

# 浙江大学

## 数据库系统实验报告

SQL 数据定义和操作

作业名称:

谢集

姓 名:

3220103501

学 号:

1436572990@qq.com

电子邮箱:

13567793981

联系电话:

孙建伶

指导老师:

2024 年 3 月 24 日

# 实验 2 SQL 数据定义和操作

## 一、实验目的

1. 掌握关系数据库语言 SQL 的使用。
2. 面向某个应用定义数据模式和操作数据。

## 二、实验环境

操作系统：Windows11 22H2。

## 三、实验流程

打开命令行，输入 `mysql -u sanaka -p`。输入密码，完成 Lab1 中所创建账号的登录。使用命令 `create database Lab1` 创建新数据库 Lab1。

我选择创建的数据库是关于图书管理的，包含三个表：

### 1、图书（books）

- `book_id`: 图书的 ID，我将它设为主键。
- `title`: 图书的标题。
- `author`: 图书的作者。

### 2、借阅记录（borrow）

- `record_id`: 借阅记录的 ID，我将它设为主键。
- `book_id`: 借出的图书 ID。
- `reader_id`: 借阅者的 ID。

### 3、读者表（readers）

- `reader_id`: 读者的 ID，我将它设为主键。
- `name`: 读者的姓名。
- `email`: 读者的邮箱。

用 SQL 代码进行表格创建：

```
create table books (  
    book_id int primary key,  
    title varchar(255),  
    author varchar(255)  
);  
  
create table borrow (  
    record_id int primary key,  
    book_id int,  
    reader_id int  
);  
  
create table readers (  
    reader_id int primary key,  
    name varchar(255),  
    email varchar(255)  
);
```

使用命令 `drop table if exists books;` 即可删除表格 books。效果如下：

```
mysql> drop table if exists books;
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> show tables;
+-----+
| Tables_in_lab1 |
+-----+
| borrow          |
| readers         |
+-----+
2 rows in set (0.00 sec)
```

使用命令 `create index id_author on books(author);` 即可创建索引，再使用命令 `drop index id_author on books;` 即可删除索引。

使用命令：

```
create view borrow_detailed as
select * from (borrow natural join books) natural join readers;
```

即可创建视图。我定义的这个视图（borrow\_detailed）的意义表示详细的查询信息，包括全部的用户信息、书本信息。使用命令 `drop view borrow_detailed;` 即可删除该视图。

接下来进行数据插入操作：

```
insert into books(book_id, title, author) values
(1, 'How to run faster', 'ZJU'),
(2, 'Database and SQL', 'Sun Jianlin'),
(3, 'Advanced Data Structures', 'Wang Can'),
(4, 'Programming in Java', 'Weng Kai');

insert into readers(reader_id, name, email) values
(1, 'CDJ', 'cdj666@qq.com'),
(2, 'YZY', 'yzy233@qq.com'),
(3, 'XJ', 'sanaka@qq.com'),
(4, 'LHH', 'lhxxx@qq.com');

insert into borrow(record_id, book_id, reader_id) values
(1, 1, 1),
(2, 2, 1),
(3, 3, 2),
(4, 4, 3),
(5, 1, 4),
(6, 2, 4);
```

我们进行单表查询进行验证：

```
mysql> select * from readers;
+-----+-----+-----+
| reader_id | name | email |
+-----+-----+-----+
| 1 | CDJ | cdj666@qq.com |
| 2 | YZY | yzy233@qq.com |
| 3 | XJ | sanaka@qq.com |
| 4 | LHH | lhhhh@qq.com |
+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select title from books
-> ;
+-----+
| title |
+-----+
| How to run faster |
| Database and SQL |
| Advanced Data Structures |
| Programming in Java |
+-----+
4 rows in set (0.00 sec)
```

接下来进行更新操作和删除操作：

```
mysql> update readers set email = 'sanaka87@zju.edu.cn' where reader_id = 3;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> delete from borrow where record_id = 6;
Query OK, 1 row affected (0.01 sec)

mysql> select email from readers
-> ;
+-----+
| email |
+-----+
| cdj666@qq.com |
| yzy233@qq.com |
| sanaka87@zju.edu.cn |
| lhhhh@qq.com |
+-----+
4 rows in set (0.00 sec)

mysql> select * from borrow
-> ;
+-----+-----+-----+
| record_id | book_id | reader_id |
+-----+-----+-----+
| 1 | 1 | 1 |
| 2 | 2 | 1 |
| 3 | 3 | 2 |
| 4 | 4 | 3 |
| 5 | 1 | 4 |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

可以发现，我们成功进行了数据的更新和删除操作。

接下来我们进行复杂的多表查询/嵌套子查询（查询每本书名被借阅的次数），代码如下：

```
select b.title, count(br.book_id) as borrow_count
  from books b natural join borrow br
 group by b.title;
```

结果完全正确，为：

title	borrow_count
How to run faster	2
Database and SQL	1
Advanced Data Structures	1
Programming in Java	1

尝试对之前创建的视图进行查询，结果也完全正确：

```
mysql> select * from borrow_detailed;
```

reader_id	book_id	record_id	title	author	name	email
1	1	1	How to run faster	ZJU	CDJ	cdj666@qq.com
1	2	2	Database and SQL	Sun Jianlin	CDJ	cdj666@qq.com
2	3	3	Advanced Data Structures	Wang Can	YZY	yzy233@qq.com
3	4	4	Programming in Java	Weng Kai	XJ	sanaka87@zju.edu.cn
4	1	5	How to run faster	ZJU	LHH	lhhhh@qq.com

5 rows in set (0.01 sec)

对一个视图进行数据更新，结果如下：

```
mysql> create view emails as (select reader_id, email from readers);
Query OK, 0 rows affected (0.01 sec)

mysql> insert into emails(reader_id, email) value (5, 'kimset@qq.com');
Query OK, 1 row affected (0.00 sec)

mysql> select * from readers
-> ;
```

reader_id	name	email
1	CDJ	cdj666@qq.com
2	YZY	yzy233@qq.com
3	XJ	sanaka87@zju.edu.cn
4	LHH	lhhhh@qq.com
5	NULL	kimset@qq.com

5 rows in set (0.00 sec)

虽然插入成功，但是未在视图中显示出的字段被设置成了 NULL。这也是符合我们课堂学过的知识的。

## 四、遇到的问题及解决方法

本次实验较为基础，因此我没有遇到太大的问题。不过我对 Lab1 中的一些基础操作（比如登录命令）还是有些陌生，这也促使我加强对 MySQL 基础知识的复习。

还有，我在写 SQL 代码时常常出现忘记加分号的情况，下一次实验一定会注意。

## 五、总结

通过本次实验，我深入理解并且亲自测试了 SQL 语言的强大功能，包括数据定义、更新、查询、视图等操作。设计并操作图书管理系统的过程，也增强了我对数据库实际应用的认识，为日后的课程学习奠定了坚实的基础。