**课程名称：**《数据库安全实训》 **专业：**网络空间安全 **班级：**21网安班 **指导老师：**张跃欣

**填写时间及地点**：2024年 6 月 12 日

**学号及姓名**：

135052021091 李文钦，121122021002 蔡展宇

121122021039 钟智恺，121122021040 李贵雄

**实训报告题目**：

图书馆管理系统数据库的实现

**实训内容及步骤**：

**1. 项目需求分析**

**1.1用户特点**

本系统的核心用户包括图书馆管理员和读者。管理员需具备一定的计算机操作技能和数据库管理知识，负责图书的增减、分类、以及借阅记录的更新等任务。读者则需具备基本的计算机操作能力，能够独立进行图书搜索、借阅和归还操作。

**1.2需求概述**

在过去的图书馆管理中，图书的借阅、归还、以及信息记录等环节主要依赖于人工操作。这种传统的方式不仅效率低下，而且在处理大量的图书和用户信息时容易出错。例如，图书的借阅和归还需要手工填写借阅卡，这不仅耗时而且容易造成信息的遗漏或错误。图书的检索也需要通过卡片目录或手工登记的方式进行，这不仅不便于快速查找，也难以保持信息的实时更新。此外，图书的库存管理、逾期罚款的计算、以及图书的损坏和遗失处理等都需要大量的人工干预，这些繁琐的工作不仅增加了图书馆管理员的负担，也影响了图书馆服务的质量和效率。

**1.3解决方案**

为解决此问题，图书馆管理系统应运而生，旨在实现管理过程的自动化、标准化和高效化。系统通过引入智能化设计，简化并优化了图书馆的日常运营流程。管理员可通过系统轻松完成新书入库、图书分类、借阅规则设定等任务。图书信息管理模块使管理员能快速准确地更新图书信息，图书类型管理便于对图书进行分类整理，方便读者检索。借阅模块详细记录读者借阅历史，使图书流通状况一目了然。归还模块确保图书归还流程顺畅，同时更新图书借阅状态。

对读者而言，使其能轻松搜索图书、查看借阅状态以及在线续借。读者信息模块建立后，每次借阅和归还都能迅速完成，极大提升读者使用体验。

综上所述，图书馆管理系统的引入不仅减轻了管理员工作负担，提高工作效率，还增强了读者服务的便捷性和准确性，全面提升了图书馆的服务质量和整体管理水平。

**2.数据库设计**

**2.1项目系统结构图**

图书馆管理系统分为三部分，管理员部分负责管理员信息的增删改查，图书管理部分主要包括图书信息、类型、借阅和归还，而读者管理负责读者信息的修改、增加、查询和修改。

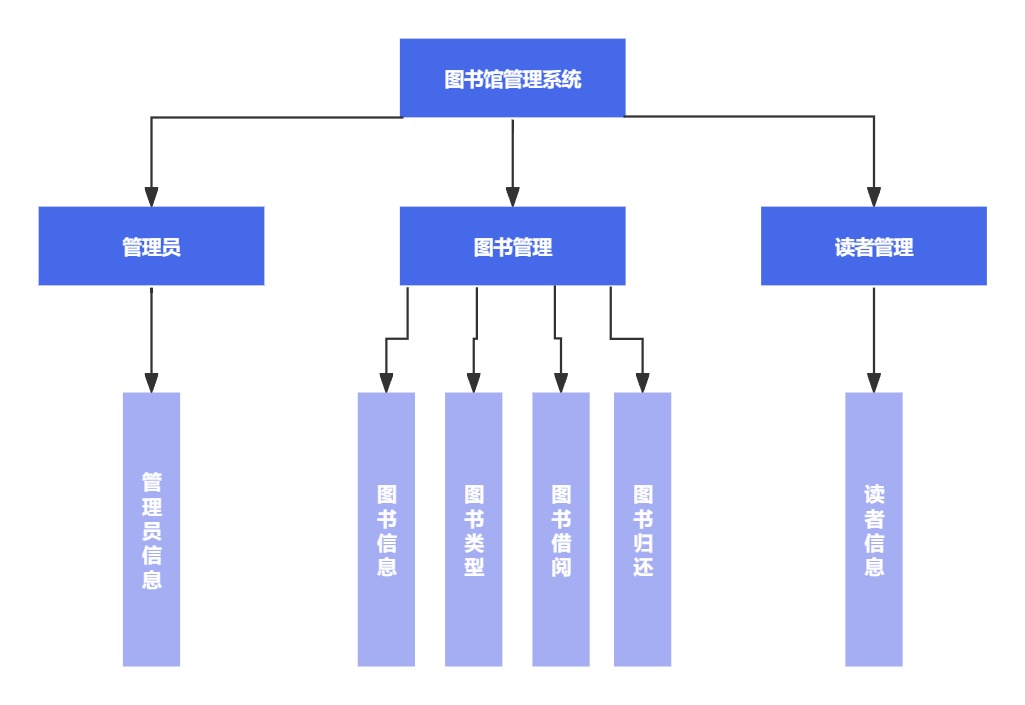


图1 项目系统结构图

**2.2实体之间的联系**

实体之间的联系如下：

（1）一个管理员可以整理多本图书信息，因此管理员和图书信息具有一对多的联系，用整理来表示管理员和图书信息的关系。

（2）一个管理员可以对多种图书类型进行归类，因此管理员和图书类型具有一对多的联系，用归类来表示管理员和图书类型的关系。

（3）一个管理员可以管理多个图书借阅信息，因此管理员和图书借阅信息具有一对多的联系，用管理来表示管理员和图书借阅信息的关系。

（4）一个管理员可以管理多个图书归还信息，因此管理员和图书归还信息具有一对多的联系，用管理来表示管理员和图书归还信息的关系。

（5）一个管理员可以管理多个读者信息，因此管理员和读者信息具有一对多的联系，用管理来表示管理员和读者信息的关系。

（6）一本图书可由多个读者借阅，一个读者也可借阅多本图书，因此读者和图书借阅具有多对多的联系，用借阅来表示读者和图书借阅的关系。

（7）一本图书可由多个读者归还，一个读者也可归还多本图书，因此读者和图书归还具有多对多的联系，用归还来表示读者和图书归还的关系。

**2.3概念结构设计**

由实体间的联系转为E-R图，概念结构E-R图设计如下：

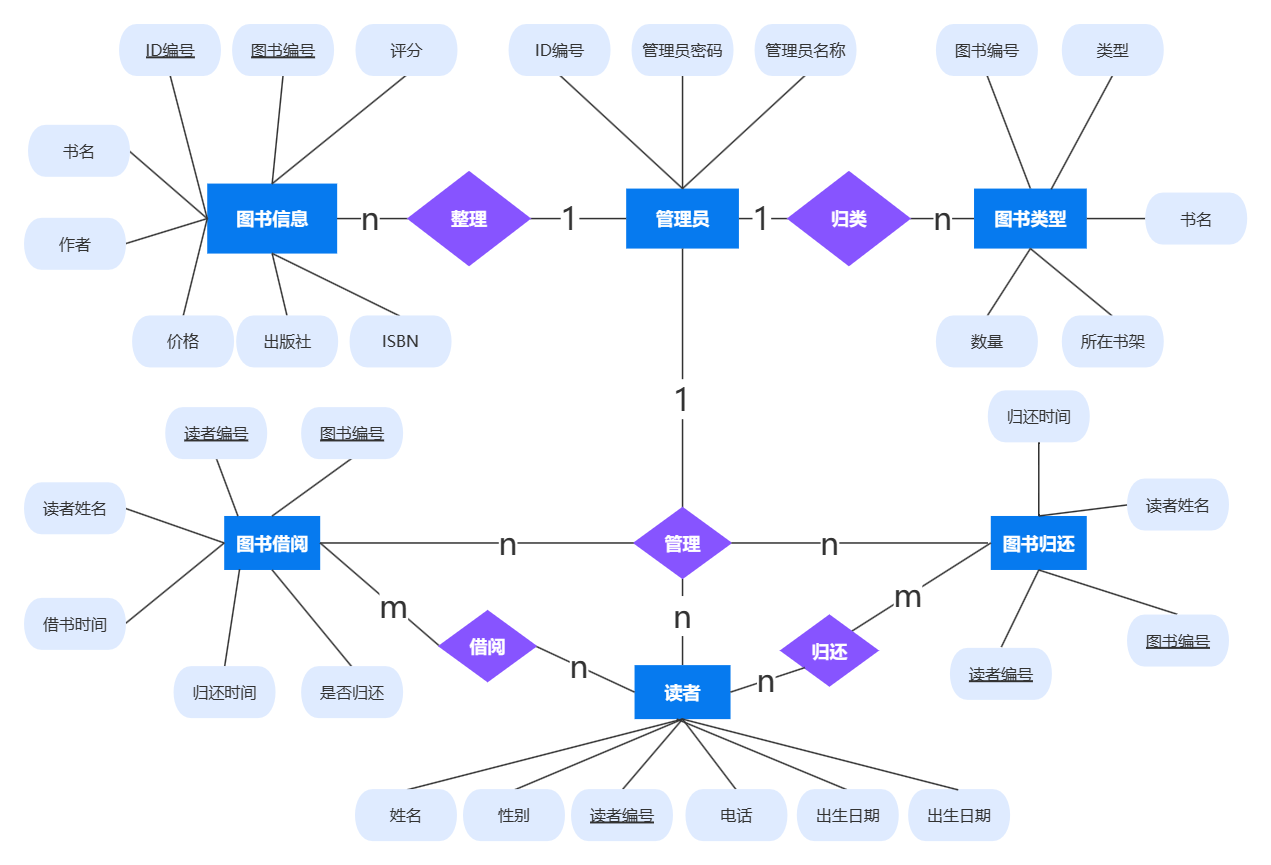


图2 E-R图

**2.4逻辑结构设计**

将E-R图转换为逻辑结构关系模型，其中有下划线的为主键（或外键）。

（1）管理员（ID编号、管理员名称、管理员密码）

（2）图书信息（ID编号、图书编号、评分、出版时间、书名、作者、价格、类型、出版社、ISBN、页数）

（3）图书类型（图书编号、书名、数量、类型、所在书架）

（4）图书借阅（读者编号、图书编号、读者姓名、借书时间、归还时间、是否归还）

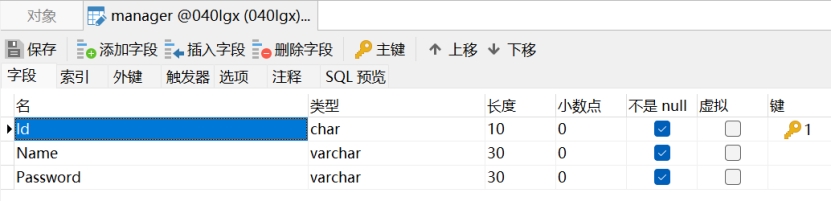
（5）图书归还（读者编号、图书编号、读者姓名、归还时间）

（6）读者（读者编号、姓名、性别、职业、出生日期、证件号码、电话、电子邮箱）

**2.5物理结构设计**

将关系模式转换为一张张数据表的物理结构，我们使用Mysql+Navicat进行实现，这里总共有六张表，如下：

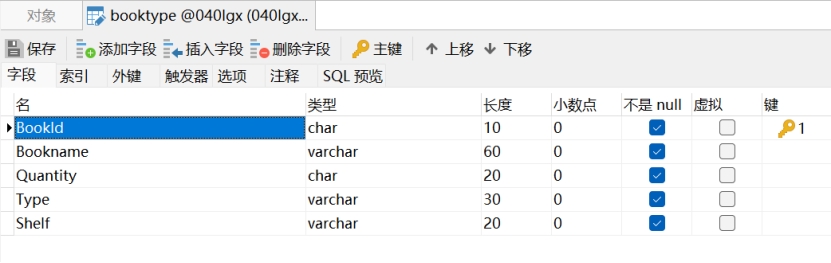
（1）创建manager(管理员信息表)



（2）创建bookinfo(图书信息表)



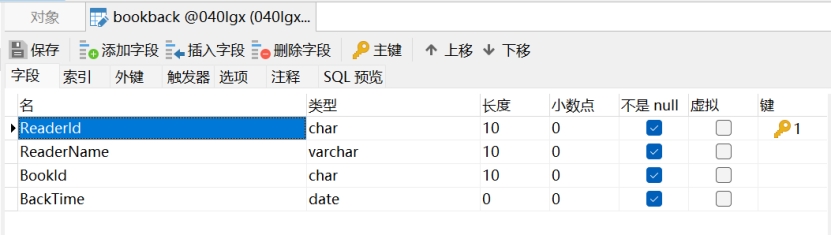
（3）创建booktype（图书类型表）



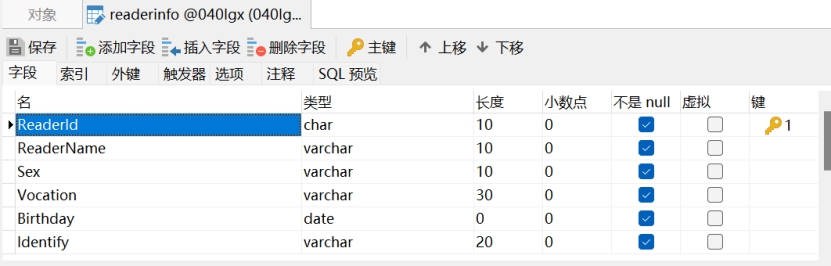
（4）创建bookborrow（图书借阅表）



（5）创建bookback（图书归还表）



（6）创建readerinfo(读者信息表)



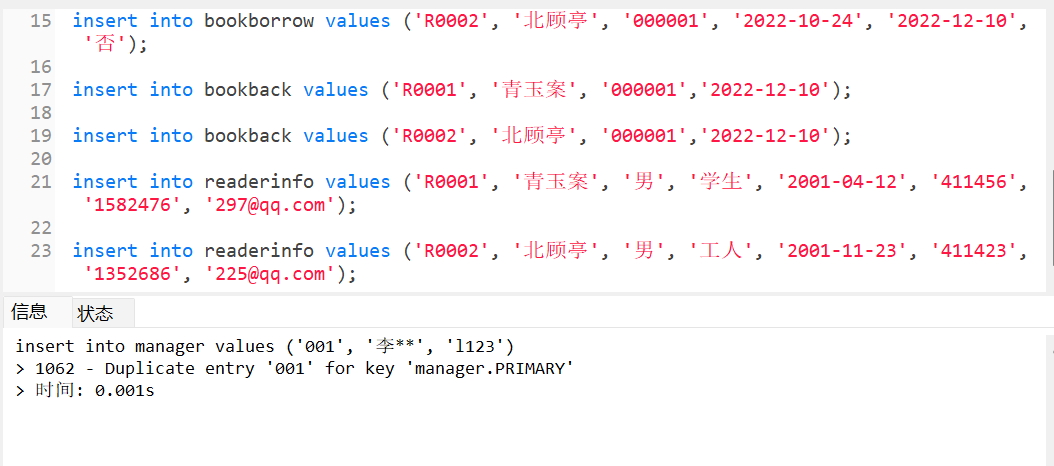
现在，我们有了最基本的数据库与表。

**3.数据库的基本实现**

**3.1数据操纵**

数据操纵是对数据库其中的对象和数据运行访问工作的编程语句。用户通过它可以实现对数据库数据的基本操作。 这种操作包括将数据插入数据库表、检索现有数据、从现有表中删除数据和更新现有数据。

（1）插入数据，在每张表里分别插入两条数据。



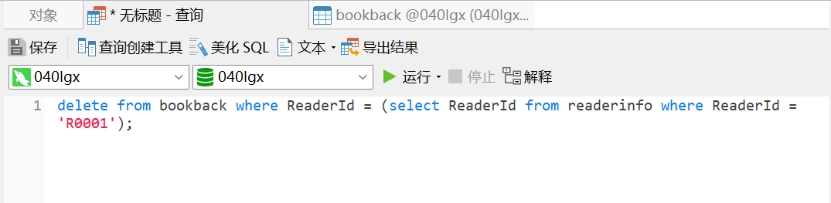
（2）修改数据，通过图书名称在类型表里查询图书编号，根据编号更改信息表里的图书价格。



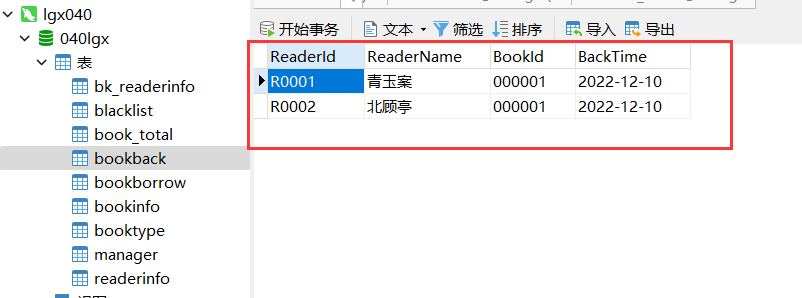
（3）查询数据，查询读者编号为R0001的读者信息和图书借阅信息。



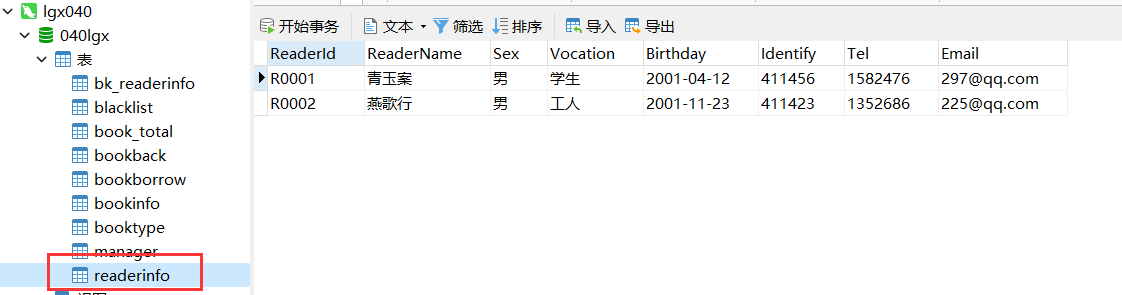
（4）删除数据，根据从读者信息表里查找的读者编号在图书归还表里删除数据。



如图所示，数据操纵后各个表格产生了相对应的变化，如图书归还表（操纵后）成功添加了信息。



如下图，读者信息表（操纵后）也成功添加了两条信息。

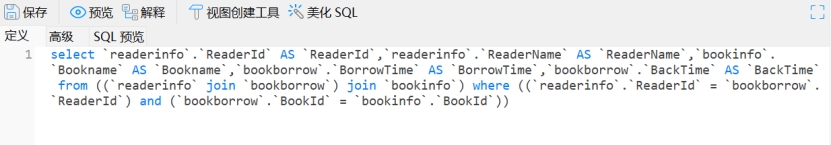


**3.2视图**

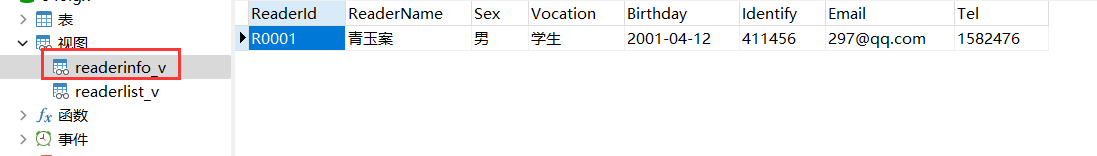
视图是从一个或几个基本表（或视图）导出的表，其作用是方便用户对数据的操作。视图是一个虚拟表，其内容由查询定义。

**（1）视图readerinfo\_v**

视图readerinfo\_v包含所有学生的信息。包含所有学生的读者编号、姓名、性别、职业、出生日期、证件号码、电话和电子邮箱。

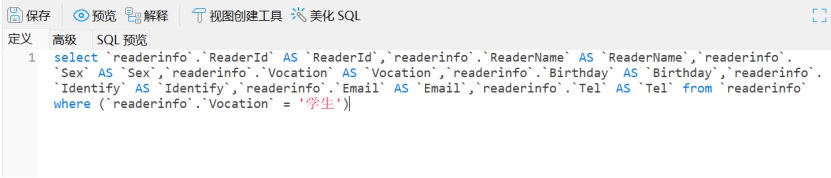


视图readerinfo\_v创建完后如下

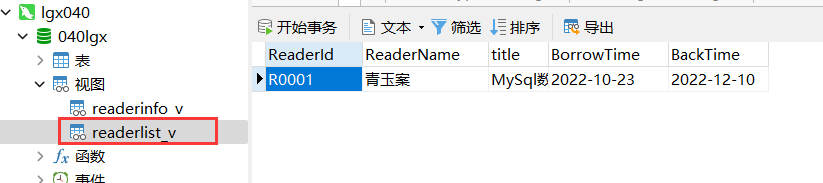


**（2）视图readerlist\_v**

创建视图readerlist\_v，包含读者编号，姓名，书名，借书时间和归还时间。



视图readerlist\_v创建完后如下。



**3.3索引**

索引是对数据库表中一列或多列的值进行排序的一种结构，是为了提高查询速度而对表字段附加的一种标识。在数据库中，索引允许数据库程序迅速地找到表中的数据，而不必扫描整个数据库。

**（1）索引manager\_index**

对管理员信息表的管理员名称创建普通索引。

****

**（2）索引bookinfo\_index**

对图书信息表中的Bookname创建唯一性索引。

****

**（3）索引combine\_index**

对读者信息表中的读者编号、姓名和电话创建组合索引。

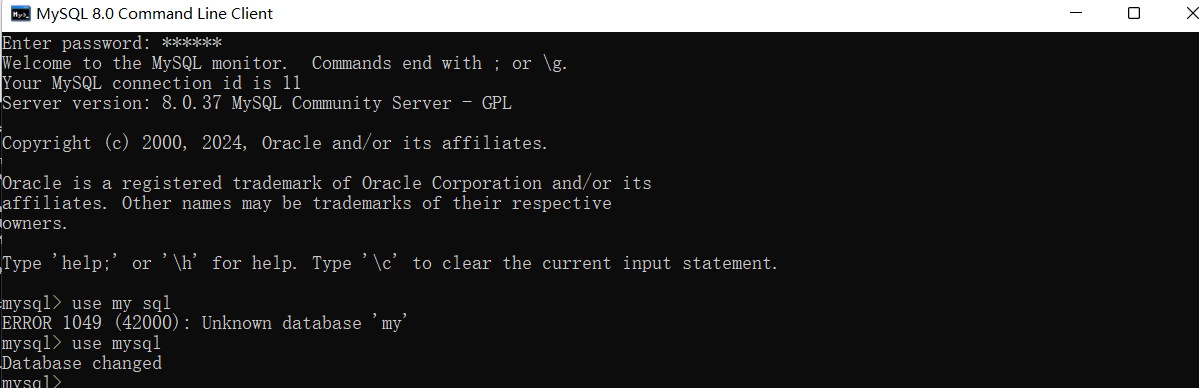


**4.数据库创建的创新**

**4.1分布式创建数据库**

在创建数据库的过程中我们发现，由一个人独立创建数据库工作量很大，并且也不公平。我们汲取了联邦学习思想，使多人能够分布式、同时地一起完成同一份数据库，大大加快了数据库创建的过程。

首先打开Mysql CLC界面。



输入如下命令，为远程登入用户授权，使得能多人共享一个数据库。

**（1）use mysql;   
（2）grant all privileges on \* . \* to ‘ceztime’@**

**‘192.168.43.233’ identified by ‘123’ with grant option;   
（3）flush privileges;**

各代码含义如下：

（1）：进入名为mysql的数据库，

（2）：ceztime、192.168.43.233、123

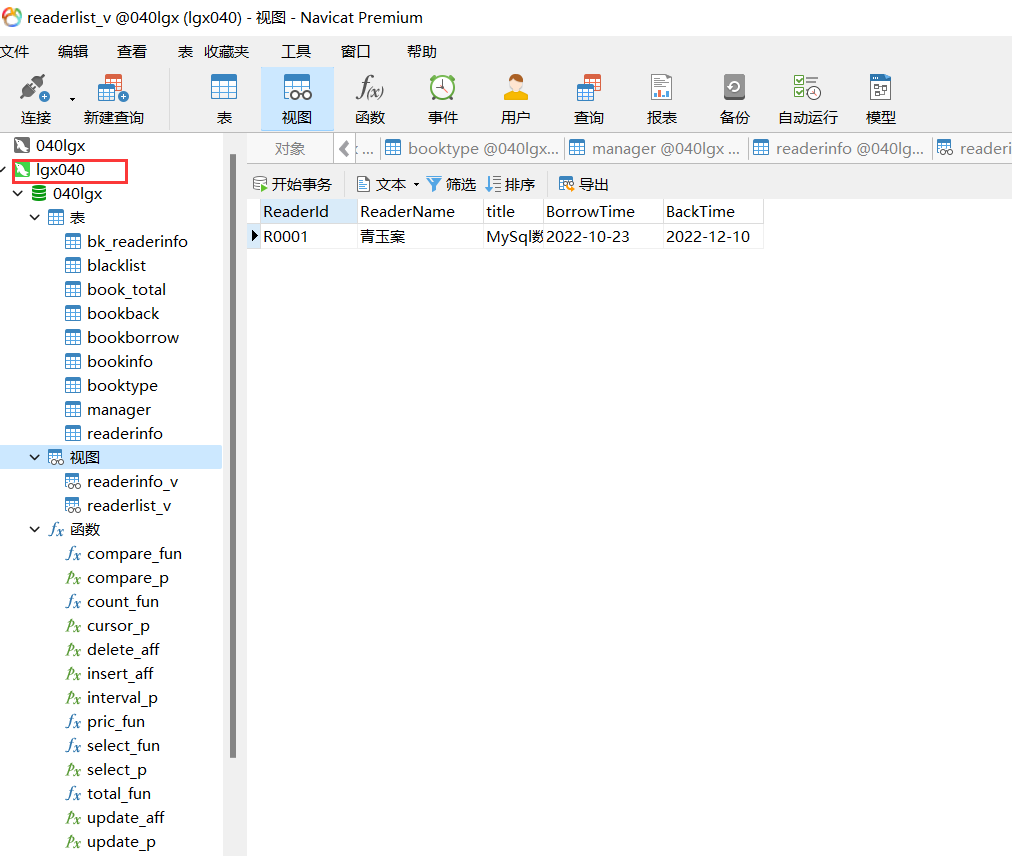
在电脑B的数据库上创建一个用户名为ceztime的账户，密码为 123，192.168.43.233就是电脑A的IP，允许它连接电脑B的数据库。

（3）：刷新权限。

之后在Navicat中按照以下格式输入相关数据。



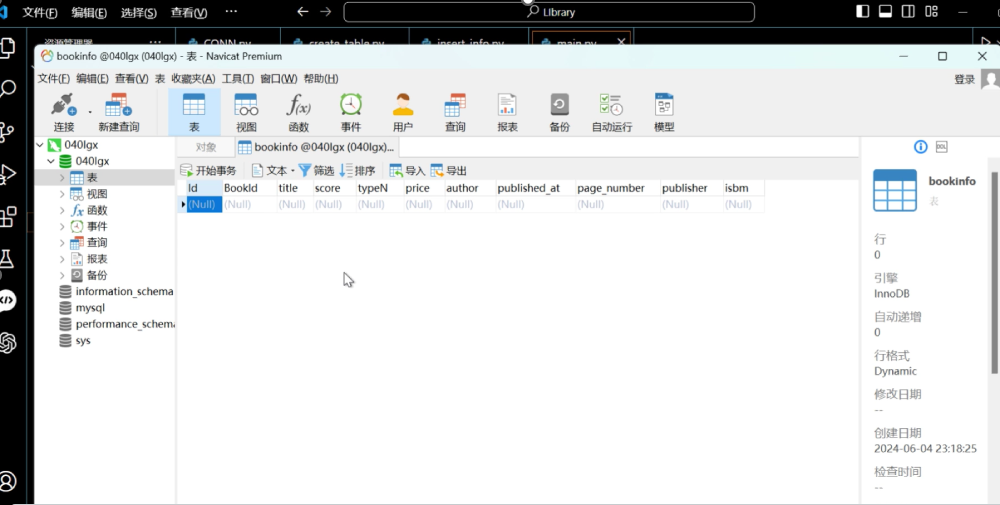
之后便可以多人共享一份数据库了，例如这里我们共享了lgx040的数据库，大家可以同时对该数据库进行操作，只需要刷新即可同步。



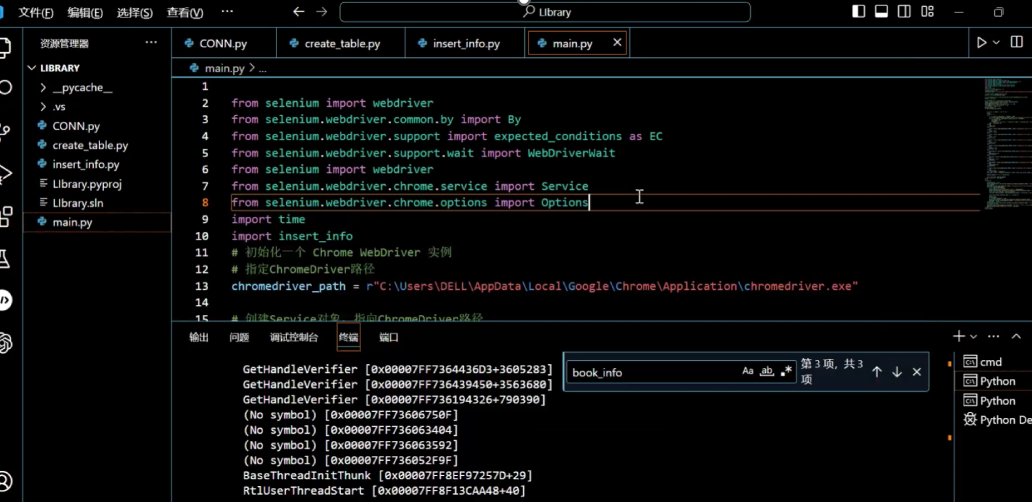
**4.2爬虫**

前面提到我们利用分布式的创建降低了工作量，但在现实生活中，对于图书信息这一动辄上千上万的庞大的数据量，即使依靠分布式，工作量仍然很大。这时我们想到了爬虫，通过爬取书籍网站的数据，达到快速创建书籍信息表的目的。

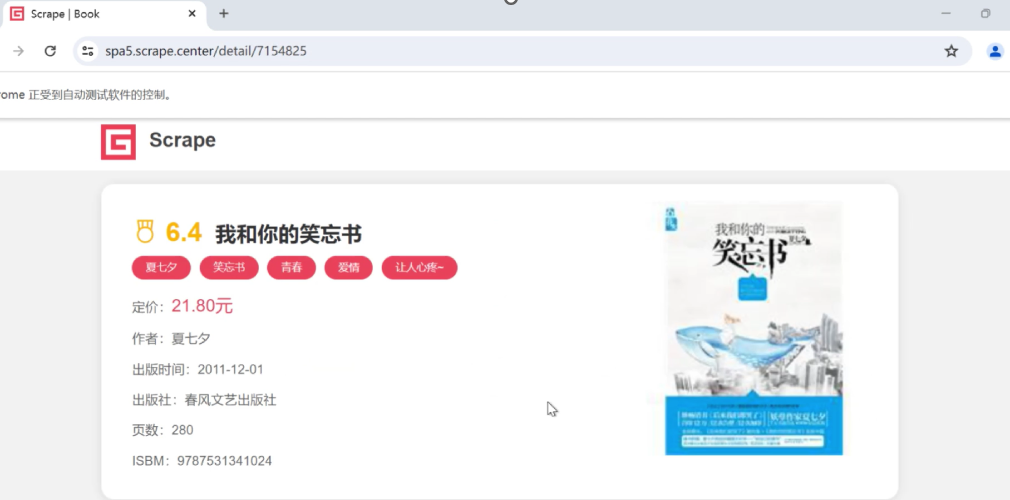
**（1）首先一开始书籍信息表为空。**

****

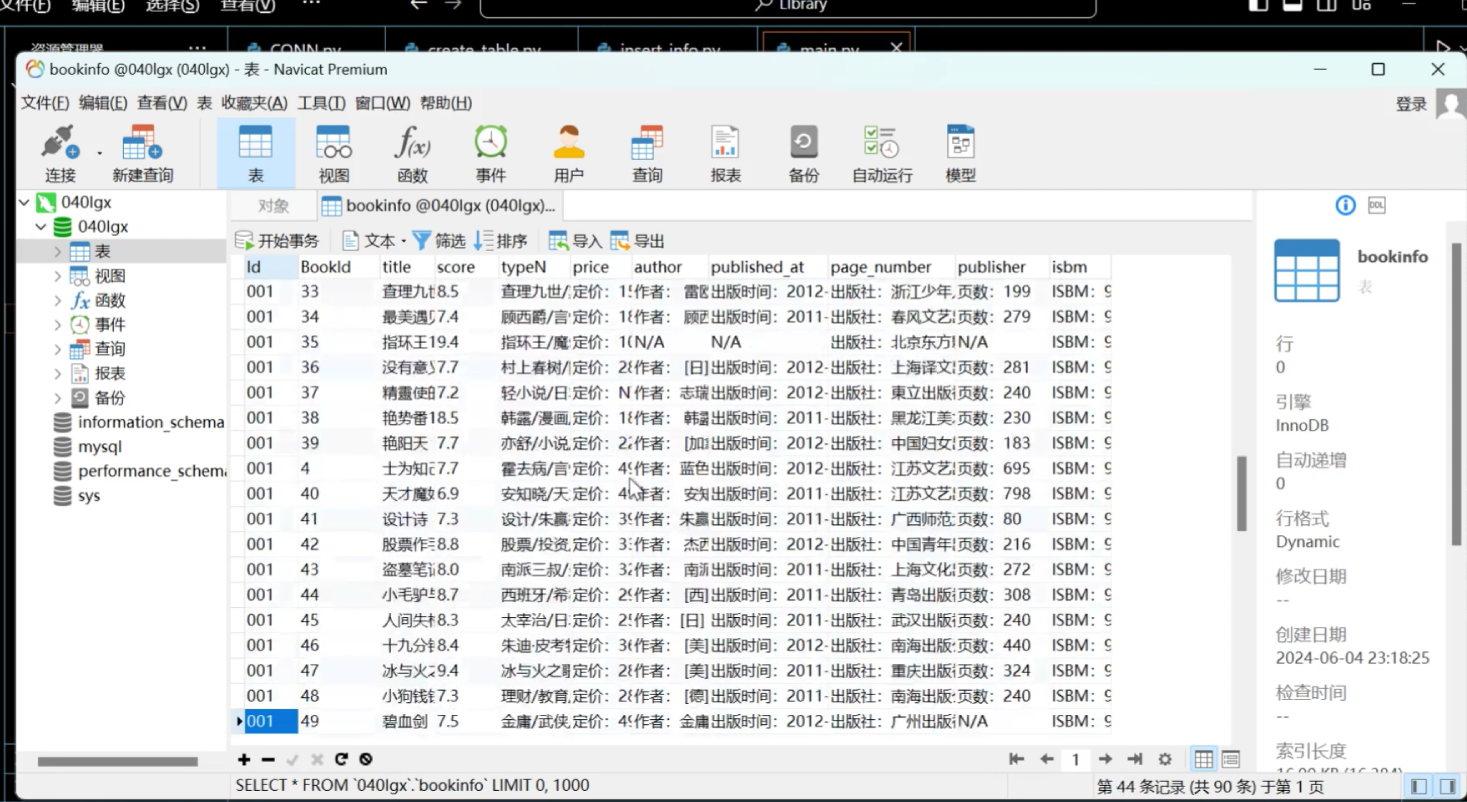
**（2）代码文件如下，它将爬取Scrape网站的书籍信息。**

****

**（3）已经开始自动爬取，这里我们设置爬取五页的书籍信息。**

****

**（4）爬取完毕，数据库里已经更新了近百条书籍信息。**



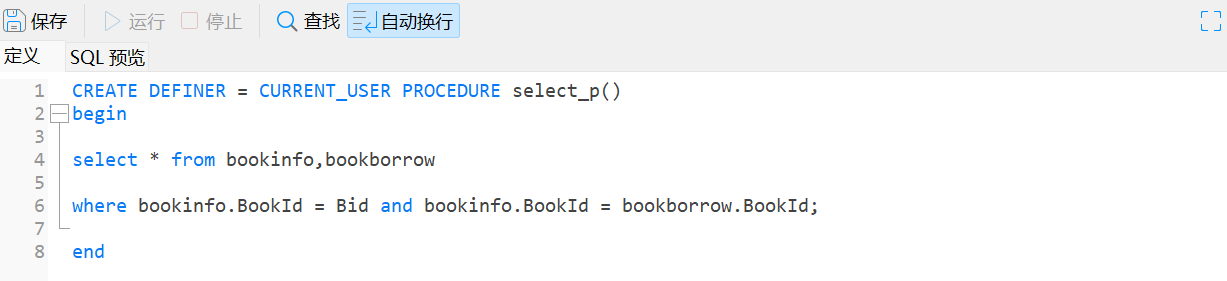
**5.数据库的更新维护**

**5.1存储过程**

存储过程是一组为了完成特定功能的SQL语句集，良好的封装性与复用性。它可以包含一系列的控制流语句，如循环和条件判断。存储过程在数据库中被保存，可以通过指定的名称来调用执行。

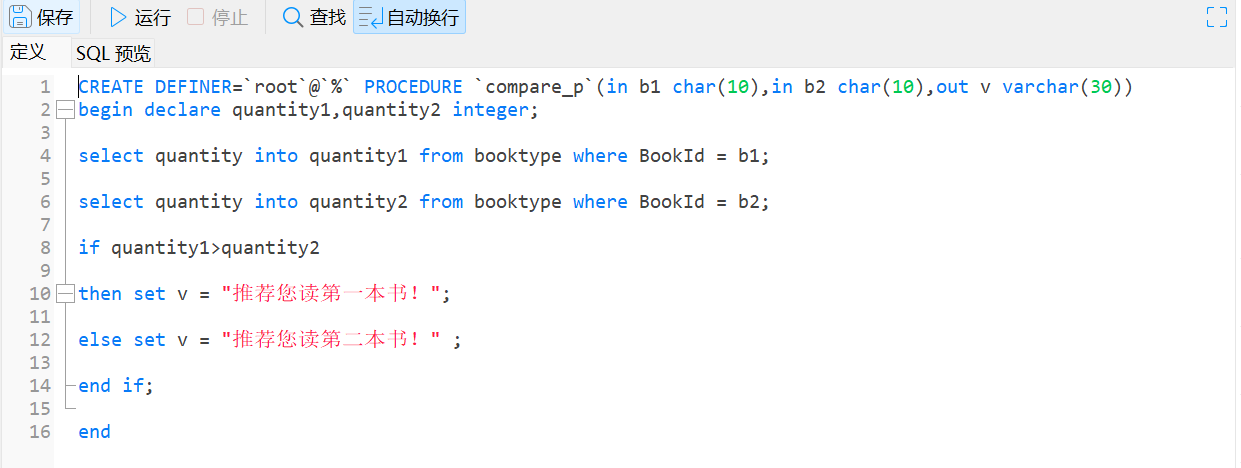
**（1）存储过程select\_p**

创建存储过程select\_p，通过输入图书编号来查询图书信息和图书借阅信息。

****

**（2）存储过程compare\_p**

创建存储过程compare\_p，通过比较两本图书的图书数量，来为读者推荐所读书籍。

****

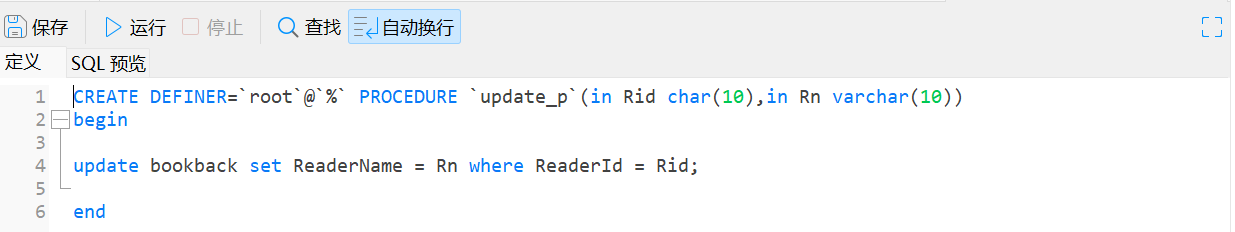
**（3）存储过程cursor\_p**

创建存储过程cursor\_p，使用fetch语句检索数据，并将游标指向从图书借阅表里查询出的结果集里

****

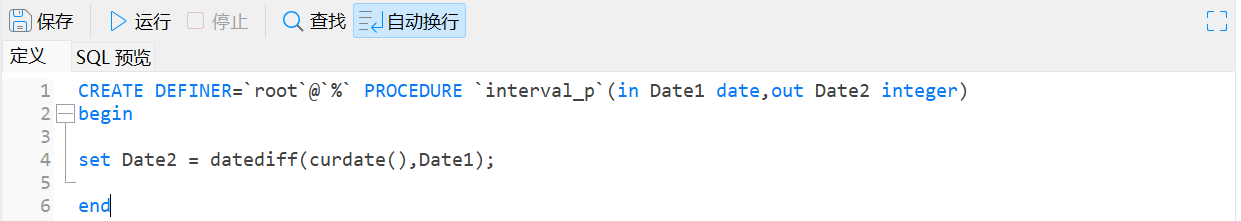
**（4）存储过程update\_p**

创建存储过程update\_p，根据输入的读者编号来更改图书归还表中的读者姓名。

****

**（5）存储过程interval\_p**

创建存储过程interval\_p，通过读者借书时间与当前时间比较，得出实际借阅天数。

****

**5.2****存储函数**

存储函数是特殊类型的存储程序，用于返回单个值。存储函数（Stored Function）和存储过程（Stored Procedure）都是数据库中预先编写并存储的代码块，它们在很多方面有相似之处，但也存在一些关键的区别：相同：预编译：存储函数和存储过程都是预先编译的，这意味着它们的执行计划在第一次创建时就已经生成，这有助于提高执行效率。不同：返回值：存储函数必须返回一个值（标量值，如整数、字符串等），而存储过程不需要返回值。

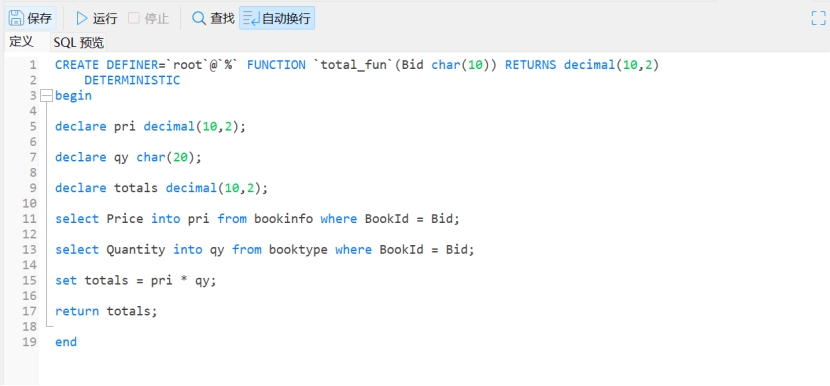
**（1）存储函数count\_fun**

创建存储函数count\_fun，返回图书价格不小于50的总数

****

**（2）存储函数total\_fun**

创建存储函数，通过输入的图书编号来计算剩余书本的总价



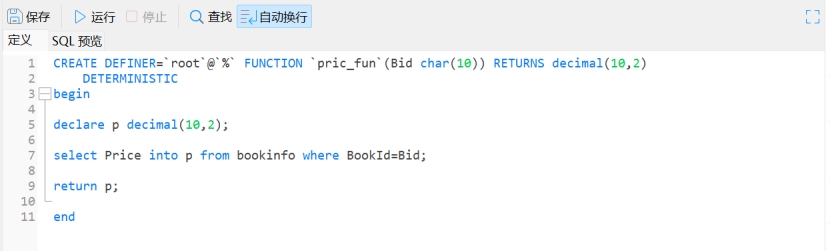
**（3）存储函数select\_fun**

创建存储函数，通过输入的读者编号来查询读者姓名，如果为空返回"未找到该读者！"，否则返回读者姓名

****

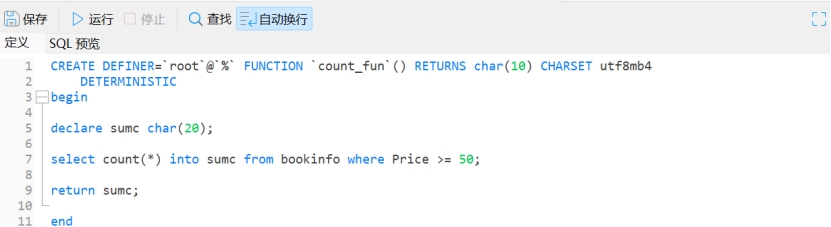
**（4）存储函数pric\_fun**

创建存储函数，通过输入两个图书编号来比较相应价格，为读者选择较便宜的书籍

****

**（5）存储函数compare\_fun**

创建存储函数，通过输入两个图书编号来比较相应价格，为读者选择较便宜的书籍

****

**5.3触发器**

触发器（被动）是当满足某些特定的条件时会自动执行的预定义操作，数据库中的触发器（Trigger）是一种特殊的存储过程，它在特定数据库事件（如INSERT、UPDATE或DELETE操作）发生时自动执行。触发器通常用于维护数据的完整性、实施复杂的业务规则、自动更新表中的数据等。

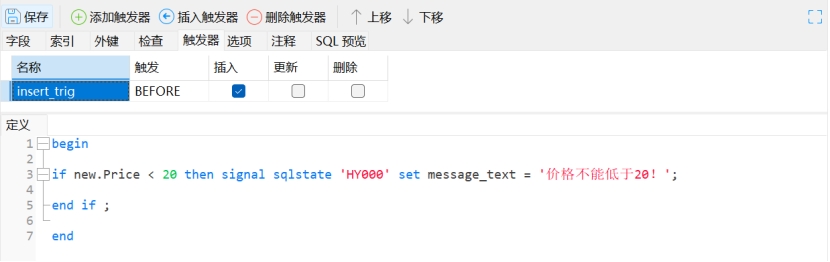
**（1）触发器update\_trig**

创建触发器，修改读者信息表中的读者姓名时同时修改图书借阅表和图书归还表中的读者姓名。

****

**（2）触发器insert\_trig**

创建触发器，限制插入数据，当图书价格低于20时插入失败并给出提示。

****

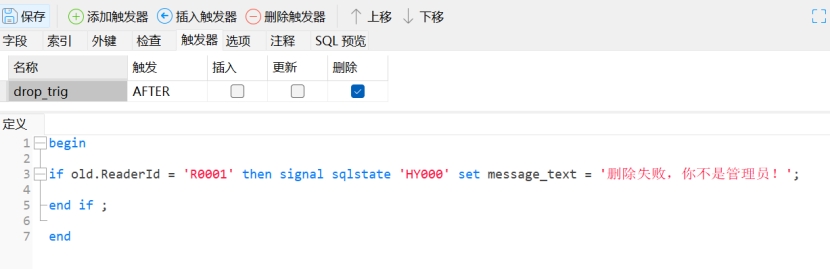
**（3）触发器delete\_trig**

创建触发器，删除读者信息表中的读者信息时同时把借阅信息与归还信息全部删除。

****

**（4）触发器drop\_trig**

创建触发器，删除图书归还表中的数据时提示“非管理员，删除失败”。

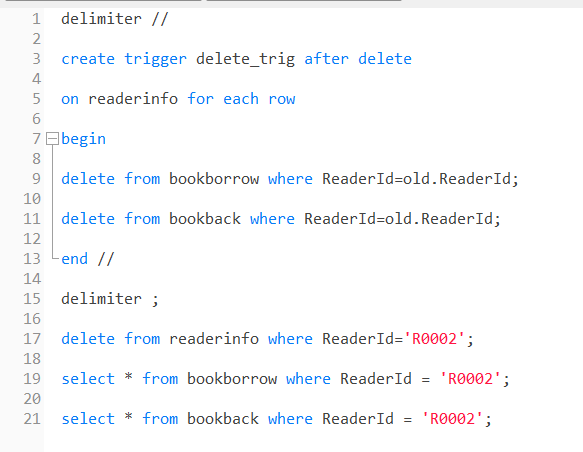


**5.4事务**

事务（主动）是一种机制、一个操作序列，包含了一组数据库操作命令。是一组不可分割的数据库操作集合，它们作为一个整体被执行，以确保数据库状态的一致性和完整性。具有原子性、一致性、隔离性和持久性。

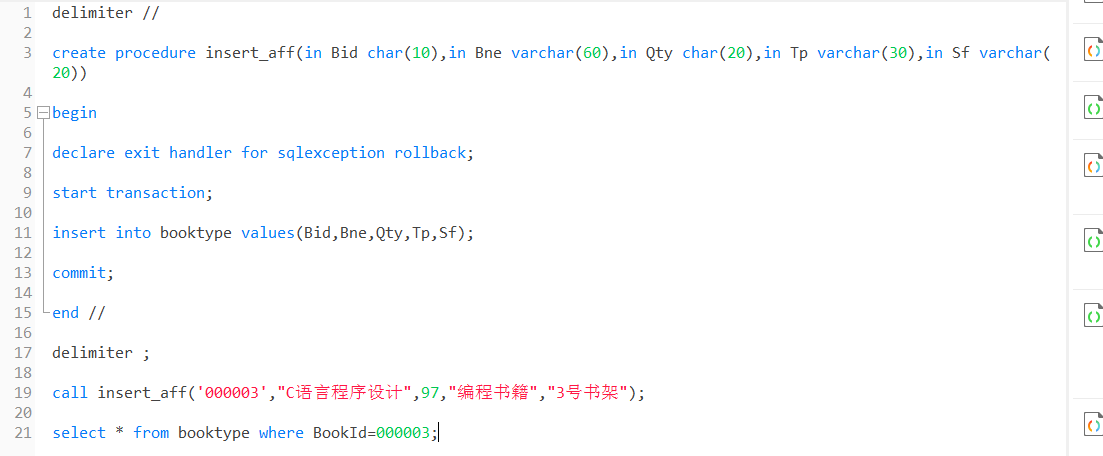
**（1）事务delete\_aff**

创建存储过程，在其中使用事务，当删除借阅表中的信息同时删除归还表的对应信息。

****

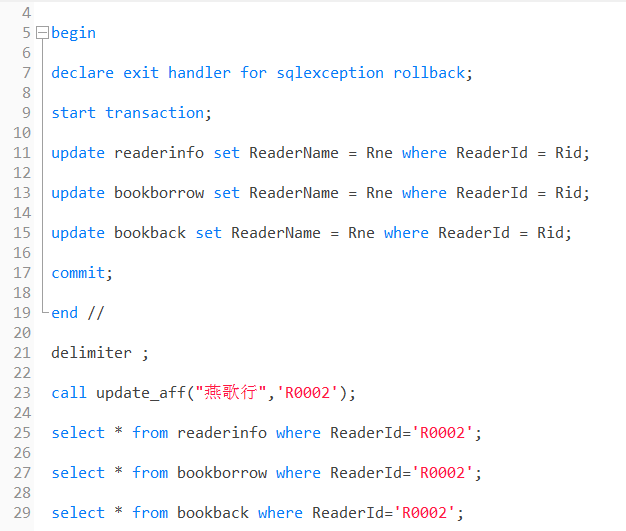
**（2）事务insert\_aff**

创建存储过程，在其中使用事务，向图书类型表中插入数据。

****

**（3）事务update\_aff**

创建存储过程，在其中使用事务，当修改读者信息表中的读者姓名同时修改借阅表和归还表中的读者姓名



**5.5事件**

事件是可以根据计划运行的任何任务，现在开始并每两年执行一次，把逾期未归还的读者的IfBack改为黑名单。

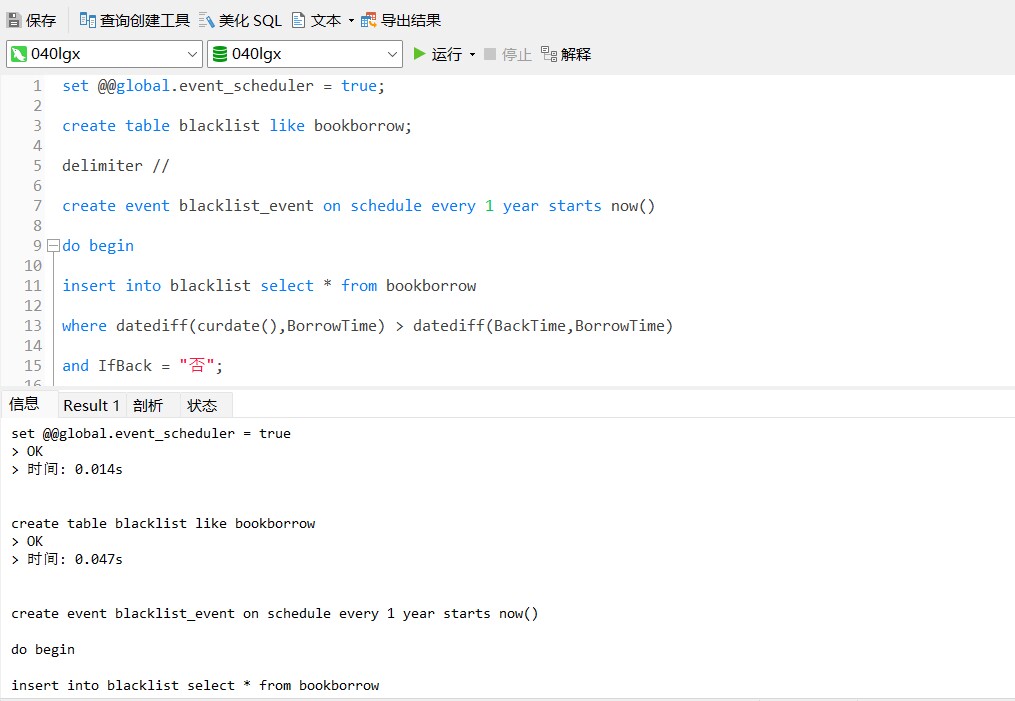
**（1）事件update\_event**

创建事件，现在开始并每一年执行一次，把逾期未归还的读者信息插入blacklist表中。



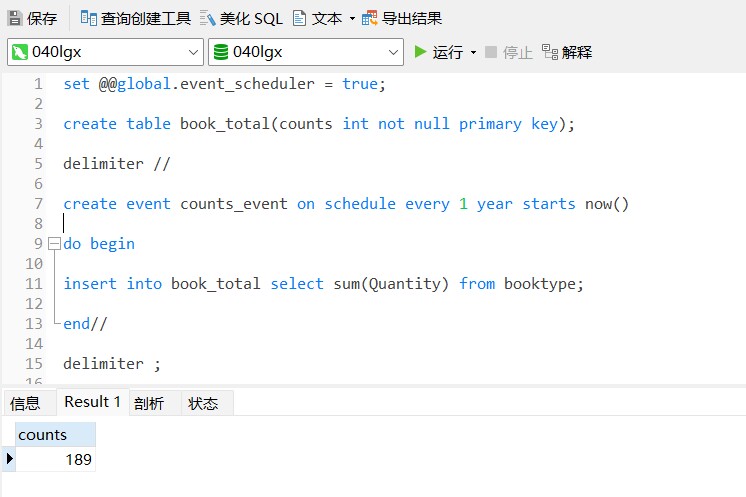
**（2）事件clean\_event**

创建事件，现在开始并每一年执行一次，合计图书总数量。



**（3）事件blacklist\_event**

创建事件，清空readerinfo表，现在开始并每15S执行一次，在规定时间终止。



**（4）事件delback\_event**

创建事件，现在开始并每一年执行一次，把变为黑名单的图书归还表信息删掉。



**（5）事件book\_total**

创建事件，现在开始并每一年执行一次，合计图书总数量。



**6.数据库安全**

**6.1数据库用户定义及权限分配**

数据库用户定义及权限分配是指在数据库管理系统中创建用户账户，并为这些用户分配相应的权限，以控制他们对数据库资源的访问和操作。这是数据库安全和访问控制的基础。

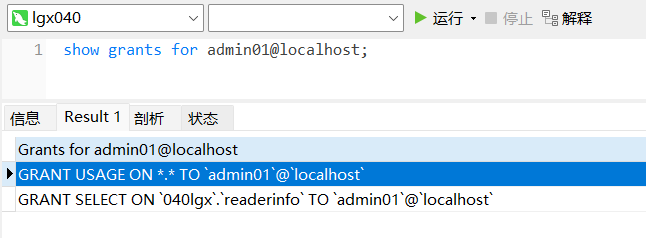
**（1）用户admin01定义及权限分配**

定义用户admin01的sql语句如下：

|  |  |  |
| --- | --- | --- |
| **用户名** | **定义用户语句** | **权限分配与回收语句** |
| **admin01** | **create user 'admin01'@'localhost' identified by '123456';** | **grant select on readerinfo to 'admin01'@'localhost';**  **grant update on bookinfo to 'admin01'@'localhost';**  **revoke update on bookinfo from 'admin01'@'localhost';**  **show grants for admin01@localhost;** |

以上语句授予了用户对读者信息表select权限，对图书信息表update权限。回收图书信息表update权限。

在Navicat中查询admin01的权限如下图，成功授予相关权限。



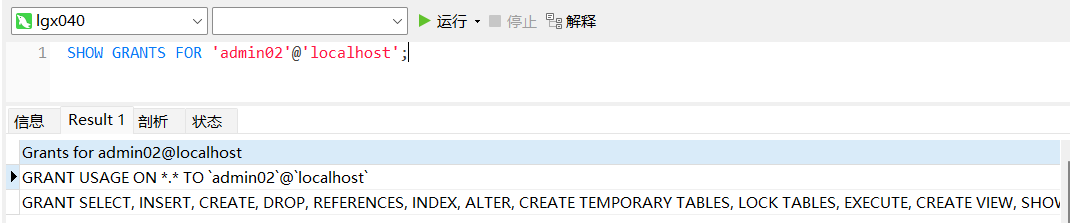
**（2）用户admin02定义及权限分配**

定义admin02的sql语句如下：

|  |  |  |
| --- | --- | --- |
| **用户名** | **定义用户语句** | **权限分配与回收语句** |
| **admin02** | **create user 'admin02'@'localhost' identified by '123456';** | **grant all on manager to 'admin02'@'localhost' identified by '123456' with grant option;**  **revoke update,delete on manager from 'admin02'@'localhost';**  **show grants for admin02@localhost;** |

以上语句授予了用户对管理员信息表的所有权限，并允许授予其他用户。回收修改和删除权限。

在Navicat中查询admin01的权限如下图，成功授予相关权限。

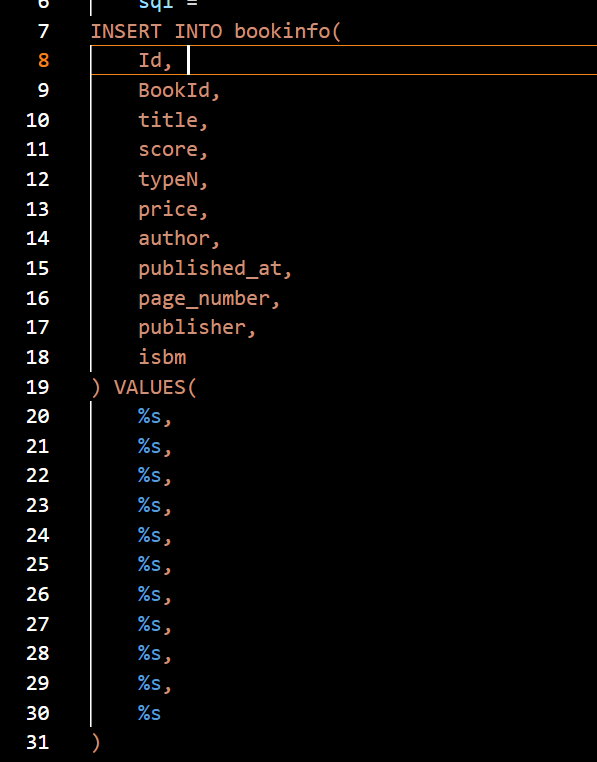


**6.2参数化查询抵抗sql注入攻击**

SQL注入是比较常见的网络攻击方式之一，它不是利用操作系统的BUG来实现攻击，而是针对程序员编写时的疏忽，通过SQL语句，实现无账号登录，甚至篡改数据库。

而使用参数化查询可以防止SQL注入攻击。在参数化查询中，应用程序将用户输入视为参数，而不是SQL代码的一部分。参数会被转义并传递到数据库中，因此不会被认为是SQL代码。参数化查询可以防止大多数SQL注入攻击。

利用python来进行数据库的增删改查sql命令，这里使用参数占位符来表示外部输入，在执行查询时被安全地传递给数据库。

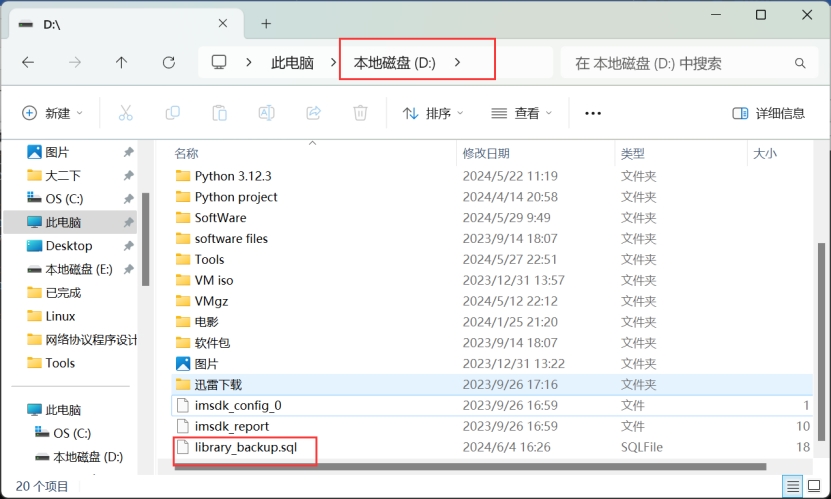
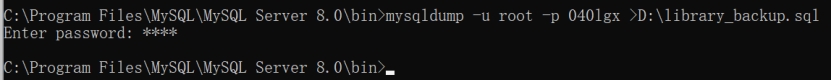


**6.3备份与恢复**

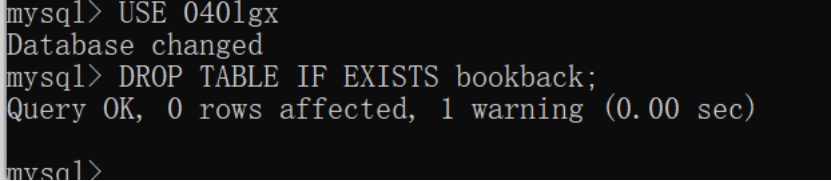
备份的目的是在原始数据发生损坏、丢失或遭到破坏时，能够恢复数据到一个已知的良好状态。可以提供更好的数据保护和容灾能力

**（1）例子一**

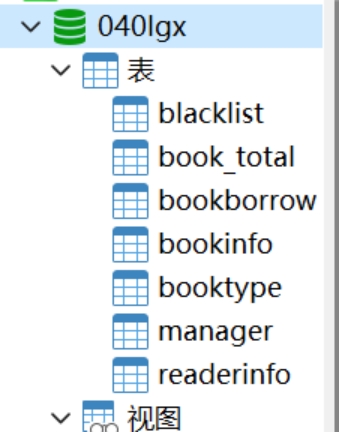
首先备份，使用mysqldump命令备份040lgx数据库中的所有表到D盘。

****

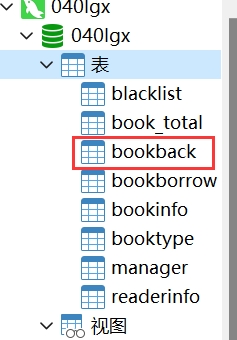
然后删除一张数据库中的表，模拟原始数据损坏。



此时已经没有bookback这张表了。



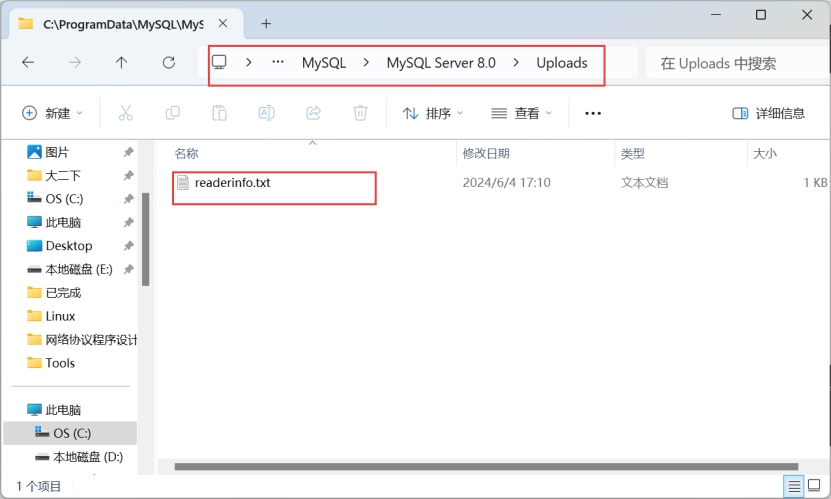
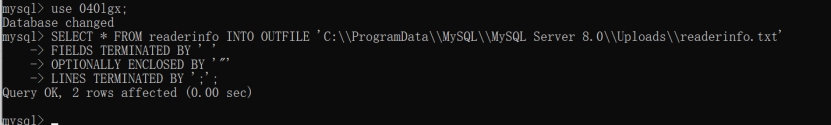
接着使用mysql或source命令将备份文件恢复到数据库，恢复bookback这张表。



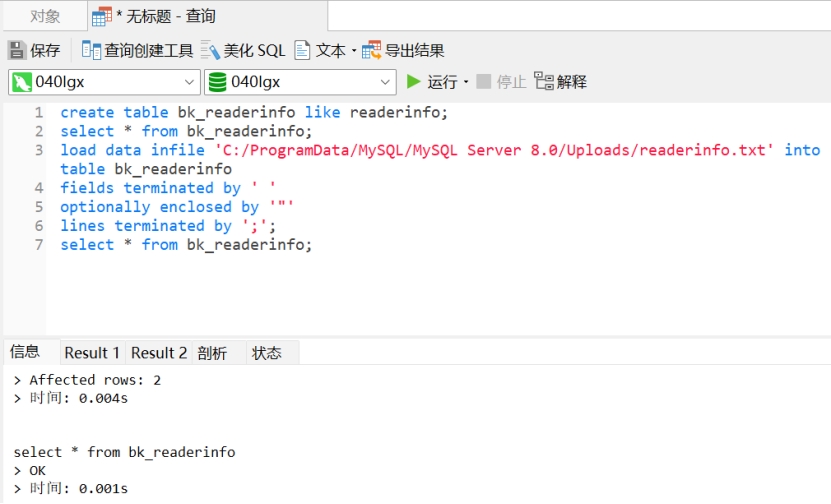
恢复成功

**（2）例子二**

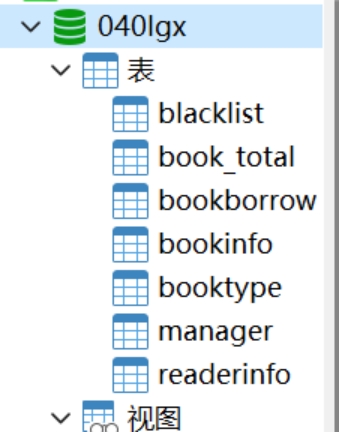
我们还可以导出备份好的表，之后再重新导入。首先导出readerinfo读者信息表的数据为txt文件。

****

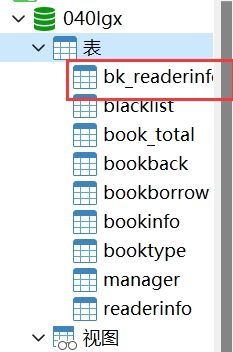
接着创建bk\_readerinfo表，将导出的数据导入到该表中



原先是readerinfo表。



现在成功导入readerinfo表的备份bk\_readerinfo表。



**7.小组成员及分工**

综上我们完成了一个图书馆管理系统的数据库安全实训，可能选题上与大家重合的比较多，并且也没有做前端，但我们认为前端应该在Java课程中实现，作为数据库安全这门课的实训报告，我们还是更注重数据库创建与数据库安全本身，我们力争将这两部分做深做透。

小组成员与分工：

李文钦 实训选题 ER图绘制 资料寻找 分布式创建数据库 爬虫 PPT展示 实验报告攥写

钟智恺 资料寻找 创建数据库 实验报告攥写

李贵雄 资料寻找 爬虫 创建数据库

蔡展宇 创建数据库ER图绘制