

数据库工程作业

要求:

- 1. 完成一个小型的数据库信息管理系统（或部分功能），并填写工程作业报告；程序和报告请在规定时间之内上传。
- 2. 开发模式（B/S 或 C/S）、开发高级语言任选，后台数据库使用大型数据库管理系统（SQL Server、Oracle、MySQL 等），不要使用桌面数据库。
- 3. 报告中所列举的四种操作，每种操作举一个例子即可。
- 4. 作业成绩按照报告中的标准评分，程序只实现报告中涉及的部分即可。
- 5. 作业完成后，请将工程作业报告和程序上传到学院网站，并联系助教老师进行系统说明和演示，回答相关问题。

工程作业报告

1. 项目信息（10 分）

学号	1711425	姓名	曹元议	专业	计算机科学与技术
项目名称	CD 出租销售店自助服务与管理系统				
必备环境	Visual Studio 2015、Microsoft SQL Server 2017、ADO 数据访问接口				
系统主要功能简介（4 分）	该系统可以实现对 CD 店顾客的自助服务和管理员对 CD 店信息的管理，顾客可以用这个系统购买（这里理解为在不注销帐户的前提下购买永久使用权）、租用（理解为购买暂时使用权）、归还、查询这个 CD 店的 CD，管理员也可以管理这个 CD 店的信息（进货，增加和更改供应商信息，查看库存，登记 CD 信息，查看店中的顾客等）。系统分为顾客和管理员两个身份，必须登录之后才能使用；新来的顾客必须先注册本店的会员以拿到帐户 ID 才能使用本系统。系统也允许顾客注销帐户。				
系统主要页面截图（6 分）	截 3-4 个页面即可 1. 登录界面：				

My CDStore - Login

×

欢迎进入我的CD出租销售店自助服务与管理系统，请先登录！
初次进入的顾客请先注册为会员！

登录区

帐户类型 ☒ 顾客 ☐ 管理员

帐户ID

密码

登录 会员注册

进入系统时间：2019-05-19 16:59:06

About

2. 顾客查看店内空闲 CD:

My CDStore - 查看空闲CD

×

输入搜索内容

CD名 作者名

类别 Search

格式（按入库时间先后排序）：CD号、CD名、作者名、类别、售价、租用价格、租用次数

3. 新顾客注册会员:

	<div data-bbox="395 197 1190 851"> <div>My CDStore - Register</div> <div>×</div> <p>欢迎注册为本店会员！首次注册必须充值不低于20元的余额！ 充值的余额仅供本店消费，不能提现。</p> <div> <div>密码</div> <input type="text"/> </div> <div> <div>确认密码</div> <input type="text"/> </div> <div> <div>充值余额</div> <input type="text"/> </div> <div> <div>确定</div> <div>取消</div> </div> </div> <p>4. 顾客购买、租用 CD:</p> <div data-bbox="395 940 1051 1509"> <div>My CDStore - 顾客购买、租用CD</div> <div>×</div> <p>如果购买自己正在租用的CD，则扣除相应的差价。</p> <p>你的余额: 11.00</p> <p>交易类型 <input checked="" type="radio"/> 购买 <input type="radio"/> 租用</p> <div> <div>CD号</div> <input type="text"/> </div> <div> <div>确定</div> <div>取消</div> </div> </div>
--	--

2. 系统配置 (10 分)

说明		(2 分) 请说明系统配置情况 (后台数据库, 高级语言); (8 分) 请使用连接串连接高级语言和数据库, 并分析字符串的各个部分。
配置 步骤 2 分	DBMS	1.Microsoft SQL Server 2017
		2.Microsoft SQL Server Management Studio 2017
		3.使用 SQL Server 身份验证和 Windows 身份验证的混合验证模式
		4.数据库默认访问端口: 1433
	高级语言	C++ (MFC 应用程序, 基于对话框)

	序号	名称	功能说明	取值
连接串分析 (6分)	1	Provider	数据源提供者(这里使用 OLE DB 数据源)	SQLOLEDB
	2	Server	要连接的数据库服务器名称(这里连接本地服务器)	(local)
	3	Database	要连接的数据库名称(即本程序要操作的数据库)	CDStore
	4	uid	数据库登录名	Bettygirl
	5	pwd	数据库登录名对应的登录密码	betty981022 cyy
连接串代码 (截图) (2分)	CCDStoreApp::InitInstance(): 数据库连接			
	<pre>54 AfxEnableControlContainer(); 55 56 //连接数据库 57 CoInitialize(NULL); //初始化OLE/COM库环境 58 HRESULT hr = NULL; 59 try 60 { 61 hr = m_pConnection.CreateInstance("ADODB.Connection"); 62 if (SUCCEEDED(hr)) 63 { 64 //初始化连接串 65 bstr_t strConnect = "Provider=SQLOLEDB;Server=(local);Database=CDStore;uid=Bettygirl;pwd=betty981022cyy"; 66 hr = m_pConnection->Open(strConnect, "", "", adModeUnknown); //打开数据库 67 if (FAILED(hr)) 68 { 69 AfxMessageBox(_T("Open Failed!")); 70 return FALSE; 71 } 72 } 73 else 74 { 75 AfxMessageBox(_T("Create instance of connection failed!")); 76 return FALSE; 77 } 78 } 79 catch (_com_error e) 80 { 81 CString temp1, temp2; 82 temp1.Format(_T("连接数据库失败! 错误信息: %s\r\n"), e.ErrorMessage()); 83 temp2.Format(e.Description()); 84 AfxMessageBox(temp1 + temp2); //显示错误信息 85 return FALSE; 86 } 87 }</pre>			
备注	以上是连接数据库的代码,连接串在上图的“strConnect”变量处。			

3. 数据库设计 (14分)

说明	(10分) 按照数据表的创建顺序,依次给出所涉及数据表的信息,其中参照字段以“(字段1, 字段2, ……, 字段n)”的形式给出,被参照字段以“表名(字段1, 字段2, ……, 字段n)”的形式给出; (4分) 一般 DBMS 都可以为数据库生成关系图,请将该图片截图并粘贴到表格中。				
数据表 (10)	创建顺序	数据表名称	主键	参照属性	被参照表及属性
	1	Supplier (CD 的供应商)	sup_id		
	2	Customer (顾客信息)	cus_id		
	3	Administrator (管理员信息)	admin_id		
	4	Genre(CD 的音乐流派,也就是音乐的分类,也相当于把 CD 放在不同的货架上)	genre_id		

	5	CDinfo (本店已登记的 CD 的信息)	cd_id	genre_id	Genre(genre_id)
	6	Stock (本店的库存)	stock_id	cd_id	CDinfo(cd_id)
				sup_id	Supplier(sup_id)
				admin_id	Administrator(admin_id)
	7	Trade (顾客的租用、购买 CD 的记录, 操作统称为交易)	trade_id	cus_id	Customer(cus_id)
				stock_id	Stock(stock_id)
关系图 (4)	<p>(截屏)</p> <pre>graph TD CDinfo[CDinfo] --- Genre[Genre] CDinfo --- Stock[Stock] Customer[Customer] --- Trade[Trade] Trade --- Stock Stock --- Supplier[Supplier] Stock --- Administrator[Administrator]</pre> <p>The diagram illustrates the database schema with the following tables and their attributes:</p> <ul style="list-style-type: none">CDinfo: cd_id (PK), cd_name, artist_name, genre_id (FK to Genre)Genre: genre_id (PK), genre_nameCustomer: cus_id (PK), cus_password, balance, register_dateTrade: trade_id (PK), cus_id (FK to Customer), stock_id (FK to Stock), trade_type, trade_dateStock: stock_id (PK), admin_id (FK to Administrator), sup_id (FK to Supplier), cd_id (FK to CDinfo), pur_price, lent_count, pur_dateSupplier: sup_id (PK), sup_name, addr, phoneAdministrator: admin_id (PK), admin_password, admin_name, register_date				
备注	<p>一些字段意义的解释: balance 是顾客帐户的余额; trade_type 表示交易操作的类型 (数据类型 bit, 0 为正在租用, 1 为已经购买); trade_date 是交易时间; stock_id 表示这单个商品 CD 在仓库中的 ID, 也就是说同一款 CD 在仓库中可以有很多个, 用 stock_id 作为这个 CD 在库存中的唯一性标识, 顾客只能购买库存中未被其他顾客购买或者正在租用的 CD 或者自己正在租用的 CD; register_date 是帐户注册的时间; lent_count 是这单个商品 CD 被租用的次数; pur_price 是单个商品 CD 的进价, 规定售价是进价的 1.2 倍, 租用价格是售价的 1/4; pur_date 是进货 (入库) 时间。</p> <p>表中任何唯一性约束都已通过 primary key, unique 或者触发器实现。</p> <p>Administrator 和 Supplier 表的应用没有展示, 但是在源程序中有 (没在这里展示的) 相关的功能设计, 具体情况可以查看源程序。</p>				

4. 含有事务应用的删除操作 (13 分)

说明	<p>(1 分) 简要说明该操作所要完成的功能;</p> <p>(2 分) 该操作会涉及的表 (必须含有两张或两张以上的关系表, 同时以 “表名” 的形式给出)</p>
----	--

	(1 分) 表连接涉及字段描述 (描述方式为“表 1. 属性=表 2. 属性”) (1 分) 删除条件涉及的字段描述 (以“表名. 属性=? ”形式给出) (4 分) 实现该操作的关键代码 (高级语言、SQL), 截图即可; (其中如果删除语句中不包含任何形式的事务应用将扣除 3 分) (4 分) 如何执行该操作, 按所述方法能够正常演示程序则给分。	
功能描述 (1 分)	<p>顾客注销帐户</p> <p>某位顾客以后再也不想去这家店了, 可以选择将自己的帐户永久注销 (以后要是想再使用这个系统必须再注册一个新帐户)。但是 CD 店规定顾客在注销帐户之前必须先归还所有正在租用的 CD 并且帐户余额低于 20 元; 如果满足了注销的条件, 则系统收回该顾客所有已购买的 CD 的使用权, 并且也不会退回 (那些低于 20 元的) 余额。</p> <p>意思是: 如果顾客曾经在这家店购买过 CD 的使用权 (Trade 表中存在与这位顾客对应的 trade_type=1 的记录), 就收回使用权, 相应的这个 CD 也成为空闲 CD。(删除 Trade 表中的这条记录后 Trade 表中没有与这个 CD 有关的记录)。如果这名顾客 (@cus_id) 的顾客没有正在租用 CD (Trade 表中不存在这位顾客的租用记录) 并且帐户余额小于 20 元 (Customer.balance<20), 就删除 Customer 表中这位顾客对应的记录。</p> <p>如果出现了任何不满足删除条件的情况, 则两处都不能删除, 也就是回滚到删除的状态之前。</p> <p>先从 Trade 表删再判断再从 Customer 表删是为了不破坏外键约束。</p>	
涉及的表 (2 分)	Customer, Trade	
表连接涉及字段 (1 分)	Customer.cus_id=Trade.cus_id	
删除条件 字段描述 (1 分)	字段	规则
	Trade.trade_type=1 Trade.cus_id=@cus_id	这个购买的交易记录是对应这位顾客的 (如果存在), 也要从 Trade 表中删除这条记录。
	Customer.balance<20 Customer.cus_id=@cus_id	这位顾客的帐户所剩余额必须小于 20 元, 且 Trade 表中不存在对应这位顾客的租用 CD 的记录, 就删除 Customer 表中这位顾客的帐户记录。
代码 (4 分)	(截屏) 1. SQL: 存储过程 sp_deletecus, 涉及到删除的事务在此定义	

```

1 create procedure sp_deletecus
2 @cus_id varchar(30),
3 @tips varchar(100) output
4 as
5 begin
6     declare @balance decimal(20,2)
7     set @balance=(select balance from Customer where cus_id=@cus_id)
8     begin transaction T1 -- 开启一个事务
9     delete from Trade where cus_id=@cus_id and trade_type=1
10    if(exists(select * from Trade where cus_id=@cus_id
11    and trade_type=0))
12    begin
13        select @tips='你还有正在租用的cd! 请先归还!'
14        rollback transaction T1 -- 不满足删除条件, 回滚事务
15        return 0
16    end
17    delete from Customer where cus_id=@cus_id
18    if(@balance>=20)
19    begin
20        select @tips='帐户余额不低于20元! 不能注销!'
21        rollback transaction T1 -- 不满足删除条件, 回滚事务
22        return 0
23    end
24    select @tips='注销成功!'
25    commit transaction T1 -- 满足删除条件, 提交事务
26    return 1
27 end

```

2. 高级语言: 执行存储过程 sp_deletecus

CDeleteCus::OnBnClickedOk()

```

41 void CDeleteCus::OnBnClickedOk()
42 {
43     // TODO: 在此添加控件通知处理程序代码
44     //CDialogEx::OnOK();
45     CString pwd;
46     m_calpwd.GetWindowTextW(pwd);
47     if (pwd.IsEmpty())
48     {
49         MessageBox(L"密码不能为空!", L"顾客注销帐户", MB_OK | MB_ICONWARNING);
50         return;
51     }
52     else
53     {
54         m_pRecordset.CreateInstance(__uuidof(Recordset)); //创建实例
55         CString sqlHasRecord;
56         sqlHasRecord.Format(L"select * from Customer where cus_id='%s' and cus_password='%s'", id, pwd);
57         try
58         {
59             HRESULT hr = m_pRecordset->Open((_variant_t)sqlHasRecord, _variant_t((IDispatch*)m_pConnection, true),
60             adOpenDynamic, adLockPessimistic, adCmdText);
61             if (SUCCEEDED(hr))
62             {
63                 if (m_pRecordset->rsEOF %& m_pRecordset->BOF) //记录集为空
64                 {
65                     MessageBox(L"密码错误!", L"顾客注销帐户", MB_OK | MB_ICONWARNING);
66                     m_pRecordset->Close();
67                     return;
68                 }
69             }
70             else
71             {
72                 AfxMessageBox(L"Create instance of recordset failed!");
73                 return;
74             }
75             m_pRecordset->Close();

```

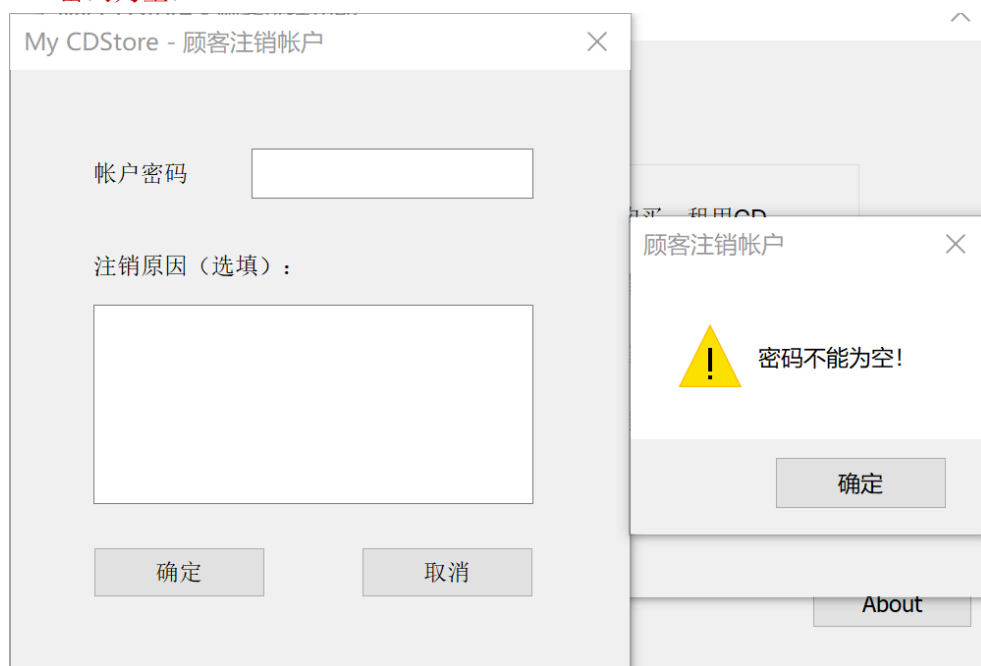
```

76     }
77     catch (_com_error e)
78     {
79         CString temp1 = e.Description(), temp2 = e.ErrorMessage();
80         AfxMessageBox(temp1 + L"\r\n" + temp2); //显示错误信息
81         return;
82     }
83 }
84 try
85 {
86     HRESULT hr = m_pCommand.CreateInstance(__uuidof(Command));
87     if (FAILED(hr))
88     {
89         AfxMessageBox(L"Create instance of command failed!!");
90         return;
91     }
92     m_pCommand->Parameters->Refresh();
93     m_pCommand->Parameters->Append(m_pCommand->CreateParameter(_bstr_t("Return"),
94         adInteger, adParamReturnValue, sizeof(int));
95     m_pCommand->Parameters->Append(m_pCommand->CreateParameter(_bstr_t("cus_id"),
96         adVarChar, adParamInput, 30, _variant_t(id));
97     m_pCommand->Parameters->Append(m_pCommand->CreateParameter(_bstr_t("tips"),
98         adVarChar, adParamOutput, 100));
99     m_pCommand->CommandText = _bstr_t("sp deletecus");
100    m_pCommand->ActiveConnection = m_pConnection;
101    m_pCommand->CommandType = adCmdStoredProc;
102    m_pConnection->BeginTrans();
103    m_pCommand->Execute(NULL, NULL, adCmdStoredProc); //执行存储过程
104    int nRet = m_pCommand->Parameters->GetItem("Return")->GetValue();
105    CString tips = m_pCommand->Parameters->GetItem("tips")->GetValue();
106    if (nRet == 0) //不满足删除条件
107    {
108        m_pConnection->RollbackTrans();
109        MessageBox(tips, L"顾客注销帐户", MB_OK | MB_ICONWARNING);
110        return;
111    }
112    else if (nRet == 1) //满足删除条件
113    {
114        m_pConnection->CommitTrans();
115        MessageBox(tips, L"顾客注销帐户", MB_OK | MB_ICONINFORMATION);
116        CDialogEx::OnOK();
117    }
118 }
119 catch (_com_error e)
120 {
121     m_pConnection->RollbackTrans();
122     CString temp1 = e.Description(), temp2 = e.ErrorMessage();
123     AfxMessageBox(temp1 + L"\r\n" + temp2); //显示错误信息
124     return;
125 }
126 }
127 }

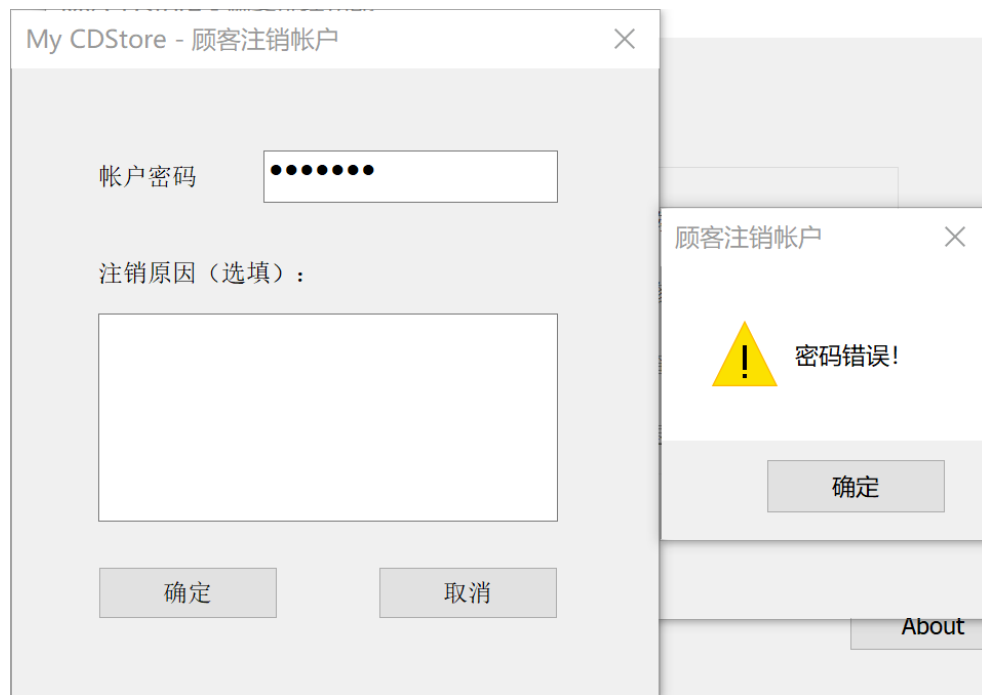
```

1. 密码为空:

程序演示
(4分)



2. 密码错误

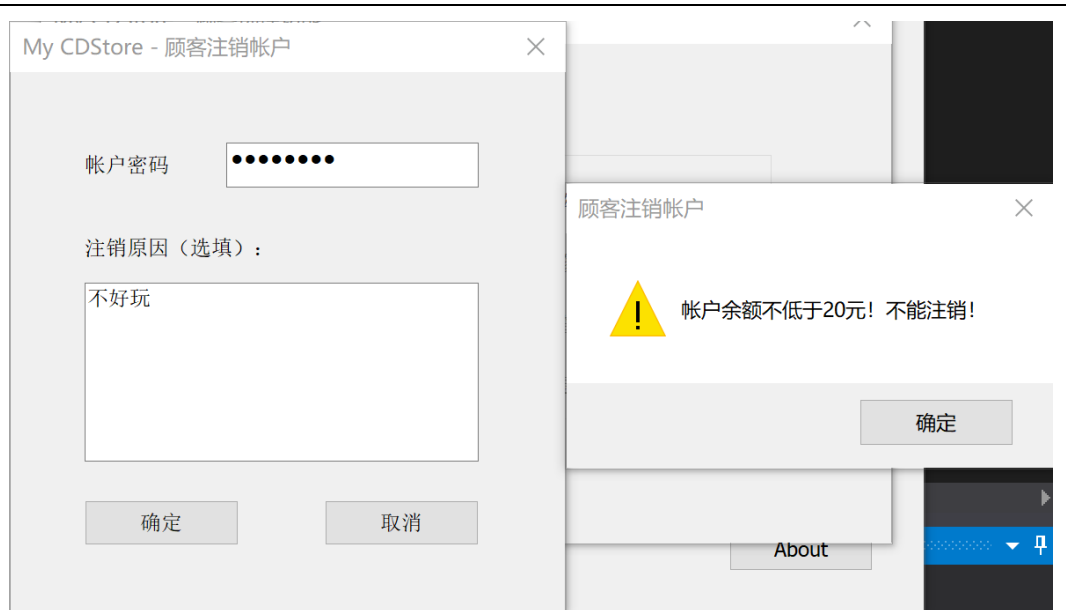


3. 密码正确，但不满足注销（删除）条件

要注销的帐户：（操作前后无变化）



注销失败:



要注销的帐户：（操作前后无变化）



注销失败：



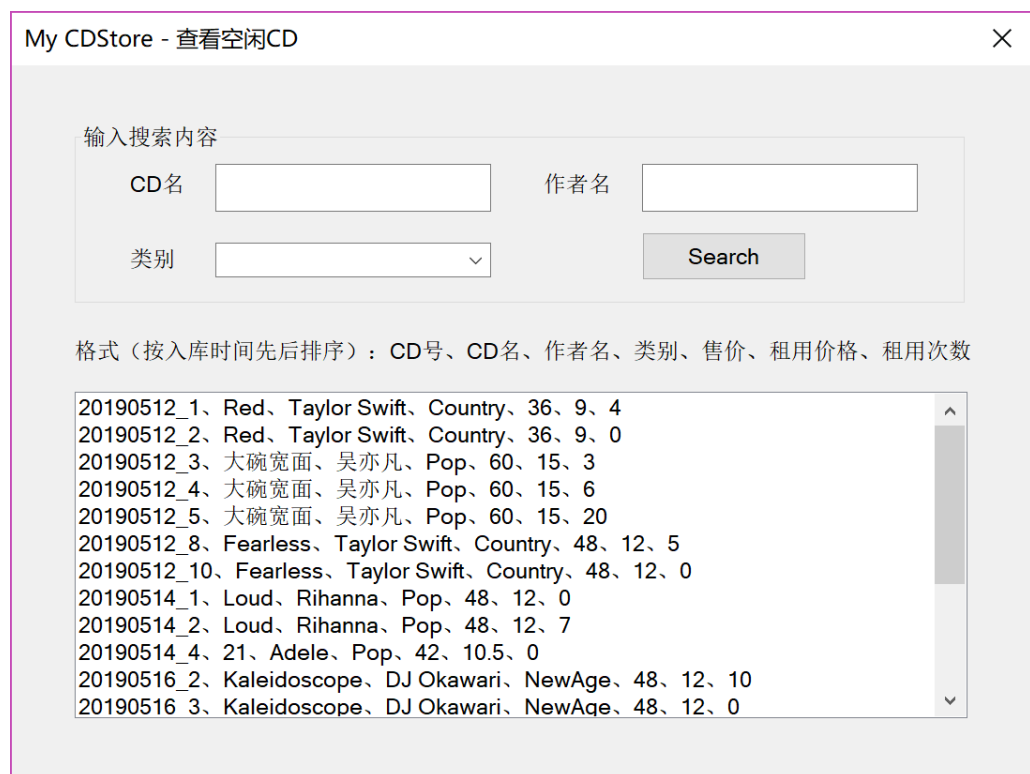
要注销的帐户：（操作前后无变化）



注销失败:



4. 密码正确，且符合注销（删除）条件
注销前的部分空闲 CD 列表：



要注销的帐户和其购买使用权的 CD：

My CDStore - 顾客查看个人帐户

帐户ID: 20190522_2

余额: 2.00

注册时间: 2019/5/22 16:08:59

我正在租用的CD:

我已经购买的CD:

20190516_1

确定

注销成功，并退出登录界面：

My CDStore - 顾客注销帐户

帐户密码:

注销原因（选填）：

没钱了

确定 取消

顾客注销帐户

注销成功!

确定

About

再次尝试登录，已经登录不上了：

5. 触发器控制下的添加操作（20 分）

说明	（1 分）简要说明该操作所要完成的功能； （2 分）简要说明该触发器所要完成的功能 （1 分）该操作会涉及的表（以“表名”的形式给出）。 （2 分）该操作输入数据以及输入数据应该满足的条件，如：数值范围、是否为空； （6 分）实现该操作的关键代码（高级语言、SQL），截图即可； （8 分）如何执行该操作，按所述方法能够正常演示程序则给分。	
功能描述 （1 分）	新顾客注册会员 通过输入密码（对应字段 <code>cus_password</code> ）和帐户余额注册新的会员帐户。CD 店规定在注册的时候必须先充值不少于 20 元的余额（对应字段 <code>balance</code> ），才能注册成功；否则注册失败。	
触发器描述 （2 分）	往表 <code>Customer</code> 中插入新元组（试图注册新会员帐户）后，触发器 <code>Register</code> 被触发，自动执行里面的语句。触发器中声明了一个 <code>decimal(20,2)</code> 类型（包括小数点后的 2 位一共 20 位数字）的局部变量 <code>@balance</code> ，并给 <code>@balance</code> 赋值为刚插入的元组中的 <code>balance</code> （余额）值。之后判断局部变量 <code>@balance</code> 值是否小于 20，如果小于 20 则违背触发器要求，回滚到插入之前的状态（相当于未插入，即注册失败）并打印一条错误信息，终止执行；否则插入元组成功（即注册成功），并打印一条注册成功的信息（源程序并未用到），结束执行。	
涉及的表 （1 分）	<code>Customer</code>	
输入数据 （2 分）	字段	规则
	<code>cus_id</code>	新注册的顾客会员的帐户 ID，类型 <code>varchar(30)</code> ，已在源程序中自动生成。
	<code>cus_password</code>	新注册的顾客会员的密码，类型 <code>varchar(30)</code> ，程序中必须输入长度为 6 到 30 的字符串，不能为空。（格式已在源程序中进行了预处理）
	<code>balance</code>	新注册的顾客会员的充值的余额，为 <code>decimal(20,2)</code> 类型（包括小数点后的 2 位一共 20 位数字），必须大于等于 0，不能为空。（格式已在源程序中进行了预处理）
	<code>register_date</code>	注册时间，类型 <code>datetime</code> 。取发生插入操作时的系统时间（在 <code>insert</code> 语句中给出）。
插入操作源码 （3 分）	（截屏） <code>CRegisterDlg::OnBnClickedOk()</code> ：	


```

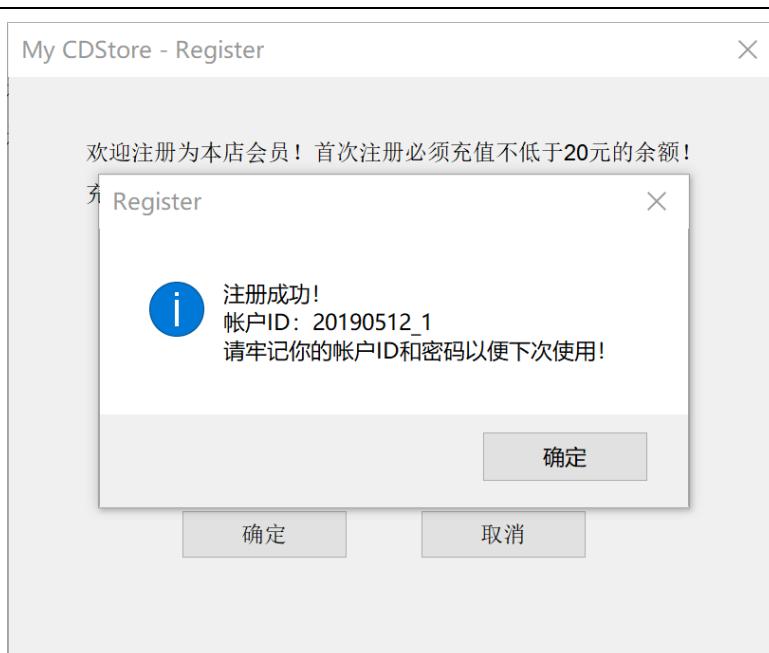
46 void CRegisterDlg::OnBnClickedOk()
47 {
48     // TODO: 在此添加控件通知处理程序代码
49     CString pwd1, pwd2, bal;
50     m_password1.GetWindowTextW(pwd1);
51     m_password2.GetWindowTextW(pwd2);
52     m_balance.GetWindowTextW(bal);
53     if (pwd1.IsEmpty())
54     {
55         MessageBox(L"密码不能为空!", L"Register", MB_OK | MB_ICONWARNING);
56     }
57     else if (pwd2.IsEmpty())
58     {
59         MessageBox(L"确认密码不能为空!", L"Register", MB_OK | MB_ICONWARNING);
60     }
61     else if (bal.IsEmpty())
62     {
63         MessageBox(L"充值余额不能为空!", L"Register", MB_OK | MB_ICONWARNING);
64     }
65     else
66     {
67         if (pwd1 != pwd2)
68         {
69             MessageBox(L"两次密码输入不一致! 请重新输入!", L"Register", MB_OK | MB_ICONWARNING);
70         }
71         else if (pwd1.GetLength() < 6 || pwd1.GetLength() > 30)
72         {
73             MessageBox(L"密码必须不少于6个字符且不多于30个字符!", L"Register", MB_OK | MB_ICONWARNING);
74         }
75     }
76     {
77         UpdateData(true);
78         CString strTime, cusid, strSql, sqlHasRecord, templ;
79         CTime tm;
80         tm = CTime::GetCurrentTime();
81         strTime = tm.Format("%Y%m%d");
82         m_pRecordset.CreateInstance(__uuidof(Recordset)); //创建实例
83         sqlHasRecord.Format(L"select * from Customer where cus_id like '%s', strTime);
84         templ = L"% ' order by register_date desc";
85         sqlHasRecord += templ;
86         HRESULT hr = m_pRecordset->Open((_variant_t)sqlHasRecord,
87             _variant_t((IDispatch*)m_pConnection, true), adOpenDynamic, adLockPessimistic, adCmdText);
88         if (SUCCEEDED(hr))
89         {
90             //生成新顾客会员ID
91             if (m_pRecordset->rsEOF && m_pRecordset->BOF) //记录集为空
92             {
93                 cusid.Format(L"%s_%d", strTime, 1);
94             }
95             else
96             {
97                 CString temp2;
98                 temp2 = m_pRecordset->GetCollect(L"cus_id");
99                 int down = temp2.Find('_');
100                 temp2 = temp2.Mid(down + 1);
101                 cusid.Format(L"%s_%d", strTime, _ttoi(temp2) + 1);
102             }
103         }
104     }
105     {
106         AfxMessageBox(L"Create instance of recordset failed!");
107         return;
108     }
109     m_pRecordset->Close();
110
111     //执行插入语句
112     strSql.Format(L"insert into Customer values('%s', '%s', %lf, getdate())",
113         cusid, m_passwordstr, m_balancedata);
114     try
115     {
116         m_pConnection->Execute(_bstr_t(strSql), 0, adCmdText);
117     }
118     catch (_com_error e)
119     {
120         //违背触发器的插入操作的异常在此处捕获
121         MessageBox(e.Description(), L"Register", MB_OK | MB_ICONWARNING);
122         return;
123     }
124     CString success;
125     success.Format(L"注册成功! \r\n帐户ID: %s\r\n请牢记你的帐户ID和密码以便下次使用!", cusid);
126     MessageBox(success, L"Register", MB_OK | MB_ICONINFORMATION);
127     CDialogEx::OnOK();
128 }
129 //CDialogEx::OnOK();
130
131

```

触发器源
码
(3分)

(截屏)

	<pre>1 create trigger Register on Customer 2 after insert 3 as 4 begin 5 declare @balance decimal(20,2) 6 set @balance=(select balance from inserted) 7 if(@balance<20) 8 begin 9 rollback transaction 10 print '注册失败! 首次充值余额不得少于20元! ' 11 end 12 else 13 begin 14 print '注册成功! ' 15 end 16 end</pre>																									
程序演示 (4分)	<p>说明：不违背触发器能够执行插入操作。</p> <p>1. 执行 insert 操作前的 Customer 表。</p> <table><tr><th></th><th>cus_id</th><th>cus_password</th><th>balance</th><th>register_date</th></tr><tr><td>1</td><td>20140209_1</td><td>123456qwerty</td><td>30.00</td><td>2014-02-09 00:00:00.000</td></tr><tr><td>2</td><td>20150320_1</td><td>zxcvbnmasf</td><td>25.00</td><td>2015-03-20 00:00:00.000</td></tr><tr><td>3</td><td>20160715_1</td><td>rtyuiop67890</td><td>40.00</td><td>2016-07-15 00:00:00.000</td></tr><tr><td>4</td><td>20170307_1</td><td>1234567</td><td>20.00</td><td>2017-03-07 00:00:00.000</td></tr></table> <p>2. 点击“确定”之前，执行 insert 操作前，下图输入的密码保证了输入密码在 6 位到 30 位之间，且两次输入密码一致。</p> <div><div>My CDStore - Register</div><div>×</div><div>欢迎注册为本店会员！首次注册必须充值不低于20元的余额！ 充值的余额仅供本店消费，不能提现。</div><div>密码 <input type="password" value="••••••••"/></div><div>确认密码 <input type="password" value="••••••~"/></div><div>充值余额 <input type="text" value="50"/></div><div>确定 取消</div></div> <p>3. 点击“确定”之后，执行 insert 操作，执行了源程序中异常处理之后的语句，表明插入成功。</p>		cus_id	cus_password	balance	register_date	1	20140209_1	123456qwerty	30.00	2014-02-09 00:00:00.000	2	20150320_1	zxcvbnmasf	25.00	2015-03-20 00:00:00.000	3	20160715_1	rtyuiop67890	40.00	2016-07-15 00:00:00.000	4	20170307_1	1234567	20.00	2017-03-07 00:00:00.000
	cus_id	cus_password	balance	register_date																						
1	20140209_1	123456qwerty	30.00	2014-02-09 00:00:00.000																						
2	20150320_1	zxcvbnmasf	25.00	2015-03-20 00:00:00.000																						
3	20160715_1	rtyuiop67890	40.00	2016-07-15 00:00:00.000																						
4	20170307_1	1234567	20.00	2017-03-07 00:00:00.000																						



4. 插入之后 Customer 表的结果。

	cus_id	cus_password	balance	register_date
1	20140209_1	123456qwerty	30.00	2014-02-09 00:00:00.000
2	20150320_1	zxcvbnmasf	25.00	2015-03-20 00:00:00.000
3	20160715_1	rtyuiop67890	40.00	2016-07-15 00:00:00.000
4	20170307_1	1234567	20.00	2017-03-07 00:00:00.000
5	20190512_1	1234567890	50.00	2019-05-12 09:22:52.300

程序演示
(4 分)

说明：违背触发器要求，不能够执行插入操作，系统报错。

1. 点击“确定”之前，执行 insert 操作前，下图输入的密码保证了输入密码在 6 位到 30 位之间，且两次输入密码一致，且输入的余额违背触发器要求。

2. 点击“确定”之后，执行 insert 操作，由于输入的余额值小于 20，违背触发器操作，触发器停止执行并输出错误信息，在源程序中作为异常被捕获。

	<div><div>My CDStore - Register</div><div>欢迎注册为本店会员！首次注册必须充值不低于20元的余额！</div><div>充值金额不得低于本店消费金额提醒</div><div><div>Register</div><div><div>!</div><div>注册失败！首次充值余额不得少于20元！</div></div><div>确定</div></div><div><div>确定</div><div>取消</div></div></div>
备注	3. insert 操作前后 Customer 表无变化。 “输入数据”理解为 insert 语句中输入的数据。

6. 存储过程控制下的更新操作（18 分）

说明	<p>（1 分）简要说明该操作所要完成的功能；</p> <p>（1 分）简要说明该存储过程所要完成的功能；</p> <p>（2 分）说明该操作涉及操作的表（必须包含两张或两张以上的关系表，以“表名形式”描述）</p> <p>（1 分）表连接涉及字段描述（描述方式为“表 1. 属性=表 2. 属性”）</p> <p>（2 分）该操作会修改字段（以“表名. 字段名”的形式给出），以及修改规则，如新数值的计算方法、在何种条件下予以修改等；</p> <p>（6 分）实现该操作的关键代码（高级语言、SQL），截图即可；</p> <p>（5 分）如何执行该操作，按所述方法能够正常演示程序则给分。</p>
功能描述 (1 分)	<p>顾客购买或租用 CD</p> <p>顾客选择交易类型 (@trade_type) (购买、租用) 并输入 CD 号 (@stock_id) 进行交易操作。顾客输入的 CD 号所对应的 CD 没有被其他顾客购买或者正在租用，或者自己没有购买此 CD，并且余额足够，则交易成功。</p> <p>允许顾客购买自己正在租用的 CD，但是要补齐差价（根据 CD 店的规定，租用的价格是购买价格的 1/4，因此补齐的差价是租用价格的 3 倍）。但是顾客不能对自己正在租用的 CD 重复租用，也不能对自己已经购买的 CD 重复做其他操作。</p>
存储过程 功能描述 (1 分)	<p>存储过程 sp_trade 有返回值 (integer 类型, 用于判断是否更新成功或者违背存储过程, 取值从 -1 到 8 的整数)、三个输入参数 (@cus_id 是当前顾客的 ID, 类型 varchar(30); @stock_id 是顾客输入的要操作的 CD 的 ID, 类型 varchar(30); @trade_type 是顾客选择的交易类型, 为 bit 类型, 0 为租用, 1 为购买)、两个输出参数 (@trade_price 是顾客要支付的余额; @trade_balance 是顾客交易后的余额; 都是为输出消息框的内容而服务的, 都是 float 类型)。</p> <p>顾客做购买或者租用的操作就是调用这个存储过程。首先 CD 号不能为空, 否则不调用存储过程。根据存储过程的返回值来判断是否违背存储过程。输入的 CD 号必须是店内库存中已有的 CD, 否则违背存储过程 (返回值为 -1)。如果输入的 CD 是库存中有的, 但是被其他顾客正在租用或已经购买, 则违背存储过程 (返回值为 0)。如果输入的 CD 号是自己已经购买的 CD, 则违背存储过程 (返回值为 1)。如果顾客选择租用自己正在租用的 CD, 则违背存储过程 (返回值为 4)。此时两个输出参数都为 0。</p> <p>其他判断都是余额足够则进行更新操作, 余额不足则违背存储过程 (输出参数 @trade_price 为交易时顾客要支付的余额, @trade_balance 为顾客的余额: 如果更新成功则是交易之后的余额): 如果顾客选择购买自己正在租用的 CD, 则补齐差价, 余额足够则 同时 更新顾客的余额 (Customer 表中对应 @cus_id 的 balance 属性) 和 Trade 表中对应的交易类型 (trade_type 改为 1) (返回值为 2), 余额不足则违背存储过程 (返回值为 3)。如果顾客购买店内空闲的 CD, 余额足够则更新顾客的余额 (返回值为 6), 余额不足则违背存储过程 (返回值为 8)。如果顾客租用店内空闲的 CD, 余额足够则 同时 更新顾客的余额和库存中此 CD 的租用次数 (Stock 表中 @stock_id 对应的 lent_count 值+1) (返回值为 5), 余额不足则违背存储过程 (返回值为 7)。</p> <p>综上, 返回值为 2、5、6 时更新成功, 不违背存储过程; 其余返回值都是违背存储过程。</p>

	Trade 表中的记录来源：如果返回值是 5 或者 6 则更新成功（不违背存储过程），后续在 Trade 表中插入一条记录。（返回值是 2 的时候不用插入新的记录，直接更新相应记录的 trade_type）	
涉及的关系表（2 分）	Customer, Stock, Trade	
表连接涉及字段（1）	Customer.cus_id=Trade.cus_id Stock.stock_id=Trade.stock_id	
更改字段（2 分）	字段	规则
	Customer.balance	<p>前提：如果顾客输入的 CD 号不为空，且是库存已有的 CD，且不是其他顾客正在租用或者已经购买的 CD。</p> <p>1. 如果顾客购买店内空闲的 CD，且余额充足（@balance>=@sell_price，@balance 是这个顾客对应的余额，@sell_price 是 Stock 表中这个 CD 对应的售价（进价 pur_price 的 1.2 倍）），就更新 Customer 表中这个顾客的 balance 属性：balance=balance-@sell_price</p> <p>2. 如果顾客租用店内空闲的 CD，且余额充足（@balance>=@lent_price，@balance 是这个顾客对应的余额，@lent_price 是 Stock 表中这个 CD 对应的租用价格（进价 pur_price 的 1.2 倍的 1/4）），就更新 Customer 表中这个顾客的 balance 属性：balance=balance-@lent_price</p> <p>3. 如果顾客购买自己正在租用的 CD，且余额充足（@balance>=3*@lent_price，@balance 是这个顾客对应的余额，@lent_price 同(2)），就更新 Customer 表中这个顾客的 balance 属性：balance=balance-3*@lent_price</p>
	Stock.lent_count	<p>前提条件同上。</p> <p>如果顾客租用店内空闲的 CD，且余额充足（条件同上面的 2），就更新 Stock 表中这个 CD 的 lent_count 属性（租用次数+1）：lent_count=lent_count+1</p>
	Trade.trade_type	<p>前提条件同上。</p> <p>如果顾客购买自己正在租用的 CD，且余额充足（条件同上面的 3），就更新 Trade 表中这个交易的 trade_type 属性（租用改为购买）：trade_type=1</p>
更新代码（3 分）	<p>（截屏）</p> <p>1. 如果顾客租用店内空闲的 CD，且余额充足：</p> <pre>update Customer set balance=balance-@lent_price where cus_id=@cus_id update Stock set lent_count=lent_count+1 where stock_id=@stock_id</pre> <p>2. 如果顾客购买店内空闲的 CD，且余额充足：</p> <pre>update Customer set balance=balance-@sell_price where cus_id=@cus_id</pre> <p>3. 如果顾客购买自己租用的 CD，且余额充足：</p>	

	<pre> update Customer set balance=balance-3*@lent_price where cus_id=@cus_id update Trade set trade_type=1 where stock_id=@stock_id and cus_id=@cus_id </pre>
<p>创建存储 过程源码 (3分)</p>	<p>(截屏)</p> <pre> 1 create procedure sp_trade 2 @cus_id varchar(30), 3 @stock_id varchar(30), 4 @trade_type bit, 5 @trade_price float output, 6 @trade_balance float output 7 as 8 begin 9 if(not exists(select * from Stock where stock_id=@stock_id)) 10 begin --店里没有相应的CD 11 select @trade_price=0 12 select @trade_balance=0 13 return -1 14 end 15 else if(exists(select * from Trade where stock_id=@stock_id and cus_id<>@cus_id)) 16 begin --其他顾客已购买或正在租用CD 17 select @trade_price=0 18 select @trade_balance=0 19 return 0 20 end 21 else if(exists(select * from Trade where cus_id=@cus_id and stock_id=@stock_id and trade_type=1)) 22 begin --自己已经购买CD 23 select @trade_price=0 24 select @trade_balance=0 25 return 1 26 end 27 else 28 begin 29 declare @balance decimal(20,2) 30 declare @sell_price decimal(20,2) 31 declare @lent_price decimal(20,2) 32 33 set @balance=(select balance from Customer where cus_id=@cus_id) 34 set @sell_price=(select pur_price from Stock where stock_id=@stock_id)*1.2 35 set @lent_price=@sell_price*0.25 36 if(exists(select * from Trade where cus_id=@cus_id and stock_id=@stock_id and trade_type=0)) 37 begin 38 if(@trade_type=1)--购买自己正在租用的CD 39 begin 40 if(@balance>=3*@lent_price) 41 begin 42 update Customer set balance=balance-3*@lent_price where cus_id=@cus_id 43 update Trade set trade_type=1 where stock_id=@stock_id and cus_id=@cus_id 44 set @balance=(select balance from Customer where cus_id=@cus_id) 45 select @trade_price=cast(3*@lent_price as float) 46 select @trade_balance=cast(@balance as float) 47 return 2 48 end 49 else 50 begin 51 select @trade_price=cast(3*@lent_price as float) 52 select @trade_balance=cast(@balance as float) 53 return 3 54 end 55 end 56 else --自己已经正在租用此CD, 不能重复租用 57 begin 58 select @trade_price=0 59 select @trade_balance=0 60 return 4 61 end 62 else if(@trade_type=0 and @balance>=@lent_price) </pre>


```

63 begin
64     update Customer set balance=balance-@lent_price where cus_id=@cus_id
65     update Stock set lent count=lent count+1 where stock_id=@stock_id
66     set @balance=(select balance from Customer where cus_id=@cus_id)
67     select @trade_price=cast(@lent_price as float)
68     select @trade_balance=cast(@balance as float)
69     return 5
70 end
71 else if (@trade_type=1 and @balance>=@sell_price)
72 begin
73     update Customer set balance=balance-@sell_price where cus_id=@cus_id
74     set @balance=(select balance from Customer where cus_id=@cus_id)
75     select @trade_price=cast(@sell_price as float)
76     select @trade_balance=cast(@balance as float)
77     return 6
78 end
79 else if (@trade_type=0 and @balance<@lent_price)
80 begin
81     select @trade_price=cast(@lent_price as float)
82     select @trade_balance=cast(@balance as float)
83     return 7
84 end
85 else if (@trade_type=1 and @balance<@sell_price)
86 begin
87     select @trade_price=cast(@sell_price as float)
88     select @trade_balance=cast(@balance as float)
89     return 8
90 end
91 end
92 end

```

(截屏)
CTrade::OnBnClickedOk()

存储过程
执行源码
(1分)

```

82 void CTrade::OnBnClickedOk()
83 {
84     // TODO: 在此添加控件通知处理程序代码
85     CString cdid;
86     m_cdid.GetWindowTextW(ccid);
87     if (ccid.IsEmpty())
88     {
89         MessageBox(L"CD号不能为空!", L"购买、租用CD", MB_OK | MB_ICONWARNING);
90         return;
91     }
92     else
93     {
94         UpdateData(true);
95         if (((CButton*)GetDlgItem(IDC_BUY))->GetCheck())
96         {
97             tradetype = 1;
98         }
99         else
100         {
101             tradetype = 0;
102         }
103         CString type;
104         type.Format(L"%d", tradetype);
105         try
106         {
107             HRESULT hr = m_pCommand.CreateInstance(__uuidof(Command));
108             if (FAILED(hr))
109             {
110                 AfxMessageBox(L"Create instance of command failed!!");
111                 return;
112             }
113             m_pCommand->Parameters->Refresh();
114             m_pCommand->Parameters->Append(m_pCommand->CreateParameter(_bstr_t("Return"),
115                 adInteger, adParamReturnValue, sizeof(int)));
116             m_pCommand->Parameters->Append(m_pCommand->CreateParameter(_bstr_t("cus_id"),
117                 adVarChar, adParamInput, 30, variant t(id)));

```



```

117         adVarChar, adParamInput, 30, _variant_t(id));
118     m_pCommand->Parameters->Append(m_pCommand->CreateParameter(_bstr_t("stock_id"),
119         adVarChar, adParamInput, 30, _variant_t(m_cdidstr)));
120     m_pCommand->Parameters->Append(m_pCommand->CreateParameter(_bstr_t("trade_type"),
121         adBoolean, adParamInput, sizeof(BOOL), _variant_t(type)));
122     m_pCommand->Parameters->Append(m_pCommand->CreateParameter(_bstr_t("trade_price"),
123         adDouble, adParamOutput, sizeof(double)));
124     m_pCommand->Parameters->Append(m_pCommand->CreateParameter(_bstr_t("trade_balance"),
125         adDouble, adParamOutput, sizeof(double)));
126     m_pCommand->CommandText = _bstr_t("sp_trade");
127     m_pCommand->ActiveConnection = m_pConnection;
128     m_pCommand->CommandType = adCmdStoredProc;
129     m_pConnection->BeginTrans();
130     m_pCommand->Execute(NULL, NULL, adCmdStoredProc); //执行存储过程
131     int nRet = m_pCommand->Parameters->GetItem("Return")->GetValue();
132     double price = m_pCommand->Parameters->GetItem("trade_price")->GetValue();
133     double bal = m_pCommand->Parameters->GetItem("trade_balance")->GetValue();
134     m_pConnection->CommitTrans();
135     if (nRet == -1) //输入库存中不存在的CD, 违背存储过程
136     {
137         MessageBox(L"库存没有此CD! ", L"购买、租用CD", MB_OK | MB_ICONWARNING);
138         return;
139     }
140     else if (nRet == 0) //输入别人租用或购买的CD, 违背存储过程
141     {
142         MessageBox(L"此CD已被其他顾客购买或者正在被租用! ", L"购买、租用CD", MB_OK | MB_ICONWARNING);
143         return;
144     }
145     else if (nRet == 1) //输入自己已购买的CD, 违背存储过程
146     {
147         MessageBox(L"你已购买此CD! ", L"购买、租用CD", MB_OK | MB_ICONWARNING);
148         return;
149     }
150     else if (nRet == 2) //购买已租用的CD成功
151     {
152         CString str;

```

根据返回值作相应的处理

```

153         str.Format(L"购买成功! \r\n你的余额: %lf", bal);
154         str = str.Left(str.Find('.') + 3);
155         MessageBox(str, L"购买、租用CD", MB_OK | MB_ICONINFORMATION);
156         return;
157     }
158     else if (nRet == 3) //购买已租用的CD, 但是余额不足, 违背存储过程
159     {
160         CString str1, str2;
161         str1.Format(L"余额不足, 请先充值! \r\n购买本CD金额: %lf", price);
162         str1 = str1.Left(str1.Find('.') + 3);
163         str2.Format(L"\r\n你的余额: %lf", bal);
164         str2 = str2.Left(str2.Find('.') + 3);
165         str1 += str2;
166         MessageBox(str1, L"购买、租用CD", MB_OK | MB_ICONWARNING);
167         return;
168     }
169     else if (nRet == 4) //重复租用自己正在租用的CD, 违背存储过程
170     {
171         MessageBox(L"你已正在租用此CD! 不能重复租用! ", L"购买、租用CD", MB_OK | MB_ICONWARNING);
172         return;
173     }
174     else if (nRet == 5) //租用空闲CD, 余额充足, 更新成功
175     {
176         CString str;
177         str.Format(L"租用成功! \r\n你的余额: %lf", bal);
178         str = str.Left(str.Find('.') + 3);
179         MessageBox(str, L"购买、租用CD", MB_OK | MB_ICONINFORMATION);
180     }
181     else if (nRet == 6) //购买空闲CD, 余额充足, 更新成功
182     {
183         CString str;
184         str.Format(L"购买成功! \r\n你的余额: %lf", bal);
185         str = str.Left(str.Find('.') + 3);
186         MessageBox(str, L"购买、租用CD", MB_OK | MB_ICONINFORMATION);
187     }

```

根据返回值作相应的处理

	<pre> 188 else if (nRet == 7) //租用空闲CD, 余额不足, 违背存储过程 189 { 190 CString str1, str2; 191 str1.Format(L"余额不足, 请先充值! \r\n租用本CD金额: %lf", price); 192 str1 = str1.Left(str1.Find('.') + 3); 193 str2.Format(L"\r\n你的余额: %lf", bal); 194 str2 = str2.Left(str2.Find('.') + 3); 195 str1 += str2; 196 MessageBox(str1, L"购买、租用CD", MB_OK MB_ICONWARNING); 197 return; 198 } 199 else if (nRet == 8) //购买空闲CD, 余额不足, 违背存储过程 200 { 201 CString str1, str2; 202 str1.Format(L"余额不足, 请先充值! \r\n购买本CD金额: %lf", price); 203 str1 = str1.Left(str1.Find('.') + 3); 204 str2.Format(L"\r\n你的余额: %lf", bal); 205 str2 = str2.Left(str2.Find('.') + 3); 206 str1 += str2; 207 MessageBox(str1, L"购买、租用CD", MB_OK MB_ICONWARNING); 208 return; 209 } 210 211 catch (_com_error e) 212 { 213 m_pConnection->RollbackTrans(); 214 CString temp1 = e.Description(), temp2 = e.ErrorMessage(); 215 AfxMessageBox(temp1 + L"\r\n" + temp2); //显示错误信息 216 return; 217 } 218 219 //插入一条记录到Trade表中 220 CString strTime, strSql, sqlHasRecord, temp1, tradeid; 221 CTime tm; 222 tm = CTime::GetCurrentTime(); 223 strTime = tm.Format("%Y%m%d"); 224 225 m_pRecordset.CreateInstance(__uuidof(Recordset)); //创建实例 226 sqlHasRecord.Format(L"select * from Trade where trade_id like '%s', strTime); 227 temp1 = L"%s' order by trade_date desc"; 228 sqlHasRecord += temp1; 229 try 230 { 231 HRESULT hr = m_pRecordset->Open((variant_t)sqlHasRecord, 232 _variant_t((IDispatch*)m_pConnection, true), adOpenDynamic, adLockPessimistic, adCmdText); 233 if (SUCCEEDED(hr)) 234 { 235 if (m_pRecordset->rsEOF && m_pRecordset->BOF) //记录集为空 236 { 237 tradeid.Format(L"%s_%d", strTime, 1); 238 } 239 else 240 { 241 CString temp2; 242 temp2 = m_pRecordset->GetCollect(L"trade_id"); 243 int down = temp2.Find('.'); 244 temp2 = temp2.Mid(down + 1); 245 tradeid.Format(L"%s_%d", strTime, _ttoi(temp2) + 1); 246 } 247 } 248 else 249 { 250 AfxMessageBox(L"Create instance of recordset failed!"); 251 return; 252 } 253 m_pRecordset->Close(); 254 255 strSql.Format(L"insert into Trade values('%s','%s','%s',%d,getdate())", 256 tradeid, id, cdid, tradetype); 257 m_pConnection->Execute(_bstr_t(strSql), 0, adCmdText); 258 } 259 catch (_com_error e) 260 { 261 CString temp1 = e.Description(), temp2 = e.ErrorMessage(); 262 AfxMessageBox(temp1 + L"\r\n" + temp2); //显示错误信息 263 return; 264 } 265 266 CDialogEx::OnOK(); 267 268 catch (_com_error e) 269 { 270 CString temp1 = e.Description(), temp2 = e.ErrorMessage(); 271 AfxMessageBox(temp1 + L"\r\n" + temp2); //显示错误信息 272 return; 273 } 274 275 CDialogEx::OnOK(); 276 277 //CDialogEx::OnOK(); 278 } </pre> <p>根据返回值作相应的处理</p>
<p>程序演示 (2分)</p>	<p>说明：不违背存储过程，能够执行更新操作</p> <p>以下只展示程序效果，如果要检查减去的余额对不对，可以在表 Stock、表 Customer 和视图 CusCDlist 中查询比对。检查租用次数是（lent_count）否加了 1 可以在表 Stock 中查询比对。检查交易类型（trade_type）是否正确可以在表 Trade 中查询比对。</p>

1. 顾客租用店内空闲的 CD，且余额充足：（返回值为 5）

操作前的帐户信息：

My CDStore - 顾客查看个人帐户

帐户ID: 20190522_1

余额: 251.00

注册时间: 2019/5/22 10:45:00

我正在租用的CD:

20190512_7

我已经购买的CD:

20190516_10
20190516_15

确定

操作成功：

My CDStore - 顾客购买、租用CD

如果购买自己正在租用的CD，则扣除相应的差价。

你的余额: 251.00

交易类型 ☐ 购买 ☒ 租用

CD号

20190516_16

确定

取消

购买、租用CD

租用成功!

你的余额: 239.00

确定

操作后的帐户信息：

My CDStore - 顾客查看个人帐户

帐户ID: 20190522_1

余额: 239.00

注册时间: 2019/5/22 10:45:00

我正在租用的CD:

20190512_7
20190516_16

我已经购买的CD:

20190516_10
20190516_15

确定

2. 顾客购买店内空闲的 CD，且余额充足：（返回值为 6）

操作前的帐户信息：

My CDStore - 顾客查看个人帐户

帐户ID: 20190522_1

余额: 239.00

注册时间: 2019/5/22 10:45:00

我正在租用的CD:

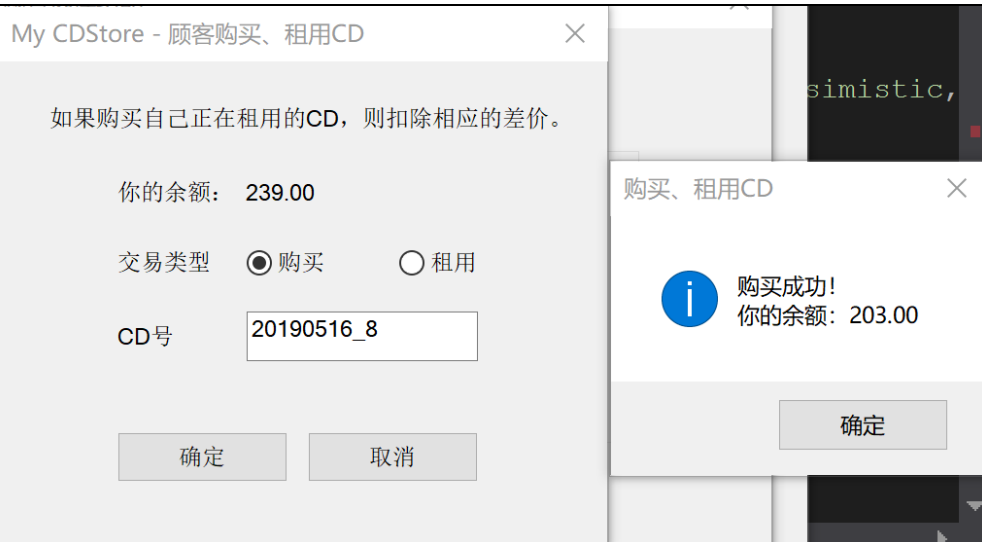
20190512_7
20190516_16

我已经购买的CD:

20190516_10
20190516_15

确定

操作成功：



操作后的帐户信息：



3. 顾客购买自己租用的 CD，且余额充足：（返回值为 2）

操作前的帐户信息和要操作的 CD：

My CDStore - 顾客查看个人帐户

帐户ID: 20190522_1

余额: 203.00

注册时间: 2019/5/22 10:45:00

我正在租用的CD:

20190512_7
20190516_16

我已经购买的CD:

20190516_10
20190516_15
20190516_8

确定

操作成功:

My CDStore - 顾客购买、租用CD

如果购买自己正在租用的CD, 则扣除相应的差价。

你的余额: 203.00

交易类型 ☒ 购买 ☐ 租用

CD号 20190512_7

确定 取消

购买、租用CD

购买成功!
你的余额: 158.00

确定

About

操作后的帐户信息和被操作的 CD:

My CDStore - 顾客查看个人帐户

帐户ID: 20190522_1

余额: 158.00

注册时间: 2019/5/22 10:45:00

我正在租用的CD:

20190516_16

我已经购买的CD:

20190516_10
20190516_15
20190512_7
20190516_8

确定

（插在这个位置是因为查询结果显示的顺序是默认按照第一次交易操作的时间先后排序，对这个 CD 的第一个操作是租用，时间晚于前两个购买了的 CD）

程序演示
(2 分)

说明：违背存储过程，系统报错；

1. 操作店内库存没有的 CD: (返回值为-1)

My CDStore - 顾客购买、租用CD

如果购买自己正在租用的CD，则扣除相应的差价。

你的余额: 158.00

交易类型 ☒ 购买 ☐ 租用

CD号

确定 取消

购买、租用CD

! 库存没有此CD!

确定

About

2. 操作其他顾客正在租用或已经购买的 CD: (返回值为 0)
要操作的 CD 和这个 CD 所对应的帐户: (操作前后无变化)

My CDStore - 顾客查看个人帐户

帐户ID: 20170307_1

余额: 63.00

注册时间: 2017/3/7

我正在租用的CD:

20190516_11
20190514_5

我已经购买的CD:

确定

要操作的帐户（与上图的帐户 20170307_1 不是一个帐户，操作前后无变化）:

My CDStore - 顾客查看个人帐户

帐户ID: 20190522_1

余额: 158.00

注册时间: 2019/5/22 10:45:00

我正在租用的CD:

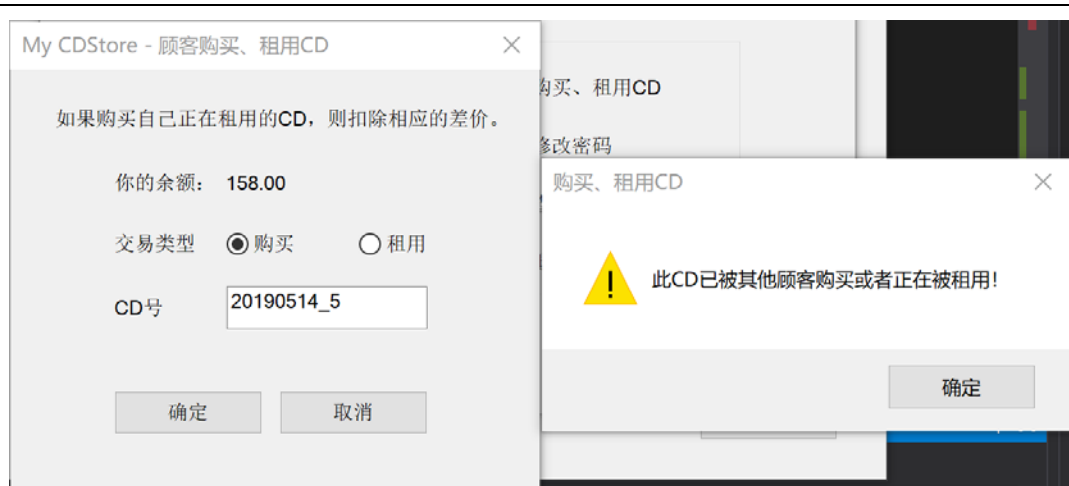
20190516_16

我已经购买的CD:

20190516_10
20190516_15
20190512_7
20190516_8

确定

在上图的帐户（20190522_1）下操作失败:



3. 操作自己已经购买的 CD: (返回值为 1)

要操作的帐户和 CD: (操作前后无变化)



在上图的帐户 (20190522_1) 下操作这个已经购买的 CD:

My CDStore - 顾客购买、租用CD

如果购买自己正在租用的CD，则扣除相应的差价。

你的余额: 158.00

交易类型 ☐ 购买 ☒ 租用

CD号

确定

取消

购买、租用CD

你已购买此CD!

确定

About

My CDStore - 顾客购买、租用CD

如果购买自己正在租用的CD，则扣除相应的差价。

你的余额: 158.00

交易类型 ☒ 购买 ☐ 租用

CD号

确定

取消

购买、租用CD

你已购买此CD!

确定

About

4. 重复租用自己正在租用的 CD: (返回值为 4)

要操作的帐户和 CD: (操作前后无变化)

My CDStore - 顾客查看个人帐户

帐户ID: 20190522_1

余额: 158.00

注册时间: 2019/5/22 10:45:00

我正在租用的CD:

20190516_16

我已经购买的CD:

20190516_10
20190516_15
20190512_7
20190516_8

确定

操作失败:

My CDStore - 顾客购买、租用CD

如果购买自己正在租用的CD, 则扣除相应的差价。

你的余额: 158.00

交易类型 ☐ 购买 ☒ 租用

CD号 20190516_16

确定 取消

你已正在租用此CD! 不能重复租用!

确定

About

5. 购买自己正在租用的 CD, 但是余额不足: (返回值为 3)

要操作的帐户和 CD: (操作前后无变化)

My CDStore - 顾客查看个人帐户

帐户ID: 20190516_1

余额: 26.00

注册时间: 2019/5/16 11:52:55

我正在租用的CD:

20190512_12

20190516_4

我已经购买的CD:

确定

操作失败:

My CDStore - 顾客购买、租用CD

如果购买自己正在租用的CD, 则扣除相应的差价。

你的余额: 26.00

交易类型 ☒ 购买 ☐ 租用

CD号

20190512_12

确定

取消

购买、租用CD

余额不足, 请先充值!

购买本CD金额: 36.00

你的余额: 26.00

确定

About

6. 购买空闲 CD, 但是余额不足: (返回值为 8)

要操作的帐户: (操作前后无变化)

My CDStore - 顾客查看个人帐户

帐户ID: 20190516_1

余额: 26.00

注册时间: 2019/5/16 11:52:55

我正在租用的CD:

20190512_12
20190516_4

我已经购买的CD:

确定

操作失败:

My CDStore - 顾客购买、租用CD

如果购买自己正在租用的CD, 则扣除相应的差价。

你的余额: 26.00

交易类型 ☒ 购买 ☐ 租用

CD号 20190512_5

确定 取消

余额不足, 请先充值!
购买本CD金额: 60.00
你的余额: 26.00

确定

7. 租用空闲 CD, 但是余额不足: (返回值为 7)

要操作的帐户: (操作前后无变化)

	<div data-bbox="335 203 1106 887"><p>My CDStore - 顾客查看个人帐户</p><p>帐户ID: 20190515_1</p><p>余额: 13.00</p><p>注册时间: 2019/5/15 14:08:06</p><p>我正在租用的CD: 20190516_5</p><p>我已经购买的CD:</p><p>确定</p></div> <p>操作失败:</p> <div data-bbox="335 947 1345 1473"><div data-bbox="335 947 930 1473"><p>My CDStore - 顾客购买、租用CD</p><p>如果购买自己正在租用的CD, 则扣除相应的差价。</p><p>你的余额: 13.00</p><p>交易类型 <input type="radio"/> 购买 <input checked="" type="radio"/> 租用</p><p>CD号 20190512_5</p><p>确定 取消</p></div><div data-bbox="930 947 1345 1473"><p>购买、租用CD</p><p>余额不足, 请先充值!</p><p>租用本CD金额: 15.00</p><p>你的余额: 13.00</p><p>确定</p><p>About</p></div></div>
--	---

	者触发器实现的。
--	----------

7. 含有视图的查询操作（15 分）

说明	<p>（1 分）简要说明该操作所要完成的功能；</p> <p>（1 分）简要说明建立的该视图的功能；</p> <p>（2 分）简要说明该操作涉及的关系数据表（以“表名”的形式给出）</p> <p>（1 分）简要说明表连接涉及的字段（以“表 1. 属性=表 2. 属性”）</p> <p>（6 分）实现该操作的关键代码（高级语言、SQL），截图即可；</p> <p>（4 分）如何执行该操作，按所述方法能够正常演示程序则给分。</p>
操作功能描述（1 分）	<p>顾客通过输入 CD 名、作者名和类别精准查询 CD 店内空闲（即尚未被购买或者正在被租用）的 CD，查询结果按照入库时间先后排序，不支持模糊查找（即输入非空的且不完整的 CD 名或者作者名就能找到相应的 CD）。如果某一项（CD 名、作者名或类别）输入的内容为空则在这一项不加用于筛选的查询条件。</p>
视图功能描述（1 分）	<p>创建一个视图 CusCDlist，该视图包含的字段有：CD 号（由 Stock 表中的 stock_id 生成）、CD 名（由 CDinfo 表中的 cd_name 生成）、作者名（由 CDinfo 表中的 artist_name 生成）、类别（由 Genre 表中的 genre_name 生成）、进价（由 Stock 表中的 pur_price*1.2 的结果生成）、租用价格（由 Stock 表中的 pur_price*1.2*0.25 的结果生成）、入库日期（由 Stock 表中的 pur_date 生成）、租用次数（由 Stock 表中的 lent_count 生成）。其他在此视图中涉及到的表的字段对顾客来说都是不可见的数据库（涉及到店的经营情况等等，要保证这些数据的安全），所以要为顾客创建一个只包含顾客可见属性或者辅助属性的视图，方便顾客进行查询。入库时间作为查询结果排序依据，不展示在实际的查询界面中。</p> <p>not exists 语句中的条件使用关联子查询，变量为 Stock 表的 s1，条件表明在 Trade 表中不存在与 s1 的 stock_id 相等的记录，意思是这个 CD 是空闲的，没有被购买或者被正在租用。</p>
涉及的关系表（2 分）	Stock s1, CDinfo, Genre
表连接字段（1 分）	<p>s1.cd_id=CDinfo.cd_id</p> <p>CDinfo.genre_id=Genre.genre_id</p>
创建视图代码（3 分）	<p>（截屏）</p> <pre> 1 create view CusCDlist as 2 select s1.stock_id as CD号,cd_name as CD名, 3 artist_name as 作者名,genre_name as 类别, 4 pur_price*1.2 as 售价,pur_price*1.2*0.25 as 租用价格, 5 pur_date as 入库时间,lent_count as 租用次数 6 from Stock s1,CDinfo,Genre 7 where s1.cd_id=CDinfo.cd_id and 8 CDinfo.genre_id=Genre.genre_id and 9 not exists(select * from Trade 10 where Trade.stock_id=s1.stock_id) </pre>
查询代码（3 分）	<p>（截屏）</p> <p>CLookCD::OnBnClickedSearch(): 一共有 8 种供查询的 SQL 语句，根据某个空是否为空来决定使用哪个语句。</p>


```

42 void CLookCD::OnBnClickedSearch()
43 {
44     // TODO: 在此添加控件通知处理程序代码
45     m_resultlist.ResetContent(); //清空上次查询的结果
46     CString cdname, artist, genre, genre2;
47     m_cdname.GetWindowTextW(cdname);
48     m_artistname.GetWindowTextW(artist);
49     int judge;
50     m_genre.GetWindowTextW(genre);
51     judge = m_genre.FindString(0, genre);
52     if (judge >= 0)
53     {
54         m_genre.GetLBText(judge, genre2);
55     }
56     if (!genre.IsEmpty() && genre != genre2)
57     {
58         MessageBox(L"没有这个CD类别!", L"查看空闲CD", MB_OK | MB_ICONWARNING);
59         return;
60     }
61     m_pRecordset.CreateInstance(__uuidof(Recordset)); //创建实例
62     CString sqlHasRecord;
63     if (cdname.IsEmpty() && !artist.IsEmpty() && !genre.IsEmpty())
64     {
65         if (judge > 0)
66         {
67             sqlHasRecord.Format(L"select * from CusCDlist where [CD名]='%s' and [作者名]='%s' and [类别]='%s'",
68                                 cdname, artist, genre);
69         }
70         else
71         {
72             sqlHasRecord.Format(L"select * from CusCDlist where [CD名]='%s' and [作者名]='%s' and [类别] "
73                                 "is null", cdname, artist);
74         }
75     }
76     else if (!cdname.IsEmpty() && !artist.IsEmpty() && genre.IsEmpty())
77     {

```

```

78         sqlHasRecord.Format(L"select * from CusCDlist where [CD名]='%s' and [作者名]='%s'", cdname, artist);
79     }
80     else if (!cdname.IsEmpty() && artist.IsEmpty() && !genre.IsEmpty())
81     {
82         if (judge > 0)
83         {
84             sqlHasRecord.Format(L"select * from CusCDlist where [CD名]='%s' and [类别]='%s'", cdname, genre);
85         }
86         else
87         {
88             sqlHasRecord.Format(L"select * from CusCDlist where [CD名]='%s' and [类别] is null", cdname);
89         }
90     }
91     else if (!cdname.IsEmpty() && artist.IsEmpty() && genre.IsEmpty())
92     {
93         sqlHasRecord.Format(L"select * from CusCDlist where [CD名]='%s'", cdname);
94     }
95     else if (cdname.IsEmpty() && !artist.IsEmpty() && !genre.IsEmpty())
96     {
97         if (judge > 0)
98         {
99             sqlHasRecord.Format(L"select * from CusCDlist where [作者名]='%s' and [类别]='%s'", artist, genre);
100         }
101         else
102         {
103             sqlHasRecord.Format(L"select * from CusCDlist where [作者名]='%s' and [类别] is null", artist);
104         }
105     }
106     else if (cdname.IsEmpty() && !artist.IsEmpty() && genre.IsEmpty())
107     {
108         sqlHasRecord.Format(L"select * from CusCDlist where [作者名]='%s'", artist);
109     }
110     else if (cdname.IsEmpty() && artist.IsEmpty() && !genre.IsEmpty())
111     {
112         if (judge > 0)
113         {
114             sqlHasRecord.Format(L"select * from CusCDlist where [类别]='%s'", genre);

```

```

115
116     else
117     {
118         sqlHasRecord.Format(L"select * from CusCDlist where [类别] is null");
119     }
120 }
121 else if (cdname.IsEmpty() && artist.IsEmpty() && genre.IsEmpty())
122 {
123     sqlHasRecord.Format(L"select * from CusCDlist");
124 }
125 sqlHasRecord += L" order by [入库时间]";
126 try
127 {
128     HRESULT hr = m_pRecordset->Open((_variant_t)sqlHasRecord,
129     _variant_t{(IDispatch*)m_pConnection, true),
130     adOpenDynamic, adLockPessimistic, adCmdText);
131     if (SUCCEEDED(hr))
132     {
133         if (m_pRecordset->rsEOF && m_pRecordset->BOF) //记录集为空, 没有与任何记录匹配上
134         {
135             MessageBox(L"库存没有相应的空闲CD!", L"查看空闲CD", MB_OK | MB_ICONWARNING);
136         }
137         else
138         {
139             while (!m_pRecordset->rsEOF) //遍历返回的每一条记录
140             {
141                 CString str, all;
142                 str = (LPCSTR) bstr_t(m_pRecordset->GetCollect("CD号"));
143                 all = all + str + L", ";
144                 _variant_t vValue = m_pRecordset->GetCollect("CD名");
145                 if (vValue.vt != VT_NULL)
146                 {
147                     str = (LPCSTR) _bstr_t(m_pRecordset->GetCollect("CD名"));
148                 }
149                 else
150                 {
151                     str = L"未知";
152                 }
153                 all = all + str + L", ";
154                 vValue = m_pRecordset->GetCollect("作者名");
155                 if (vValue.vt != VT_NULL)
156                 {
157                     str = (LPCSTR) _bstr_t(m_pRecordset->GetCollect("作者名"));
158                 }
159                 else
160                 {
161                     str = L"匿名";
162                 }
163                 all = all + str + L", ";
164                 vValue = m_pRecordset->GetCollect("类别");
165                 if (vValue.vt != VT_NULL)
166                 {
167                     str = (LPCSTR) _bstr_t(m_pRecordset->GetCollect("类别"));
168                 }
169                 else
170                 {
171                     str = L"未知类别";
172                 }
173                 all = all + str + L", ";
174                 str = (LPCSTR) _bstr_t(m_pRecordset->GetCollect("售价"));
175                 all = all + str + L", ";
176                 str = (LPCSTR) _bstr_t(m_pRecordset->GetCollect("租用价格"));
177                 all = all + str + L", ";
178                 str = (LPCSTR) _bstr_t(m_pRecordset->GetCollect("租用次数"));
179                 all = all + str;
180                 m_resultlist.InsertString(m_resultlist.GetCount(), all);
181                 m_pRecordset->MoveNext();
182             }
183         }
184     }
185     else
186     {
187         AfxMessageBox(L"Create instance of recordset failed!");
188         return;
189     }
190     m_pRecordset->Close();
191 }
192 catch (_com_error e)
193 {
194     CString temp1 = e.Description(), temp2 = e.ErrorMessage();
195     AfxMessageBox(temp1 + L"\r\n" + temp2); //显示错误信息
196     return;
197 }
198 }

```

排序依据

程序演示
(4分)

这里演示的结果, 由于和前面的演示的时间点不同, 不作为前面部分演示的依据 (CD 是否空闲)。

1. 什么都不输入, 点击 “Search” 返回所有空闲 CD。

My CDStore - 查看空闲CD

输入搜索内容

CD名

作者名

类别

Search

格式（按入库时间先后排序）：CD号、CD名、作者名、类别、售价、租用价格、租用次数

20190512_1、Red、Taylor Swift、Country、36、9、4

20190512_2、Red、Taylor Swift、Country、36、9、0

20190512_3、大碗宽面、吴亦凡、Pop、60、15、3

20190512_4、大碗宽面、吴亦凡、Pop、60、15、6

20190512_5、大碗宽面、吴亦凡、Pop、60、15、20

20190512_7、大碗宽面、吴亦凡、Pop、60、15、1

20190512_8、Fearless、Taylor Swift、Country、48、12、5

20190512_10、Fearless、Taylor Swift、Country、48、12、0

20190514_1、Loud、Rihanna、Pop、48、12、0

20190514_2、Loud、Rihanna、Pop、48、12、7

20190514_4、21、Adele、Pop、42、10.5、0

20190516_1、Kaleidoscope、DJ Okawari、NewAge、48、12、0

2. 输入特定条件，按条件筛选空闲 CD（只展示一部分例子）

My CDStore - 查看空闲CD

输入搜索内容

CD名

大碗宽面

作者名

吴亦凡

类别

Search

格式（按入库时间先后排序）：CD号、CD名、作者名、类别、售价、租用价格、租用次数

20190512_3、大碗宽面、吴亦凡、Pop、60、15、3

20190512_4、大碗宽面、吴亦凡、Pop、60、15、6

20190512_5、大碗宽面、吴亦凡、Pop、60、15、20

20190512_7、大碗宽面、吴亦凡、Pop、60、15、1

My CDStore - 查看空闲CD



输入搜索内容

CD名

作者名

类别

Search

格式（按入库时间先后排序）：CD号、CD名、作者名、类别、售价、租用价格、租用次数

20190512_1、Red、Taylor Swift、Country、36、9、4
20190512_2、Red、Taylor Swift、Country、36、9、0
20190512_8、Fearless、Taylor Swift、Country、48、12、5
20190512_10、Fearless、Taylor Swift、Country、48、12、0

My CDStore - 查看空闲CD



输入搜索内容

CD名

Kaleidoscope

作者名

DJ Okawari

类别

NewAge

Search

格式（按入库时间先后排序）：CD号、CD名、作者名、类别、售价、租用价格、租用次数

20190516_1、Kaleidoscope、DJ Okawari、NewAge、48、12、0
20190516_2、Kaleidoscope、DJ Okawari、NewAge、48、12、10
20190516_3、Kaleidoscope、DJ Okawari、NewAge、48、12、0

My CDStore - 查看空闲CD

输入搜索内容

CD名

作者名

Rihanna

类别

Search

格式（按入库时间先后排序）：CD号、CD名、作者名、类别、售价、租用价格、租用次数

20190514_1、Loud、Rihanna、Pop、48、12、0
20190514_2、Loud、Rihanna、Pop、48、12、7

3. 如果输入了库存里根本没有的 CD，或者库存里本来有的 CD 但都是非空闲的 CD，但是类别没有输入或者输入的是店里已有的 CD 类别（可以在下拉菜单中选择，也可以自己输入）的时候；则会提示“库存中没有相应的空闲 CD”并且结果区空白。（只展示一部分例子）

My CDStore - 查看空闲CD

输入搜索内容

CD名

作者名

类别

R&B

Search

格式（按入库时间先后排序）：CD号、CD名、作者名、类别、售价、租用价格、租用次数

查看空闲CD

!

库存没有相应的空闲CD!

确定

My CDStore - 查看空闲CD

输入搜索内容

CD名

Bangerz

作者名


Miley Cyrus

类别

Search

格式（按入库时间先后排序）：CD号、CD名、作者名、类别、售价、租用价格、租用次数

查看空闲CD

 库存没有相应的空闲CD!

确定

My CDStore - 查看空闲CD

输入搜索内容

CD名

出山

作者名

花粥

类别

Search

格式（按入库时间先后排序）：CD号、CD名、作者名、类别、售价、租用价格、租用次数

查看空闲CD

 库存没有相应的空闲CD!

确定

My CDStore - 查看空闲CD

输入搜索内容

CD名

作者名

那英

类别

Search

格式（按入库时间先后排序）：CD号、CD名、作者名、类别、售价、租用价格、租用次数

查看空闲CD

!

库存没有相应的空闲CD!

确定

4. 如果输入了店内没有的 CD 类别，无论其他地方有没有输入，都会提示“没有这个 CD 类别”并且结果区空白。

My CDStore - 查看空闲CD

输入搜索内容

CD名

作者名

类别

Search

格式（按入库时间先后排序）：CD号、CD名、作者名、类别、售价、租用价格、租用次数

未知类别

Pop

Rock

Dance

Country

Blues

R&B

Jazz

NewAge

Metal

Hip-Hop

Rap

Instrumental

Vocal

Other

Disco

Folk

	<div data-bbox="336 203 1311 929"><div>My CDStore - 查看空闲CD</div><div>输入搜索内容</div><div><div>CD名</div><div>作者名</div><div>类别</div><div>Good</div><div>Search</div></div><div>格式（按入库时间先后排序）：CD号、CD名、作者名、类别、售价、租用价格、租用次数</div><div><div>查看空闲CD</div><div><div>!</div><div>没有这个CD类别!</div></div><div>确定</div></div></div> <div data-bbox="336 949 1311 1675"><div>My CDStore - 查看空闲CD</div><div>输入搜索内容</div><div><div>CD名</div><div>aaa</div><div>作者名</div><div>类别</div><div>Good</div><div>Search</div></div><div>格式（按入库时间先后排序）：CD号、CD名、作者名、类别、售价、租用价格、租用次数</div><div><div>查看空闲CD</div><div><div>!</div><div>没有这个CD类别!</div></div><div>确定</div></div></div>
备注	表连接字段：sl 是 Stock 表的别名。