

# 《数据库系统原理》课程实验指导

## 数据库接口实验指导书



2023年9月

## 目录

1. 实验介绍.....	3
1.1 实验环境说明.....	3
1.2 实验目的.....	3
1.3 实验原理.....	3
1.4 实验内容.....	4
2. 实验要求.....	4
2.1 实验内容要求.....	4
3. 实验步骤.....	5
3.1 实验准备.....	5
3.2 环境配置.....	5
3.3 数据库连接及访问.....	8
4. JDBC 访问.....	8
4.1 连接准备.....	8
4.2 执行 java 程序访问数据库.....	9
4.2.1 示例代码.....	9
4.2.1 运行结果.....	11
4.3 对数据库进行其他操作.....	11
5. ODBC 访问.....	12
5.1 连接准备.....	12
5.1.1 驱动程序安装.....	12

5.1.2 配置 ODBC 数据源 .....	12
5.1.3 编译器设置 .....	15
5.2 执行 C 程序访问数据库 .....	15
5.2.1 示例代码 .....	15
5.2.2 运行结果 .....	17
5.3 对数据库进行其他操作 .....	18
6. 实验总结 .....	18

## 1. 实验介绍

### 1.1 实验环境说明

本实验环境为 GaussDB(for openGauss)云数据库或 openGauss 本地数据库。

当以 GaussDB(for openGauss)云数据库为实验平台时，IP 地址为 116.205.151.39，端口号为 8000，数据库名为 postgres。当以 openGauss 本地数据库为实验平台时，以上信息根据自己设置的来连接数据库，并且要修改 openGauss 的配置，设置好监听信息，才能让 openGauss 本地数据库接受远程访问。

### 1.2 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。

### 1.3 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。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。

## 1.4 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、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 TPC-H 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。

## 2. 实验要求

### 2.1 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。ODBC 接口实验流程复杂，且容易报错，推荐选择 JDBC 接口实验。
2. 实验时选取 TPC-H 数据库作为数据源，可参照后文中以 nation 表为数据源的访问样

例。

3. 实验中需要完成如下数据库访问操作。

- 1) 查询。选取一张表执行查询操作，并打印出数据。
- 2) 插入。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。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。选取一张表执行删除操作，并检查操作是否成功。

## 3. 实验步骤

### 3.1 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](https://support.huaweicloud.com/devg-opengauss/opengauss_devg_0065.html)

### 3.2 环境配置

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

使用 openGauss 本地数据库需要配置好监听信息：

步骤1：

更改连接设置，将监听地址改为“\*”（即全部）或者将连接的IP地址加入到服务器的监听列表中

方法一：通过gs命令语句进行修改

```
gs_guc reload -N nodeName -l all -c "listen_addresses='*'"
```

方法二：编辑修改postgresql文件

```
vi /gaussdb/data/db1/postgresql.conf
```

```

# -----
# PostgreSQL configuration file
# -----
#
# This file consists of lines of the form:
#
#   name = value
#
# (The "=" is optional.)  Whitespace may be used.  Comments are introduced with
# "#" anywhere on a line.  The complete list of parameter names and allowed
# values can be found in the PostgreSQL documentation.
#
# The commented-out settings shown in this file represent the default values.
# Re-commenting a setting is NOT sufficient to revert it to the default value;
# you need to reload the server.
#
# This file is read on server startup and when the server receives a SIGHUP
# signal.  If you edit the file on a running system, you have to SIGHUP the
# server for the changes to take effect, or use "pg_ctl reload".  Some
# parameters, which are marked below, require a server shutdown and restart to
# take effect.
#
# Any parameter can also be given as a command-line option to the server, e.g.,
# "postgres -c log_connections=on".  Some parameters can be changed at run time
# with the "SET" SQL command.
#
# Memory units:  kB = kilobytes           Time units:  ms = milliseconds
#                  MB = megabytes           s = seconds
#                  GB = gigabytes           min = minutes
#                                           h = hours
#                                           d = days
#
#-----
# FILE LOCATIONS
#-----
#
# The default values of these variables are driven from the -D command-line
# option or PGDATA environment variable, represented here as ConfigDir.
#
#data_directory = 'ConfigDir'           # use data in another directory
#                                          # (change requires restart)
#hba_file = 'ConfigDir/pg_hba.conf'     # host-based authentication file
#                                          # (change requires restart)
#ident_file = 'ConfigDir/pg_ident.conf' # ident configuration file
#                                          # (change requires restart)
#
# If external_pid_file is not explicitly set, no extra PID file is written.
#external_pid_file = ''                 # write an extra PID file
#                                          # (change requires restart)
#
#-----
# CONNECTIONS AND AUTHENTICATION
#-----
# - Connection Settings -
listen_addresses = '10.0.3.15'          # what IP address(es) to listen on;
#                                          # comma-separated list of addresses;
#                                          # defaults to 'localhost'; use '*' for all
#                                          # (change requires restart)

```

将listen addresses = '10.0.3.15' 改为listen addresses = '\*'

```
listen_addresses = '*'
```

注：连接的默认设置是只监听本地地址，这样只能接收本地的连接，改为监听所有之后，便能接收外部的连接。如果不改，在通过JDBC连接数据库时连接失败，并报错。

Connection to 192.168.56.102:26000 refused. Check that the hostname and port are correct and that the postmaster is accepting TCP/IP connections.

## 步骤2:

在数据库主节点配置文件中加入与IP地址对应的认证规则，从而放行客户端

方法一：通过gs命令语句进行修改

```
gs_guc reload -N all -I all -h "host all all 192.168.56.1/32 sha256"
```

方法二：编辑修改pg\_hba.conf文件

```
vi /gaussdb/data/db1/pg_hba.conf
```

```
# TYPE DATABASE USER ADDRESS METHOD
# "local" is for Unix domain socket connections only
local all all trust
# IPv4 local connections:
host all all 127.0.0.1/32 trust
host all all 10.0.3.15/32 trust
# IPv6 local connections:
host all all ::1/128 trust
# Allow replication connections from localhost, by a user with the
# replication privilege.
#local replication omm trust
#host replication omm 127.0.0.1/32 trust
#host replication omm ::1/128 trust
```

在127.0.0.1/32的下面加入host all all 192.168.56.1/32 sha256

```
# TYPE DATABASE USER ADDRESS METHOD
# "local" is for Unix domain socket connections only
local all all trust
# IPv4 local connections:
host all all 127.0.0.1/32 trust
host all all 10.0.3.15/32 trust
host all all 192.168.56.1/32 sha256
# IPv6 local connections:
host all all ::1/128 trust
# Allow replication connections from localhost, by a user with the
# replication privilege.
#local replication omm trust
#host replication omm 127.0.0.1/32 trust
#host replication omm ::1/128 trust
-- INSERT --
```

注：如果不加入认证规则，则连接亦会失败，并报错

**FATAL: no pg\_hba.conf entry for host "192.168.56.1".**

如果是远端连接，则认证方式不能为trust，可以设置成sha256，如果设置为trust，则连接失败并报错

**FATAL: Forbid remote connection with trust method!**

以上的IP地址为远端连接的IP地址，请跟自己的实际IP地址填写

以上配置文件的路径请根据实际安装的位置查找。

### 3.3 数据库连接及访问

针对 TPC-H 实验数据库，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。







## 4. JDBC 访问

### 4.1 连接准备

(1) 准备好 Java 1.8 环境和 IDE (eclipse 或 IDEA)。通过以下链接，下载驱动包

[https://dbs-download.obs.cn-north-1.myhuaweicloud.com/rds/GaussDB\\_opengauss\\_client\\_tools.zip](https://dbs-download.obs.cn-north-1.myhuaweicloud.com/rds/GaussDB_opengauss_client_tools.zip)

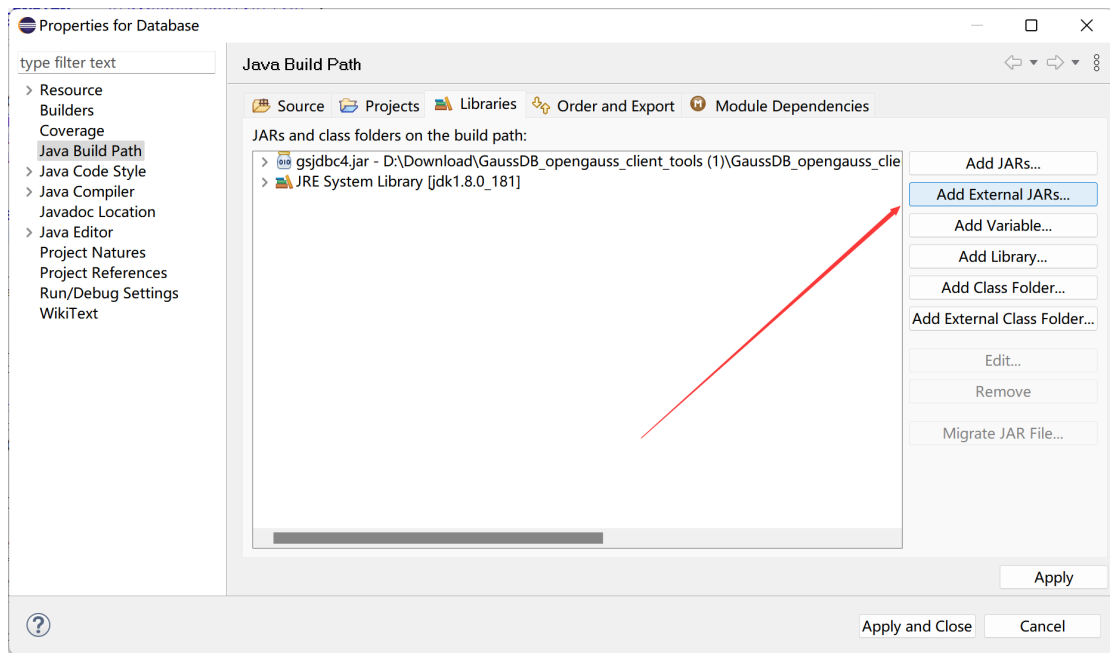
(2) 选择 X86 版驱动版本，然后选择 Jdbc 包进行解压。保存 gsjdbc4.jar 架包。

Data (D:) > download > GaussDB_opengauss_client_tools > Euler2.5_X86_64		
名称		修改日期
 GaussDB-Kernel-V500R001C20-EULER-64bit-Gds.tar		2021/5/13 11:47
 GaussDB-Kernel-V500R001C20-EULER-64bit-gsql.tar		2021/5/13 11:47
 GaussDB-Kernel-V500R001C20-EULER-64bit-Jdbc.tar		2021/5/13 11:47
 GaussDB-Kernel-V500R001C20-EULER-64bit-Libpq.tar		2021/5/13 11:47
 GaussDB-Kernel-V500R001C20-EULER-64bit-Odbc.tar		2021/5/13 11:47
 GaussDB-Kernel-V500R001C20-EULER-64bit-Python.tar		2021/5/13 11:47

(3) 在 eclipse 中右击工程名，点击 “build path” ----> “configure build path”，然后点击 “Add

External JARs”，把 gsjdbc4.jar 添加进去。如下图所示：





## 4.2 执行 java 程序访问数据库

### 4.2.1 示例代码

**注意：**红色部分需要替换。其中云数据库的 IP 为 116.205.151.39，端口号为 8000，数据库为 postgres。本地数据库的需要根据自己的设置。

```
import java.sql.*;

public class Database{

    static final String JDBC_DRIVER = "org.postgresql.Driver";
    static final String DB_URL = "jdbc:postgresql://IP:Port/Database?ApplicationName=app1";

    // 数据库的用户名与密码，需要根据自己的设置
    static final String USER = "XXX";
    static final String PASS = "XXXXXXXXXX";

    public static void main(String[] args) {

        System.out.println("java版本号: "+ System.getProperty("java.version"));

        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 n_nationkey ,n_name, n_regionkey, n_comment from nation";
ResultSet rs = stmt.executeQuery(sql);
System.out.print("nationkey   name   regionkey   comment\n" );

// 展开结果集数据库
while(rs.next()){
    // 通过字段检索
    int nationkey  = rs.getInt("n_nationkey");
    String name = rs.getString("n_name");
    int regionkey  = rs.getInt("n_regionkey");
    String comment = rs.getString("n_comment");

    System.out.printf("   %-5d   %-15s %-5d %s\n",nationkey ,name.trim(), regionkey,
comment);
}
// 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){
    }
    try{
        if(conn!=null) conn.close();
    }catch(SQLException se){
        se.printStackTrace();
    }
}
}
}
}

```

## 4.2.1 运行结果

```
Problems @ Javadoc Declaration Console ×
<terminated> Database (1) [Java Application] C:\Program Files\Java\jdk1.8.0_181\bin\javaw.exe (2022年10月31日 下午3:29:35)
java版本号: 1.8.0_181
连接数据库...
十月 31, 2022 3:29:40 下午 org.postgresql.core.v3.ConnectionFactoryImpl.openConnectionImpl
信息: [0fed2643-ba4f-46b2-9a97-7577bc93fb8e] Try to connect. IP: 116.205.151.39:8000
十月 31, 2022 3:29:40 下午 org.postgresql.core.v3.ConnectionFactoryImpl.openConnectionImpl
信息: [10.128.144.243:55527/116.205.151.39:8000] Connection is established. ID: 0fed2643-b
十月 31, 2022 3:29:40 下午 org.postgresql.core.v3.ConnectionFactoryImpl.openConnectionImpl
信息: Connect complete. ID: 0fed2643-ba4f-46b2-9a97-7577bc93fb8e
实例化Statement对象...
nationkey name regionkey comment
0 ALGERIA 0 posits use carefully pending accounts. special deposits ha
1 ARGENTINA 1 ly bold instructions haggle quickly across the blithely cl
2 BRAZIL 1 carefully regular dependencies are quickly. stealthily icro
3 CANADA 1 packages must are. quickly regular requests among the slyl
4 EGYPT 4 e furiously silent packages. furiously ironic accounts af
5 ETHIOPIA 0 . ironic foxes haggle slyly. slyly special accounts nod am
6 FRANCE 3 encies. carefully even accounts after the asymptotes are i
7 GERMANY 3 usual hockey players against the unusual, ironic packages n
8 INDIA 2 the even, regular accounts. slyly special pinto bea
9 INDONESIA 2 y across the final foxes. requests are fluffily near
10 IRAN 4 equests. packages are ironic, regular theodolites. careful
11 IRAQ 4 cording to the quickly regular platelets. carefully ironic
12 JAPAN 2 ites integrate across the requests. slyly pending depths n
13 JORDAN 4 nag furiously. carefully unusual pinto beans against the f
14 KENYA 0 foxes wake fluffily around the fluffily unusual grouches.
15 MOROCCO 0 ular accounts wake carefully carefully close frays. furiou
16 MOZAMBIQUE 0 blithely regular instructions haggle qui
17 PERU 1 the requests. regular foxes sleep furiously. final reques
18 CHINA 2 he blithely express theodolites. carefully final deposits
19 ROMANIA 3 express, even deposits. unusual, final ideas along
20 SAUDI ARABIA 4 ole against the slyly unusual platelets. regular accounts
21 VIETNAM 2 r the carefully special pinto b
22 RUSSIA 3 lly silent excuses. accounts are quickly
23 UNITED KINGDOM 3 riously after the even, express ideas. slyly final theodol
24 UNITED STATES 1 e bold requests. carefully unusual packages cajole blithel
```

## 4.3 对数据库进行其他操作

(1) 增加。添加一条 `n_nationkey` 为 25, `n_name` 为 South Korea, `n_regionkey` 为 3 的数据到数据库中。并根据 `n_nationkey` 查询是否插入成功。

(2) 更新, 将 (1) 中插入的数据中的 `n_regionkey` 更新为 2。并根据 `n_nationkey` 查询是否更新成功。

(3) 删除, 删除 (1) 插入的信息。根据 `n_nationkey` 查询是否删除成功。

注: 以上增删改操作可通过 `stmt.execute(sql)`方法执行。

## 5. ODBC 访问

### 5.1 连接准备

#### 5.1.1 驱动程序安装

(1) 通过以下链接，下载驱动包

[https://dbs-download.obs.cn-north-1.myhuaweicloud.com/rds/GaussDB\\_opengauss\\_client\\_tools.zip](https://dbs-download.obs.cn-north-1.myhuaweicloud.com/rds/GaussDB_opengauss_client_tools.zip)

(2) 选择 X86 版驱动版本，然后选择 windows-Odbc 包进行解压。双击其中的 psqlodbc.msi 文件。



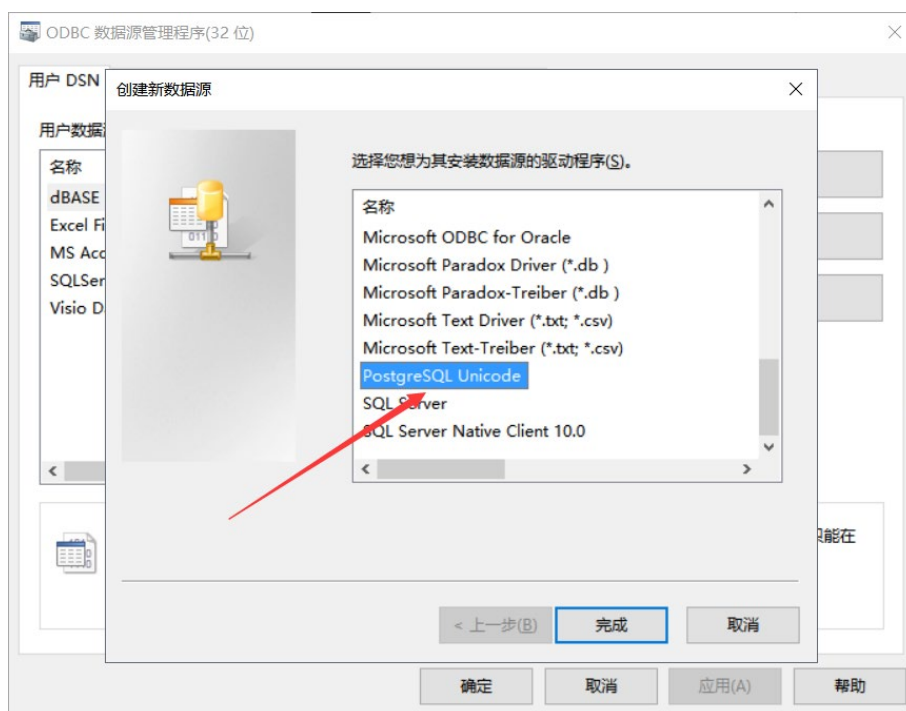
#### 5.1.2 配置 ODBC 数据源

(1) 在 windows 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)

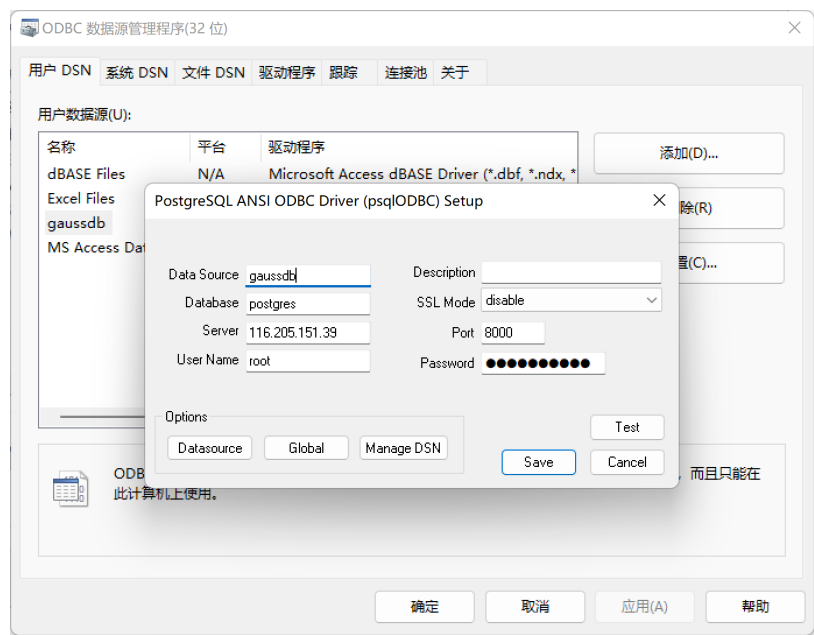


(2) 打开 ODBC 数据源后点击“添加”，选择驱动程序 PostgreSQL Unicode

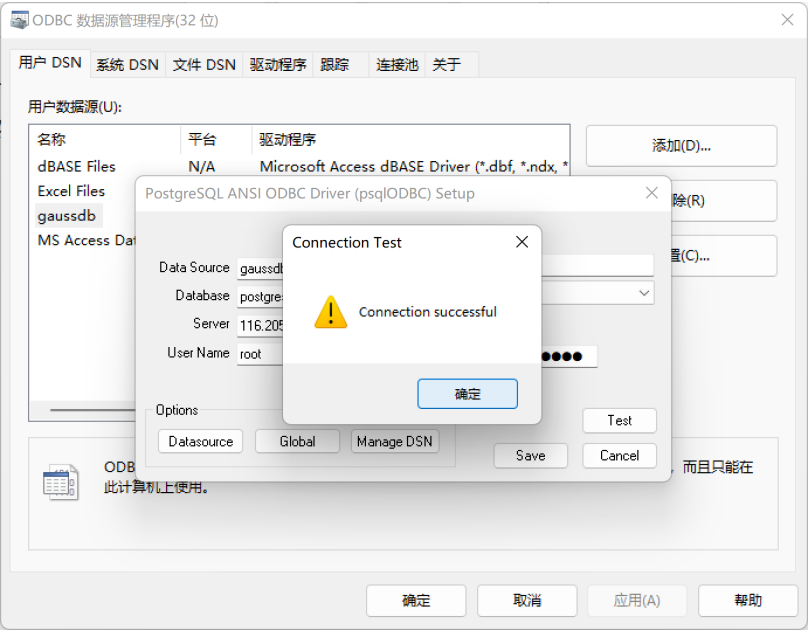


(3) 添加后，把 Data Source 的值改为“gaussdb”（名字不限，但注意要与后面程序中的名字对应），云数据库的 Server(即 IP)为 116.205.151.39, Port 为 8000, Database 为 postgres。

本地数据库的 IP、Port 和 Database 需要根据自己的设置。User Name 和 Password 填写为自己的数据库用户名和密码。



(4) 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。



### 5.1.3 编译器设置

下载安装Dev-C++, 下载地址: <https://sourceforge.net/projects/orwelldevcpp/>

安装完成后, 点击上方的“工具”——“编译选项”, 把“设定编译器配置”改为32-bit Release, 将“在连接器命令行加入以下命令”的值改为“-static-libgcc -lodb32”。



## 5.2 执行 C 程序访问数据库

### 5.2.1 示例代码

**注意：**需要将以下代码保存为C源文件，不是后缀为cpp的C++源文件。

```
#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
SQLINTEGER     V_OD_erg,V_OD_buffer,V_OD_err,nationkey,regionkey;
SQLCHAR        name[100],comment[100],name1[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, (void*)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 n_nationkey ,n_name, n_regionkey, n_comment from nation";
    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。

printf("nationkey  name  regionkey  comment\n");
//printf("-----\n");

while(V_OD_erg != SQL_NO_DATA)
{
    SQLGetData(V_OD_hstmt,1,SQL_C_SLONG,(SQLPOINTER)&nationkey,100,NULL);
    SQLGetData(V_OD_hstmt,2,SQL_C_CHAR,(SQLPOINTER)&name,100,NULL);
    SQLGetData(V_OD_hstmt,3,SQL_C_SLONG,(SQLPOINTER)&regionkey,100,NULL);
    SQLGetData(V_OD_hstmt,4,SQL_C_CHAR,(SQLPOINTER)&comment,100,NULL);

    strncpy(name1, name, 13);

    printf(" %5d  %-15s %-5d %s\n",nationkey,name1,regionkey,comment);
    V_OD_erg=SQLFetch(V_OD_hstmt);
}

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);
}
```

## 5.2.2 运行结果

```
Connected !
nationkey  name  regionkey  comment
0  ALGERIA  0  posits use carefully pending accounts. special deposits haggle. ironic, silent accounts are furio
1  ARGENTINA  1  ly bold instructions haggle quickly across the blithely close dep
2  BRAZIL  1  carefully regular dependencies are quickly. stealthily ironic platelets sleep
3  CANADA  1  packages must are. quickly regular requests among the slyly pending theodolites cajole quickly foxe
4  EGYPT  4  e furiously silent packages. furiously ironic accounts af
5  ETHIOPIA  0  . ironic foxes haggle slyly. slyly special accounts nod among the furiously express de
6  FRANCE  3  encies. carefully even accounts after the asymptotes are ironic, ironic accoun
7  GERMANY  3  sual hockey players against the unusual, ironic packages nag carefully against the carefully ironic
8  INDIA  2  the even, regular accounts. slyly special pinto bea
9  INDONESIA  2  y across the final foxes. requests are fluffily near
10  IRAN  4  equests. packages are ironic, regular theodolites. carefully regular ideas sleep slyly final, ex
11  IRAQ  4  cording to the quickly regular platelets. carefully ironic pinto beans against the slyly unusual th
12  JAPAN  2  ites integrate across the requests. slyly pending depths n
13  JORDAN  4  nag furiously. carefully unusual pinto beans against the fluffily bold req
14  KENYA  0  foxes wake fluffily around the fluffily unusual grouches. carefully unusual theodolites are slyly
15  MOROCCO  0  ular accounts wake carefully carefully close frays. furiously express dugouts above the furiously p
16  MOZAMBIQUE  0  blithely regular instructions haggle qui
17  PERU  1  the requests. regular foxes sleep furiously. final requests integrate carefully about the slyly re
18  CHINA  2  he blithely express theodolites. carefully final deposits after the blithely stealthy instructions
19  ROMANIA  3  express, even deposits. unusual, final ideas along
20  SAUDI ARABIA  4  ole against the slyly unusual platelets. regular accounts after the blithel
21  VIETNAM  2  r the carefully special pinto b
22  RUSSIA  3  lly silent excuses. accounts are quickly
23  UNITED KINGDO  3  riously after the even, express ideas. slyly final theodolites are slyly deposits. blithely fina
24  UNITED STATES  1  e bold requests. carefully unusual packages cajole blithely regular Tiresias. ironic requests nag b

Done !
-----
```

## 5.3 对数据库进行其他操作

(1) 增加。添加一条 n\_nationkey 为 25, n\_name 为 South Korea, n\_regionkey 为 3 的数据到数据库中。并根据 n\_nationkey 查询是否插入成功。

(2) 更新。将 (1) 中插入的数据中的 n\_regionkey 更新为 2。并根据 n\_nationkey 查询是否更新成功。

(3) 删除。删除 (1) 插入的信息。根据 n\_nationkey 查询是否删除成功。

## 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。