

浙江大学

数据库系统实验报告

作业名称：数据库程序设计

姓 名：林馨怡

学 号：3170101791

电子邮箱：xin751687372@163.com

联系电话：17326088652

指导老师：孙建伶

2019 年 4 月 17 日

实验名称

一、 实验目的

1. 掌握数据库应用开发程序设计方法

二、 系统需求

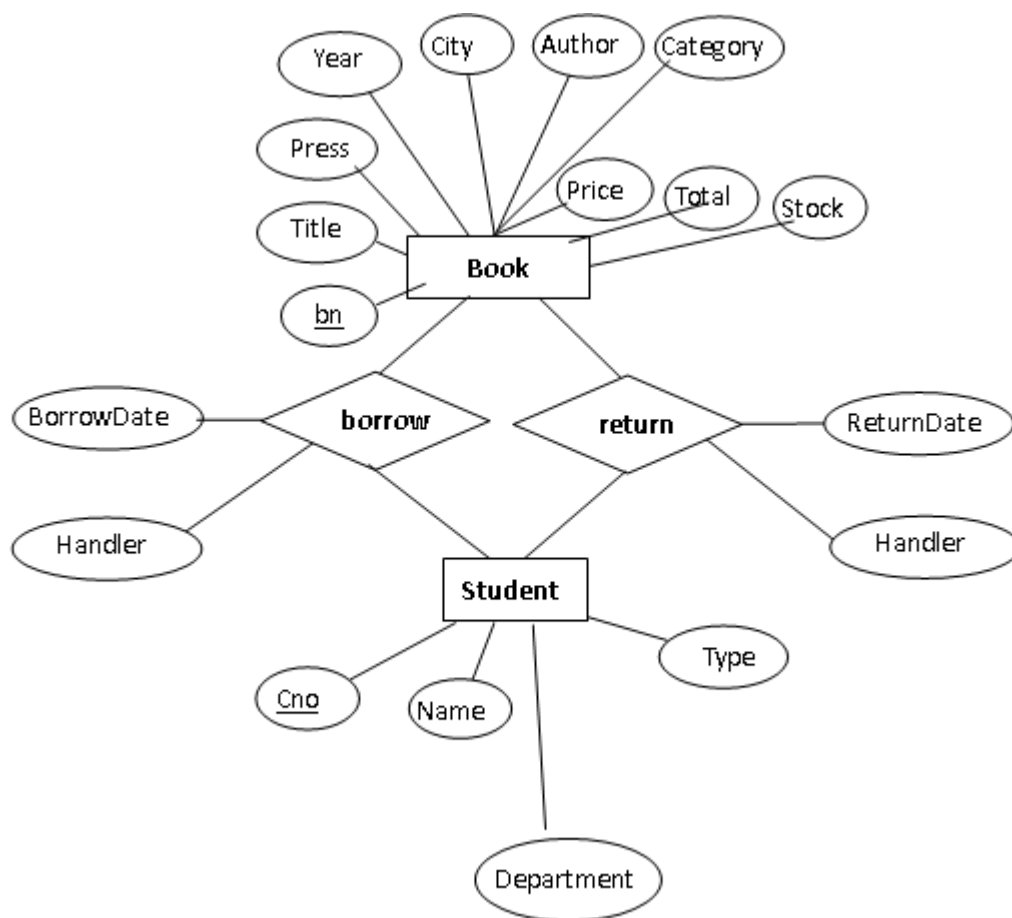
1. 实现图书查询，包括查询全部和对书编号、书名的查询
2. 实现图书借阅，借阅时输入借阅证编号，借阅后相应的书的库存量减少
3. 实现图书归还，归还时输入借阅证编号，归还后相应的书的库存量增加
4. 实现图书插入，插入时需输入图书的所有信息，系统依据图书编号先判断库中是否已有该书信息，如有则更改库中该书库存使其加一，否则添加一条新书的记录
5. 实现借阅证的管理，包括依据借阅证编号进行的借阅证的删除、插入和修改，其中借阅证的修改可以实现修改借阅证姓名、借阅证所属系，以及借阅证类型，但无法修改借阅证编号

三、 实验环境

1. 数据库管理系统：MySQL
2. 开发语言：C
3. 开发工具：Visual Studio 2017
4. 交互界面：控制台字符界面

四、 系统设计与实现

1. 实体之间的关系 E-R 图



2. 数据库逻辑结构设计

```
create table book
(bno char(10),
category varchar(10),
title varchar(20),
press varchar(20),
year int,
author varchar(10),
price decimal(7,2),
total int,
stock int,
primary key(bno));
```

```
create table card
(cno char(7),
```

```
name varchar(10),
department varchar(40),
type char(1),
primary key(cno),
check(type in('T','S')));

create table borrow
(cno char(7),
bno char(8),
borrow_date int,
return_date int,
primary key(cno,bno),
foreign key (cno) references card(cno),
foreign key (bno) references book(bno));
```

3. 程序运行结果场景以及截图说明

a. 图书入库和修改:

1. 检验库中是否已存在编号相同的书的信息

先调用 SQLAllocStmt 函数分配语句句柄，再输入想要插入的图书信息。

至于为什么要分配句柄，是因为通常使用 ODBC API 开发数据库应用程序需要经过如下步骤：

- 连接数据源。
- 分配语句句柄。
- 准备并执行 SQL 语句。
- 获取结果集。
- 提交事务。
- 断开数据源连接并释放环境句柄。

为了连接数据源，必须要建立一个数据源连接的环境句柄。通过调用 SQLAllocEnv 函数实现对环境句柄的分配，在 ODBC 3.0 里，这个函数已经被 SQLAllocHandle 取代，但是熟悉 ODBC API 的开发人员还是习惯用这个函数建

立环境句柄，因为 VC++ 开发平台有一个映射服务，这个服务将程序代码对函数 SQLAllocEnv 的调用转向对函数 SQLAllocHandle 的调用。

句柄是指向一个特殊结构的指针，而环境指的是驱动程序管理器需要为该驱动程序存储的有关系统和数据源的一般信息。由于这个时候还没有建立同数据源的连接，驱动程序还并不知道该使用哪一个驱动程序来完成这个任务，所以这个任务只能由驱动程序管理器来完成，利用这个环境句柄保留信息直到被使用。

句柄是指向一个特殊结构的指针，而环境指的是驱动程序管理器需要为该驱动程序存储的有关系统和数据源的一般信息。由于这个时候还没有建立同数据源的连接，驱动程序还并不知道该使用哪一个驱动程序来完成这个任务，所以这个任务只能由驱动程序管理器来完成，利用这个环境句柄保留信息直到被使用。

接下来，我将要执行的 SQL 语句 select 以字符串形式赋值给指针 query, 在 query 语句中，未知的要查询的图书编号用 “?” 来代替，绑定参数后调用 SQLExecute 函数执行该 SQL 语句。如果找到了该书的编号，就用 update 语句将该书的库存加一。

插入前的 book 表

Result Grid									
		Filter Rows:		Edit:		Export/Import:		Wrap Cell C	
	bno	category	title	press	year	author	price	total	stock
▶	00000001	A1	Database_system	China Machine	2012	Abraham	99.00	24893	22305
	00000002	A1	Data_structure	press_1	2019	author_1	98.00	1	0
	00000003	A1	CG	press_2	2019	author_2	97.00	2	4
	00000004	A1	ADS	press_3	2019	author_3	96.00	1	1
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

插入的图书信息

```
1. 图书查询 2. 借书 3. 还书 4. 图书入库 5. 借阅证管理 0. 退出系统
请输入需要的服务编号：
4
请输入书名, 作者, 出版社, 价格
00000002 A1 Data_structure press_1 2019 author_1 98.00 1 1
入库成功!
```

插入后的 book 表

	bno	category	title	press	year	author	price	total	stock
▶	00000001	A1	Database_system	China Machine	2012	Abraham	99.00	24893	22305
	00000002	A1	Data_structure	press_1	2019	author_1	98.00	1	1
	00000003	A1	CG	press_2	2019	author_2	97.00	2	4
	00000004	A1	ADS	press_3	2019	author_3	96.00	1	1
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

可以看到，编号为 00000002 的图书库存比原来多了 1。

2. 库中不存在编号相同的书的信息

先调用 SQLPrepare 函数，再调用 SQLBindParameter 函数对每一个属性的参数进行绑定，然后执行插入语句 insert，将所有信息输入到库中。

插入前的 book 表

	bno	category	title	press	year	author	price	total	stock
▶	00000001	A1	Database_system	China Machine	2012	Abraham	99.00	24893	22305
	00000002	A1	Data_structure	press_1	2019	author_1	98.00	1	1
	00000003	A1	CG	press_2	2019	author_2	97.00	2	4
	00000004	A1	ADS	press_3	2019	author_3	96.00	1	1
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

插入的图书信息

```

*****
                        图书管理系统
*****
1. 图书查询 2. 借书 3. 还书 4. 图书入库 5. 借阅证管理 0. 退出系统
请输入需要的服务编号：
4
请输入书名, 作者, 出版社, 价格
00000005 A1 Python press_4 2019 author_4 67.00 1 1
入库成功!
  
```

插入后的 book 表

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell

	bno	category	title	press	year	author	price	total	stock
▶	00000001	A1	Database_system	China Machine	2012	Abraham	99.00	24893	22305
	00000002	A1	Data_structure	press_1	2019	author_1	98.00	1	1
	00000003	A1	CG	press_2	2019	author_2	97.00	2	4
	00000004	A1	ADS	press_3	2019	author_3	96.00	1	1
	00000005	A1	Python	press_4	2019	author_4	67.00	1	1
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

可以看到，编号为 00000005 的图书信息被成功插入 book 表中。

b. 借书：

1. 检验要借阅的图书库存是否为 0

输入要借阅的图书编号和借阅证编号。先调用 SQL 的 select 语句查询该编号的图书库存是否为 0。如果库存不为 0，就调用 update 语句将该书的库存减一，并显示借阅成功。

借阅前的 book 表

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell

	bno	category	title	press	year	author	price	total	stock
▶	00000001	A1	Database_system	China Machine	2012	Abraham	99.00	24893	22305
	00000002	A1	Data_structure	press_1	2019	author_1	98.00	1	1
	00000003	A1	CG	press_2	2019	author_2	97.00	2	4
	00000004	A1	ADS	press_3	2019	author_3	96.00	1	1
	00000005	A1	Python	press_4	2019	author_4	67.00	1	1
✱	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

借阅图书的语句

```
*****
图书管理系统
*****
1. 图书查询 2. 借书 3. 还书 4. 图书入库 5. 借阅证管理 0. 退出系统
请输入需要的服务编号：
2
请输入选择：1. 借阅的书名及借阅证编号 0. 退出系统
1
00000001 0000002
借阅成功！
*****
```

借阅后的 book 表

	bno	category	title	press	year	author	price	total	stock
▶	00000001	A1	Database_system	China Machine	2012	Abraham	99.00	24893	22304
	00000002	A1	Data_structure	press_1	2019	author_1	98.00	1	1
	00000003	A1	CG	press_2	2019	author_2	97.00	2	4
	00000004	A1	ADS	press_3	2019	author_3	96.00	1	1
	00000005	A1	Python	press_4	2019	author_4	67.00	1	2
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

可以看到，编号为 00000001 的图书库存少了 1。

2. 要借阅的图书库存为 0，借阅失败

如果查询后用 SQLFetch 函数取出来的元组库存为 0，则显示借阅失败。

借阅前的 book 表

	bno	category	title	press	year	author	price	total	stock
▶	00000001	A1	Database_system	China Machine	2012	Abraham	99.00	24893	22304
	00000002	A1	Data_structure	press_1	2019	author_1	98.00	1	0
	00000003	A1	CG	press_2	2019	author_2	97.00	2	4
	00000004	A1	ADS	press_3	2019	author_3	96.00	1	1
	00000005	A1	Python	press_4	2019	author_4	67.00	1	2
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

借阅图书的语句

```
1. 图书查询 2. 借书 3. 还书 4. 图书入库 5. 借阅证管理 0. 退出系统
请输入需要的服务编号：
2
请输入选择：1. 借阅的书名及借阅证编号 0. 退出系统
1
00000002 00000002
该书库存为零！
```


借阅后的 book 表

	bno	category	title	press	year	author	price	total	stock
▶	00000001	A1	Database_system	China Machine	2012	Abraham	99.00	24893	22304
	00000002	A1	Data_structure	press_1	2019	author_1	98.00	1	0
	00000003	A1	CG	press_2	2019	author_2	97.00	2	4
	00000004	A1	ADS	press_3	2019	author_3	96.00	1	1
	00000005	A1	Python	press_4	2019	author_4	67.00	1	2
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

可以看到，由于编号为 00000002 的图书借阅前库存为 0，无法借阅该书，显示借阅失败。

c. 还书：

输入要归还的书的编号和借阅证编号，如果归还成功就显示归还成功。

归还前的 book 表

	bno	category	title	press	year	author	price	total	stock
▶	00000001	A1	Database_system	China Machine	2012	Abraham	99.00	24893	22304
	00000002	A1	Data_structure	press_1	2019	author_1	98.00	1	0
	00000003	A1	CG	press_2	2019	author_2	97.00	2	4
	00000004	A1	ADS	press_3	2019	author_3	96.00	1	1
	00000005	A1	Python	press_4	2019	author_4	67.00	1	2
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

归还图书的语句

```

*****
                        图书管理系统
*****
1. 图书查询 2. 借书 3. 还书 4. 图书入库 5. 借阅证管理 0. 退出系统
请输入需要的服务编号：
3
请输入选择：1. 归还的书名及借阅证编号 0. 退出系统
1
00000003 0000002
归还成功！

```

归还后的 book 表

Result Grid									
		Filter Rows:		Edit:		Export/Import:		Wrap Cell	
	bno	category	title	press	year	author	price	total	stock
▶	00000001	A1	Database_system	China Machine	2012	Abraham	99.00	24893	22304
	00000002	A1	Data_structure	press_1	2019	author_1	98.00	1	0
	00000003	A1	CG	press_2	2019	author_2	97.00	2	5
	00000004	A1	ADS	press_3	2019	author_3	96.00	1	1
	00000005	A1	Python	press_4	2019	author_4	67.00	1	2
✱	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

可以看到，编号为 00000003 的图书库存比原来多了 1，显示归还成功。

d. 借书证管理：

1. 借书证的删除

输入要删除的借书证编号，并调用 SQL 的 delete 语句将该编号的借书证删除，并显示删除成功。

删除前的 card 表

Result Grid				
		Filter Rows:		
	cno	name	department	type
▶	00000002	Liuxian	Computer	S
	00000003	Linhaotong	Computer	S
✱	NULL	NULL	NULL	NULL

删除借阅证语句

```
1. 图书查询 2. 借书 3. 还书 4. 图书入库 5. 借阅证管理 0. 退出系统
请输入需要的服务编号：
5
1. 删除借阅证 2. 增加借阅证 3. 借阅证修改 0. 退出系统
1
请输入要删除的借阅证编号
00000003
删除借阅证成功！
```

删除后的 card 表

Result Grid				
	cno	name	department	type
▶	0000002	Liuxian	Computer	S
*	NULL	NULL	NULL	NULL

2. 借书证的增加

输入要添加的借书证编号、姓名、系名和类型，使用预备语句并执行 SQL 的 insert 语句对 card 表进行插入。

插入前的 card 表

Result Grid				
	cno	name	department	type
▶	0000001	Linxinyi	Computer	S
	0000002	Liuxian	Computer	S
	NULL	NULL	NULL	NULL

插入借阅证的语句

```
*****
                        图书管理系统
*****
1. 图书查询 2. 借书 3. 还书 4. 图书入库 5. 借阅证管理 0. 退出系统
请输入需要的服务编号：
5
1. 删除借阅证 2. 增加借阅证 3. 借阅证修改 0. 退出系统
2
0000003 Linhaotong Computer S
增加借阅证成功！
```

插入后的 card 表

Result Grid				
	cno	name	department	type
▶	0000001	Linxinyi	Computer	S
	0000002	Liuxian	Computer	S
	0000003	Linhaotong	Computer	S
	NULL	NULL	NULL	NULL

3. 借书证的修改 (编号)

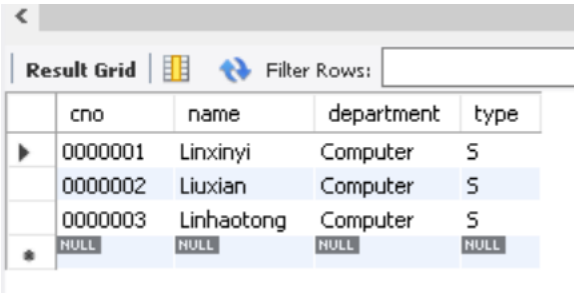
借书证的编号应无法修改，因此不让输入要修改的借书证编号。

```
*****
                        图书管理系统
*****
1. 图书查询 2. 借书 3. 还书 4. 图书入库 5. 借阅证管理 0. 退出系统
请输入需要的服务编号：
5
1. 删除借阅证 2. 增加借阅证 3. 借阅证修改 0. 退出系统
3
请选择要修改的内容：1. 编号 2. 姓名 3. 部门 4. 类型
1
无法修改借阅证编号！
```

4. 借书证的修改 (姓名)

输入要修改的借阅证的编号和新的姓名，再调用 SQL 的 update 语句对该编号的借阅证进行修改。

修改前的 card 表



	cno	name	department	type
▶	0000001	Linxinyi	Computer	S
	0000002	Liuxian	Computer	S
	0000003	Linhaotong	Computer	S
*	NULL	NULL	NULL	NULL

修改借阅证的语句

```
1. 图书查询 2. 借书 3. 还书 4. 图书入库 5. 借阅证管理 0. 退出系统
请输入需要的服务编号：
5
1. 删除借阅证 2. 增加借阅证 3. 借阅证修改 0. 退出系统
3
请选择要修改的内容：1. 编号 2. 姓名 3. 部门 4. 类型
2
请输入借阅证编号和新的姓名
0000003 dhwkk
修改成功！
```

修改后的 card 表

Result Grid				
Filter Rows:				
	cno	name	department	type
▶	0000001	Linxinyi	Computer	S
	0000002	Liuxian	Computer	S
	0000003	dhwkk	Computer	S
✱	NULL	NULL	NULL	NULL

可以看到，编号为 0000003 的借阅证的姓名变成了 dhwkk，修改成功。

5. 借书证的修改（系名）

输入要修改的借阅证的编号和新的系名，再调用 SQL 的 update 语句对该编号的借阅证进行修改。

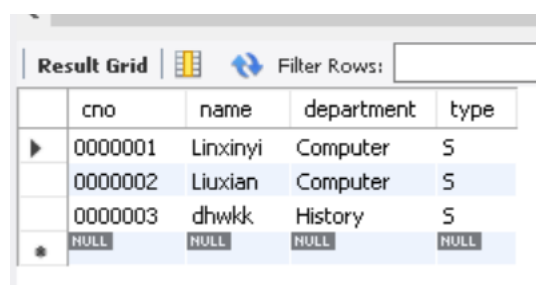
修改前的 card 表

Result Grid				
Filter Rows:				
	cno	name	department	type
▶	0000001	Linxinyi	Computer	S
	0000002	Liuxian	Computer	S
	0000003	dhwkk	Computer	S
✱	NULL	NULL	NULL	NULL

修改借阅证的语句

```
*****
                        图书管理系统
*****
1. 图书查询 2. 借书 3. 还书 4. 图书入库 5. 借阅证管理 0. 退出系统
请输入需要的服务编号：
5
1. 删除借阅证 2. 增加借阅证 3. 借阅证修改 0. 退出系统
3
请选择要修改的内容：1. 编号 2. 姓名 3. 部门 4. 类型
3
请输入借阅证编号和新的部门
0000003 History
修改成功！
```

修改后的 card 表



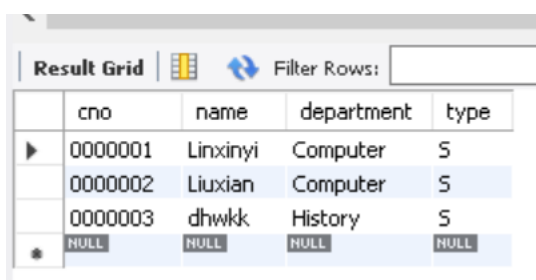
	cno	name	department	type
▶	0000001	Linxinyi	Computer	S
	0000002	Liuxian	Computer	S
	0000003	dhwkk	History	S
•	NULL	NULL	NULL	NULL

可以看到，编号为 0000003 的借阅证系名改为了 History，修改成功。

6. 借书证的修改（类型）

输入要修改的借阅证的编号和新的系名，再调用 SQL 的 update 语句对该编号的借阅证进行修改。

修改前的 card 表



	cno	name	department	type
▶	0000001	Linxinyi	Computer	S
	0000002	Liuxian	Computer	S
	0000003	dhwkk	History	S
•	NULL	NULL	NULL	NULL

修改借阅证的语句

```
1. 图书查询 2. 借书 3. 还书 4. 图书入库 5. 借阅证管理 0. 退出系统
请输入需要的服务编号：
5
1. 删除借阅证 2. 增加借阅证 3. 借阅证修改 0. 退出系统
3
请选择要修改的内容：1. 编号 2. 姓名 3. 部门 4. 类型
4
请输入借阅证编号和新的类型
0000003 T
修改成功！
```

修改后的 card 表

Result Grid				
Filter Rows:				
	cno	name	department	type
▶	0000001	Linxinyi	Computer	S
	0000002	Liuxian	Computer	S
	0000003	dhwkk	History	T
✱	NULL	NULL	NULL	NULL

可以看到，编号为 0000003 的借阅证的类型变为了 T，修改成功。

e. 图书查询：

1. 查询全部

先调用 `SQLAllocStmt` 函数分配语句句柄，再输入想要插入的图书信息，将要执行的 SQL 语句 `select` 以字符串形式赋值给指针 `query`，然后调用 `SQLExecute` 函数执行该 SQL 语句。调用 `SQLBindCol` 函数将每一列属性与预设变量——绑定，并不断用 `SQLFetch` 函数取出每一个表 `book` 中的元组，将每一列属性的值打印出来。

查询全部语句

```
*****
***** 图书管理系统 *****
*****
1. 图书查询 2. 借书 3. 还书 4. 图书入库 5. 借阅证管理 0. 退出系统
请输入需要的服务编号：
1
1. 查询全部 2. 按书名查询 3. 按书号查询 0. 退出
请输入需要的服务编号
1
查询结果如下
*****
书号      类别      书名      出版社      年份      作者      价格      总藏书量      库存
*****
00000001      A1      Database_system China Machine      2012      Abraham      99.00      24893      22304
00000002      A1      Data_structure press_1 2019      author_1      98.00      1      0
00000003      A1      CG      press_2 2019      author_2      97.00      2      5
00000004      A1      ADS      press_3 2019      author_3      96.00      1      1
00000005      A1      Python      press_4 2019      author_4      67.00      1      2
*****
```

2. 按书名查询

输入要查询的图书的名字进行查询。

按书名查询语句

```
1. 查询全部 2. 按书名查询 3. 按书号查询 0. 退出
请输入需要的服务编号
2
CG
查询结果如下
*****
书号 类别 书名 出版社 年份 作者 价格 总藏书量 库存
*****
00000003 A1 CG press_2 2019 author_2 97.00 2 5
```

3. 按书号查询

输入要查询的图书的编号进行查询。

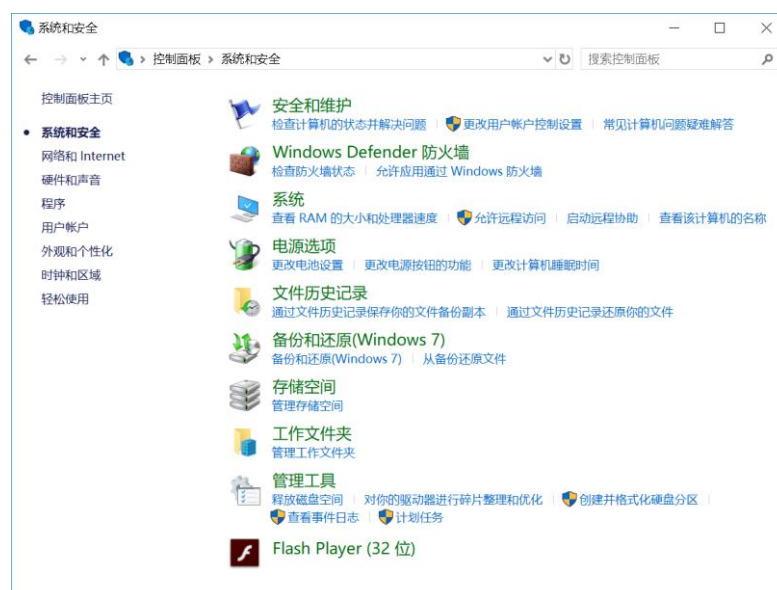
按书号查询语句

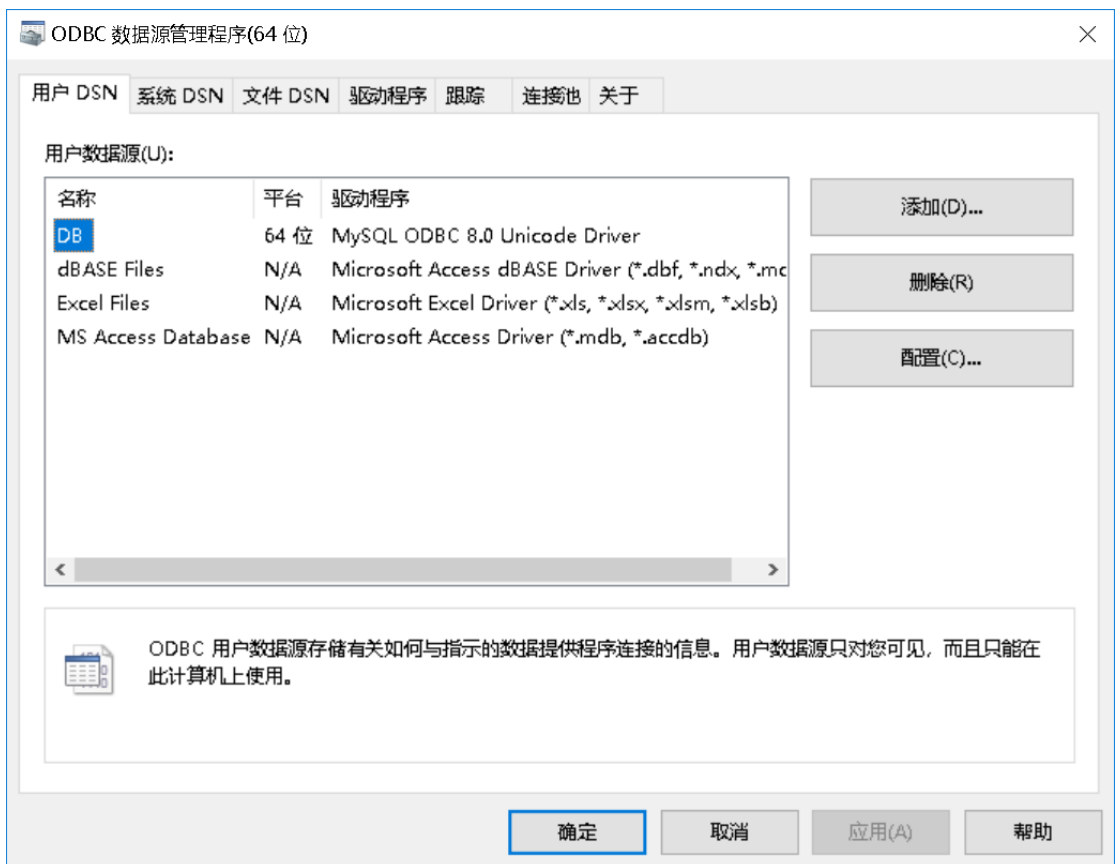
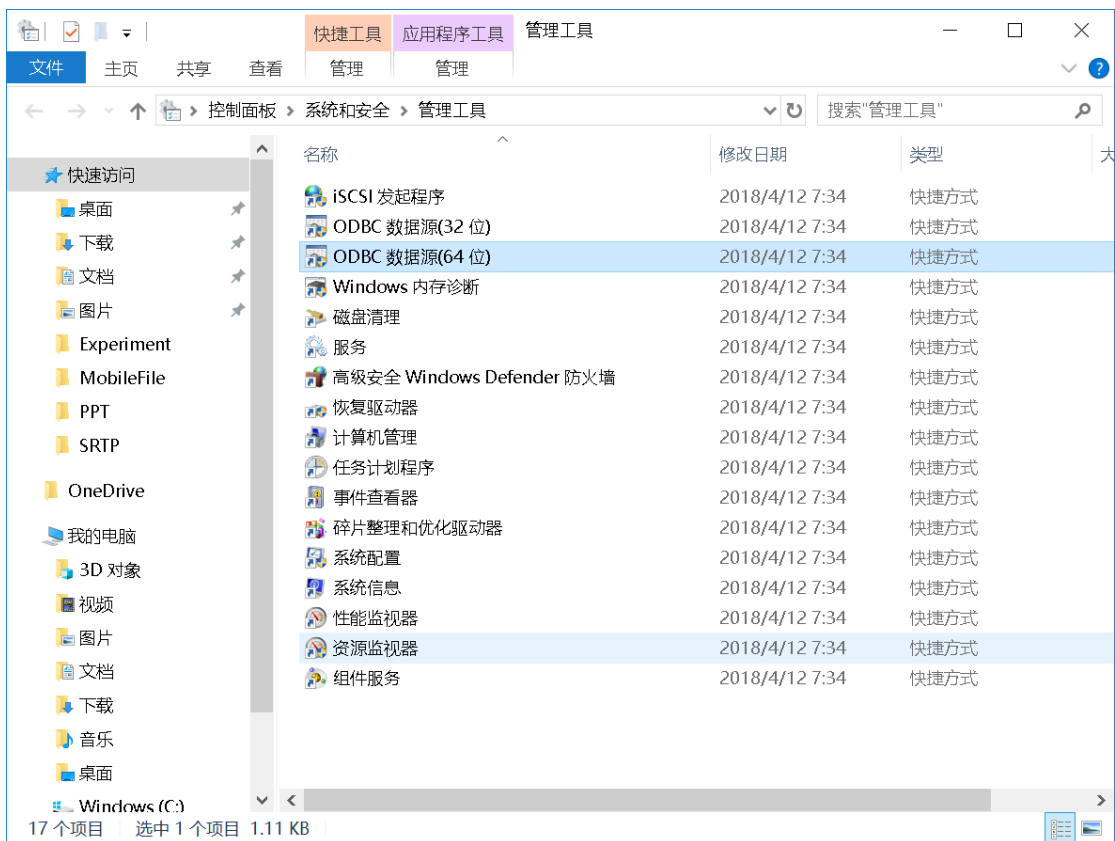
```
1. 查询全部 2. 按书名查询 3. 按书号查询 0. 退出
请输入需要的服务编号
3
00000002
查询结果如下
*****
书号 类别 书名 出版社 年份 作者 价格 总藏书量 库存
*****
00000002 A1 Data_structure press_1 2019 author_1 98.00 1
```

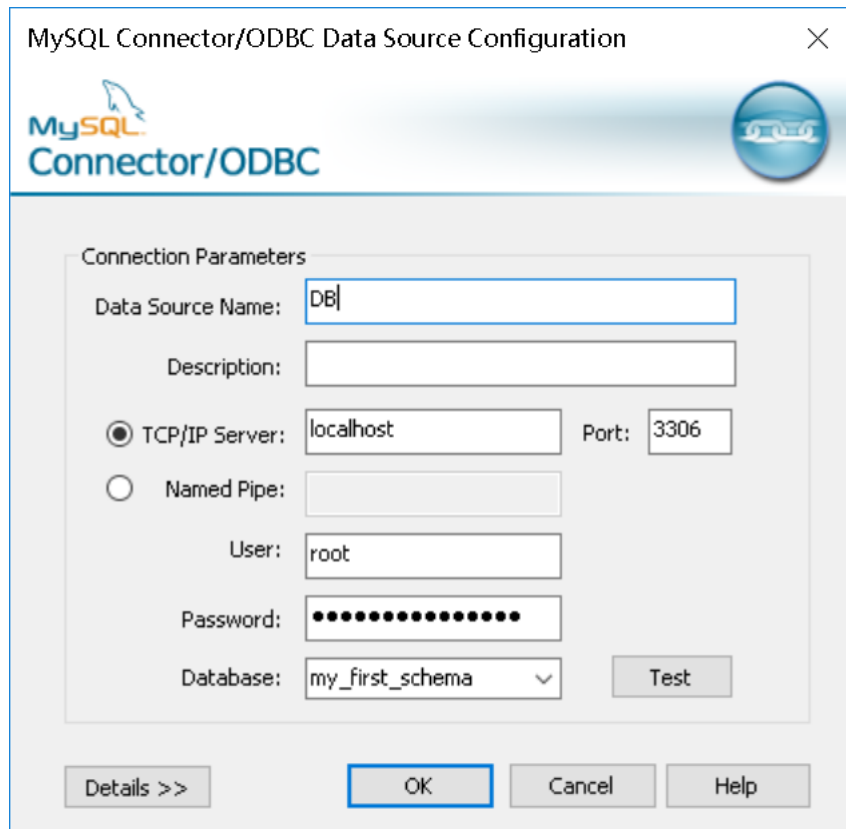
五、 遇到的问题及解决方法

1. 配置环境问题

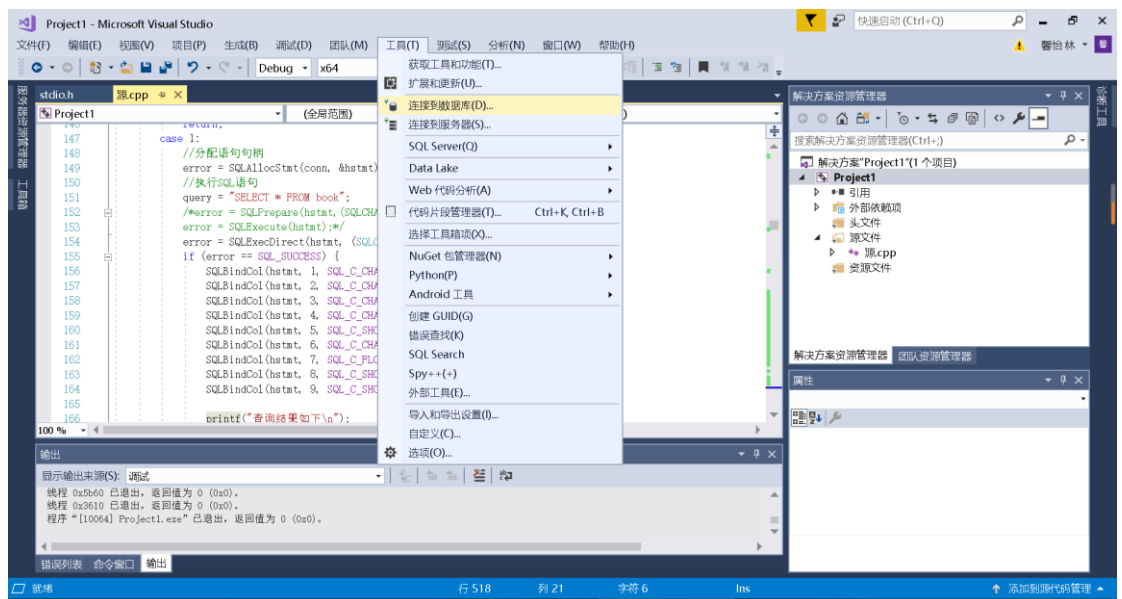
一开始我一直连不上数据库，我在控制面板里添加了数据库，步骤如下：







接下来，我在 MySQL Workbench 里新建了一个模式命名为 my_first_schema，在 VS 里连接到数据库





添加连接

输入信息以连接到选定的数据源，或单击“更改”选择另一个数据源和/或提供程序。

数据源(S):

MySQL Database (MySQL Data Provider) 更改(C)...

Server name: localhost

User name: root

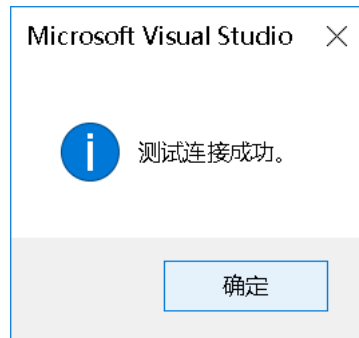
Password: ●●●●●●●●●●

☒ Save my password

Database name: my_first_schema

高级(V)...

测试连接(T) 确定 取消



但是，当我建表的时候，发现依然无法连接到数据库，建表一直不成功。

后来，我发现我应该把 Debug 改成 x64，因为 x64 是 64 位，而 x86 是 32 位，如果我用 32 位的编译器，就无法连接到 64 位的数据库了。



改完之后，成功连接上数据库。

2. 字符串的输入

我在输入数据时，需要连续输入多个字符串，并将每个字符串存储到相应的变量中，这样才能完成 SQL 语句的执行。但是无论我用 `gets_s` 函数、`scanf_s` 函数还是 `sscanf_s` 函数，都无法正确地把变量读入并存储进来。于是，我最后用了 `cin`，成功将变量读入并存储。

六、 总结

我觉得通过这次实验，我更加了解了 ODBC 的运行机制，也知道了该如何对一个数据模式进行具体的数据库操作。ODBC 确实可以完成构建一个数据库的工作，但是可能并不是最佳选择。

同时，我也更加熟悉了 Visual Studio 的使用。Visual Studio 的 `scanf_s` 和 `sscanf_s` 有一定的优势，但在多个连续字符串的读入和存储上有很大的局限性，而这些局限性用 `cin`

就可以成功避免。

在写 ODBC 时，要注意的是不能将 SQL 语句写错，如果写错了，执行结果会变成-1，而刚开始很难发现哪里写错了，只能知道程序不能正确运行。

这次写一个大程序确实让我学到了很多，也对数据库本身有了更深的理解，我也希望通过后续的学习更加深入地了解数据库的其他操作，更好地实现一个完整的数据库。