

Database Design

Content:

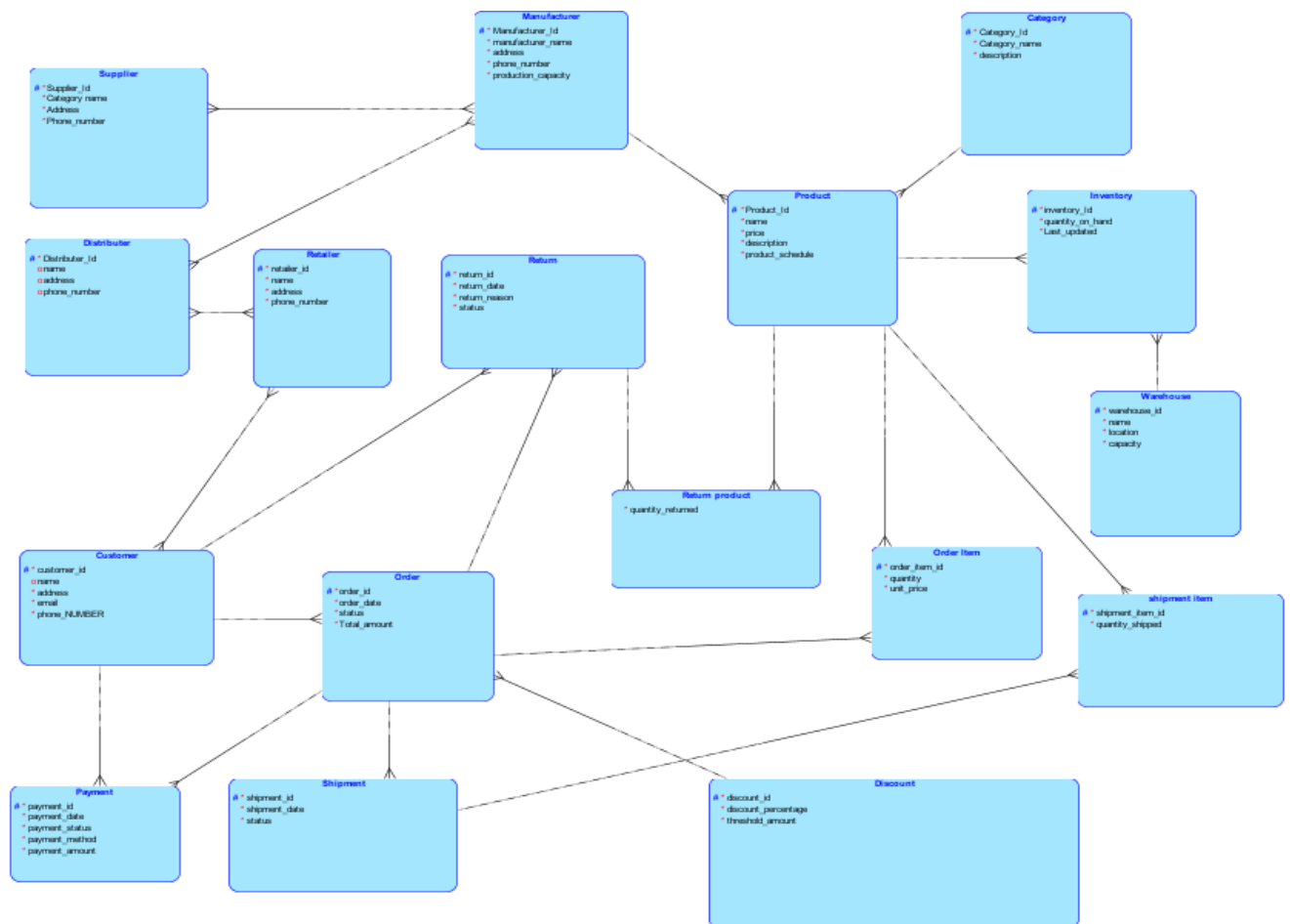
- I. Use Case**
- II. Conceptual Model Diagram**
- III. Logical model Diagram**
- IV. SQL queries**

Project Done by: Sara Alayan-Lynn Haidar

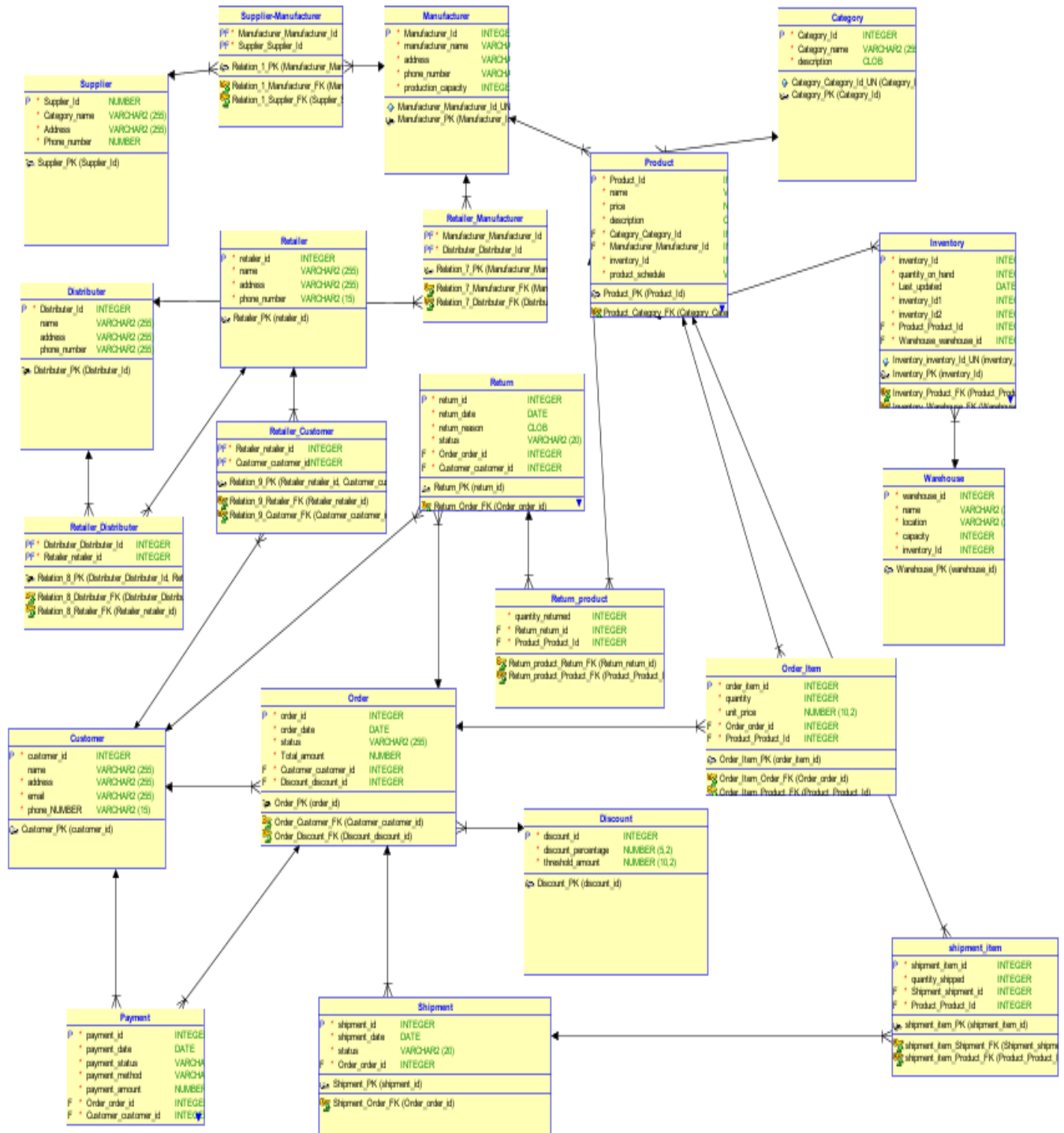
I. Supply Chain Management:

This database optimizes the supply chain by tracking inventory, suppliers, orders, shipments, and production schedules, improving efficiency and reducing costs. Keywords: Inventory, Supplier, Order, Shipment, Production Schedule, Logistics, Cost

II. Conceptual Model Diagram



III. Logical Model Diagram



IV. SQL QUERIES

```
CREATE TABLE Supplier (  
    supplier_id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(255) NOT NULL,  
    address VARCHAR(255) NOT NULL,  
    phone_number VARCHAR(15) NOT NULL  
);  
CREATE TABLE Manufacturer (  
    manufacturer_id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(255) NOT NULL,  
    address VARCHAR(255) NOT NULL,  
    phone_number VARCHAR(15) NOT NULL,  
    production_capacity INT NOT NULL  
);  
CREATE TABLE Supplier_Manufacturer (  
    supplier_id INT NOT NULL,  
    manufacturer_id INT NOT NULL,  
    PRIMARY KEY (supplier_id, manufacturer_id),  
    FOREIGN KEY (supplier_id) REFERENCES Supplier(supplier_id) ON DELETE  
CASCADE,  
    FOREIGN KEY (manufacturer_id) REFERENCES Manufacturer(manufacturer_id) ON  
DELETE CASCADE  
);  
CREATE TABLE Category (  
    category_id INT AUTO_INCREMENT PRIMARY KEY,  
    category_name VARCHAR(255) NOT NULL,  
    description TEXT  
);  
CREATE TABLE Product (  
    product_id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(255) NOT NULL,  
    description TEXT,  
    price DECIMAL(10, 2) NOT NULL,  
    manufacturer_id INT NOT NULL,  
    category_id INT NOT NULL,  
    FOREIGN KEY (manufacturer_id) REFERENCES Manufacturer(manufacturer_id) ON  
DELETE CASCADE,  
    FOREIGN KEY (category_id) REFERENCES Category(category_id) ON DELETE  
CASCADE
```

);

```
CREATE TABLE Warehouse (  
    warehouse_id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(255) NOT NULL,  
    location VARCHAR(255) NOT NULL,  
    capacity INT NOT NULL
```

);

```
CREATE TABLE Inventory (  
    inventory_id INT AUTO_INCREMENT PRIMARY KEY,  
    product_id INT NOT NULL,  
    warehouse_id INT NOT NULL,  
    quantity_on_hand INT NOT NULL,  
    last_updated DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE  
CURRENT_TIMESTAMP,  
    FOREIGN KEY (product_id) REFERENCES Product(product_id) ON DELETE CASCADE,  
    FOREIGN KEY (warehouse_id) REFERENCES Warehouse(warehouse_id) ON DELETE  
CASCADE
```

);

```
CREATE TABLE Distributor (  
    distributor_id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(255) NOT NULL,  
    address VARCHAR(255) NOT NULL,  
    phone_number VARCHAR(15) NOT NULL
```

);

select * from distributor;

```
CREATE TABLE Retailer (  
    retailer_id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(255) NOT NULL,  
    address VARCHAR(255) NOT NULL,  
    phone_number VARCHAR(15) NOT NULL
```

);

```
CREATE TABLE Distributor_Retailer (  
    distributor_id INT NOT NULL,  
    retailer_id INT NOT NULL,  
    PRIMARY KEY (distributor_id, retailer_id),  
    FOREIGN KEY (distributor_id) REFERENCES Distributor(distributor_id) ON DELETE  
CASCADE,  
    FOREIGN KEY (retailer_id) REFERENCES Retailer(retailer_id) ON DELETE CASCADE
```

);

```
CREATE TABLE Customer (  
    customer_id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(255) NOT NULL,  
    address VARCHAR(255) NOT NULL,  
    phone_number VARCHAR(15) NOT NULL
```

```

customer_id INT AUTO_INCREMENT PRIMARY KEY,
name VARCHAR(255) NOT NULL,
address VARCHAR(255) NOT NULL,
email VARCHAR(255) NOT NULL,
phone_number VARCHAR(15) NOT NULL
);
CREATE TABLE Retailer_Customer (
    retailer_id INT NOT NULL,
    customer_id INT NOT NULL,
    PRIMARY KEY (retailer_id, customer_id),
    FOREIGN KEY (retailer_id) REFERENCES Retailer(retailer_id) ON DELETE CASCADE,
    FOREIGN KEY (customer_id) REFERENCES Customer(customer_id) ON DELETE
    CASCADE
);
CREATE TABLE `Order` (
    order_id INT AUTO_INCREMENT PRIMARY KEY,
    customer_id INT NOT NULL,
    order_date DATETIME DEFAULT CURRENT_TIMESTAMP,
    status VARCHAR(20) NOT NULL, -- e.g., pending, shipped, delivered
    total_amount DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (customer_id) REFERENCES Customer(customer_id) ON DELETE
    CASCADE
);
CREATE TABLE Discount (
    discount_id INT AUTO_INCREMENT PRIMARY KEY,
    order_id INT NOT NULL,
    discount_type VARCHAR(50), -- e.g., 'percentage', 'fixed'
    discount_value DECIMAL(10, 2), -- e.g., 10 for percentage or $10 for fixed
    description VARCHAR(255),
    FOREIGN KEY (order_id) REFERENCES `Order` (order_id) ON DELETE CASCADE
);
ALTER TABLE `Order`
ADD COLUMN discounted_amount DECIMAL(10, 2) DEFAULT NULL;
CREATE TABLE Order_Item (
    order_item_id INT AUTO_INCREMENT PRIMARY KEY,
    order_id INT NOT NULL,
    product_id INT NOT NULL,
    quantity INT NOT NULL,
    unit_price DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (order_id) REFERENCES `Order` (order_id) ON DELETE CASCADE,
    FOREIGN KEY (product_id) REFERENCES Product(product_id) ON DELETE CASCADE

```

```

);
CREATE TABLE Payment (
    payment_id INT AUTO_INCREMENT PRIMARY KEY,
    order_id INT NOT NULL,
    customer_id INT NOT NULL,
    payment_date DATETIME DEFAULT CURRENT_TIMESTAMP,
    payment_amount DECIMAL(10, 2) NOT NULL,
    payment_status VARCHAR(20) NOT NULL, -- e.g., successful, failed
    payment_method VARCHAR(50) NOT NULL,
    FOREIGN KEY (order_id) REFERENCES `Order` (order_id) ON DELETE CASCADE,
    FOREIGN KEY (customer_id) REFERENCES Customer(customer_id) ON DELETE
CASCADE
);
CREATE TABLE Shipment (
    shipment_id INT AUTO_INCREMENT PRIMARY KEY,
    order_id INT NOT NULL,
    shipment_date DATETIME DEFAULT CURRENT_TIMESTAMP,
    status VARCHAR(20) NOT NULL, -- e.g., in transit, delivered
    FOREIGN KEY (order_id) REFERENCES `Order` (order_id) ON DELETE CASCADE
);
CREATE TABLE Shipment_Item (
    shipment_item_id INT AUTO_INCREMENT PRIMARY KEY,
    shipment_id INT NOT NULL,
    product_id INT NOT NULL,
    quantity_shipped INT NOT NULL,
    FOREIGN KEY (shipment_id) REFERENCES Shipment(shipment_id) ON DELETE
CASCADE,
    FOREIGN KEY (product_id) REFERENCES Product(product_id) ON DELETE CASCADE
);
CREATE TABLE Returnn (
    return_id INT AUTO_INCREMENT PRIMARY KEY,
    order_id INT NOT NULL,
    customer_id INT NOT NULL,
    return_date DATETIME DEFAULT CURRENT_TIMESTAMP,
    return_reason TEXT NOT NULL,
    status VARCHAR(20) NOT NULL, -- e.g., pending, approved, rejected
    FOREIGN KEY (order_id) REFERENCES `Order` (order_id) ON DELETE CASCADE,
    FOREIGN KEY (customer_id) REFERENCES Customer(customer_id) ON DELETE
CASCADE
);
CREATE TABLE Return_Product (

```

```

    return_id INT NOT NULL,
    product_id INT NOT NULL,
    quantity_returned INT NOT NULL,
    PRIMARY KEY (return_id, product_id),
    FOREIGN KEY (return_id) REFERENCES Returnn(return_id) ON DELETE CASCADE,
    FOREIGN KEY (product_id) REFERENCES Product(product_id) ON DELETE CASCADE
);
CREATE TABLE Manufacturer_Distributor (
    manufacturer_id INT NOT NULL,
    distributor_id INT NOT NULL,
    partnership_date DATE,
    PRIMARY KEY (manufacturer_id, distributor_id),
    FOREIGN KEY (manufacturer_id) REFERENCES Manufacturer(manufacturer_id) ON
DELETE CASCADE,
    FOREIGN KEY (distributor_id) REFERENCES Distributor(distributor_id) ON DELETE
CASCADE
);

```

```

INSERT INTO Customer (customer_id, name, address, email, phone_number)
VALUES
(1, 'sara alayan', 'street,baabda, lebanon', 'sara@gmail.com', '81123456'),
(2, 'Jane Smith', 'street, choueifat, lebanon', 'lynnh@gmail.com', '81123123');

```

```

INSERT INTO Category (category_id, category_name, description)
VALUES
(1, 'Electronics', 'Devices and gadgets'),
(2, 'Furniture', 'Home and office furniture');

```

```

INSERT INTO Manufacturer (manufacturer_id, name, address, phone_number,
production_capacity)
VALUES
(1, 'Tech', 'street, City, Country', '05434343', 5000),
(2, 'FurniMakers', 'street, City, Country', '01424242', 3000);

```

```

INSERT INTO Product (product_id, name, description, price, manufacturer_id,
category_id)
VALUES
(1, 'Smartphone', 'Latest model with advanced features', 700.00, 1, 1),
(2, 'Laptop', 'High-performance laptop for professionals', 1299.99, 1, 1),
(3, 'Office Chair', 'Ergonomic office chair', 199.99, 2, 2);

```



```
INSERT INTO Warehouse (warehouse_id, name, location, capacity)
VALUES
(1, 'Main Warehouse', 'City, Country', 10000),
(2, 'Secondary Warehouse', 'Suburb, Country', 5000);
INSERT INTO Inventory (inventory_id, product_id, quantity_on_hand, last_updated,
warehouse_id)
VALUES
(1, 1, 50, NOW(), 1),
(2, 2, 30, NOW(), 1),
(3, 3, 100, NOW(), 2);
```

```
INSERT INTO Distributor (distributor_id, name, address, phone_number)
VALUES
(1, 'Global Distributors', '123 Distribute St, City, Country', '70123123'),
(2, 'FastShip', 'street, City, Country', '70123456');
```

```
INSERT INTO Retailer (retailer_id, name, address, phone_number)
VALUES
(1, 'Retailer A', 'street st, City, Country', '05414243'),
(2, 'Retailer B', 'street st, City, Country', '01414243');
```

```
INSERT INTO `Order` (order_id, customer_id, order_date, status, total_amount)
VALUES
(1, 1, NOW(), 'pending', 899.98),
(2, 2, NOW(), 'shipped', 1399.99);
```

```
INSERT INTO Order_Item (order_item_id, order_id, product_id, quantity, unit_price)
VALUES
(1, 1, 1, 1, 699.99),
(2, 1, 3, 1, 199.99),
(3, 2, 2, 1, 1299.99);
```

```
INSERT INTO Payment (payment_id, order_id, customer_id, payment_date,
payment_amount, payment_status, payment_method)
VALUES
(1, 1, 1, NOW(), 899.98, 'successful', 'Credit Card'),
(2, 2, 2, NOW(), 1399.99, 'successful', 'OMT');
```

```
INSERT INTO Shipment (shipment_id, order_id, shipment_date, status)
VALUES
```

```
(1, 1, NOW(), 'shipped'),  
(2, 2, NOW(), 'in transit');
```

```
INSERT INTO Shipment_Item (shipment_item_id, shipment_id, product_id,  
quantity_shipped)  
VALUES  
(1, 1, 1, 1),  
(2, 1, 3, 1),  
(3, 2, 2, 1);
```

```
INSERT INTO Returnn (return_id, order_id, customer_id, return_date, return_reason,  
status)  
VALUES  
(1, 1, 1, NOW(), 'Defective product', 'pending'),  
(2, 2, 2, NOW(), 'Wrong item shipped', 'approved');
```

```
INSERT INTO Return_Product (return_id, product_id, quantity_returned)  
VALUES  
(1, 1, 1),  
(2, 2, 1);  
INSERT INTO Discount (order_id, discount_type, discount_value, description)  
VALUES  
(1, 'percentage', 10, '10% discount for holiday sale'),  
(2, 'fixed', 20, 'Flat $20 discount for promotion');
```

```
INSERT INTO Manufacturer_Distributor (manufacturer_id, distributor_id,  
partnership_date)  
VALUES  
(1, 1, '2022-01-15'),  
(1, 2, '2023-03-10'),  
(2, 1, '2021-07-20');
```

```
ALTER TABLE Product  
ADD COLUMN product_schedule VARCHAR(255) DEFAULT NULL;
```

```
UPDATE Product  
SET product_schedule = 'Monthly Production'  
WHERE product_id = 1;
```

```
UPDATE Product  
SET product_schedule = 'Weekly Production'
```

```
WHERE product_id = 2;
```

```
UPDATE Product  
SET product_schedule = 'On-Demand Production'  
WHERE product_id = 3;
```

```
SELECT p.product_id, p.name, p.description, p.price  
FROM Product p  
JOIN Category c ON p.category_id = c.category_id  
WHERE c.category_name = 'Electronics';
```

```
SELECT o.customer_id, SUM(o.total_amount) AS total_spent  
FROM `Order` o  
GROUP BY o.customer_id;
```

```
SELECT p.name, i.quantity_on_hand, w.name AS warehouse  
FROM Inventory i  
JOIN Product p ON i.product_id = p.product_id  
JOIN Warehouse w ON i.warehouse_id = w.warehouse_id  
WHERE p.product_id = 1;
```

```
SELECT o.order_id, o.order_date, o.status AS order_status, p.payment_status  
FROM `Order` o  
JOIN Payment p ON o.order_id = p.order_id;
```

```
SELECT DISTINCT p.name  
FROM Returnn r  
JOIN Return_Product rp ON r.return_id = rp.return_id  
JOIN Product p ON rp.product_id = p.product_id;
```

```
SELECT s.shipment_id, SUM(si.quantity_shipped) AS total_shipped  
FROM Shipment s  
JOIN Shipment_Item si ON s.shipment_id = si.shipment_id  
GROUP BY s.shipment_id;
```

```
SELECT DISTINCT o.customer_id, c.name  
FROM `Order` o  
JOIN Customer c ON o.customer_id = c.customer_id  
WHERE o.total_amount > 1000;
```

```
SELECT o.order_id, o.order_date, oi.quantity, oi.unit_price
FROM `Order` o
JOIN Order_Item oi ON o.order_id = oi.order_id
WHERE oi.product_id = 2;
```

```
SELECT product_schedule, COUNT(*) AS product_count
FROM Product
GROUP BY product_schedule;
```

```
SELECT d.name AS Distributor
FROM Distributor d
JOIN Manufacturer_Distributor md ON d.distributor_id = md.distributor_id
JOIN Manufacturer m ON md.manufacturer_id = m.manufacturer_id
WHERE m.name = 'Samsung';
```