|  |
| --- |
|  |

**数据库原理与实践期末项目**

**学号：**

**姓名：**



数据库原理与实践期末项目

20 24 —20 25 学年 一 学期

|  |  |
| --- | --- |
| 课程名称： | 数据库原理及应用 |
| 专业学院： | 现代技术学院 |
| 专业班级： | 计科B2215 |
| 学 号： |  |
| 姓 名： |  |
| 指导教师： | 李小蓉 |

目 录

[1、 项目背景及需求分析 1](#_Toc153742448)

[1.1项目背景 1](#_Toc153742449)

[1.2需求分析 1](#_Toc153742450)

[2、 概念结构设计 2](#_Toc153742451)

[2.1抽象出系统实体 2](#_Toc153742452)

[2.2局部E-R图 2](#_Toc153742453)

[2.3全局E-R图 2](#_Toc153742454)

[3、 逻辑结构设计 3](#_Toc153742455)

[3.1简单说明 3](#_Toc153742456)

[3.2关系模式 3](#_Toc153742457)

[3.3表结构 3](#_Toc153742458)

[4、 创建数据库及SQL相关操作 4](#_Toc153742459)

[4.1 创建数据库 4](#_Toc153742460)

[4.1.1 创建表 4](#_Toc153742461)

[4.1.2 添加记录 4](#_Toc153742462)

[4.1.3 修改记录 5](#_Toc153742463)

[4.2 删除记录 5](#_Toc153742464)

[4.3查询问题及查询结果 5](#_Toc153742465)

[4.3.1 选择列 5](#_Toc153742466)

[4.3.2函数使用 5](#_Toc153742467)

[4.3.3 LIKE子句 6](#_Toc153742468)

[4.3.4 条件查询 6](#_Toc153742469)

[4.4.5多表查询 6](#_Toc153742470)

[4.4.6子查询 6](#_Toc153742471)

[4.5建立视图 7](#_Toc153742472)

[4.5.1 定义视图 7](#_Toc153742473)

[4.5.2利用视图插入数据 7](#_Toc153742474)

[5、建立存储过程及函数 8](#_Toc153742475)

[5.1创建存储过程 8](#_Toc153742476)

[5.2存储过程应用 8](#_Toc153742477)

[6、建立触发器 9](#_Toc153742478)

[7、项目设计总结 10](#_Toc153742479)

# 1、 项目背景及需求分析

## 1.1项目背景

随着信息技术的迅猛发展，报刊管理系统作为一种现代化管理工具，对报刊出版社和发行机构具有重要意义。传统的报刊管理方式通常依赖于人工操作和纸质记录，存在着诸多问题，如信息不准确、处理效率低下、数据难以统计和分析等。而引入计算机技术和数据库管理系统，可以极大地提高报刊管理的效率和准确性，实现对报刊的全面管理和控制。

首先，报刊管理系统能够提高报刊管理的效率。传统的报刊管理需要大量的人工操作和繁琐的记录工作，容易出现错误和遗漏。而通过引入计算机技术，可以实现自动化的数据录入、处理和存储，大大减少了人力成本和时间消耗。管理员可以通过系统快速地查询和更新报刊信息，提高工作效率。

其次，报刊管理系统能够提高报刊管理的准确性。在传统的管理方式下，由于人工操作的不确定性和繁重的工作负担，容易出现信息错误和数据不一致的问题。而通过引入数据库管理系统，可以确保数据的一致性和准确性。管理员可以通过系统对报刊信息进行规范化和标准化的管理，避免了人为因素对数据的影响，提高了管理的准确性。

此外，报刊管理系统还能够实现对报刊的全面管理和控制。通过系统的设计，可以对报刊的出版、订阅和发行等环节进行全面的监控和管理。管理员可以及时获取报刊的销售情况、订阅情况和发行情况等数据，进行业务分析和决策支持。同时，系统还可以提供统计和查询功能，帮助管理员快速获取所需信息，提高管理效率。

总之，报刊管理系统的引入对报刊出版社和发行机构具有重要意义。它可以提高报刊管理的效率和准确性，实现对报刊的全面管理和控制。通过系统的帮助，报刊出版社和发行机构能够更好地应对市场竞争，提升服务质量，满足读者的需求，从而提升其竞争力和市场地位。

## 1.2需求分析

• 报刊信息管理：包括报刊的基本信息（如名称、出版社、发行周期等）、订阅信息和发行信息的管理。

• 订阅管理：管理订阅者的信息，包括个人信息、订阅期限和付费情况等。

• 发行管理：管理报刊的发行情况，包括发行日期、发行数量和发行地点等。

• 统计与查询：对报刊信息、订阅信息和发行信息进行统计和查询，以便进行业务分析和决策支持。

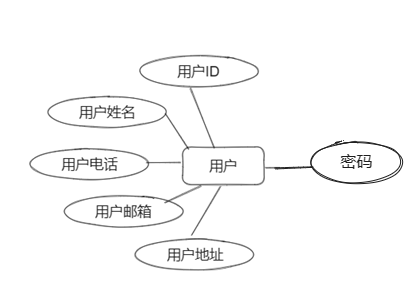
# 2、 概念结构设计

## 2.1抽象出系统实体

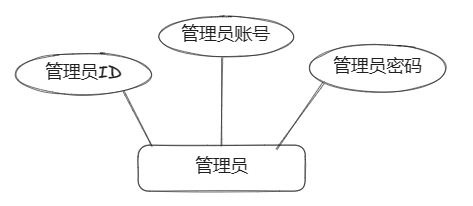
实体包括：用户、管理员、报刊、

## 2.2局部E-R图

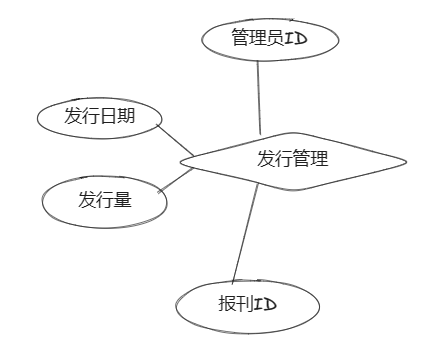
用户：



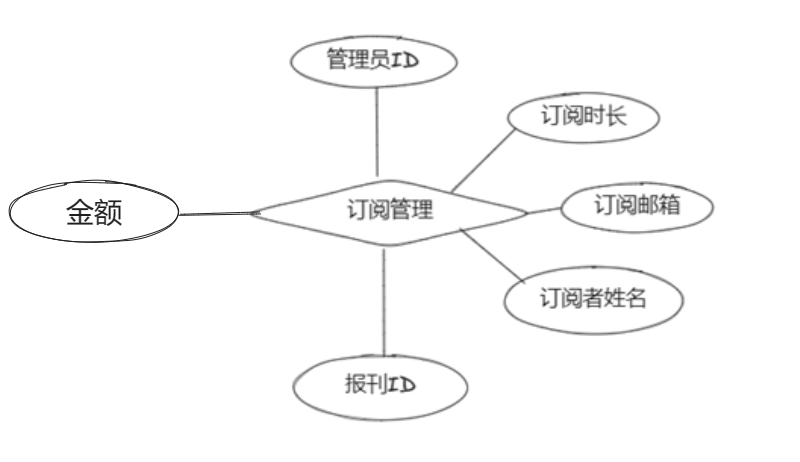
管理员



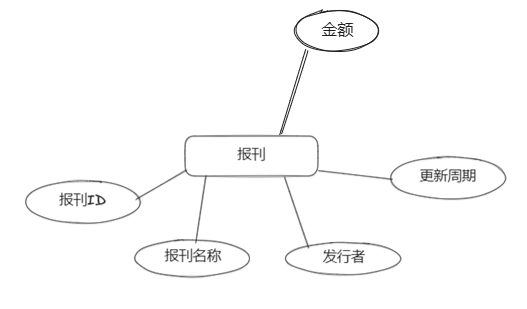
发行管理



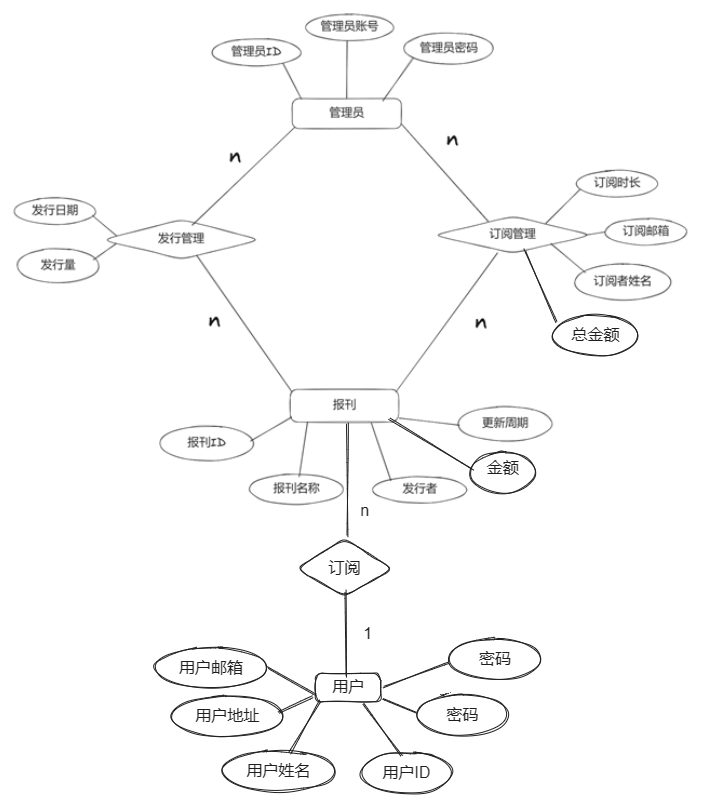
订阅管理



报刊



## 2.3全局E-R图



# 3、 逻辑结构设计

## 3.1简单说明

在第三范式关系模式中，每个关系模式中的字段都是原子的，消除了冗余和数据依赖性。报刊信息表（newspaper）中的发布周期（publication\_cycle）字段被移除，因为它可以通过订阅信息表（subscription）和发行信息表（distribution）进行关联查询获得。这样可以减少数据冗余并提高数据的一致性。

## 3.2关系模式

user(user\_id, user\_name, user\_address, user\_phone, user\_email)

admin(admin\_id, username, password)

newspaper(newspaper\_id, newspaper\_name, publisher, publication\_cycle)

subscription(subscription\_id, subscriber\_name, contact\_info, subscription\_period, user\_id)

distribution(distribution\_id, newspaper\_id, distribution\_date, distribution\_quantity)

## 3.3表结构

用户信息表（User）

| 列名 | 数据类型 | 主键 | 非空 | 描述 |
| --- | --- | --- | --- | --- |
| user\_id | INT | 主键 | 是 | 用户ID |
| user\_name | VARCHAR |  | 是 | 用户姓名 |
| user\_address | VARCHAR |  | 是 | 用户地址 |
| user\_phone | VARCHAR |  | 是 | 用户电话 |
| user\_email | VARCHAR |  | 是 | 用户电子邮件 |
| user\_password | VARCHAR |  | 是 | 用户密码 |

管理员信息表(admin)

| 列名 | 数据类型 | 主键 | 唯一 | 非空 | 描述 |
| --- | --- | --- | --- | --- | --- |
| admin\_id | INT | 主键 |  |  | 管理员ID |
| username | VARCHAR |  | 唯一 | 是 | 用户名 |
| password | VARCHAR |  |  | 是 | 密码 |

报刊信息表 (newspaper)

| 列名 | 数据类型 | 主键 | 非空 | 描述 |
| --- | --- | --- | --- | --- |
| newspaper\_id | INT | 主键 | 是 | 报刊ID |
| newspaper\_name | VARCHAR |  | 是 | 报刊名称 |
| publisher | VARCHAR |  | 是 | 出版商 |
| publication\_cycle | VARCHAR |  | 是 | 出版周期 |
| amount | DECIMAL |  | 是 | 报刊金额 |

订阅信息表 (subscription)

| 列名 | 数据类型 | 主键 | 非空 | 描述 |
| --- | --- | --- | --- | --- |
| subscription\_id | INT | 主键 | 是 | 订阅ID |
| user\_id | INT |  | 是 | 用户ID |
| newspaper\_id | INT |  | 是 | 报刊ID |
| subscriber\_name | VARCHAR |  | 是 | 订阅者姓名 |
| contact\_info | VARCHAR |  | 是 | 联系信息 |
| subscription\_period | VARCHAR |  | 是 | 订阅周期 |
| total\_amount | DECIMAL |  | 是 | 总金额 |

发行信息表 (distribution)

| 列名 | 数据类型 | 主键 | 非空 | 描述 |
| --- | --- | --- | --- | --- |
| distribution\_id | INT | 主键 | 是 | 发行ID |
| newspaper\_id | INT |  | 是 | 报刊ID |
| distribution\_date | DATE |  | 是 | 发行日期 |
| distribution\_quantity | INT |  | 是 | 发行数量 |

# 4、 创建数据库及SQL相关操作

## 4.1 创建数据库

建立数据库，数据库名称自取，设置字符集为UTF-8,排序规则为utf8\_general\_ci。

代码：

-- 创建报刊管理系统数据库

CREATE DATABASE newspaper\_management

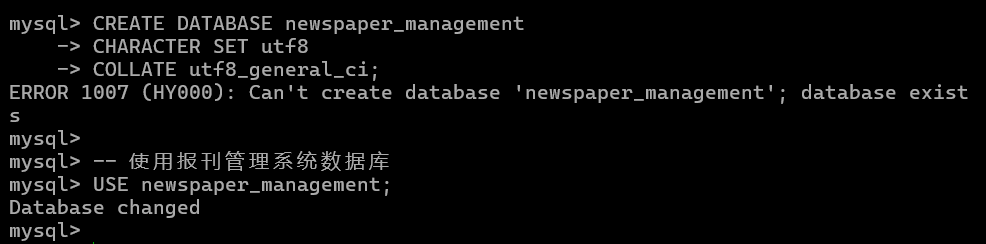
CHARACTER SET utf8

COLLATE utf8\_general\_ci;

-- 使用报刊管理系统数据库

USE newspaper\_management;

截图结果：



### 4.1.1 创建表

-- 创建管理员用户表

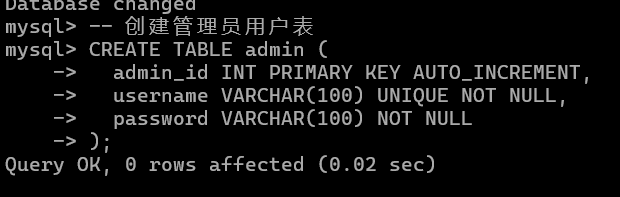
CREATE TABLE admin (

admin\_id INT PRIMARY KEY AUTO\_INCREMENT,

username VARCHAR(100) UNIQUE NOT NULL,

password VARCHAR(100) NOT NULL

);



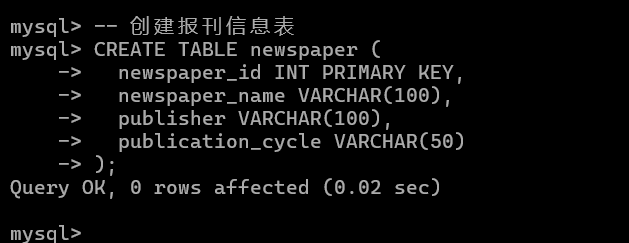
-- 创建报刊信息表

CREATE TABLE newspaper (

newspaper\_id INT PRIMARY KEY,

newspaper\_name VARCHAR(100),

publisher VARCHAR(100),

 publication\_cycle VARCHAR(50)

);

-- 创建订阅信息表

CREATE TABLE subscription (

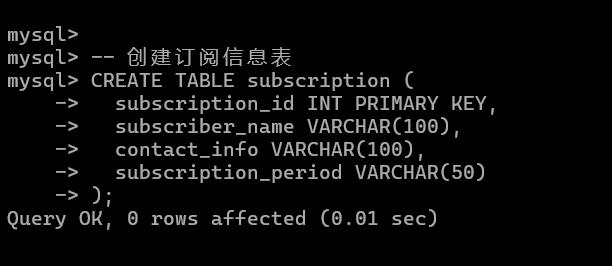
subscription\_id INT PRIMARY KEY,

subscriber\_name VARCHAR(100),

contact\_info VARCHAR(100),

subscription\_period VARCHAR(50)

);



-- 创建发行信息表

CREATE TABLE distribution (

distribution\_id INT PRIMARY KEY,

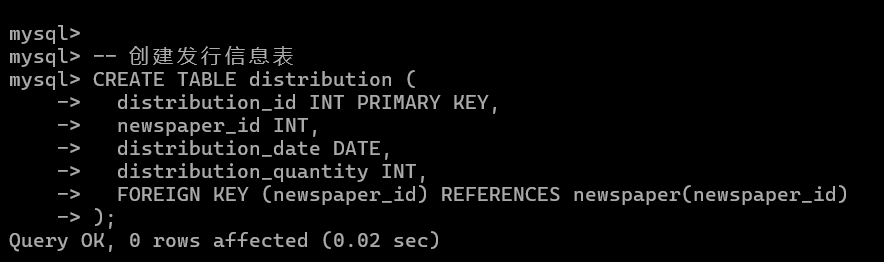
newspaper\_id INT,

distribution\_date DATE,

distribution\_quantity INT,

FOREIGN KEY (newspaper\_id) REFERENCES newspaper(newspaper\_id)

);



### 4.1.2 添加记录

-- 插入管理员用户表数据

INSERT INTO admin (username, password) VALUES

('admin1', 'password1'),

('admin2', 'password2'),

('admin3', 'password3'),

('admin4', 'password4'),

('admin5', 'password5'),

('admin6', 'password6'),

('admin7', 'password7'),

('admin8', 'password8'),

('admin9', 'password9'),

('admin10', 'password10');



-- 插入报刊信息表数据

INSERT INTO newspaper (newspaper\_id, newspaper\_name, publisher, publication\_cycle) VALUES

(1, 'Newspaper1', 'Publisher1', 'Monthly'),

(2, 'Newspaper2', 'Publisher2', 'Monthly'),

(3, 'Newspaper3', 'Publisher3', 'Monthly'),

(4, 'Newspaper4', 'Publisher4', 'Monthly'),

(5, 'Newspaper5', 'Publisher5', 'Monthly'),

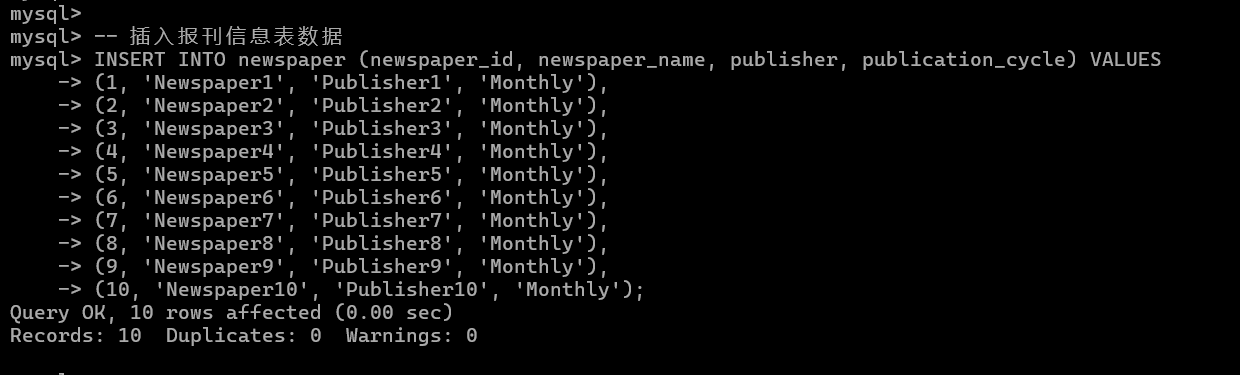
(6, 'Newspaper6', 'Publisher6', 'Monthly'),

(7, 'Newspaper7', 'Publisher7', 'Monthly'),

(8, 'Newspaper8', 'Publisher8', 'Monthly'),

(9, 'Newspaper9', 'Publisher9', 'Monthly'),

(10, 'Newspaper10', 'Publisher10', 'Monthly');



-- 插入订阅信息表数据

INSERT INTO subscription (subscription\_id, subscriber\_name, contact\_info, subscription\_period) VALUES

(1, 'Subscriber1', 'contact1@example.com', '1 Year'),

(2, 'Subscriber2', 'contact2@example.com', '1 Year'),

(3, 'Subscriber3', 'contact3@example.com', '1 Year'),

(4, 'Subscriber4', 'contact4@example.com', '1 Year'),

(5, 'Subscriber5', 'contact5@example.com', '1 Year'),

(6, 'Subscriber6', 'contact6@example.com', '1 Year'),

(7, 'Subscriber7', 'contact7@example.com', '1 Year'),

(8, 'Subscriber8', 'contact8@example.com', '1 Year'),

(9, 'Subscriber9', 'contact9@example.com', '1 Year'),

(10, 'Subscriber10', 'contact10@example.com', '1 Year');



-- 插入发行信息表数据

INSERT INTO distribution (distribution\_id, newspaper\_id, distribution\_date, distribution\_quantity) VALUES

(1, 1, '2024-12-01', 100),

(2, 2, '2024-12-02', 200),

(3, 3, '2024-12-03', 300),

(4, 4, '2024-12-04', 400),

(5, 5, '2024-12-05', 500),

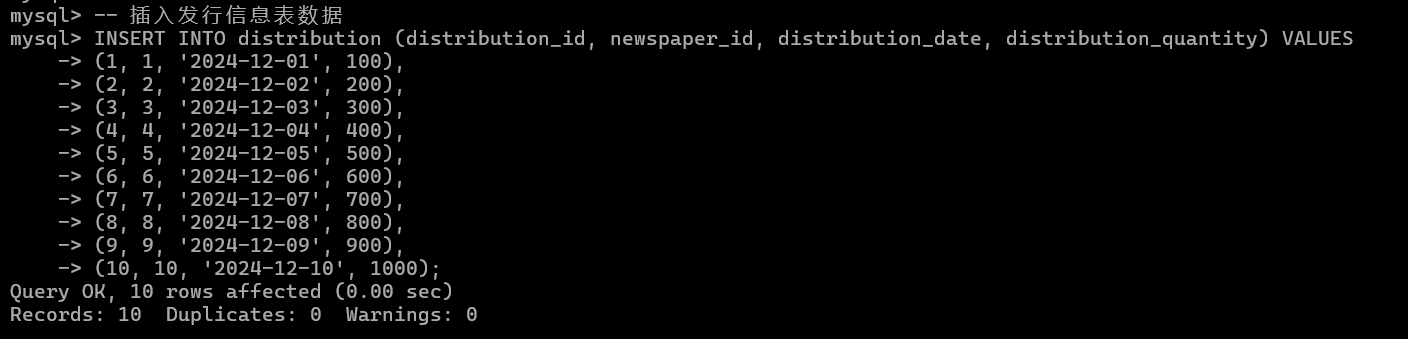
(6, 6, '2024-12-06', 600),

(7, 7, '2024-12-07', 700),

(8, 8, '2024-12-08', 800),

(9, 9, '2024-12-09', 900),

(10, 10, '2024-12-10', 1000);



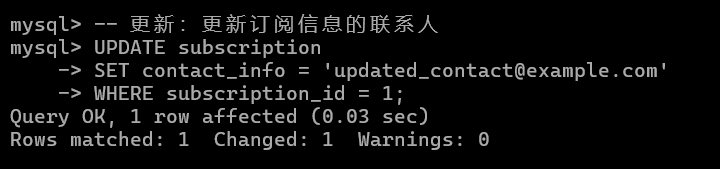
### 4.1.3 修改记录

-- 更新：更新订阅信息的联系人

UPDATE subscription

SET contact\_info = 'updated\_contact@example.com'

WHERE subscription\_id = 1;



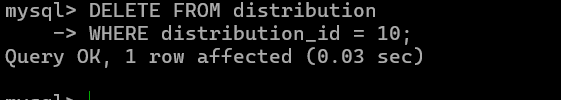
-

## 4.2 删除记录

- 删除：删除某条发行信息

DELETE FROM distribution

WHERE distribution\_id = 10;



## 4.3查询问题及查询结果

### 4.3.1 选择列

自行设计查询单表中的几列，并用别名代替。

代码：

-- 查询订阅表中的订阅者姓名和联系方式，并设置别名

SELECT subscriber\_name AS name, contact\_info AS contact

FROM subscription;

结果截图：



### 4.3.2函数使用

计算满足条件的平均值，最大值，最小值。

代码：

-- 计算发行量的平均值、最大值和最小值

SELECT

AVG(distribution\_quantity) AS average\_quantity,

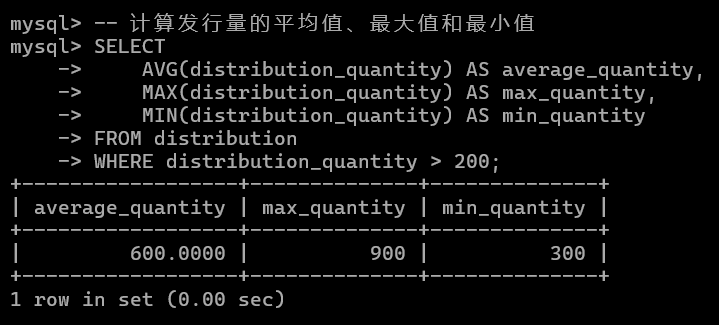
MAX(distribution\_quantity) AS max\_quantity,

MIN(distribution\_quantity) AS min\_quantity

FROM distribution

WHERE distribution\_quantity > 200;

结果截图：



### 4.3.3 LIKE子句

用LIKE子句实现模糊查询

说明实现的功能：

代码：

-- 查询订阅者名称中包含 'Subscriber' 的记录

SELECT \*

FROM subscription

WHERE subscriber\_name LIKE '%Subscriber%';

结果截图：



### 4.3.4 条件查询

用给定条件，查询结果

说明实现的功能：

代码：

-- 查询发行日期在 2024 年 12 月 1 日之后的发行信息

SELECT \*

FROM distribution

WHERE distribution\_date > '2024-12-01';

结果截图：



### 4.4.5多表查询

至少牵扯两个表，根据给定条件，查询结果

说明实现的功能：

代码：

-- 查询发行表和报刊表中的报刊名称以及发行数量

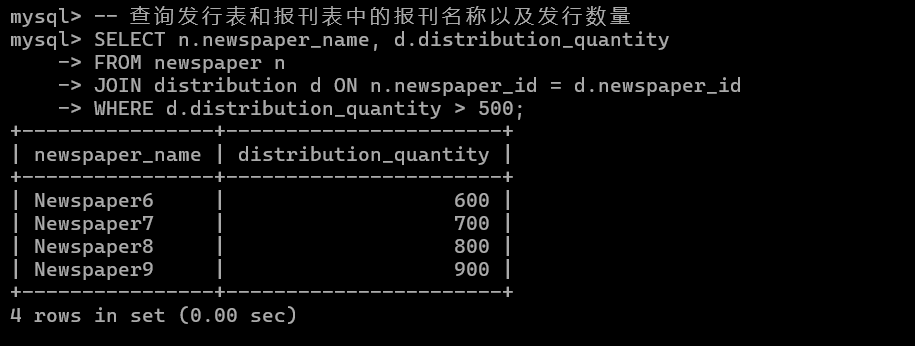
SELECT n.newspaper\_name, d.distribution\_quantity

FROM newspaper n

JOIN distribution d ON n.newspaper\_id = d.newspaper\_id

WHERE d.distribution\_quantity > 500;

结果截图：



### 4.4.6子查询

单行子查询举例

说明实现的功能：

代码：

-- 查询发行量最高的报刊名称

SELECT newspaper\_name

FROM newspaper

WHERE newspaper\_id = (

SELECT newspaper\_id

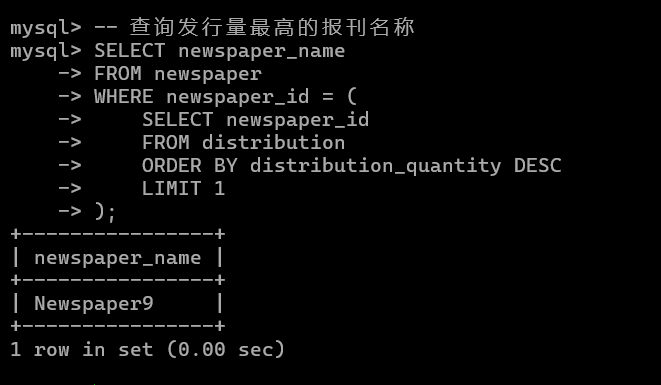
FROM distribution

ORDER BY distribution\_quantity DESC

LIMIT 1

);

结果截图：



多行子查询举例（用in，any,all任意一个均可）

说明实现的功能：

代码：

-- 查询订阅发行量大于 500 的订阅者姓名

SELECT subscriber\_name

FROM subscription

WHERE subscription\_id IN (

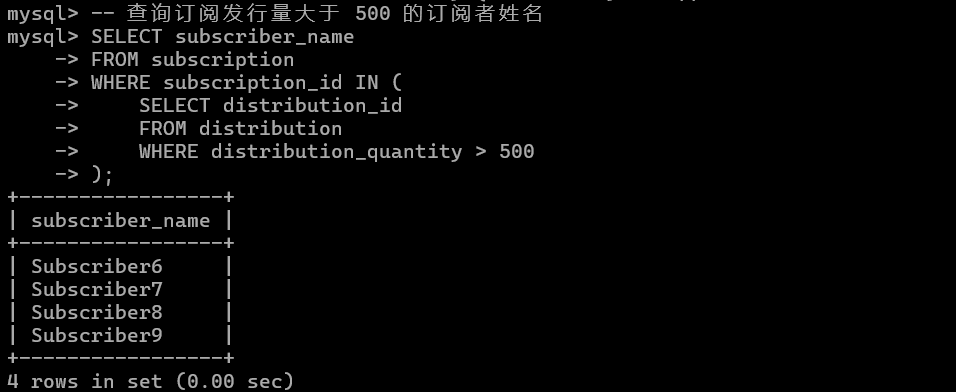
SELECT distribution\_id

FROM distribution

WHERE distribution\_quantity > 500

);

结果截图：



## 4.5建立视图

建立一个用户user01，给user01账户授予建立视图的权限。

代码：

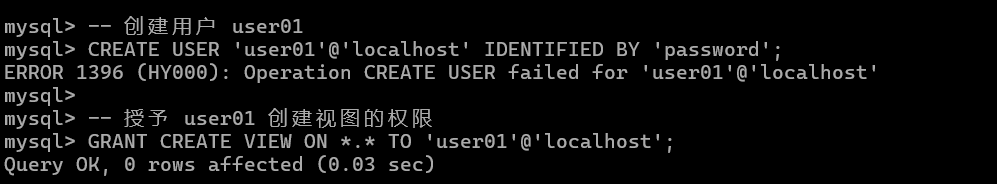
-- 创建用户 user01

CREATE USER 'user01'@'localhost' IDENTIFIED BY 'password';

-- 授予 user01 创建视图的权限

GRANT CREATE VIEW ON \*.\* TO 'user01'@'localhost';

结果截图：



### 4.5.1 定义视图

自定义一个视图，要求实现多表查询。

代码：

-- 创建一个视图，展示订阅者和对应发行的报刊名称

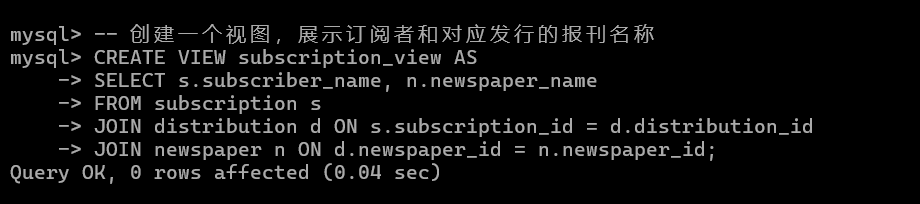
CREATE VIEW subscription\_view AS

SELECT s.subscriber\_name, n.newspaper\_name

FROM subscription s

JOIN distribution d ON s.subscription\_id = d.distribution\_id

JOIN newspaper n ON d.newspaper\_id = n.newspaper\_id;



### 4.5.2利用视图插入数据

创建简单视图代码：

CREATE VIEW simple\_subscription\_view AS

SELECT s.subscription\_id, s.subscriber\_name, n.newspaper\_name

FROM subscription s

JOIN distribution d ON s.subscription\_id = d.distribution\_id

JOIN newspaper n ON d.newspaper\_id = n.newspaper\_id;

-- 插入记录（假设视图支持更新）

-- 由于视图中包含 JOIN，无法直接通过视图插入记录（MySQL 限制）。因此，可以通过基础表进行插入：

-- 插入订阅表

INSERT INTO subscription (subscription\_id, subscriber\_name, contact\_info, subscription\_period)

VALUES (11, 'New Subscriber', 'new\_contact@example.com', '6 Months');

-- 插入发行表

INSERT INTO distribution (distribution\_id, newspaper\_id, distribution\_date, distribution\_quantity)

VALUES (11, 5, '2024-12-22', 150);



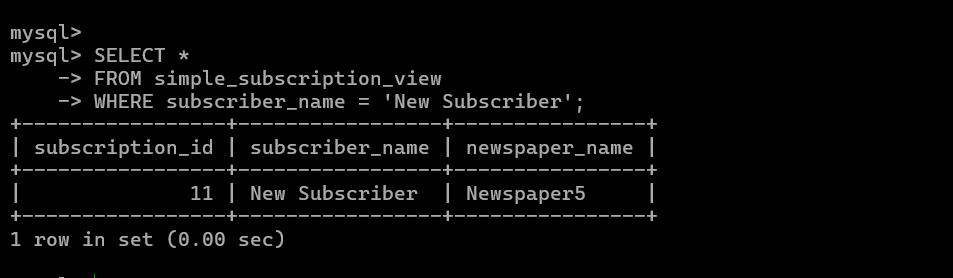
查询自定义的视图代码：

SELECT \*

FROM simple\_subscription\_view

WHERE subscriber\_name = 'New Subscriber';

结果截图：



删除插入的结果代码：

-- 删除订阅信息

DELETE FROM subscription

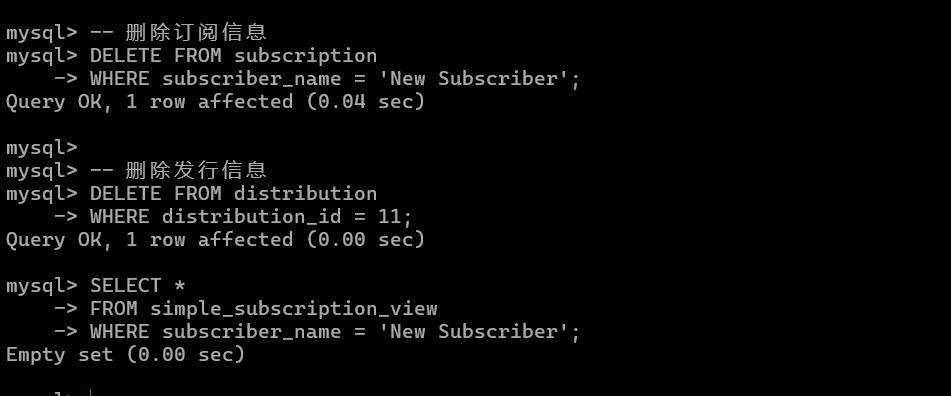
WHERE subscriber\_name = 'New Subscriber';

-- 删除发行信息

DELETE FROM distribution

WHERE distribution\_id = 11;

删除结果截图：



# 5、建立存储过程及函数

## 5.1创建存储过程

创建一个存储过程，该存储过程实现输入某一个参数可以返回对应行上另外一个参数的值（如给定学号，返回选修课程信息等）。

代码：

-- 创建存储过程，根据订阅 ID 返回对应的订阅者姓名

DELIMITER //

CREATE PROCEDURE GetSubscriberName(IN sub\_id INT, OUT sub\_name VARCHAR(100))

BEGIN

SELECT subscriber\_name INTO sub\_name

FROM subscription

WHERE subscription\_id = sub\_id;

END;

//

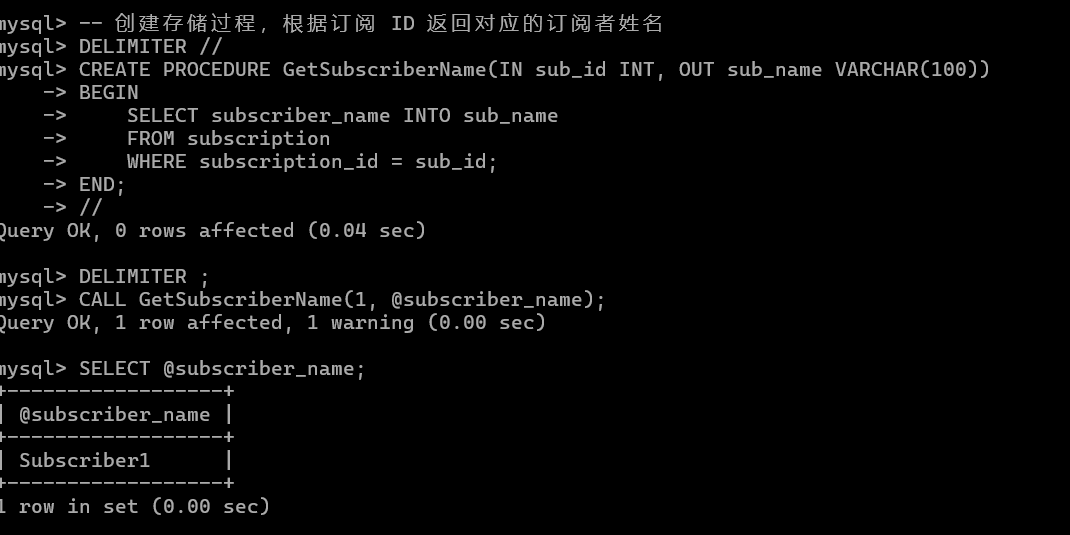
DELIMITER ;

执行存储过程代码：

CALL GetSubscriberName(1, @subscriber\_name);

SELECT @subscriber\_name;

结果截图：



## 5.2存储过程应用

创建一个函数过程，实现统计功能（统计个数或平均值等）。

代码：

-- 创建一个函数，统计发行表中发行量大于指定值的记录数

DELIMITER //

CREATE FUNCTION CountLargeDistributions(threshold INT)

RETURNS INT

DETERMINISTIC

BEGIN

DECLARE count INT;

SELECT COUNT(\*) INTO count

FROM distribution

WHERE distribution\_quantity > threshold;

RETURN count;

END;

//

DELIMITER ;

执行函数过程代码：

SELECT CountLargeDistributions(500);

结果截图：



# 6、建立触发器

-- 创建触发器，在向订阅表插入数据时记录日志

DELIMITER //

CREATE TRIGGER after\_subscription\_insert

AFTER INSERT ON subscription

FOR EACH ROW

BEGIN

INSERT INTO admin (username, password)

VALUES (CONCAT('Log\_', NEW.subscriber\_name), 'default\_password');

END;

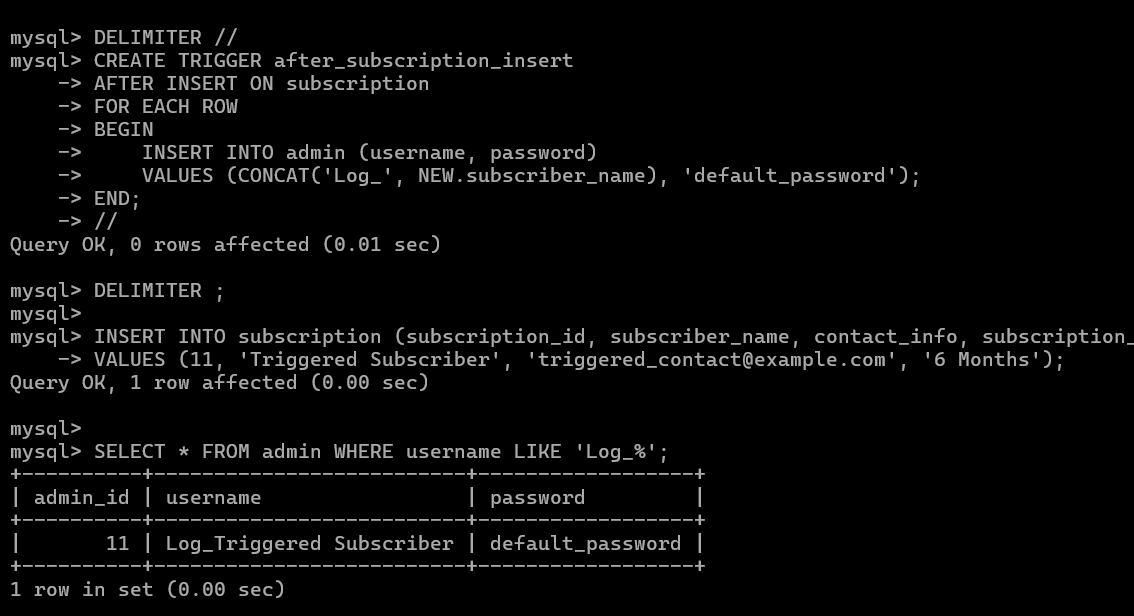
//

DELIMITER ;

INSERT INTO subscription (subscription\_id, subscriber\_name, contact\_info, subscription\_period)

VALUES (11, 'Triggered Subscriber', 'triggered\_contact@example.com', '6 Months');

SELECT \* FROM admin WHERE username LIKE 'Log\_%';



# 7、项目设计总结

在设计报刊管理系统时，我们需要考虑到系统的需求、功能和安全性等方面。通过使用数据库管理系统（如MySQL）和软件工程原理，我们可以设计出一个高效、可靠和安全的报刊管理系统。

在设计过程中，我们首先明确了报刊管理系统的选题背景和意义，以及课程设计涉及的核心理论和技术。然后，我们通过分析系统需求和功能需求，设计了数据库的表结构和关系模式，并给出了数据字典以及相应的SQL代码。为了保证数据库的完整性和安全性，我们提出了相应的措施，如定义主键和外键、设置数据类型和约束，以及使用触发器和存储过程等。

通过本次设计，我们深入了解了报刊管理系统的设计过程和关键技术，提高了对数据库管理系统和软件工程原理的理解。同时，我们也意识到了设计和实施一个完整的系统需要综合考虑多个方面，包括需求分析、系统设计、编码实现和维护操作等。这对我们今后的软件开发和系统设计工作具有重要的指导意义。