#include "stdafx.h"

#include <windows.h>

#include <iostream>

#include <sql.h>

#include <sqlext.h>

#include <sqltypes.h>

#include <string.h>

using namespace std;

int main()

{

SQLHENV serverhenv; //环境句柄

SQLHDBC serverhdbc; //连接句柄

SQLHSTMT serverhstmt;//语句句柄

SQLRETURN ret; //结果句柄

SQLCHAR Sno[10] = { 0 }, Sname[10] = { 0 }, Grade[5] = {0};

SQLLEN length;

//分配环境句柄

ret = SQLAllocHandle(SQL\_HANDLE\_ENV, SQL\_NULL\_HANDLE, &serverhenv);

//设置环境句柄

ret = SQLSetEnvAttr(serverhenv,SQL\_ATTR\_ODBC\_VERSION,(void\*)SQL\_OV\_ODBC3,0);

if (!SQL\_SUCCEEDED(ret)) {

cout << "AllocEnvHandle error!" << endl;

system("pause");

}

//分配连接句柄

ret = SQLAllocHandle(SQL\_HANDLE\_DBC, serverhenv, &serverhdbc);

if (!SQL\_SUCCEEDED(ret)) {

cout << "AllocDbcHandle error!" << endl;

system("pause");

}

//连接数据库

ret = SQLConnect(serverhdbc, (SQLWCHAR \*)"ODBC\_TEST", SQL\_NTS, (SQLWCHAR\*)"root", SQL\_NTS, (SQLWCHAR\*)"root", SQL\_NTS);

if (!SQL\_SUCCEEDED(ret)) {

cout << "SQL\_Connect error!" << endl;

system("pause");

}

//分配执行语句句柄

ret = SQLAllocHandle(SQL\_HANDLE\_STMT, serverhdbc, &serverhstmt);

//执行SQL语句

ret = SQLExecDirect(serverhstmt,(SQLWCHAR\*)"INSERT INTO TEST VALUES(160004,'王兰花',120);", SQL\_NTS);

ret = SQLExecDirect(serverhstmt, (SQLWCHAR\*)"SELECT Sno,Sname,Grade FROM TEST;", SQL\_NTS); //查询结果也可能为空

if (ret == SQL\_SUCCESS || ret == SQL\_SUCCESS\_WITH\_INFO) {

//绑定数据（定义游标区，让游标知道怎么去移动取数据，比如移动Sno长度就是取Sno，移动了length就是取了一行数据）

SQLBindCol(serverhstmt, 1, SQL\_C\_CHAR, (void\*)Sno, sizeof(Sno), &length); //推测这里length是一行的长度，每次就要加上去

SQLBindCol(serverhstmt, 2, SQL\_C\_CHAR, (void\*)Sname, sizeof(Sname), &length);

SQLBindCol(serverhstmt, 3, SQL\_C\_CHAR, (void\*)Grade, sizeof(Grade), &length);

//将光标移动到下一行，获取下一行数据

while (SQL\_NO\_DATA != SQLFetch(serverhstmt)) {

//只要还有数据就向后推进

cout << "学生学号：" << Sno << " 学生姓名：" << Sname << " 学生成绩：" << Grade << endl;

}

}

//释放语句句柄

ret = SQLFreeHandle(SQL\_HANDLE\_STMT, serverhstmt);

if (SQL\_SUCCESS != ret && SQL\_SUCCESS\_WITH\_INFO != ret) //如果没有显示成功并且没有显示有数据就表示出错

cout << "free hstmt error!" << endl;

//断开数据库连接

ret = SQLDisconnect(serverhdbc);

if (ret != SQL\_SUCCESS && ret != SQL\_SUCCESS\_WITH\_INFO) //同上

cout << "disconnect error!" << endl;

//释放连接句柄

ret = SQLFreeHandle(SQL\_HANDLE\_DBC, serverhdbc);

if (ret != SQL\_SUCCESS && ret != SQL\_SUCCESS\_WITH\_INFO)

cout << "free hdbc error !" << endl;

//释放环境句柄

ret = SQLFreeHandle(SQL\_HANDLE\_ENV, serverhenv);

if (ret != SQL\_SUCCESS && ret != SQL\_SUCCESS\_WITH\_INFO)

cout << "free henv error !" << endl;

system("pause");

return 0;

}

private static Connection getConn() {

String driver = "com.mysql.jdbc.Driver";

String url = "jdbc:mysql://localhost:3306/samp\_db";

String username = "root";

String password = "";

Connection conn = null;

try {

Class.forName(driver); //classLoader,加载对应驱动

conn = (Connection) DriverManager.getConnection(url, username, password);

} catch (ClassNotFoundException e) {

e.printStackTrace();

} catch (SQLException e) {

e.printStackTrace();

}

return conn;

}

存储过程：

创建查看fruits表的存储过程

Delimiter // #结束符修改

create procedure proc()

begin

select \* from fruits;

end //

delimiter ;

Call proc();

创建一个名为 num\_sc 的存储过程，统计某位同学的考试门数，代码如下。

mysql>delimiter //

mysql>create procedure num\_sc ( in tmp\_sno char(10), out count\_num int )

->begin

-> select count(\*) into count\_num from sc

-> where sno=tmp\_sno;

->end //

mysql>delimiter ;

Call num\_sc(‘20201245152’,@count\_num);

Select @count\_num;

函数：

create function NameByZip()

returns char(50)

return (select s\_name from suppliers where s\_call ='48075');

创建一个名为funct\_name的存储函数返回某班级的辅导员姓名

mysql> delimiter &&

mysql> create function func\_name (class\_no varchar(8))

-> returns varchar(8)

-> begin

-> return ( select header from class

-> where classno=class\_no);

-> end &&

mysql> delimiter ;

Call func\_name(‘1001’);

mysql> delimiter &&

mysql> create procedure proc\_name (in class\_no varchar(8),out hname varchar(8))

-> begin

-> select header into hname from class

-> where classno=class\_no;

-> end &&

mysql> delimiter ;

