在做实验五之前，请先仔细阅读5.1节内容。实验五数据库平台SQL Server或MySQL，开发语言C/C++、Java，开发工具任选。

提交物：实验报告（对程序功能运行结果截图说明）、代码打包提交。

关于具体实验内容，参看《实验五-图书管理程序实验指导书》。

下面**以数据库平台MySQL和开发语言Java为例，JDBC标准，开发工具为Eclipse；以数据库平台SQL Server和开发语言C为例，ODBC标准，开发工具为gcc**介绍。

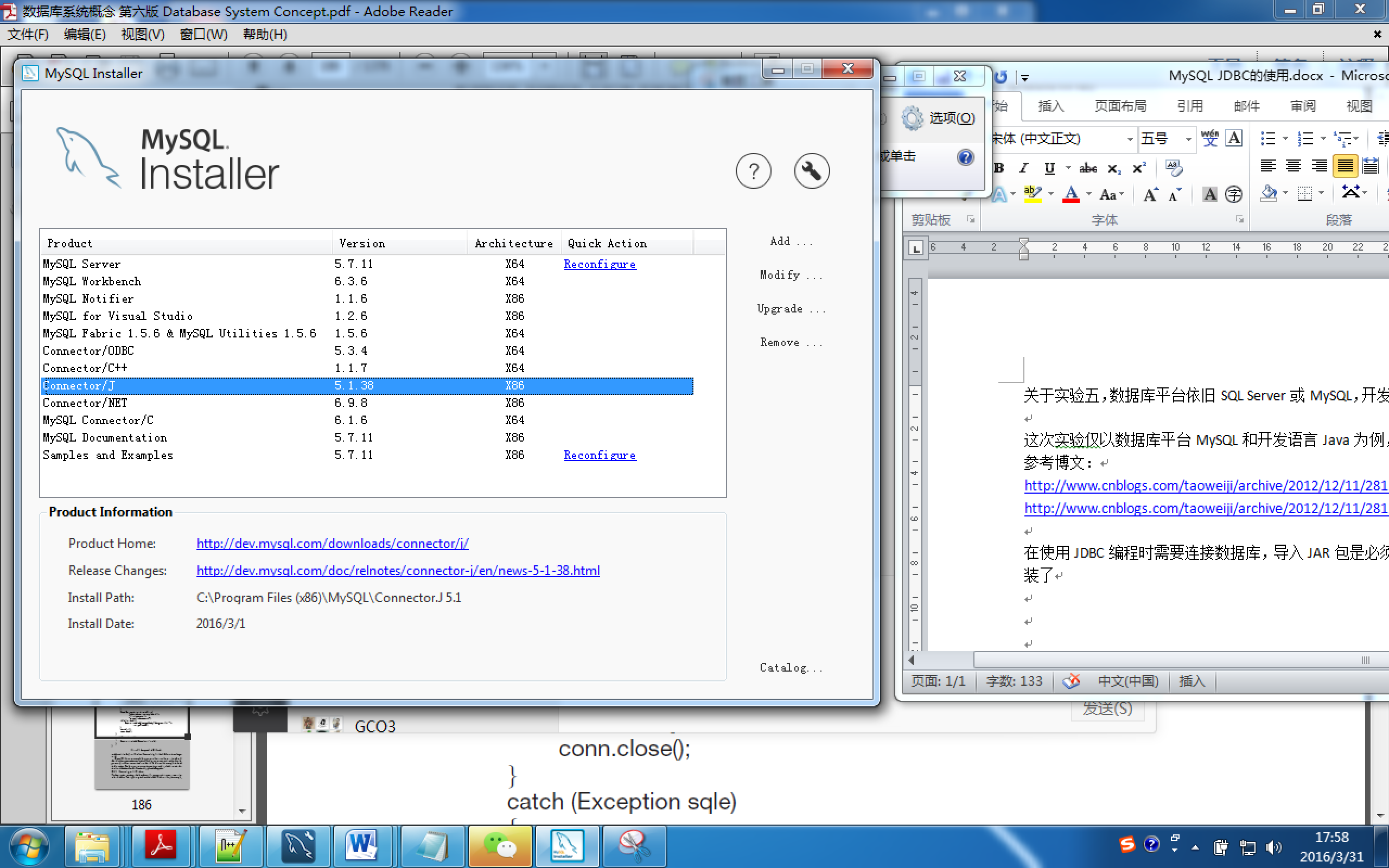
1. **以数据库平台MySQL和开发语言Java为例，JDBC标准，开发工具为Eclipse。**

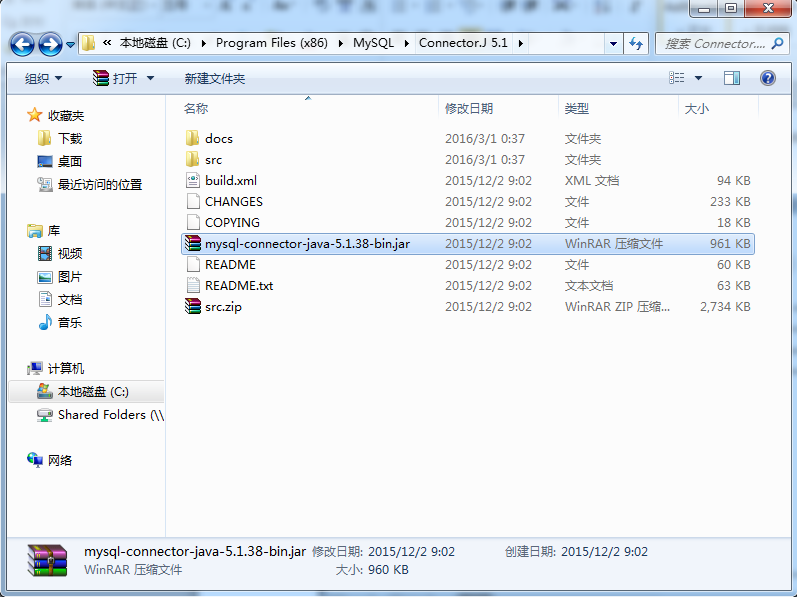
参考博文：

<http://www.cnblogs.com/taoweiji/archive/2012/12/11/2812295.html>

<http://www.cnblogs.com/taoweiji/archive/2012/12/11/2812852.html>

在使用JDBC编程时需要连接数据库，导入JAR包是必须的。如果你在安装MySQL时已经安装了connectors/J，可以找到Install Path，把mysql-connector-java-5.1.38-bin.jar 单独copy到Java项目的目录下，然后用第一篇博文中的方式导入。如果没有安装connectors/J，也可直接去官网下载，<http://dev.mysql.com/downloads/connector/j/>。





JDBC标准定义了Java程序连接数据库服务器的应用程序接口（API）。代码编写请参考第二篇博文和书上Figure 5.1 An example of JDBC code。

如果连接MySQL出现如下警告: Establishing SSL connection without server's identity verification is not recommended. According to MySQL 5.5.45+, 5.6.26+ and 5.7.6+ requirements SSL connection must be established by default if explicit option isn't set…原因是高版本MySQL需要指明是否进行SSL连接。解决方案是在URL中加入ssl=true或者false即可，如String url=”jdbc:mysql://127.0.0.1:3306/newtest?characterEncoding=utf8&useSSL=true”。

如果你需要连接远程MySQL服务器，因为MySQL为了安全性，在默认情况下用户只允许在本地登录，因此为了使其可以远程，用root用户登录MySQL进行授权操作：

GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'%' IDENTIFIED BY 'youpassword' WITH GRANT OPTION;

FLUSH PRIVILEGES;

参考程序：

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**import** java.util.Scanner;

**public** **class** JDBCexample {

**static** Connection *conn*;

**public** **static** **void** main(String args[]){

*start*("root","123456");

}

**public** **static** **void** start(String userid, String passwd) {

**try**{

Class.*forName* ("com.mysql.jdbc.Driver");

*conn* = DriverManager.*getConnection*(

"jdbc:mysql://192.168.2.159:3306/newtest?characterEncoding=utf8&useSSL=true",

userid, passwd);

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.***out***.println("\t\t\t图书管理系统");

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

Scanner reader=**new** Scanner(System.***in***);

**while**(**true**){

System.***out***.println("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");

System.***out***.println("请输入需要的服务编号:");

**int** choice=reader.nextInt();

**switch**(choice)

{

**case** 0 :

*conn*.close();

**return**;

**case** 1:

*check\_Book*();

**break**;

**case** 2 :

*borrow\_Book*();

**break**;

**case** 3:

*return\_Book*();

**break**;

**case** 4:

*add\_Book*();

**break**;

**case** 5:

*proof\_Manag*();

**break**;

**default**:

System.***out***.println("服务编号错误");

}

}

}**catch** (Exception sqle){

System.***out***.println("Exception : " + sqle);

}

}

**static** **void** check\_Book() **throws** SQLException

{

String query;//查询语句

**int** choice=0;//存放用户选项

Scanner reader=**new** Scanner(System.***in***);

**while**(**true**){

System.***out***.println("1.查询全部 2.按书名查询 3.按书号查询 0.退出");

System.***out***.println("请输入需要的服务编号");

choice=reader.nextInt();

**switch**(choice)

{

**case** 0 :

**return**;

**case** 1:

//执行SQL语句

query="SELECT \* FROM book";

Statement stmt = *conn*.createStatement();

ResultSet rset = stmt.executeQuery(query);

System.***out***.println("查询结果如下");

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.***out***.println("书号\t类别\t书名\t出版社\t年份\t作者\t价格\t总藏书量\t库存");

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

**while** (rset.next()) {

System.***out***.println(rset.getString("bno")+"\t"+rset.getString("category")+

"\t"+rset.getString("title")+"\t"+rset.getString("press")+

"\t"+rset.getInt("year")+"\t"+rset.getString("author")+

"\t"+rset.getDouble("price")+"\t"+rset.getInt("total")+"\t"+rset.getInt("stock"));

}

stmt.close();

**break**;

**default**:

System.***out***.println("服务编号错误");

}

}

}

**static** **void** borrow\_Book()

{

System.***out***.println("请输入书名及借阅证编号(0退出系统)");

}

**static** **void** return\_Book()

{

System.***out***.println("请输入书名及借阅证编号(0退出系统)");

}

**static** **void** add\_Book()

{

System.***out***.println("请输入书名,作者,出版社,价格(0退出系统)");

}

**static** **void** proof\_Manag()

{

System.***out***.println("1.删除借阅证 2.增加借阅证 3.借阅证修改 0.退出系统");

}

}

1. **以数据库平台SQL Server和开发语言C为例，ODBC标准，开发工具为gcc。**

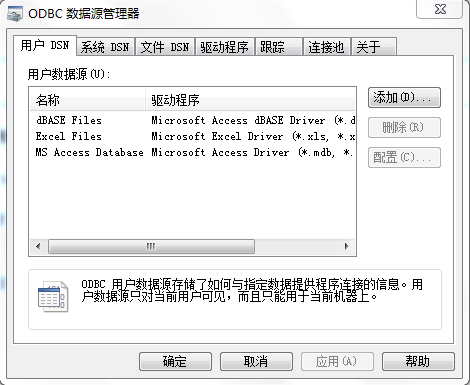
（如果你也是用gcc编译，加上-lodbc32，否则可能会出现

C:\Users\ADMINI~1\AppData\Local\Temp\cc4mkRuf.o:test.c:(.text+0x2c): undefined reference to `SQLAllocEnv@4'等问题。即

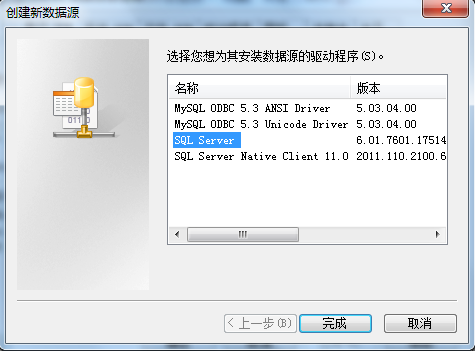
Z:\> gcc -o test test.c -lodbc32）

首先配置ODBC数据源：

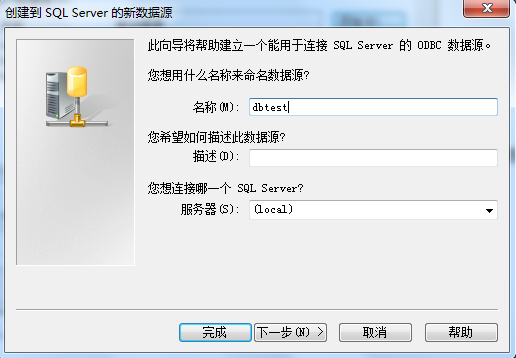
打开 控制面板\系统和安全\管理工具\ODBC数据源管理器



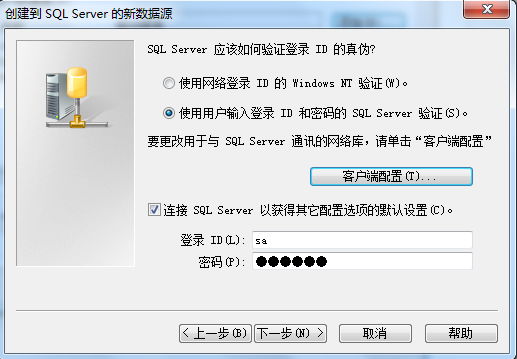
添加（D）…

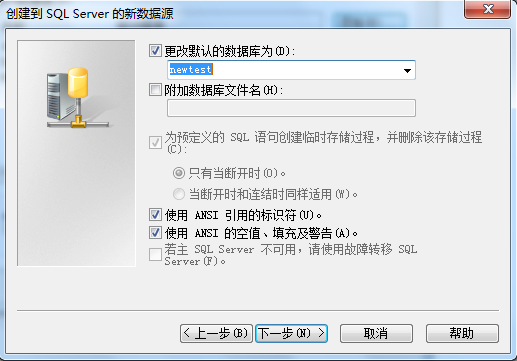


起一个DSN名字，配置SQLSERVER服务器名，由于我使用的是本机数据库，所以服务器中输入(local)，要是连接其他，那么就在服务器中输入IP地址。

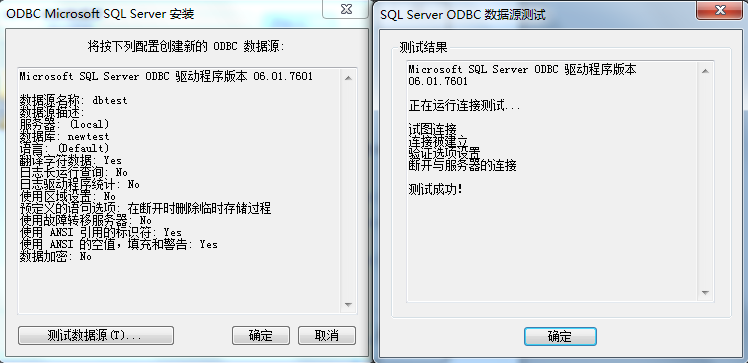


设置登录ID和密码。





测试一下，看是否能够连上。



连接数据库函数

SQLConnect(conn,"dbtest",SQL\_NTS,"sa",SQL\_NTS,"123456",SQL\_NTS);

第二个参数就是DSN名，第四个参数就是登录ID，第六个参数就是密码。

ODBC标准代码编写请参考书上Figure 5.4 ODBC code example。

参考程序：

#include <windows.h>

#include <sql.h>

#include <stdio.h>

#include <sqlext.h>

#include <sqltypes.h>

#define LOGIN\_TIMEOUT 30

#define MAXBUFLEN 255

RETCODE error;

HENV henv = NULL; //环境句柄

HDBC conn = NULL; //连接句柄

HSTMT hstmt = NULL; //语句句柄

void DBTest();

void check\_Book();

void borrow\_Book();

void return\_Book();

void add\_Book();

void proof\_Manag();

int main()

{

DBTest();

return 0;

}

void DBTest()

{

int choice=0;//存放用户选项

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*初始部分\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//分配环境句柄

error=SQLAllocEnv(&henv);

//分配连接句柄

error=SQLAllocConnect(henv, &conn);

//连接数据库

error=SQLConnect(conn,"dbtest",SQL\_NTS,"sa",SQL\_NTS,"123456",SQL\_NTS);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*主体部分\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\t\t\t图书管理系统\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

while(1){

printf("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\n");

printf("请输入需要的服务编号:\n");

scanf("%d",&choice);

switch(choice)

{

case 0 :

//断开连接

SQLDisconnect(conn);

SQLFreeConnect(conn);

SQLFreeEnv(henv);

exit(0);

case 1:

check\_Book();

break;

case 2 :

borrow\_Book();

break;

case 3:

return\_Book();

break;

case 4:

add\_Book();

break;

case 5:

proof\_Manag();

break;

default:

printf("服务编号错误\n");

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*结束部分\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

MessageBox(NULL,TEXT("执行成功"),TEXT("标题"),MB\_OK);

}

void check\_Book()

{

char\* query;//查询语句

int choice=0;//存放用户选项

//存放图书信息的变量

char book\_ID[10]={'0','0','0','0','0','0','0','0','0','0'};

char book\_category[10]={'0','0','0','0','0','0','0','0','0','0'};

char book\_name[20]={'0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0'};

char press[20]={'0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0'};

int year=0;

char writer[10]={'0','0','0','0','0','0','0','0','0','0'};

float price=0.0;

int total\_Amount=0;

int now\_Amount=0;

SQLINTEGER lent1,lent2,lent3,lent4,lent5,lent6,lent7,lent8,lent9;

while(1){

printf("1.查询全部 2.按书名查询 3.按书号查询 0.退出\n");

printf("请输入需要的服务编号\n");

scanf("%d",&choice);

switch(choice)

{

case 0 :

return;

case 1:

//分配语句句柄

error=SQLAllocStmt(conn, &hstmt);

//执行SQL语句

query="SELECT \* FROM book";

//SQLPrepare(hstmt,query,SQL\_NTS);

//SQLExecute(hstmt);

error=SQLExecDirect(hstmt,query,SQL\_NTS);

if (error==SQL\_SUCCESS) {

SQLBindCol(hstmt,1,SQL\_C\_CHAR,book\_ID,10,&lent1);

SQLBindCol(hstmt,2,SQL\_C\_CHAR,book\_category,10,&lent2);

SQLBindCol(hstmt,3,SQL\_C\_CHAR,book\_name,20,&lent3);

SQLBindCol(hstmt,4,SQL\_C\_CHAR,press,20,&lent4);

SQLBindCol(hstmt,5,SQL\_C\_SHORT,&year,0,&lent5);

SQLBindCol(hstmt,6,SQL\_C\_CHAR,writer,10,&lent6);

SQLBindCol(hstmt,7,SQL\_C\_FLOAT,&price,0,&lent7);

SQLBindCol(hstmt,8,SQL\_C\_SHORT,&total\_Amount,0,&lent8);

SQLBindCol(hstmt,9,SQL\_C\_SHORT,&now\_Amount,0,&lent9);

printf("查询结果如下\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("书号\t类别\t书名\t出版社\t年份\t作者\t价格\t总藏书量\t库存\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

while(SQLFetch(hstmt)!=SQL\_NO\_DATA\_FOUND)

{

printf("%s\t%s\t%s\t%s\t%d\t%s\t%.2f\t%d\t%d\n",book\_ID,book\_category,book\_name,press,year,writer,price,total\_Amount,now\_Amount);

}

}

//释放语句句柄

SQLFreeStmt(hstmt,SQL\_CLOSE);

break;

default:

printf("服务编号错误\n");

}

}

}

void borrow\_Book()

{

printf("请输入书名及借阅证编号(0退出系统)\n");

}

void return\_Book()

{

printf("请输入书名及借阅证编号(0退出系统)\n");

}

void add\_Book()

{

printf("请输入书名,作者,出版社,价格(0退出系统)\n");

}

void proof\_Manag()

{

printf("1.删除借阅证 2.增加借阅证 3.借阅证修改 0.退出系统\n");

}