GaussDB(for openGauss)
数据库接口实验指导书

1 . 目录

前　言

实验环境说明

本实验环境为华为云环境。

采用 GaussDB(for openGauss)数据库管理系统作为实验平台，数据库访问接口为 ODBC、JDBC，编程语言可采用 C、C++、Java 等。

当以云数据库 GaussDB(for openGauss)为实验平台时，通过公网/IP 登录方式访问数据库。

2． 实验介绍

2.1 实验目的

通过编写数据库应用程序，培养数据库应用程序开发能力。

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。

2.2 实验原理

动态 SQL 与数据库应用编程接口

数据库应用程序设计是数据库应用开发的一个重要方面。数据库系统用户通过两种方式访问数据库：1）直接通过 DBMS 利用 SQL 语句交互式访问数据库；2）通过数据库应用程序，借助嵌入式 SQL 和高级程序设计语言，访问数据库

DBMS 支持 SQL 语言直接访问数据库，但与高级程序设计语言（例如 C、C++、Java 等）相比，SQL 语言数据处理能力较弱，因此需要将 SQL 与高级程序设计语言结合起来，利用 SQL 访问数据，并将数据传递给高级语言程序进行处理，处理结果再利用 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），ESQL 随着应用程序执行被调用，完成数据库数据读写等数据管理功能，而高级语言程序则负责对数据库中数据进行统计分析等深层次处理。ESQL 分成两种：

（1）静态 ESQL，在程序执行前 SQL 的结构就已经确定，但可以在执行时传递一些数值参数。

1.1 数据库系统执行静态 ESQL 时，首先利用预编译分离 C、C++、Java 等宿主程序语言中的 SQL 语句，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。

（2）动态 ESQL，数据库应用程序执行时才确定所执行的 SQL 语句的结构和参数，通过数据库应用编程接口 ODBC、JDBC、Connector/Python、ADO 等访问数据库。

数据库系统执行动态 SQL 时，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。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。

本次实验面向动态 SQL，应用程序采用 ODBC、JDBC 两种接口访问数据库。

2.3 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、C++、Java 等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，访问 LTE 网络数据库，执行查找、增加、删除、更新等操作，掌握基于应用编程接口的数据库访问方法。

3． 实验要求

3.1 实验内容要求

1. 基于 JDBC 接口或基于 ODBC 接口的数据库访问实验，二选一完成一个即可
2. 实验时选取 TD-LTE 数据库作为数据源，可参照后文中以 tbcell 表为数据源的访问样例，自行设计访问 TLE 网络数据库访问操作；
3. 实验中需要完成如下数据库访问操作。

1) 查询

i. 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）。

2) 插入

i. 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。

3) 更新

i. 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。

4) 删除

i. 选取一张表执行删除操作，并检查操作是否成功。

4． 实验步骤

按照下述步骤完成本实验。

步骤 1.实验准备：

以课堂所学关于 SQL 语言相关内容为基础，课后查阅、学习 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（可参考华为官方文档：https://support.huaweicloud.com/devg-opengauss/opengauss_devg_0065.html）。

步骤 2. 数据库访问接口环境配置

根据实验所选的应用编程接口 ODBC、JDBC，分别从不同网站下载接口驱动程序，安装配置接口环境，为后续实验做准备。

具体样例参照 5.1、5.2、5.3 各节。

步骤 3. 数据库连接及访问

参照 5.1、5.2、5.3 各节给出的样例，针对实验二建立的 LTE 网络数据库，编写 C/C++、Java 应用程序，通过 ODBC、JDBC 接口，连接数据库，对数据库内容进行查询、插入、删除、更新等操作，观察记录实验结果。

考虑到本实验可选择两种不同的数据库接口，后面将分别给出采用 ODBC、JDBC 接口时的程序示例。这些程序参照了 TLE 提供的以 tbcell 表据库为访问对象的 ODBC、JDBC 接口访问程序。同学们可以参考这

些示例，针对本次实验对 LTE 网络数据库的具体访问要求，编写程序各项实验内容。

5． 示例

5.1 ODBC 访问

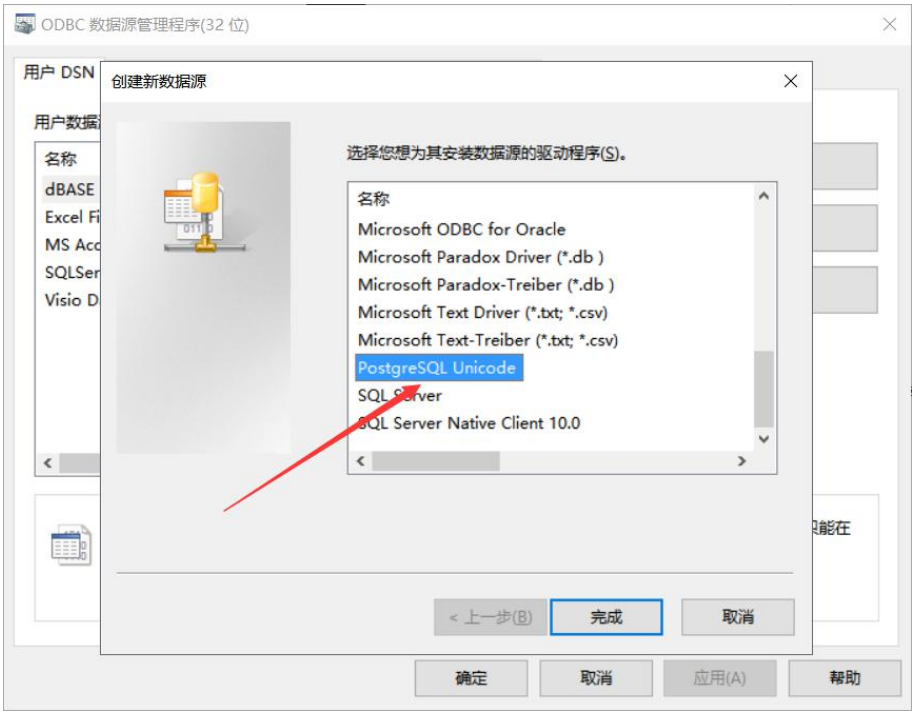
5.1.1 环境配置

双击实验指导书附带的 psqlodbc.msi 文件，点击安装，安装选项默认就好。（也可以从网址 https://dbs-download.obs.cn-north-1.myhuaweicloud.com/GaussDB/1620888486639/GaussDB_opengauss_client_tools.zip 下载，并解压其中的 Odbc 压缩包）。

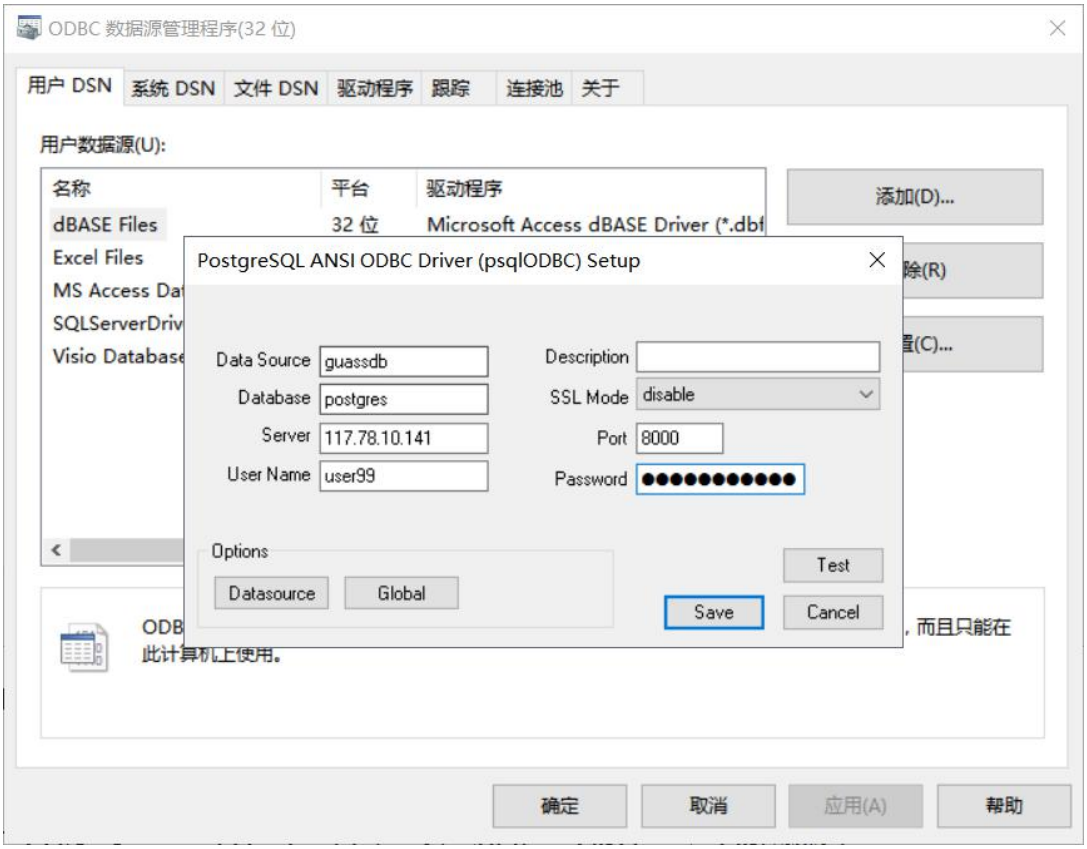
在 win10 中搜索 odbc，点击打开“ODBC 数据源(32 位)”，64 位的会报错！

(或者直接找到 C:\Windows\SysWOW64\odbcad32.exe 并双击)

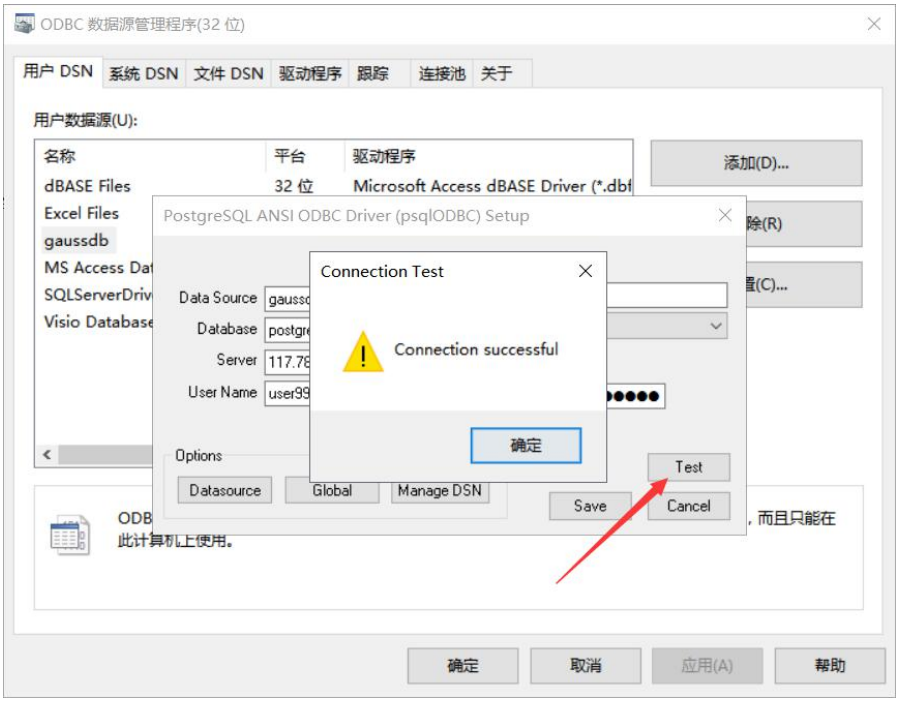
打开 ODBC 数据源后点击“添加”，选择驱动程序 PostgreSQL Unicode



添加后，把 Data Source 的值改为“gaussdb”（名字不限，但注意要与后面程序中的名字对应），Database 的值为 postgres，Server 的值为 117.78.10.141, Port 为 8000。User Name 和 Password 填写为自己的数据库用户名和密码。



填写完成后，点击 Test 进行测试，出现如下图所示“Connection successful”的弹窗说明输入数据库信息无误,然后保存确定后，即可继续进行实验。



5.1.2 编写 C/C++程序访问数据库

开发流程

编译器：Dev-C++（也可尝试用其他软件），下载网址：https://sourceforge.net/projects/orwelldevcpp/

安装完成后，点击上方的“工具”——“编译选项”，将“在连接器命令行加入以下命令”的值改为“-static-libgcc -lodbc32”

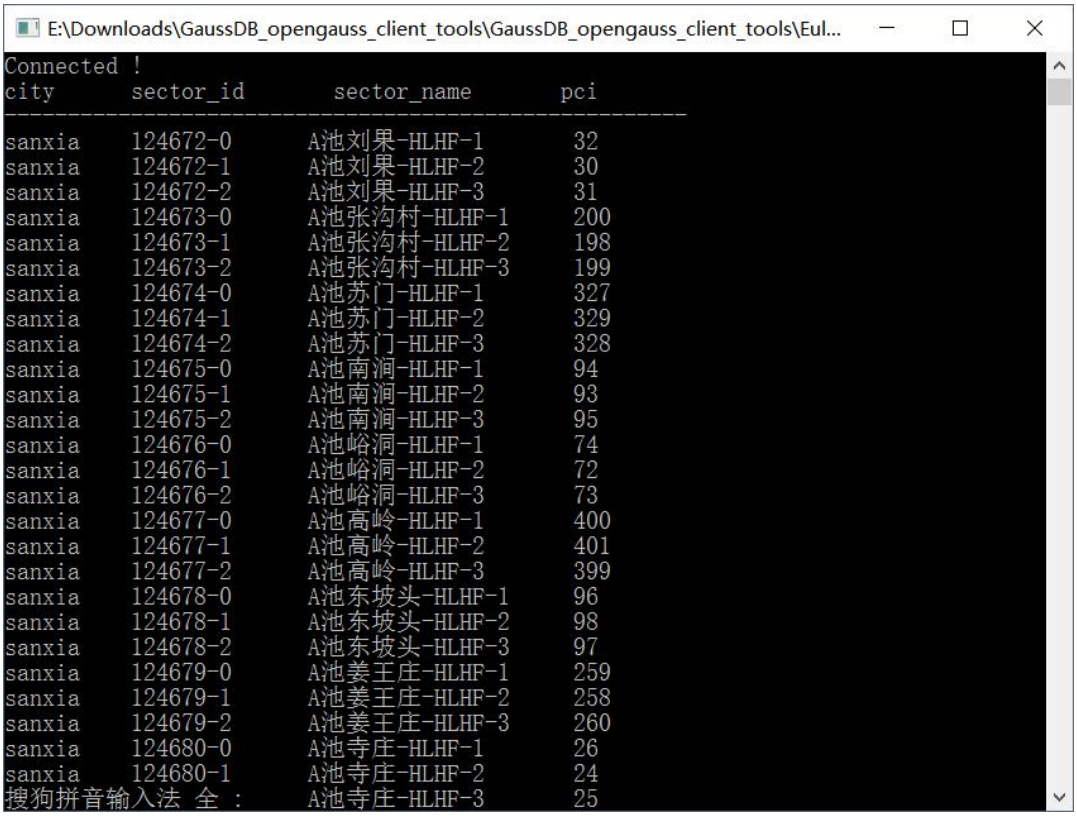
示例代码：

```
#include <windows.h>
#include <stdlib.h>
#include <stdio.h>
#include <sqlext.h>
SQLHENV          V_OD_Env;          // Handle ODBC environment
SQLHSTMT         V_OD_hstmt;        // Handle statement
SQLHDBC          V_OD_hdbc;         // Handle connection
char             typename[100];
SQLINTEGER       value = 100;
SQLINTEGER       V_OD_erg,V_OD_buffer,V_OD_err,pci;
SQLCHAR          sector_id[100],sector_name[100],city[100];
int main(int argc,char *argv[])
{
    // 1. 申请环境句柄
    V_OD_erg = SQLAllocHandle(SQL_HANDLE_ENV,SQL_NULL_HANDLE,&V_OD_Env);
    if ((V_OD_erg != SQL_SUCCESS) && (V_OD_erg != SQL_SUCCESS_WITH_INFO))
    {
```

```
        printf("Error AllocHandle\n");
        exit(0);
    }
    // 2. 设置环境属性（版本信息）
    SQLSetEnvAttr(V_OD_Env, SQL_ATTR_ODBC_VERSION, (void*)SQL_OV_ODBC3, 0);
    // 3. 申请连接句柄
    V_OD_erg = SQLAllocHandle(SQL_HANDLE_DBC, V_OD_Env, &V_OD_hdbc);
    if ((V_OD_erg != SQL_SUCCESS) && (V_OD_erg != SQL_SUCCESS_WITH_INFO))
    {
        SQLFreeHandle(SQL_HANDLE_ENV, V_OD_Env);
        exit(0);
    }
    // 4. 设置连接属性
    SQLSetConnectAttr(V_OD_hdbc, SQL_ATTR_AUTOCOMMIT, SQL_AUTOCOMMIT_ON, 0);
   //5. 连接数据源，这里的 “gaussdb” 需与之前配置数据源时的 Data Source 值对应
    V_OD_erg = SQLConnect(V_OD_hdbc, (SQLCHAR*) "gaussdb", SQL_NTS,
                          (SQLCHAR*) "", SQL_NTS,  (SQLCHAR*) "", SQL_NTS);
    if ((V_OD_erg != SQL_SUCCESS) && (V_OD_erg != SQL_SUCCESS_WITH_INFO))
    {
        printf("Error SQLConnect %d\n",V_OD_erg);
        SQLFreeHandle(SQL_HANDLE_ENV, V_OD_Env);
        exit(0);
    }
    printf("Connected !\n");
    // 6. 设置语句属性
    SQLSetStmtAttr(V_OD_hstmt,SQL_ATTR_QUERY_TIMEOUT,(SQLPOINTER *)3,0);
    // 7. 申请语句句柄
    SQLAllocHandle(SQL_HANDLE_STMT, V_OD_hdbc, &V_OD_hstmt);
    // 8. 直接执行 SQL 语句
    char *sqlquery = "SELECT city, sector_id, sector_name,pci FROM \"tbCell\"";
   SQLExecDirect(V_OD_hstmt,sqlquery,SQL_NTS);
    V_OD_erg=SQLFetch(V_OD_hstmt);
    // 9. 通过 SQLGetData 获取并返回数据。
   int i=0;
   printf("city        sector_id        sector_name        pci      \n" );
   printf("-------------------------------------------------\n");
    while(V_OD_erg != SQL_NO_DATA)
    {
        SQLGetData(V_OD_hstmt,1,SQL_C_CHAR,(SQLPOINTER)&city,100,NULL);
        SQLGetData(V_OD_hstmt,2,SQL_C_CHAR,(SQLPOINTER)&sector_id,100,NULL);
```

```
            SQLGetData(V_OD_hstmt,3,SQL_C_CHAR,(SQLPOINTER)&sector_name,100,NULL);
            SQLGetData(V_OD_hstmt,4,SQL_C_SLONG,(SQLPOINTER)&pci,100,NULL);
            printf("%-9s %-13s %-20s %-8d\n",city,sector_id,sector_name,pci    );
            V_OD_erg=SQLFetch(V_OD_hstmt);
            i++;
            if(i>30)
            {break; }
        };
        printf("Done !\n");
        // 10. 断开数据源连接并释放句柄资源
        SQLFreeHandle(SQL_HANDLE_STMT,V_OD_hstmt);
        SQLDisconnect(V_OD_hdbc);
        SQLFreeHandle(SQL_HANDLE_DBC,V_OD_hdbc);
        SQLFreeHandle(SQL_HANDLE_ENV, V_OD_Env);
        return(0);
}
```

运行结果



对数据库进行其他操作

（1）添加一条'beijing','000001', 'haidian-HLHF-1',000001,'haidian-HLHF'的数据到数据库。

```
char *sqlquery =”insert into \”tbCell\”city, sector_id, sector_name, enodebid, ENODEB_NAME)
values('beijing','000001', 'haidian-HLHF-1',000001,'haidian-HLHF')”;
```

（2）查询，查 sanxia 城市中，经度（longitude）大于 111.5 的所有基站 id 和名字，经度（enodebid，enodeb_name,longitude),且按照经度降序排列注意相同的只显示一次。

```
char *sqlquery = “select ENODEBID,enodeb_name,LONGITUDE from    \”tbCell\” where LONGITUDE>111.5
group by ENODEBID order by LONGITUDE desc”;
```

（3）更新，将（1）中插入的数据，中的 earfcn，pci，pss，sss，tac 更新为 12345,32,1,10,10000,并打印该行信息

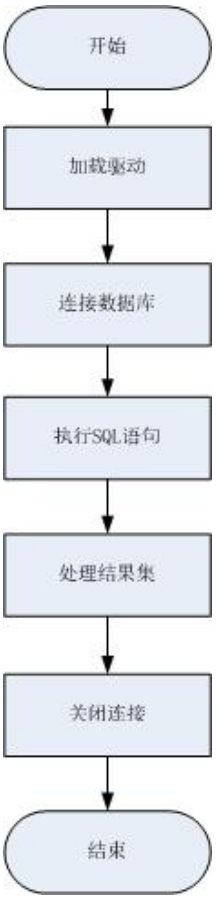
```
char *sqlquery=”update \”tbCell\” set EARFCN=12345,pci=32,pss=1,sss=10,tac=10000 where enodebid=1”;
```

（4）删除，删除（1）插入的信息，并打印整张表。

```
char *sqlquery=”delete from \”tbCell\” where CITY='beijing'”;
```

5.2 JDBC 接口访问。

5.2.3 开发流程



5.2.4 本指导书提供在 eclipse（或自行尝试其他 IDE）进行实验的步骤

首先创建新的 java 项目。右击项目名，点击“构建路径”——“配置构建路径”，然后点击“添加外部 JAR”，找到实验指导书附带的 gsjdbc4.jar 文件，添加进去即可（或者从 https://dbs-download.obs.cn-north-1.myhuaweicloud.com/GaussDB/1620888486639/GaussDB_opengauss_client_tools.zip 下载，并解压其中的 JDBC 压缩包）。

5.2.5 编写 java 程序访问数据库

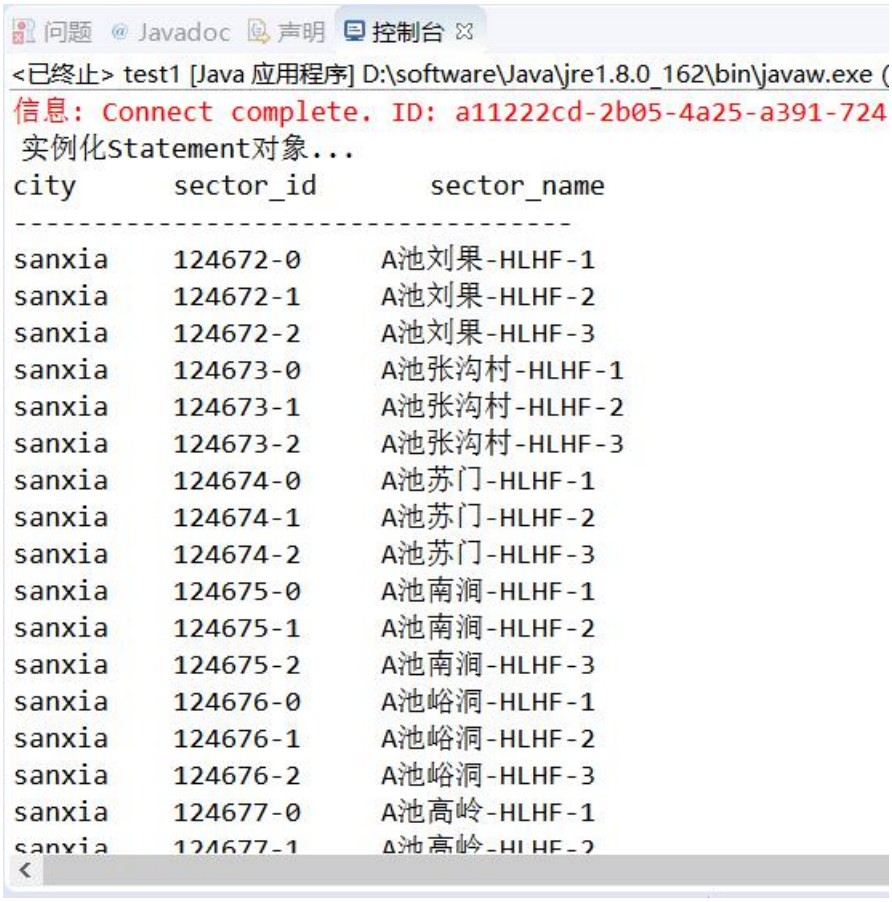
全部代码为：

```
import java.sql.*;
public class test1 {
    //  JDBC 驱动名及数据库 URL
    static final String JDBC_DRIVER = "org.postgresql.Driver";
    static final String DB_URL ="jdbc:postgresql://117.78.10.141:8000/postgres";
    // 数据库的用户名与密码，需要根据自己的设置
    static final String USER = "user99";
    static final String PASS = "user99@bupt";

    public static void main(String[] args) {
        Connection conn = null;
        Statement stmt = null;
        try{
            // 注册 JDBC 驱动
            Class.forName(JDBC_DRIVER);
            // 打开链接
            System.out.println("连接数据库...");
```

```
            conn = DriverManager.getConnection(DB_URL,USER,PASS);
            // 执行查询
            System.out.println(" 实例化 Statement 对象...");
            stmt = conn.createStatement();
            String sql;
            sql = "select * from \"tbCell\";";
            ResultSet rs = stmt.executeQuery(sql);
            // 展开结果集数据库
            System.out.print("city         sector_id            sector_name\n" );
            System.out.print("---------------------------------\n");
            int i=0;
            while(rs.next()){
                // 只输出前 30 条
             if(i>30){
                    break;
             }
                String city    = rs.getString("city");
                String sector_id = rs.getString("sector_id");
                String sector_name = rs.getString("sector_name");

                // 输出数据
                System.out.printf("%-10s%-13s%-35s\n",city,sector_id,sector_name);
                i++;
            }
            // 完成后关闭
            rs.close();
            stmt.close();
            conn.close();
        }catch(SQLException se){
            // 处理 JDBC 错误
            se.printStackTrace();
        }catch(Exception e){
            // 处理 Class.forName 错误
            e.printStackTrace();
        }finally{
            // 关闭资源
            try{
                if(stmt!=null) stmt.close();
            }catch(SQLException se2){
```

```
            }// 什么都不做
            try{
                if(conn!=null) conn.close();
            }catch(SQLException se){
                se.printStackTrace();
            }
        }
        System.out.println("Goodbye!");
    }
}
```

运行结果：



5.2.6 对数据库进行操作其他操作

（1）添加一条'beijing','000001', 'haidian-HLHF-1',000001,'haidian-HLHF'的数据到数据库。

```
sql="insert into \"tbCell\" (city, sector_id, sector_name, enodebid, ENODEB_NAME) values('beijing','000001',
'haidian-HLHF-1',000001,'haidian-HLHF');";
```

	CITY	SECTOR_ID	SECTOR_NAME	ENODEBID	ENODEB_NAME	EARFCN	PCI	PSS
1	beijing	000001	haidian-HLHF-1	1	haidian-HLHF	<null>	<null>	<null>

（2）查询，查 sanxia 城市中，经度（longitude）大于 111.5 的所有基站 id 和名字，经度（enodebid，enodeb_name,longitude),且按照经度降序排列注意相同的只显示一次。

	ENODEBID	enodeb_name	LONGITUDE
1	246333	G安水泥厂F-HLH	113.034
2	246506	G安芦院学校D-HLH	113.029
3	254642	G安铁门营业厅D-HLH	113.025
4	11429	G安玉梅F-HLH	113.024
5	236141	G安刘扬F-HLH	113.019
6	246341	G安千唐志斋F-HLH	113.018
7	246335	G安铁门镇F-HLH	113.018
8	254572	G安铁门村F-HLH	113.01
9	246703	F阳盐镇北F-HLH	113.006

```
sql="select ENODEBID,enodeb_name,LONGITUDE from \"tbCell\"`where LONGITUDE>111.5 group by ENODEBID order by LONGITUDE desc;";
```

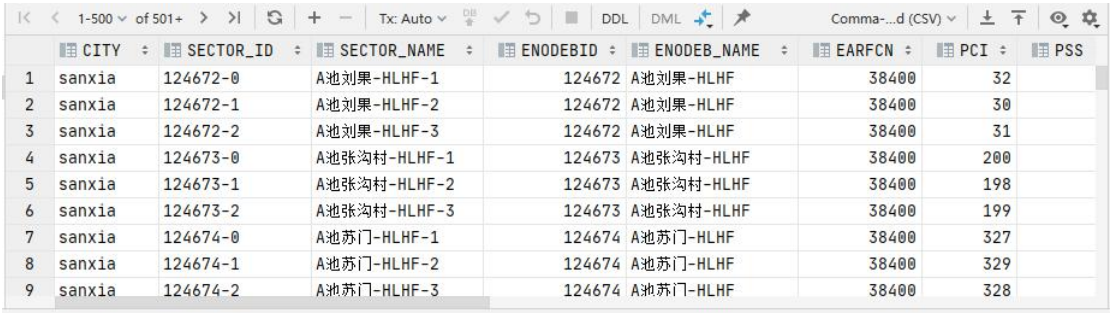
（3）更新，将（1）中插入的数据，中的 earfcn，pci，pss，sss，tac 更新为 12345,32,1,10,10000,并打印该行信息

```
sql="update \"tbCell\"` set EARFCN=12345,pci=32,pss=1,sss=10,tac=10000 where enodebid=1"
```

	CITY	SECTOR_ID	SECTOR_NAME	ENODEBID	ENODEB_NAME	EARFCN	PCI	PSS	SSS	TAC
1	beijing	000001	haidian-HLHF-1	1	haidian-HLHF	12345	32	1	10	10000

（4） 删除，删除（1）插入的信息，并打印整张表。

```
sql="delete from \"tbCell\" where CITY='beijing';"
```



	CITY	SECTOR_ID	SECTOR_NAME	ENODEBID	ENODEB_NAME	EARFCN	PCI	PSS
1	sanxia	124672-0	A池刘果-HLHF-1	124672	A池刘果-HLHF	38400	32	
2	sanxia	124672-1	A池刘果-HLHF-2	124672	A池刘果-HLHF	38400	30	
3	sanxia	124672-2	A池刘果-HLHF-3	124672	A池刘果-HLHF	38400	31	
4	sanxia	124673-0	A池张沟村-HLHF-1	124673	A池张沟村-HLHF	38400	200	
5	sanxia	124673-1	A池张沟村-HLHF-2	124673	A池张沟村-HLHF	38400	198	
6	sanxia	124673-2	A池张沟村-HLHF-3	124673	A池张沟村-HLHF	38400	199	
7	sanxia	124674-0	A池苏门-HLHF-1	124674	A池苏门-HLHF	38400	327	
8	sanxia	124674-1	A池苏门-HLHF-2	124674	A池苏门-HLHF	38400	329	
9	sanxia	124674-2	A池苏门-HLHF-3	124674	A池苏门-HLHF	38400	328	

6． 实验总结

在实验中有哪些重要问题或者事件？你如何处理的？你的收获是什么？有何建议和意见等等。