**E-Commerce Database Schema**

**Drop tables if they already exist (for reset)**

DROP TABLE IF EXISTS order\_items;

DROP TABLE IF EXISTS orders;

DROP TABLE IF EXISTS products;

DROP TABLE IF EXISTS users;

**1. User Table**

CREATE TABLE users (

id BIGINT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(100) NOT NULL,

email VARCHAR(100) UNIQUE NOT NULL,

password VARCHAR(255) NOT NULL,

role ENUM('CUSTOMER', 'ADMIN') DEFAULT 'CUSTOMER',

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

**Notes**:

* Email is unique.
* Role differentiates between customers and admins.

**2. Product Table**

CREATE TABLE products (

id BIGINT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(100) NOT NULL,

description TEXT,

price DECIMAL(10,2) NOT NULL CHECK (price > 0),

stock INT DEFAULT 0,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

**Notes:**

* Price must be positive.
* Description is optional.

**3. Order Table**

CREATE TABLE orders (

id BIGINT PRIMARY KEY AUTO\_INCREMENT,

user\_id BIGINT NOT NULL,

order\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

status ENUM('PENDING', 'PROCESSING', 'SHIPPED', 'DELIVERED', 'CANCELLED') DEFAULT 'PENDING',

total\_amount DECIMAL(10,2) NOT NULL CHECK (total\_amount >= 0),

FOREIGN KEY (user\_id) REFERENCES users(id) ON DELETE CASCADE

);

**Notes**:

* Links each order to a user.
* Tracks status of the order.

**4. OrderItem Table**

CREATE TABLE order\_items (

id BIGINT PRIMARY KEY AUTO\_INCREMENT,

order\_id BIGINT NOT NULL,

product\_id BIGINT NOT NULL,

quantity INT NOT NULL CHECK (quantity > 0),

unit\_price DECIMAL(10,2) NOT NULL CHECK (unit\_price >= 0),

FOREIGN KEY (order\_id) REFERENCES orders(id) ON DELETE CASCADE,

FOREIGN KEY (product\_id) REFERENCES products(id) ON DELETE CASCADE

);

**Notes**:

* Stores individual product entries per order.
* unit\_price is recorded at the time of order (in case product price changes later).
* Deleting an order will delete its order items.

**Entity Relationships Summary**

**User (1) ─────< Order (N)**

**Order (1) ─────< OrderItem (N)**

**OrderItem (N) ─────> Product (1)**

**Insert Users**

INSERT INTO users (name, email, password, role) VALUES

('Alice Johnson', 'alice@example.com', 'pass123', 'CUSTOMER'),

('Bob Smith', 'bob@example.com', 'pass456', 'ADMIN');

**Insert Products**

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

('Laptop', '14 inch Laptop with 16GB RAM', 999.99, 50),

('Wireless Mouse', 'Ergonomic wireless mouse', 25.00, 200),

('Monitor', '24 inch HD monitor', 150.75, 75);

**Insert Orders**

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

(1, 'PENDING', 1175.74),

(1, 'PROCESSING', 50.00);

**Insert OrderItems**

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

(1, 1, 1, 999.99),

(1, 3, 1, 150.75),

(2, 2, 2, 25.00);