北京邮电大学

实验报告

课程名称：数据库系统原理

实验名称：数据库接口实验

计算机系2015211312班 姓名刘佳鑫

计算机系2015211312班 姓名万诗婕

教师吴起凡 成绩\_\_\_\_\_\_\_\_\_

# 第七章 实验六 数据库接口实验

### 实验目的

1. 通过编写数据库应用程序，培养数据库应用程序开发能力。
2. 熟悉数据库应用程序设计的多种接口的配置，培养相关的软件配置能力。

### 实验原理

数据库应用程序设计是数据库应用的一个重要方面。SQL语言除了以用户交互的方式使用外，还可以被数据库应用程序直接访问。

通常DBMS支持的能直接访问数据库的是SQL语言，而SQL语言不像高级语言（例如C、C++等）具备很好的数据处理能力。通常的情况是需要将两种语言结合起来，利用SQL访问数据，而将数据传递给某种高级语言程序，处理后又利用SQL写回数据库。

这种嵌在高级语言程序中的SQL语句称为嵌入式SQL（或者称为ESQL），和以前使用的交互式SQL（或者称为ISQL）不一样，它们随着程序执行被调用，辅助程序完成数据库数据读写的功能，而高级语言程序则负责对数据库中的数据的分析处理转换等操作。

ESQL又分成两种，如果在程序执行前SQL的结构就已经确定，最多是可以在执行时传递一些数值参数，那么这种ESQL语句称为静态ESQL。如果程序执行时才知道执行的SQL语句时，称为动态ESQL。

SQL SERVER中执行静态ESQL（比如SQLJ）时，首先利用预编译将宿主程序语言中的SQL语句分离出来，代之以过程或函数调用，然后对剩余程序正常编译和连接库函数等。分离出来的SQL语句则在数据库端进行处理，进行语法检查、安全性检查和优化执行策略，绑定在数据库上形成包（packet）供应用程序调用。

而通过ODBC、JDBC、OLEDB等访问数据库都是动态ESQL。SQL SERVER执行它们时因为无法事先确实知道是什么样的SQL语句，从而无法进行静态绑定，而这种绑定过程只能在程序执行过程中生成了确定的要执行的SQL语句时才能进行，称为动态绑定。

应用程序的接口则有JDBC、ADO、OLEDB、ODBC等多种。微软在SQL SERVER 2005开始还增加了几项功能：一是service broker，围绕发送和接收消息而设计，处理通信中的消息收发工作，使开发人员集中精力于解决实际问题。二是CLR（Common Language Runtime）组件，就是Microsoft.NET Framework中的公共语言运行库组件，使开发人员可用任何CLR语言来编写过程、函数、触发器等，甚至自定义数据类型。三是对T-SQL做出了许多扩展。

### 实验环境

采用Microsoft SQL Server 2012数据库管理系统作为实验平台。

数据库系统概念设计工具采用的Sybase Power Designer设计工具。

### 实验内容

1. 了解通用数据库应用编程接口（例如JDBC、ODBC等）的配置方法。
2. 利用C语言(或其它支持某种数据库应用程序接口的高级程序设计语言)编程实现简单的数据库应用程序，掌握基于ODBC、JDBC接口的数据库访问的基本原理和方法。
3. 掌握获取、修改数据库连接时长的方法。
4. 通过ODBC、JDBC接口，在数据库应用程序中，执行查找、增加、删除、更新等操作。

### 实验步骤

1. 实验准备：

以教科书第四章关于SQL语言相关内容为基础，课后查阅、自学ODBC/JDBC等接口有关内容，包括体系结构、工作原理、数据访问过程、主要API接口的语法和使用方法等。

1. 在Windows控制面板中通过管理工具下的ODBC数据源工具在客户端新建连接到SQL SERVER数据库服务器的ODBC数据源，测试连接通过后保存，注意名字要和应用程序中引用的数据源一样。**另外要注意创建ODBC数据源时使用的驱动程序必须是支持SQL SERVER的，而不可选择其它驱动程序。**
2. 以实验二建立的数据库为基础，编写 C语言(或其它支持ODBC/JDBC等接口的高级程序设计语言) 数据库应用程序，利用SQLExecDirect语句，实现数据库应用程序对数据库中表（有数据）进行数据查询、删除、插入、更新等操作。**注意：SQL SERVER应设置为应用程序可以访问。**

*实验1 ODBC访问*

**1.1环境配置**

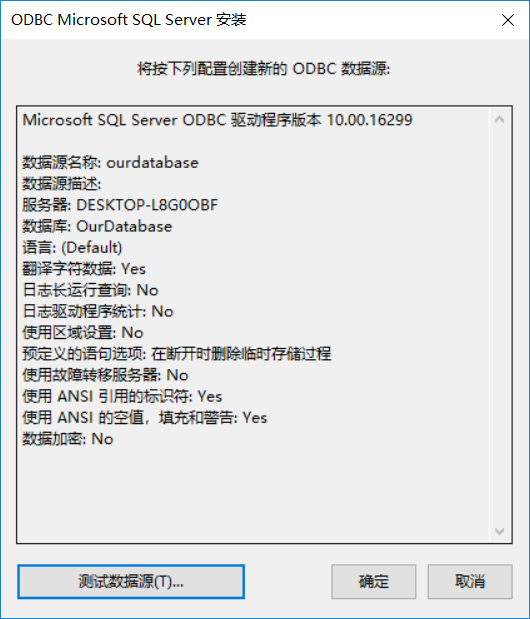
（1）打开ODBC数据源后点击“添加”，添加一个名为“ourdatabase”的数据源；

（2）设置数据源名称，服务器选择本机名称；

（3）设置数据库的用户名和密码（在此之前确保SQLServer的登录方式为SQLServer身份验证登录）；

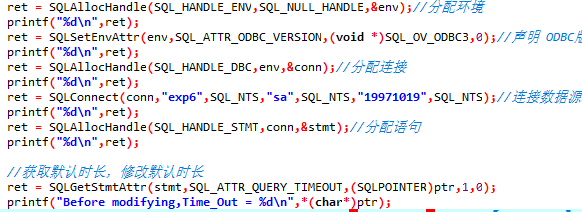
（4）更改默认数据库为要进行操作的数据库（在此之前SQLServer里已建立好university数据库，并将相应的表导入其中）；

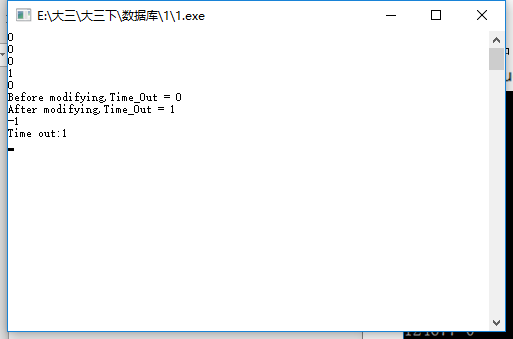
（5）测试数据源，如果测试成功，则表示数据源设置成功。



**1.2 连接时长的获取和修改**

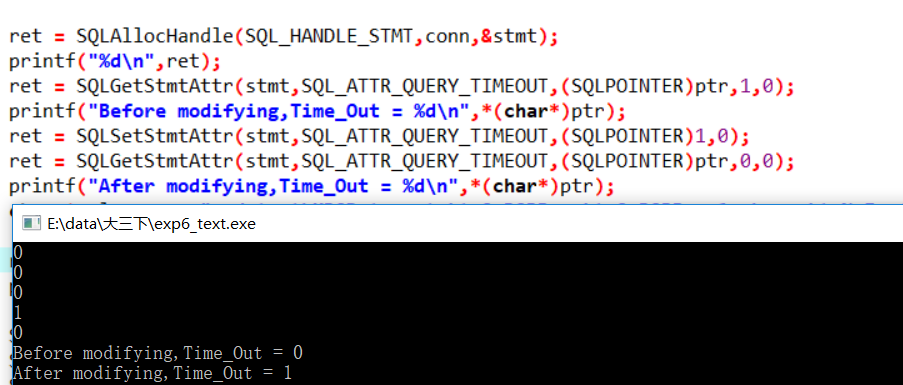
* 1. 通过API函数获取数据库默认连接时间；



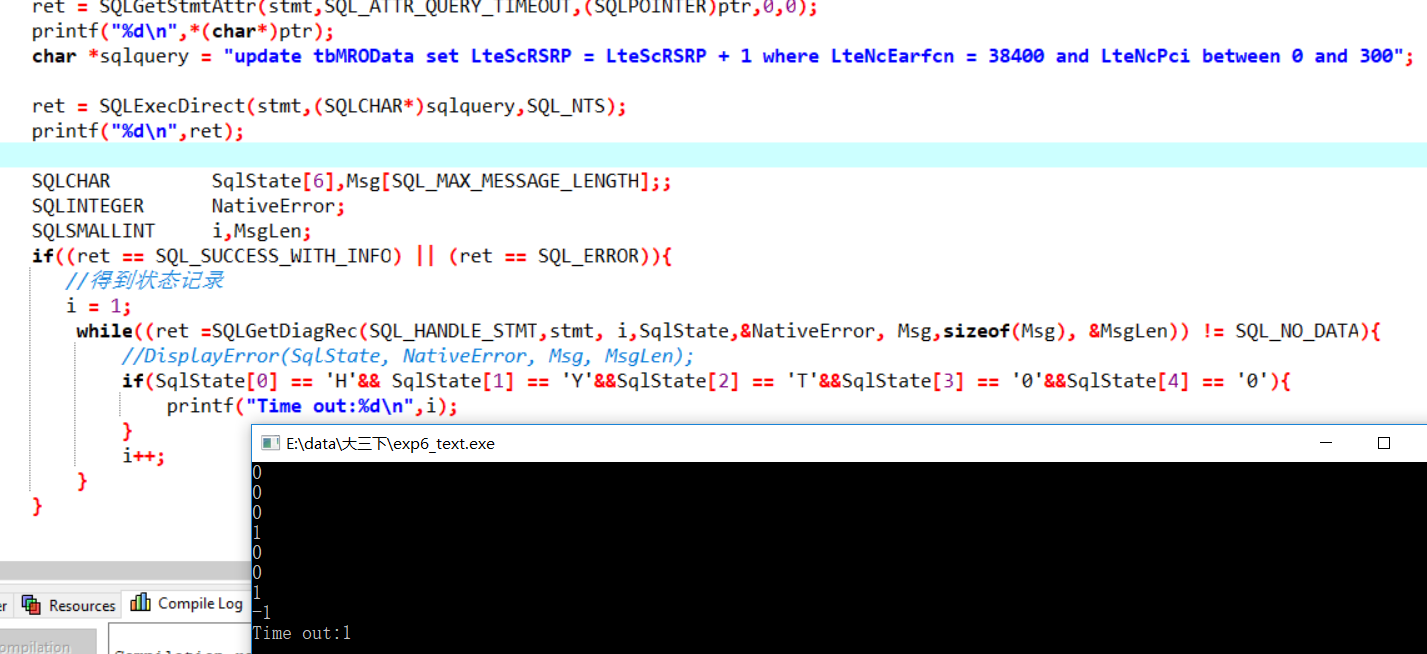


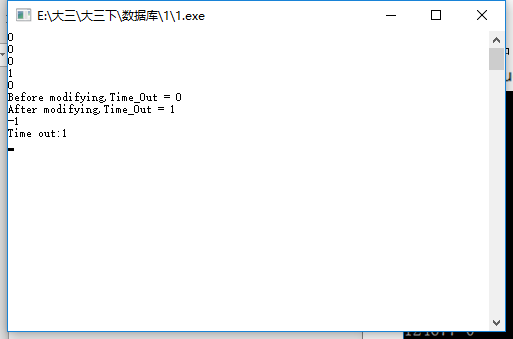
不修改连接时长时，此时长没有限制，会显示为0

2) 在默认连接时间下，完成“4.（建议的）对TD-LTE配置数据库的访问操作”中给出的执行时间较长的查询（三选一），观察是否会超时；



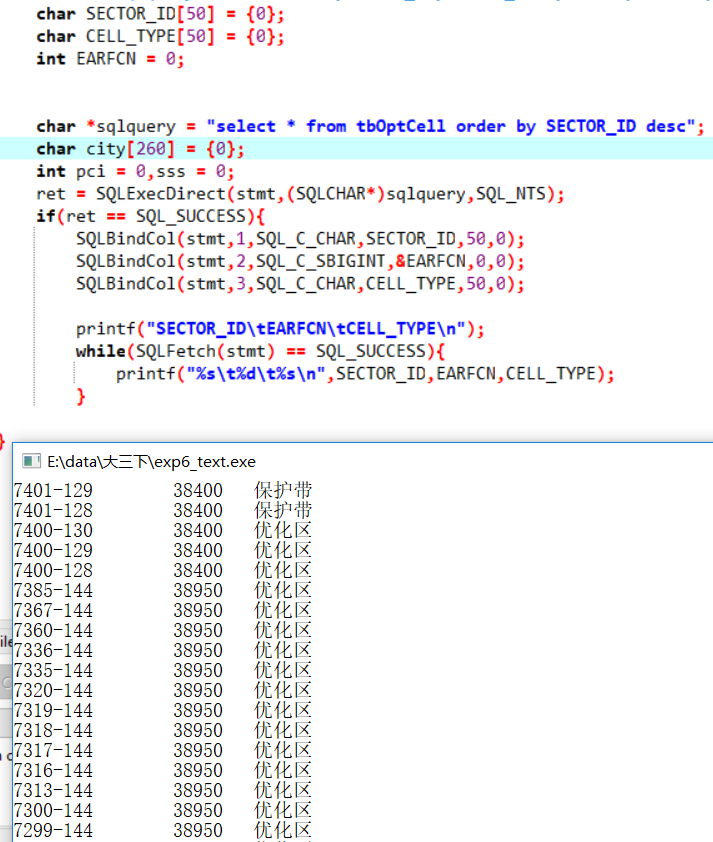
在默认连接时间下，发现并未超时。

3) 增大、降低默认连接时间，观察是否超时。



连接时长修改为1后，发现连接超时。

**1.3 代码实现：查询并进行排序**

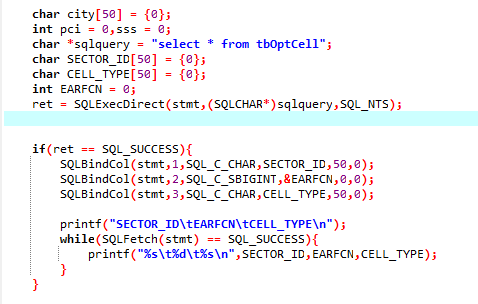
此处选择的是tbOptCell表格，按照SECTOR\_ID降序排列

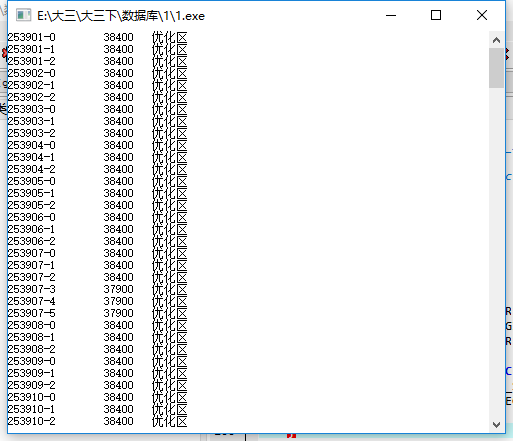
**1.4其他操作**

（1）查询表中所有数据

选取一张表或几张表执行查询操作，并打印出数据（几行数据即可）。

此处选择的是tbOptCell表格

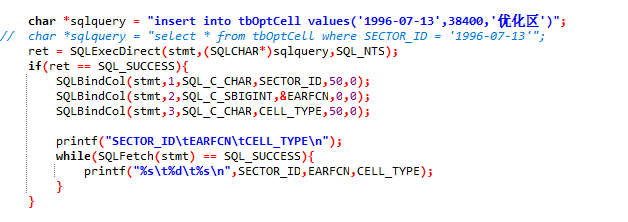




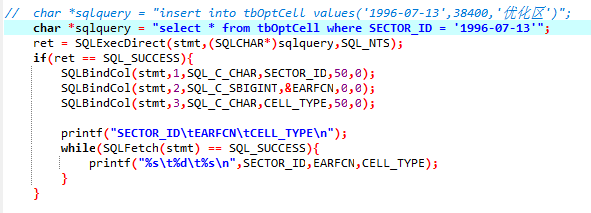
（2）添加一条数据到数据库

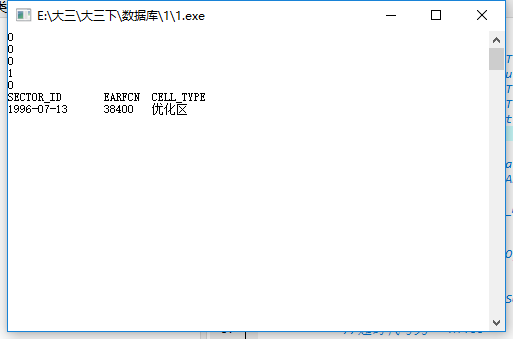
选取一张表执行插入操作，插入成功后打印出新插入的数据。

插入：



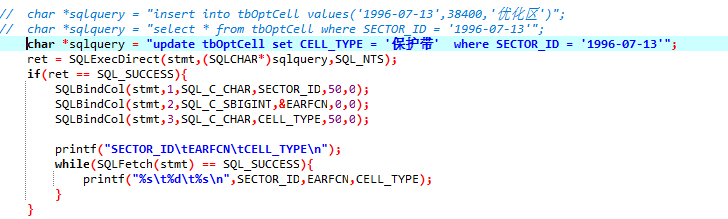
查看新插入的数据：



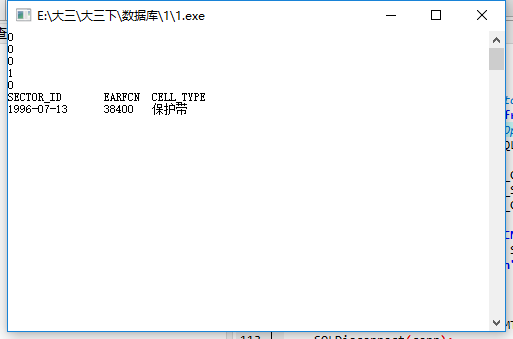


（3）更新

选取一张表执行更新操作，更新成功后打印出更新后的数据。

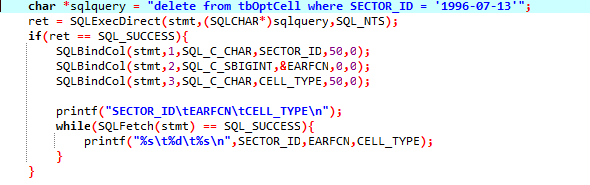


**查看更新后数据：**

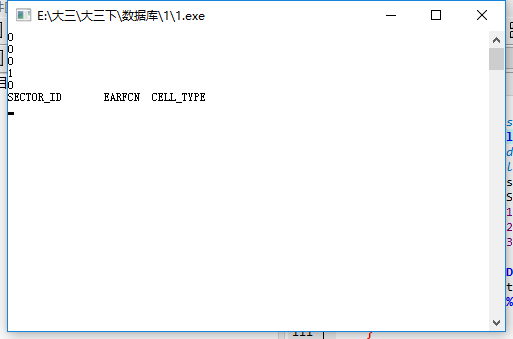


（4）删除

选取一张表执行删除操作，并检查操作是否成功。



删除后再查询，查不到结果

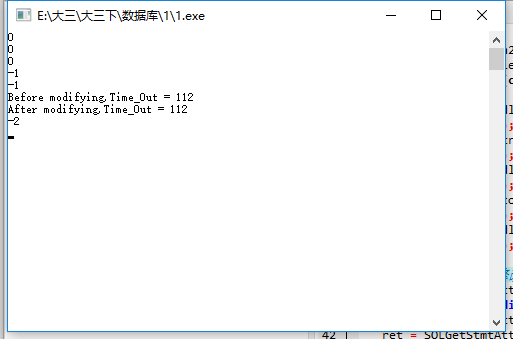


### 实验总结

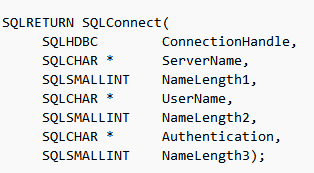
我们遇到的问题：

1．开始

写的是数据库名称，一直都连接不成功



后来应该为ODBC设置的名称。



2. 出现的报错如下：

报错1：

undefined reference to `SQLSetEnvAttr'

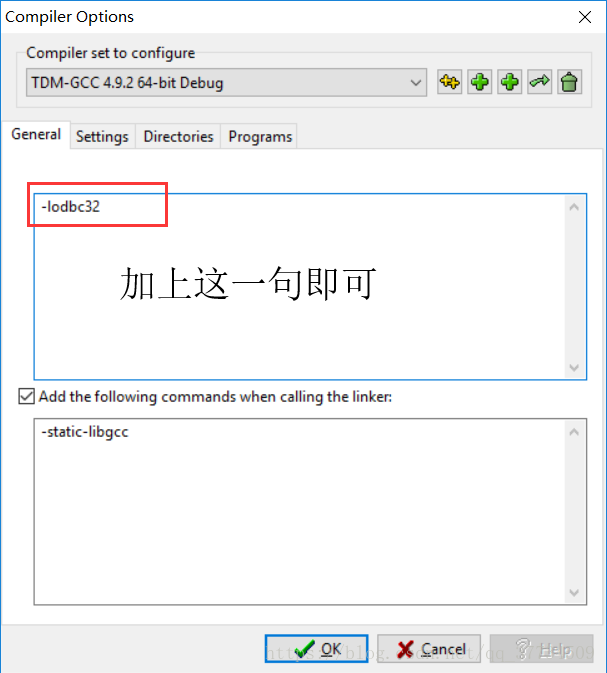
undefined reference to `SQLAllocHandle'

undefined reference to `SQLConnect'

undefined reference to `SQLAllocHandle'

``````

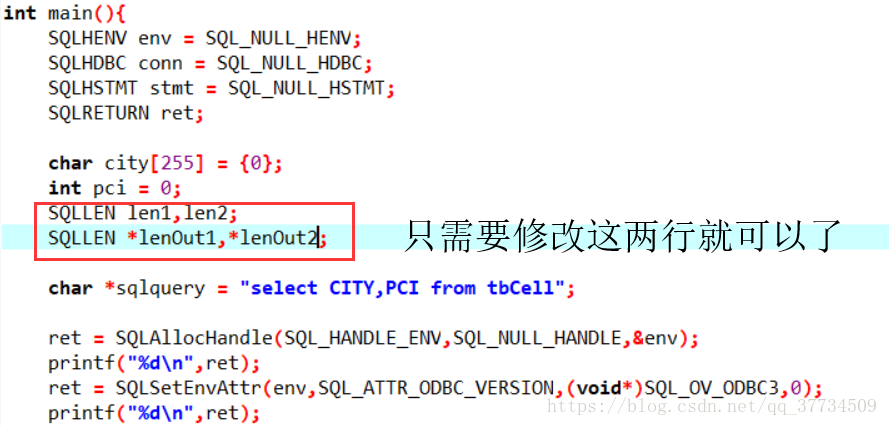
解决方法：   
在Dev的菜单栏中选择【工具/tool】->【编译器选项/compiler options】就可以看见如下窗口 :

   
-lodbc32

报错2：

[Warning] passing argument 6 of 'SQLBindCol' from incompatible pointer type

[Note] expected 'SQLLEN \*' but argument is of type 'int \*'

解决方法如图所示   


其实就是数据类型的不匹配，查看原函数的参数类型，照着改一改就可以。

3. 收获

通过这次实验学会了如何在C语言中去调用数据库中的数据，将数据库和程序链接在一起，可以极大的扩展程序的功能和适用范围，同时也可以增加数据库的使用范围。使更多的工作得到更好的解决方案。学习ODBC等数据库接口，对我们的编程能力的提高很有帮助