date BETWEEN '2025-05-05' AND '2025-05-10';

-- Showing the Current Stock for Each Product available in the
data
SELECT p.product_id, p.name,
       SUM(CASE WHEN t.transaction_type = 'purchase' THEN
t.quantity
                WHEN t.transaction_type = 'sale' THEN -t.quantity
                ELSE 0 END) AS stock_adjustment,
       p.quantity AS recorded_quantity
FROM Products p
LEFT JOIN Inventory_Transactions t ON p.product_id = t.product_id
GROUP BY p.product_id, p.name, p.quantity;
```