CREATE DATABASE if not exists online\_Store;

USE online\_Store;

-- Users Table

CREATE TABLE users (

user\_id INT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(100),

email VARCHAR(100) UNIQUE,

password VARCHAR(100),

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

-- Products Table

CREATE TABLE products (

product\_id INT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(100),

description TEXT,

price DECIMAL(10, 2),

stock INT,

category VARCHAR(50)

);

-- Orders Table

CREATE TABLE orders (

order\_id INT PRIMARY KEY AUTO\_INCREMENT,

user\_id INT,

order\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

total\_amount DECIMAL(10, 2),

status VARCHAR(20) DEFAULT 'Pending',

FOREIGN KEY (user\_id) REFERENCES users(user\_id)

);

-- Order Items Table

CREATE TABLE order\_items (

order\_item\_id INT PRIMARY KEY AUTO\_INCREMENT,

order\_id INT,

product\_id INT,

quantity INT,

price DECIMAL(10, 2),

FOREIGN KEY (order\_id) REFERENCES orders(order\_id),

FOREIGN KEY (product\_id) REFERENCES products(product\_id)

);

-- Payments Table (Optional)

CREATE TABLE payments (

payment\_id INT PRIMARY KEY AUTO\_INCREMENT,

order\_id INT,

payment\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

amount DECIMAL(10, 2),

method VARCHAR(50),

status VARCHAR(20),

FOREIGN KEY (order\_id) REFERENCES orders(order\_id)

);

-- Insert Users

INSERT INTO users (name, email, password) VALUES

('priya', 'priya@example.com', 'createpassword1'),

('rameshT', 'rameshT@example.com', 'createpassword2');

-- Insert Products

INSERT INTO products (name, description, price, stock, category) VALUES

('Laptop', '14 inch laptop with 16GB RAM', 40000, 10, 'Electronics'),

('Wireless Mouse', ' wireless mike', 500, 50, 'Accessories'),

('Coffee Mug', 'Ceramic mug - 350ml', 250, 100, 'Home');

-- Create Order

INSERT INTO orders (user\_id, total\_amount, status) VALUES

(1, 42000, 'Processing');

-- Order Items

INSERT INTO order\_items (order\_id, product\_id, quantity, price) VALUES

(1, 1, 1, 40000),

(1, 2, 1, 100 ),

(1, 3, 1, 250);

-- Payment

INSERT INTO payments (order\_id, amount, method, status) VALUES

(1, 42000, 'Credit Card', 'Completed');

-- List all users

SELECT \* FROM users;

-- List all products in stock

SELECT \* FROM products WHERE stock > 0;

-- Get order details with items

SELECT

o.order\_id, u.name AS customer\_name, p.name AS product\_name, oi.quantity, oi.price

FROM orders o

JOIN users u ON o.user\_id = u.user\_id

JOIN order\_items oi ON o.order\_id = oi.order\_id

JOIN products p ON oi.product\_id = p.product\_id

WHERE o.order\_id = 1;

-- Total sales

SELECT SUM(amount) AS total\_sales FROM payments WHERE status = 'Completed';