**<<数据库系统>>**

**实验报告一**

**(2023年度春季学期)**

实验二

# 实验目的

在熟练掌握MySQL基本命令、SQL语言以及用C语言编写MySQL操作程序的基础上，学习简单数据库系统的设计方法，包括数据库概要设计、逻辑设计。

# 实验环境

Windows 10操作系统、MySQL8.0.32版本，Microsoft Visual C++编译器。

# 实验过程及结果

## 3.1构建概念数据库

3.1.1 实体：

（1）用户（User）

（2）订单（Order）

（3）产品（Product）

（4）供应商（Supplier）

（5）支付（Payment）

（6）配送员（Delivery Person）

（7）地址（Address）

（8）评论（Comment）

3.1.2 联系：

（1）用户和订单之间的一对多联系

（2）订单和产品之间的多对多联系

（3）产品和供应商之间的多对一联系

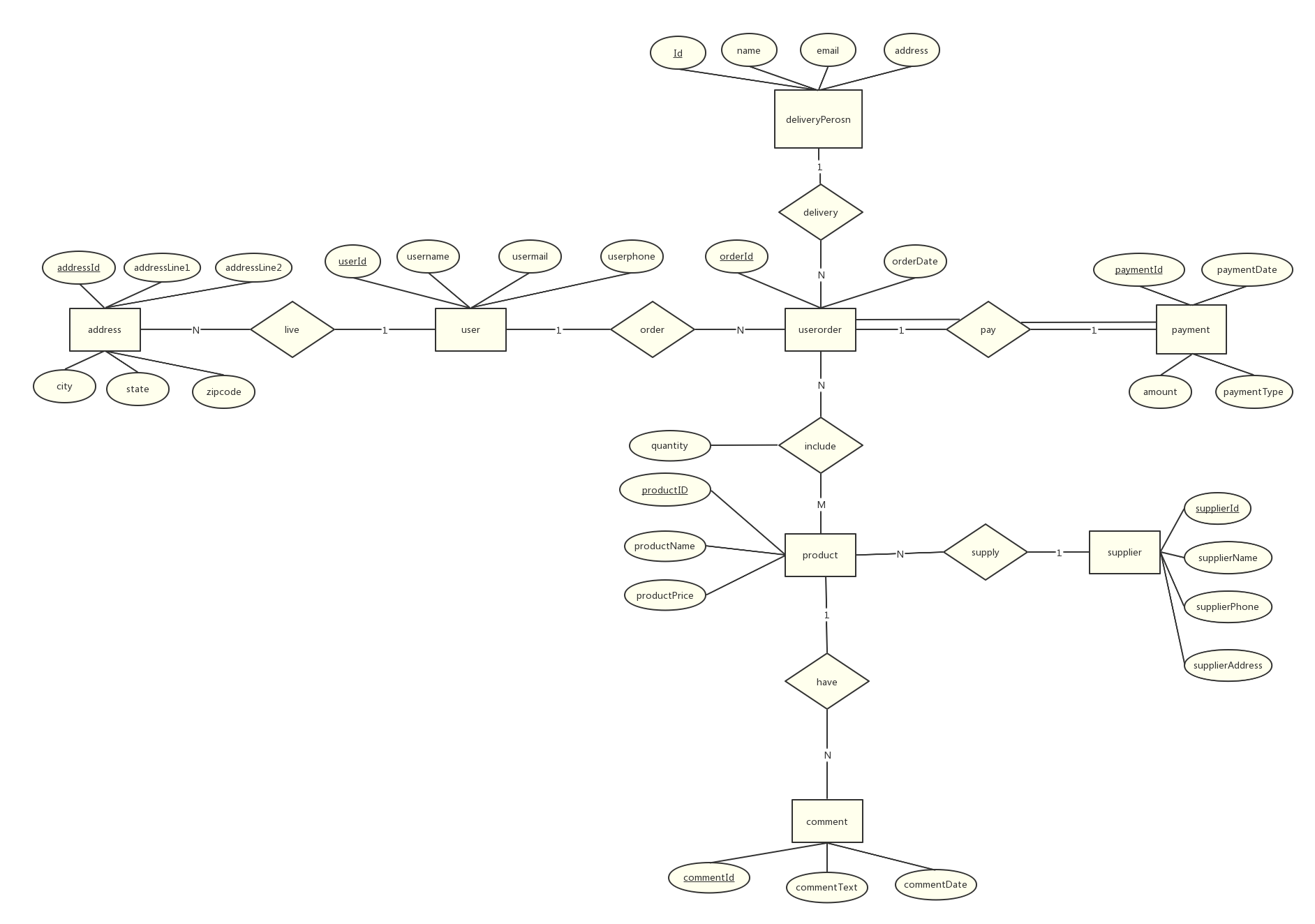
（4）订单和支付之间的一对一联系

（5）订单和配送员之间的多对一联系

（6）用户和地址之间的一对多联系

（7）产品和评论之间的一对多联系

### 3.1.3 绘制E-R图



## 3.2 设计逻辑数据库

### 3.2.1 初始关系数据库模式

User (UserID, UserName, UserEmail, UserPhone)

主键：UserID

UserOrder (OrderID, OrderDate, UserID, DeliveryPersonID, PaymentID)

主键：OrderID

外键：UserID（参照User关系的主键），DeliveryPersonID（参照Delivery Person关系的主键），PaymentID（参照Payment关系的主键）

OrderProduct (OrderID, ProductID, Quantity)

主键：(OrderID, ProductID)

外键：OrderID（参照Order关系的主键），ProductID（参照Product关系的主键）

Product (ProductID, ProductName, Price, SupplierID)

主键：ProductID

外键：SupplierID（参照Supplier关系的主键）

Supplier (SupplierID, SupplierName, SupplierEmail, SupplierPhone)

主键：SupplierID

Payment (PaymentID, PaymentDate, Amount, PaymentType)

主键：PaymentID

DeliveryPerson (DeliveryPersonID, DeliveryPersonName, DeliveryPersonEmail, DeliveryPersonPhone)

主键：DeliveryPersonID

Address (AddressID, UserID, AddressLine1, AddressLine2, City, State, ZipCode)

主键：AddressID

外键：UserID（参照User关系的主键）

Comment (CommentID, ProductID, CommentText, CommentDate)

主键：CommentID

外键：ProductID（参照Product关系模式的主键）

### 3.2.2定义关系上的完整性和安全性约束

• 主码约束：主码不允许重复

• 外码约束：满足参照完整性

• 空值约束：本关系数据库系统部分属性值非空

### 3.2.3 子模式定义

（1）supplierProduct视图：提供每个供应商及其提供的产品列表。

CREATE VIEW SupplierProduct AS

SELECT S.SupplierID, S.SupplierName, P.ProductID, P.ProductName, P.Price FROM Supplier S

JOIN Product P ON S.SupplierID = P.SupplierID;

1. User表的username列创建索引

CREATE INDEX idx\_User\_Name ON user(username);

## 3.3数据库系统实现

### 3.3.1 关系表建立

-- 创建 User 表

CREATE TABLE User (

UserID INT PRIMARY KEY,

UserName VARCHAR(50),

UserEmail VARCHAR(100),

UserPhone VARCHAR(20)

);

-- 创建 UserOrder 表

CREATE TABLE UserOrder (

OrderID INT PRIMARY KEY,

OrderDate DATE,

UserID INT,

DeliveryPersonID INT,

PaymentID INT,

FOREIGN KEY (UserID) REFERENCES User(UserID),

FOREIGN KEY (DeliveryPersonID) REFERENCES DeliveryPerson(DeliveryPersonID),

FOREIGN KEY (PaymentID) REFERENCES Payment(PaymentID)

);

-- 创建 OrderProduct 表

CREATE TABLE OrderProduct (

OrderID INT,

ProductID INT,

Quantity INT,

PRIMARY KEY (OrderID, ProductID),

FOREIGN KEY (OrderID) REFERENCES UserOrder(OrderID),

FOREIGN KEY (ProductID) REFERENCES Product(ProductID)

);

-- 创建 Product 表

CREATE TABLE Product (

ProductID INT PRIMARY KEY,

ProductName VARCHAR(50),

Price DECIMAL(10,2),

SupplierID INT,

FOREIGN KEY (SupplierID) REFERENCES Supplier(SupplierID)

);

-- 创建 Supplier 表

CREATE TABLE Supplier (

SupplierID INT PRIMARY KEY,

SupplierName VARCHAR(50),

SupplierEmail VARCHAR(100),

SupplierPhone VARCHAR(20)

);

-- 创建 Payment 表

CREATE TABLE Payment (

PaymentID INT PRIMARY KEY,

PaymentDate DATE,

Amount DECIMAL(10,2),

PaymentType VARCHAR(50)

);

-- 创建 DeliveryPerson 表

CREATE TABLE DeliveryPerson (

DeliveryPersonID INT PRIMARY KEY,

DeliveryPersonName VARCHAR(50),

DeliveryPersonEmail VARCHAR(100),

DeliveryPersonPhone VARCHAR(20)

);

-- 创建 Address 表

CREATE TABLE Address (

AddressID INT PRIMARY KEY,

UserID INT,

AddressLine1 VARCHAR(100),

AddressLine2 VARCHAR(100),

City VARCHAR(50),

State VARCHAR(50),

ZipCode VARCHAR(10),

FOREIGN KEY (UserID) REFERENCES User(UserID)

);

-- 创建 Comment 表

CREATE TABLE Comment (

CommentID INT PRIMARY KEY,

ProductID INT,

CommentText VARCHAR(1000),

CommentDate DATE,

FOREIGN KEY (ProductID) REFERENCES Product(ProductID)

);

### 3.3.2 视图与索引建立

Mysql语句已在上文给出；

### 3.3.3对数据库查询

(1)查询订单数量超过两单的用户姓名和email

SELECT UserName, UserEmail FROM User WHERE UserID = (SELECT UserID FROM userorder GROUP BY UserID HAVING COUNT(\*)>=2)"

## 使用了嵌套查询和分组查询

（2）展示各个供货商提供的货物数量，平均价格，最高价格

SELECT S.SupplierName, COUNT(\*) AS NumOfProducts, AVG(P.Price) AS AvgPrice, MAX(P.Price) AS MaxPrice FROM Supplier S JOIN Product P ON S.SupplierID = P.SupplierID GROUP BY S.SupplierName

INSERT INTO Comment (ProductID, CommentText, CommentDate) VALUES (NULL, '1', NOW())

### 3.3.4 插入删除操作及完整性约束

(1)向comment关系表中插入元组。由于ProductId为Comment关系的主键，所以ProductId必须非空，则上述操作会失败

INSERT INTO Comment (ProductID, CommentText, CommentDate) VALUES (NULL, '1', NOW())



(2)在product关系中删除productID为1的元组。由于存在外键约束，上述操作会失败。

delete from product where productid = 1

## 

## 四、实验心得

1.E-R图的绘制较为繁琐，但也很重要，需要认真绘制。

2.将E-R图转为关系模式的过程还不够熟练。

3.不太熟悉前端设计，以后可以进一步加强。