1. 安装MySQL connector/ODBC，下载处：

https://dev.mysql.com/downloads/connector/odbc/

1. 配置ODBC数据源，建立名字为DSN1的数据源，具体参考“百度经验”：

http://jingyan.baidu.com/article/8065f87f38b31423312498e4.html

1. 运行C++代码，如下：

|  |
| --- |
| #include <windows.h>  #include <stdio.h>  #include <stdlib.h>  #include "sql.h"  #include "sqltypes.h"  #include "sqlext.h"  #include"sqlucode.h"  #include"odbcinst.h"  #include<iostream>  #include<string>  using namespace std;  RETCODE retcode;//结果返回集  SQLHDBC hdbc;//定义链接句柄  void SQL(string);//执行SQL语句子程序  int main()  {  string str;  SQLHANDLE henv; //定义环境句柄  unsigned char SY[]="DSN1";//ODBC数据源名称  unsigned char db2[]="root";//用户名  unsigned char pass[]="1234";//密码  //分配ODBC环境  retcode=SQLAllocHandle(SQL\_HANDLE\_ENV, SQL\_NULL\_HANDLE, &henv);  if(retcode==SQL\_SUCCESS||retcode==SQL\_SUCCESS\_WITH\_INFO) //声明环境  retcode=SQLSetEnvAttr(henv, SQL\_ATTR\_ODBC\_VERSION,(void\*)SQL\_OV\_ODBC3, 0);  if(retcode == SQL\_SUCCESS || retcode == SQL\_SUCCESS\_WITH\_INFO)  retcode = SQLAllocHandle(SQL\_HANDLE\_DBC, henv, &hdbc); //分配连接句柄  if(retcode == SQL\_SUCCESS || retcode == SQL\_SUCCESS\_WITH\_INFO)  retcode=SQLConnect(hdbc,SY,SQL\_NTS,db2,SQL\_NTS,pass,SQL\_NTS);//链接  if(retcode == SQL\_SUCCESS || retcode == SQL\_SUCCESS\_WITH\_INFO)  {  while(1)  {  cout<<"please input the query:"<<endl;  getline(cin,str);  if(str=="exit") return 0;  SQL(str);  }  }  SQLFreeConnect(hdbc); ////释放链接句柄  SQLFreeEnv(henv); // 释放ODBC环境句柄  system("pause");  return 0;  }    void SQL(string aaa)  {  char L1[50]={'\0'};  char L2[50]={'\0'};  char L3[50]={'\0'};  char L4[50]={'\0'};  char L5[50]={'\0'};  char L6[50]={'\0'};  long lenOut1,lenOut2,lenOut3,lenOut4,lenOut5,lenOut6;  SQLHSTMT hstmt;//定义语句句柄  retcode=SQLAllocHandle(SQL\_HANDLE\_STMT, hdbc, &hstmt); //分配语义句柄  if(retcode==SQL\_SUCCESS)  {  retcode=SQLExecDirect(hstmt,(SQLCHAR \*)(aaa.c\_str()),SQL\_NTS);  if(retcode == SQL\_SUCCESS||retcode == SQL\_SUCCESS\_WITH\_INFO)  { //将结果集中的属性列一一绑定至变量  retcode=SQLBindCol(hstmt,1,SQL\_C\_CHAR,L1,sizeof(L1),&lenOut1);  retcode=SQLBindCol(hstmt,2,SQL\_C\_CHAR,L2,sizeof(L2),&lenOut2);  retcode=SQLBindCol(hstmt,3,SQL\_C\_CHAR,L3,sizeof(L3),&lenOut3);  retcode=SQLBindCol(hstmt,4,SQL\_C\_CHAR,L4,sizeof(L4),&lenOut4);  retcode=SQLBindCol(hstmt,5,SQL\_C\_CHAR,L5,sizeof(L5),&lenOut5);  retcode=SQLBindCol(hstmt,6,SQL\_C\_CHAR,L6,sizeof(L6),&lenOut6);  //把所有捆绑过的数据字段的数据拷贝到相应的缓冲区  retcode=SQLFetch(hstmt);  while(retcode == SQL\_SUCCESS||retcode == SQL\_SUCCESS\_WITH\_INFO)  { if(L2[0]=='\0')  cout<<(string)L1<<endl;  else if(L3[0]=='\0')  cout<<(string)L1<<"\t"<<(string)L2<<endl;  else if(L4[0]=='\0')  cout<<(string)L1<<" "<<(string)L2<<" "<<(string)L3<<endl;  else if(L5[0]=='\0')  cout<<(string)L1<<" "<<(string)L2<<" "<<(string)L3<<" "<<(string)L4<<endl;  else if(L6[0]=='\0')  cout<<(string)L1<<" "<<(string)L2<<" "<<(string)L4<<" "<<(string)L4<<" "<<(string)L5<<endl;  else  cout<<(string)L1<<" "<<(string)L2<<" "<<(string)L4<<" "<<(string)L4<<" "<<(string)L5<<" "<<(string)L6<<endl;  retcode=SQLFetch(hstmt);  }  }  //return 0;  }  } |