

《数据库》

Windows 下应用程序连接 GaussDB 实验指导



HUAWEI

华为技术有限公司

目录

前 言	3
简介	3
实验环境说明	3
1 应用程序连接前准备	4
1.1 购买 GaussDB 实例.....	4
1.2 绑定弹性公网 IP 及开放 8000 端口.....	4
2 使用 python 连接数据库	8
2.1 下载并安装 Python	8
2.2 连接准备	10
2.3 连接和使用数据库	12
3 使用 JDBC 连接数据库	14
3.1 下载并安装 JDK.....	14
3.2 配置 JDK 环境变量	15
3.3 连接准备	18
3.3.1 准备实验数据	18
3.3.2 下载 GaussDB 的 Java 驱动包.....	21
3.4 连接 GaussDB 并执行 Java 代码.....	22
4 使用 ODBC 连接数据库	25
4.1 加载 GaussDB ODBC 驱动包.....	25
4.2 配置数据源	27
4.3 执行示例代码连接并操作数据库.....	30

前言

简介

本实验指导书介绍使用驱动实现 Python、Java、C/C++连接 GaussDB 的方法，主要目的是为了让读者学习 GaussDB 数据库的应用程序连接方法。

实验环境说明

- 组网说明

本实验环境为华为云 GaussDB 数据库。

- 设备介绍

本实验采用以下配置：

设备明细表

设备名称	设备型号	软件版本
数据库	GaussDB 4 核 16 GB	GaussDB服务

- 实验环境预置

购买 GaussDB 数据库实例。

绑定弹性公网 IP 及开放 8000 端口。

1

应用程序连接前准备

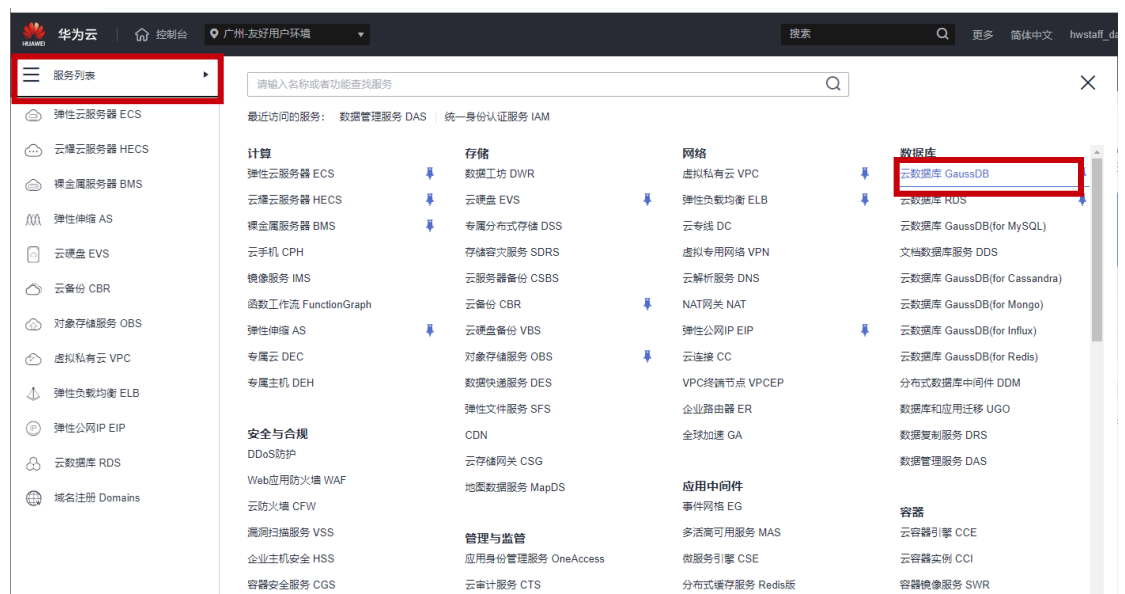
1.1 购买 GaussDB 实例

如果已经购买了 GaussDB 数据库可以跳过此节。如果还未购买，请参照《GaussDB 数据库实验环境搭建指导手册》进行购买。

1.2 绑定弹性公网 IP 及开放 8000 端口

如果 GaussDB 数据库实例已经绑定弹性公网 IP，且开放了 8000 端口，可以跳过此节。如果没有绑定，请参照如下步骤进行绑定。

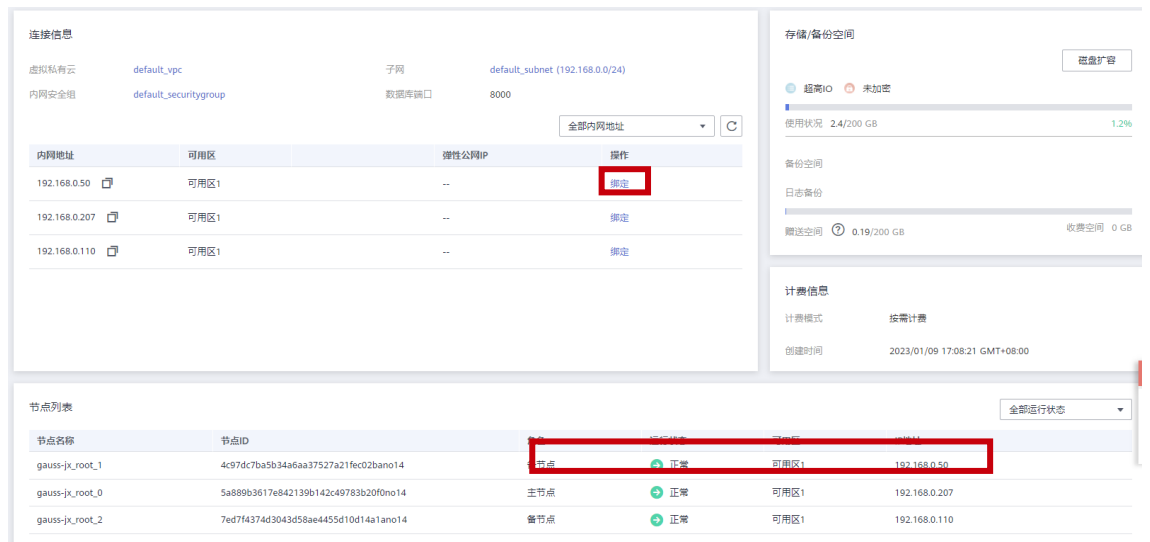
步骤 1 单击打开服务列表，然后选择“数据库”下的云数据库 GaussDB，具体如下：



在控制台选择 GaussDB 进入如下页面，单击实例名称，进入基本信息页面。



下拉到“节点列表”区域查询主节点，向上返回至“连接信息”区域，单击主节点处的“绑定”。



在弹出的“绑定弹性公网IP”的窗口。在此窗口上选择购买的弹性公网IP地址，然后单击“确定”。



绑定成功，请记录下此绑定的弹性公网 IP 地址，后面连接会用上此 IP。



弹性公网 IP 绑定成功。

步骤 2 找到内网安全组，单击进入安全组进行 8000 端口设置，具体如下：



选择入方向规则，然后添加规则，具体如下：

添加入方向规则 教我设置

安全组入方向规则为白名单（允许），放通入方向网络流量。

安全组 default_securitygroup

如您要添加多条规则，建议单击[导入规则](#)以进行批量导入。

优先级 ?	策略	协议端口 ?	类型	源地址 ?	描述	操作
1	允许	TCP 8000	IPv4	IP地址 0.0.0.0/0		复制 删除

增加1条规则

确定 取消

填写好端口 8000 后，单击“确定”。

添加规则 快速添加规则 删除 一键放通

入方向规则：6 教我设置

<input type="checkbox"/>	优先级...	策略 ?	协议端口 ?	类型	源地址 ?
<input type="checkbox"/>	1	允许	TCP : 8000	IPv4	0.0.0.0/0 ?
<input type="checkbox"/>	1	允许	全部	IPv6	default_securitygroup ?
<input type="checkbox"/>	1	允许	全部	IPv4	192.168.0.0/24
<input type="checkbox"/>	1	允许	TCP : 22	IPv4	0.0.0.0/0 ?
<input type="checkbox"/>	1	允许	TCP : 3389	IPv4	0.0.0.0/0 ?
<input type="checkbox"/>	1	允许	全部	IPv4	default_securitygroup ?

8000 端口放开成功。

2 使用 python 连接数据库

2.1 下载并安装 Python

此实验需要执行连接的计算机上安装 Python，如果计算机上已安装 Python，可以跳过此节。

步骤 1 下载 Python 解释器。

在如下链接中下载 Python 解释器：

<https://www.python.org/downloads/windows/>

根据自己电脑的实际选择安装包。

Stable Releases


- [Python 3.9.10 - Jan. 14, 2022](#)

Note that Python 3.9.10 cannot be used on Windows 7 or earlier.

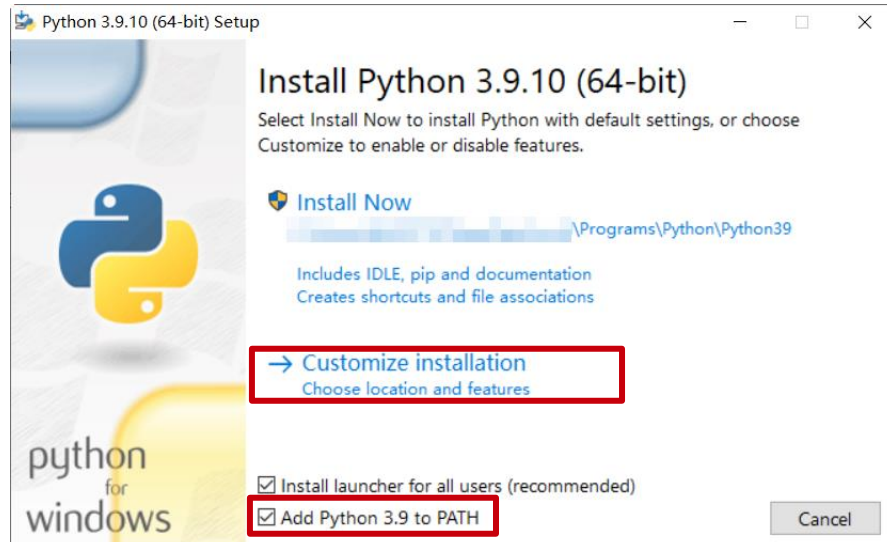
- Download [Windows embeddable package \(32-bit\)](#)
- Download [Windows embeddable package \(64-bit\)](#)
- Download [Windows help file](#)
- Download [Windows installer \(32-bit\)](#)
- Download [Windows installer \(64-bit\)](#)

步骤 2 安装 Python 解释器

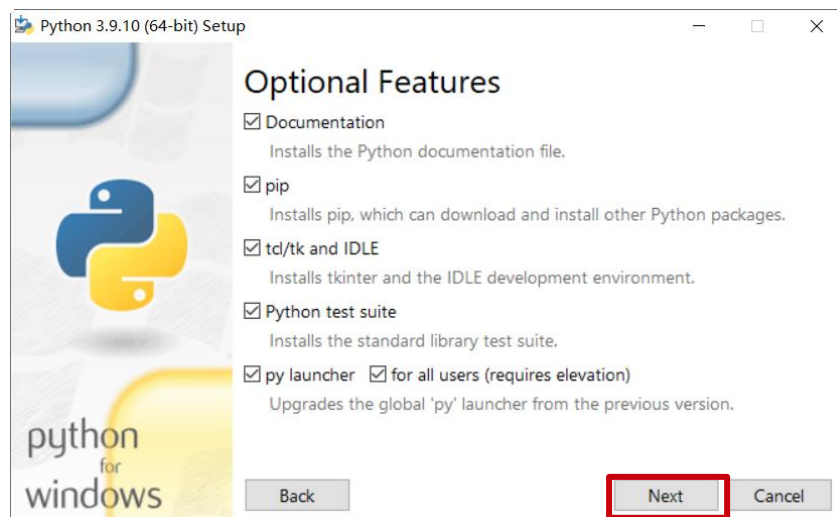
单击安装文件进行安装。

 **python-3.9.10-amd64.exe**

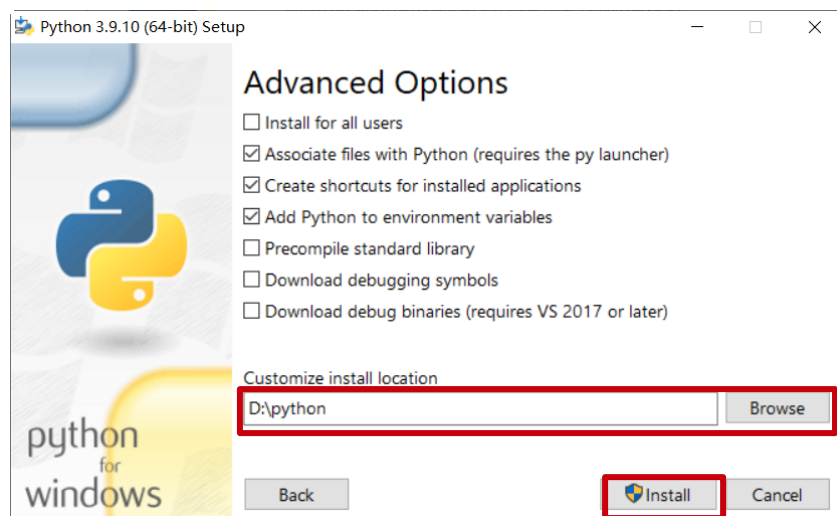
勾选 “Add Python 3.9 to PATH” ，单击 “Customize installation” 。



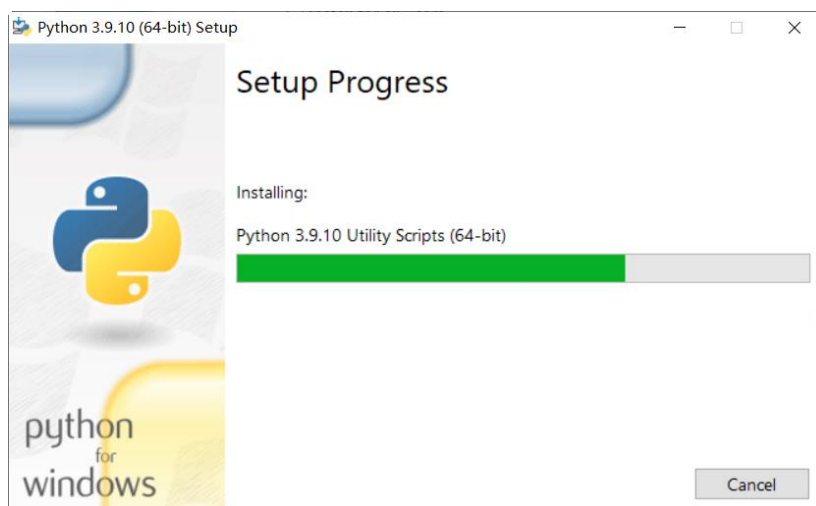
步骤 3 单击 “Next” 。



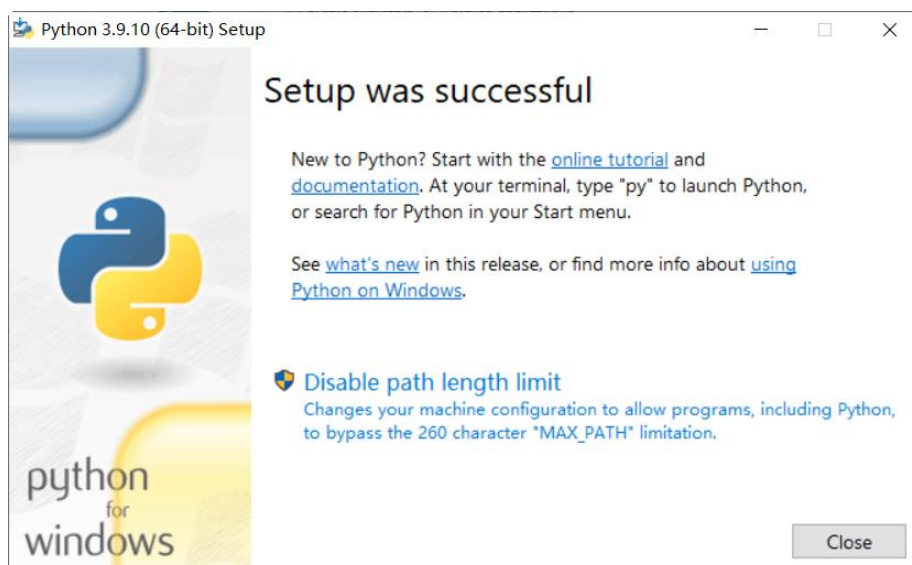
步骤 4 选择安装路径，单击 “Install” 。



步骤 5 等待安装完成。



步骤 6 安装成功后，单击 Close。



步骤 7 测试是否安装成功

打开 CMD 命令行，输入 Python，进入 Python 环境则安装成功。

```
Python 3.8.8 (default, Apr 13 2021, 15:08:03) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
```

2.2 连接准备

Psycopg2 是一种用于执行 SQL 语句的 PythonAPI，可以为 PostgreSQL、GaussDB 数据库提供统一访问接口，应用程序可基于它进行数据操作。

步骤 1 安装通用 psycopg2 包。

到有访问权限的 python 镜像服务器中下载并安装 psycopg2，此实验中，使用清华的镜像。

在确认对话框中单击“是”。



步骤 3 变更了通信协议加密方式后，创建新用户，让 Python 解释器使用该用户连接数据库，或者重置下已有的数据库帐号密码。

数据库中是不会记录任何形式的用户密码的，只有密码的 hash 值，在密码校验时使用的是加盐后的 hash 值比对（也是业界的通用做法）。所以在改变了密码的校验策略后，数据库中并没有存储原密码的 MD5 值，是仍然无法采用开源兼容的 MD5 校验的。

使用 DAS、gsqll 等数据库客户端连接数据库后，执行如下语句创建新用户：

```
CREATE USER python_user IDENTIFIED BY 'python_user@123';
```

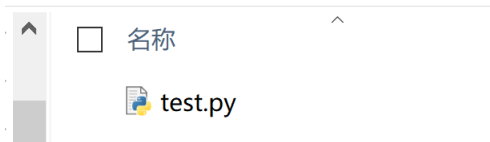
执行结果如下：

```
NOTICE: The encrypted password contains MD5 ciphertext, which is not secure.  
CREATE ROLE
```

2.3 连接和使用数据库

步骤 1 新建一个文本文档，将文件命名为 test.py，并将其保存到你喜欢的路径下，比如 D:\gaussdb。

此电脑 > Data (D:) > gaussdb



在 test.py 文件中加入如下内容。对于斜体部分的内容说明如下：

- user 及 password 的值分别为前面创建的用户及其密码；
- host 请输入所要连接的数据库实例公网 IP；
- port 为实例的端口号，在实例创建时通常默认使用 8000，如果做了变更，请使用对应的端口。

```
import pycpg2  
#创建连接对象
```

```
conn=psycopg2.connect(database="postgres",user=" python_user",password="
python_user@123",host="数据库实例的公网 IP",port=8000)

cur=conn.cursor() #创建指针对象

# 创建表
cur.execute("CREATE TABLE student(id integer,name varchar,sex varchar);")

#插入数据
cur.execute("INSERT INTO student(id,name,sex) VALUES(%s,%s,%s)",(1,'Aspirin','M'))
cur.execute("INSERT INTO student(id,name,sex) VALUES(%s,%s,%s)",(2,'Taxol','F'))
cur.execute("INSERT INTO student(id,name,sex) VALUES(%s,%s,%s)",(3,'Dixheral','M'))

# 获取结果
cur.execute('SELECT * FROM student')
results=cur.fetchall()
print (results)

conn.commit()
conn.close()
```

步骤 2 进入 test.py 所在目录。

```
cd /d D:\gaussdb
```

步骤 3 运行 test.py 脚本

```
python test.py
```

返回如下结果表示执行成功。

```
[(1, 'Aspirin', 'M'), (2, 'Taxol', 'F'), (3, 'Dixheral', 'M')]
```

3 使用 JDBC 连接数据库

3.1 下载并安装 JDK

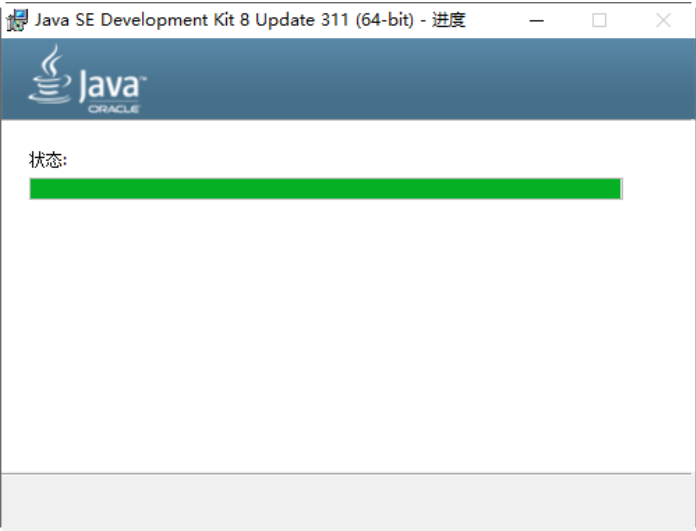
步骤 1 下载 JDK，推荐安装 java8。根据自己计算机的实际选择安装包

<https://www.oracle.com/java/technologies/downloads/#java8>

Linux macOS Solaris Windows		
Product/file description	File size	Download
x86 installer	157.37 MB	jdk-8u311-windows-i586.exe
x64 installer	170.57 MB	jdk-8u311-windows-x64.exe

步骤 2 双击 jdk-8u311-windows-x64.exe 进行安装。

默认设置即可，出现安装进度。

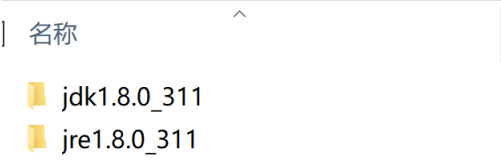


如下显示表示安装成功：



步骤 3 查看安装目录。

‣ SystemDisk (C:) ‣ Program Files ‣ Java



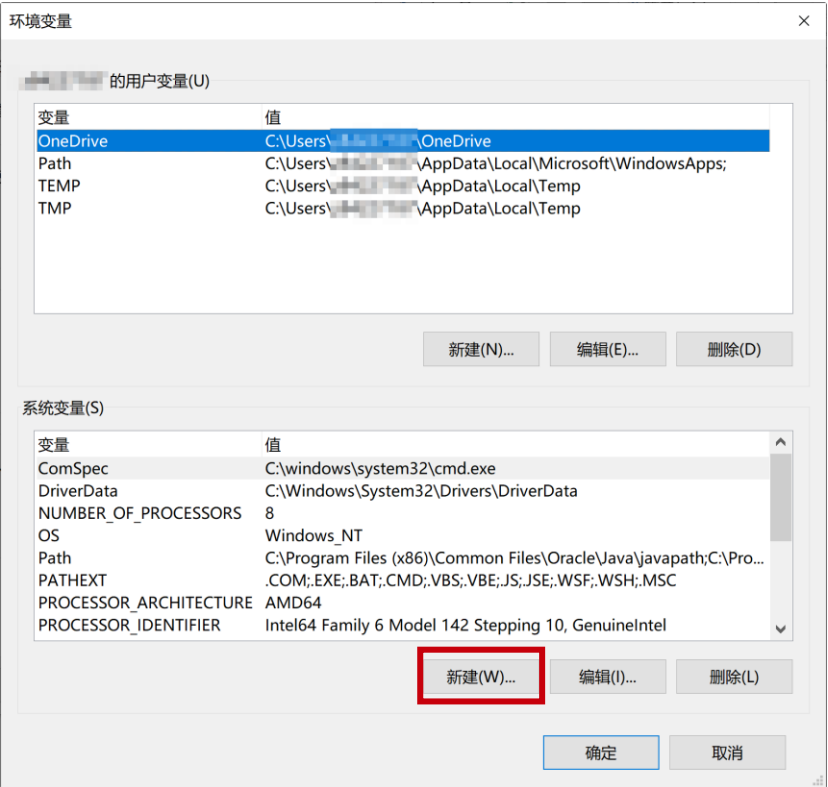
3.2 配置 JDK 环境变量

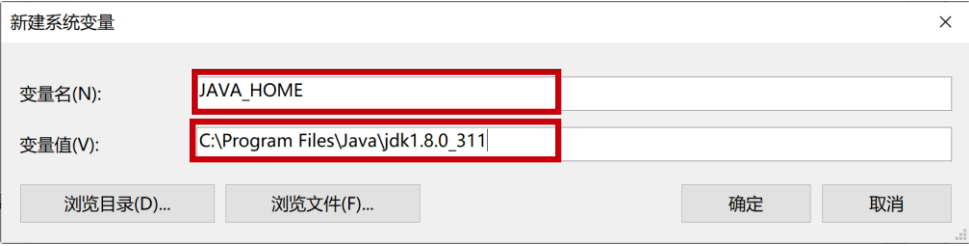
步骤 1 右击“此电脑”选择“属性”，单击“高级系统设置”。





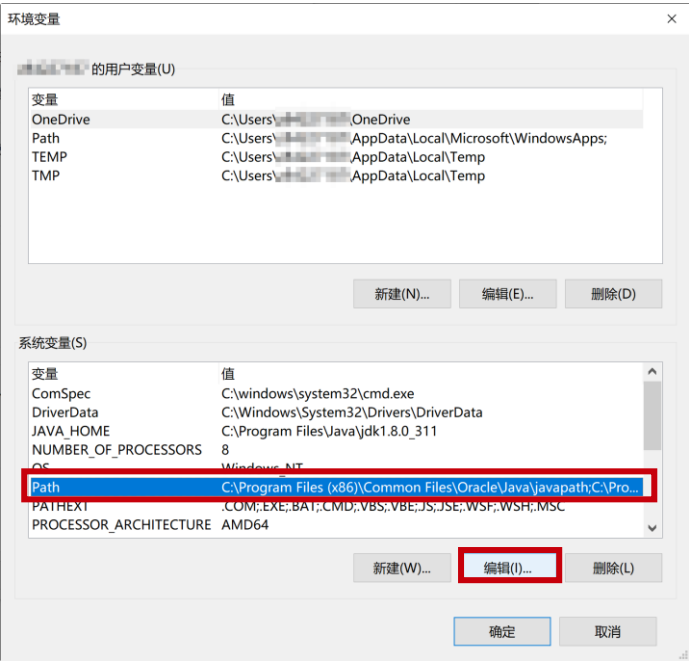
步骤 2 单击“环境变量”，新建系统变量“JAVA_HOME”，输入 JDK 安装目录。



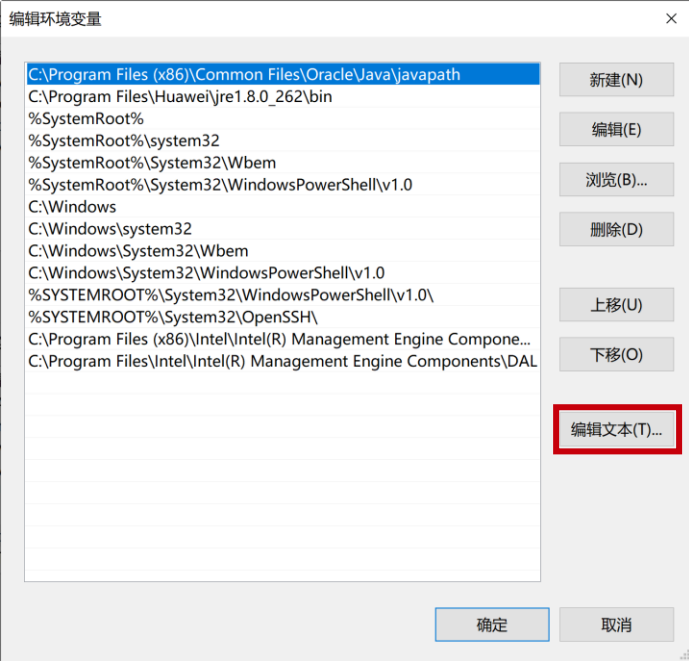


“C:\Program Files\Java\jdk1.8.0_311”为 JDK 安装目录。

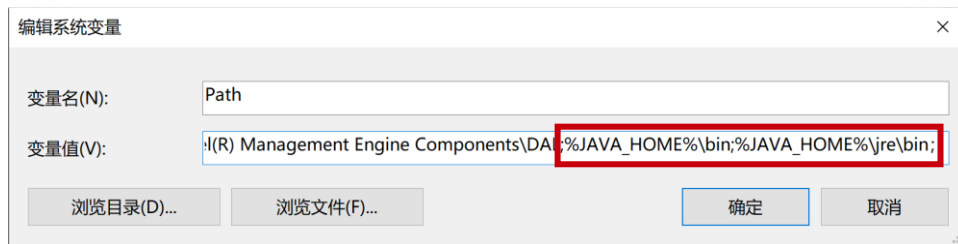
步骤 3 编辑系统变量“path”。



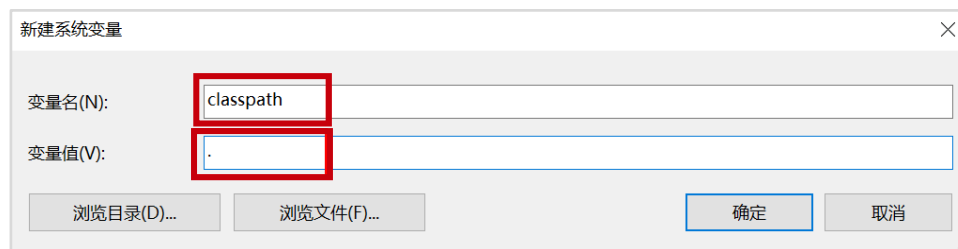
单击“编辑文本”。



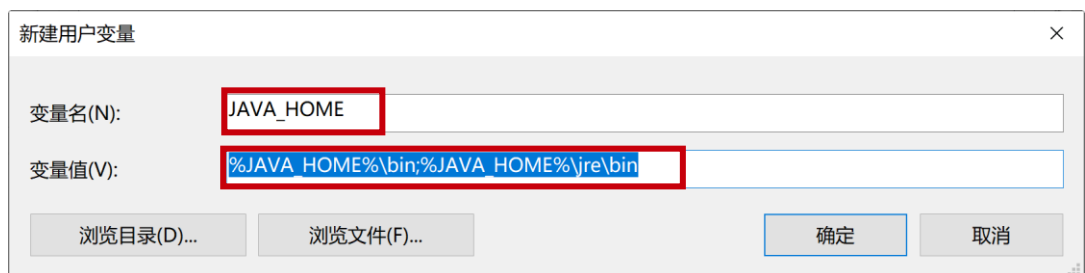
在变量值最后输入： %JAVA_HOME%\bin;%JAVA_HOME%\jre\bin; **（注意原来 Path 的变量值末尾有没有分号，如果没有，先输入分号再输入上面的代码）。**



步骤 4 新建系统变量“CLASSPATH”变量，输入“.”即可。



步骤 5 新建用户变量“JAVA_HOME”，变量值为： %JAVA_HOME%\bin;%JAVA_HOME%\jre\bin。



步骤 6 系统变量配置完毕，查询检验是否配置成功，运行 cmd 输入 java -version（java 和 -version 之间有空格）。

```
C:\Users\xxxxx>java -version
```

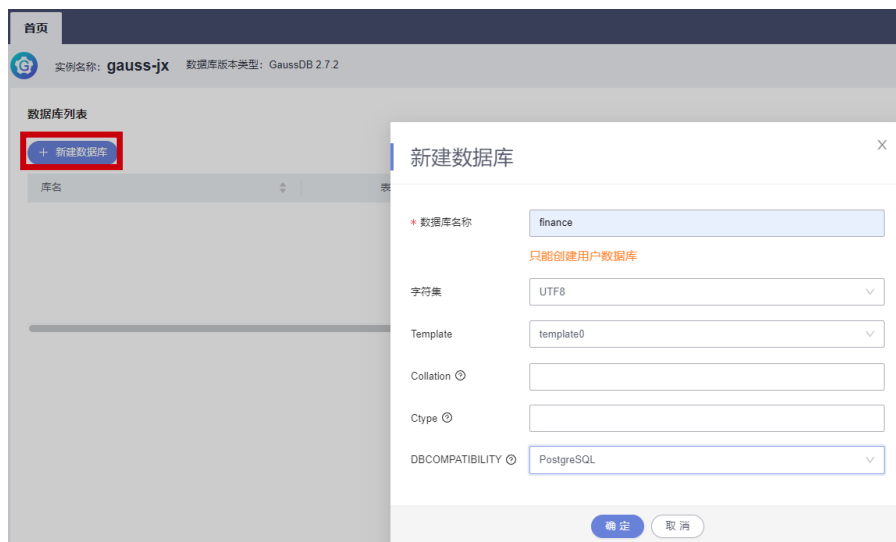
如下所示，显示版本信息，则说明安装和配置成功。

```
java version "1.8.0_311"  
Java(TM) SE Runtime Environment (build 1.8.0_311-b11)  
Java HotSpot(TM) 64-Bit Server VM (build 25.311-b11, mixed mode)
```

3.3 连接准备

3.3.1 准备实验数据

步骤 1 登录数据库实例后，单击“新建数据库”按钮。



输入新建数据库的数据库名 finance，选择对应的数据库字符集（默认使用 utf8）后，单击“确定”按钮。

步骤 2 在右侧会提示“创建数据库 finance 成功”，并在下方的数据库列表中，显示出 finance 数据库，以上执行完成，表示数据库创建成功。

数据库列表						用户库
+ 新建数据库						
库名	表数量	表大小	索引大小	字符集	操作	
finance	--	--	--	UTF8	库管理 SQL查询 删除库	

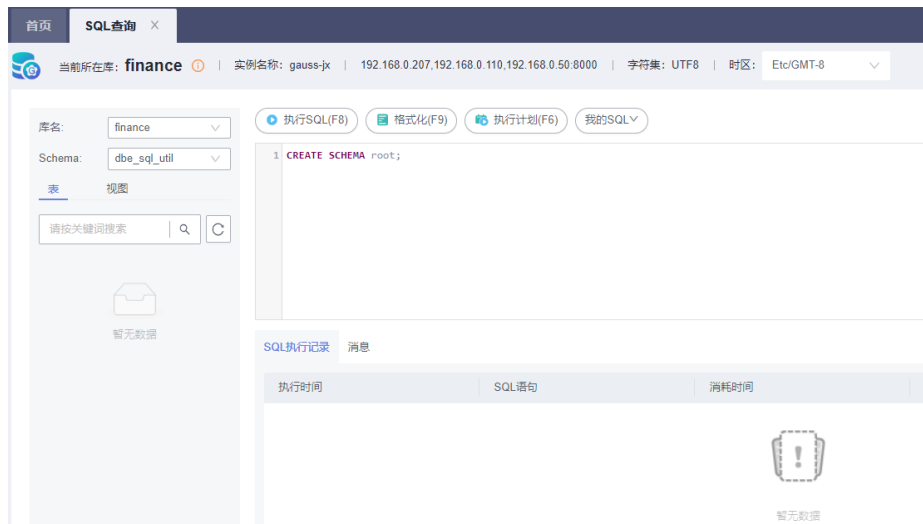
步骤 3 单击“SQL 查询”，进入 SQL 交互界面。

数据库列表						用户库
+ 新建数据库						
库名	表数量	表大小	索引大小	字符集	操作	
finance	--	--	--	UTF8	库管理 SQL查询 删除库	

步骤 4 在 finance 数据库下创建 root 用户同名的 Schema。

将 SQL 文本帖入文本框中，单击“执行 SQL”按钮或者使用 F8 执行 SQL，在“消息”界面将展示执行结果。

```
CREATE SCHEMA root;
```



root Schema 创建成功后，将数据库切换至对应的 Schema，如下：



步骤 5 在 finance 库下创建测试表 websites。



语句如下：

```
CREATE TABLE websites (  
  id int NOT NULL,  
  name char(20) NOT NULL DEFAULT "",  
  url varchar(255) NOT NULL DEFAULT "",  
  PRIMARY KEY (id)  
);  
COMMENT ON COLUMN websites.name IS '站点名称';
```

步骤 6 插入数据。

```
INSERT INTO websites VALUES
('1', 'openGauss', 'https://opengauss.org/zh/'),
('2', '华为云', 'https://www.huaweicloud.com/'),
('3', 'openEuler', 'https://openeuler.org/zh/'),
('4', '华为 support 中心', 'https://support.huaweicloud.com/');
```

3.3.2 下载 GaussDB 的 Java 驱动包

步骤 1 下载连接 GaussDB 的 Java 驱动包。

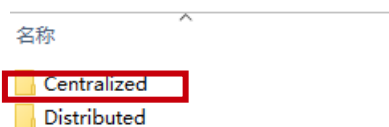
从如下地址下载 GaussDB JDBC 驱动。

https://dbs-download.obs.cn-north-1.myhuaweicloud.com/GaussDB/1660794000209/GaussDB_opengauss_client_tools.zip

将 GaussDB_opengauss_client_tools.zip 包下载到 d:\download 目录下。

对此压缩包进行解压后会有两个文件目录，对于分布式 GaussDB 实例，请选择 Distributed；对于主备实例，请选择 Centralized。

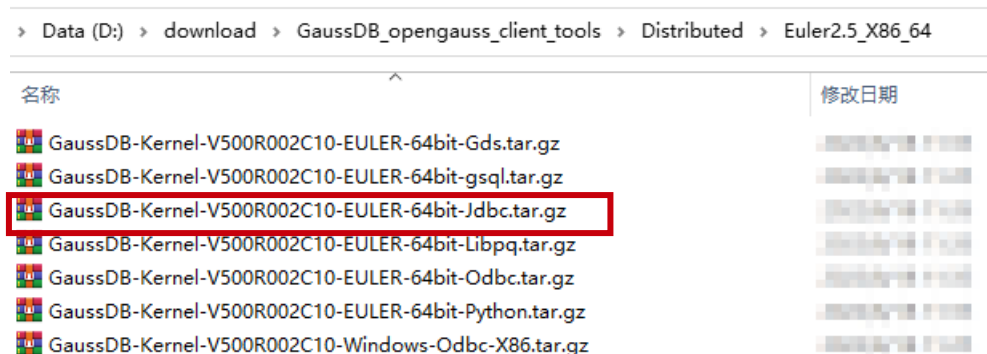
ad\GaussDB_opengauss_client_tools\



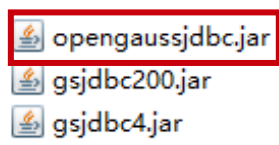
进入文件夹后，具体如下：



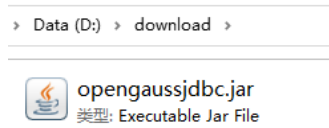
选择与云数据库相应的 X86 版驱动版本，双击 Euler2.5_X86_64 进入。



选择 GaussDB-Kernel-V500R002C10-EULER-64bit-Jdbc.tar.gz 包进行解压，解压后内容如下：



最后将 opengaussjdbc.jar 复制到 d:\download 目录下。



3.4 连接 GaussDB 并执行 Java 代码

步骤 1 使用 Java 程序连接数据库并进行查询。

在 d:\download\中创建 GaussDBDemo.java 文件，拷贝如下内容到文中，注意红色斜体部分要按照实际情况进行替换。

```
import java.sql.*;
public class GaussDBDemo{

    static final String JDBC_DRIVER = "com.huawei.opengauss.jdbc.Driver";
    static final String DB_URL = "jdbc:opengauss://弹性公网
IP:8000/finance?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 = "xxxxxxxxx";
    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 id, name, url FROM root.websites";
            ResultSet rs = stmt.executeQuery(sql);
```

```
// 展开结果集数据库
while(rs.next()){
    // 通过字段检索
    int id = rs.getInt("id");
    String name = rs.getString("name");
    String url = rs.getString("url");

    // 输出数据
    System.out.print("ID: " + id);
    System.out.print(", 站点名称: " + name);
    System.out.print(", 站点 URL: " + url);
    System.out.print("\n");
}
// 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!");
}
```

步骤 2 在安装 Java 的本机，打开 cmd 对 Java 程序编译后执行。

在 cmd 中，进入 d:\download\目录，先对 Java 程序进行编译（进入 Java 程序的目录）

```
cd /d d:\download
javac -encoding utf-8 -cp d:\download\opengaussjdbc.jar GaussDBDemo.java
```

再执行以下命令

```
java -cp ./d:/download/opengaussjdbc.jar GaussDBDemo
```

步骤 3 执行结果。

执行结果如下：

连接数据库...

一月 16, 2023 3:20:39 下午 com.huawei.opengauss.jdbc.core.v3.ConnectionFactoryImpl
openConnectionImpl

信息: [c3ce6517-bede-4356-9742-724c64286df4] Try to connect. IP: 116.205.150.217:8000

一月 16, 2023 3:20:40 下午 com.huawei.opengauss.jdbc.core.v3.ConnectionFactoryImpl
openConnectionImpl

信息: [7.250.106.88:59145/116.205.150.217:8000] Connection is established. ID: c3ce6517-
bede-4356-9742-724c64286df4

一月 16, 2023 3:20:40 下午 com.huawei.opengauss.jdbc.core.v3.ConnectionFactoryImpl
openConnectionImpl

信息: Connect complete. ID: c3ce6517-bede-4356-9742-724c64286df4

实例化 Statement 对象...

ID: 1, 站点名称: openGauss , 站点 URL: <https://opengauss.org/zh/>

ID: 2, 站点名称: 华为云 , 站点 URL: <https://www.huaweicloud.com/>

ID: 3, 站点名称: openEuler , 站点 URL: <https://openeuler.org/zh/>

ID: 4, 站点名称: 华为 support 中心 , 站点 URL: <https://support.huaweicloud.com/>

Goodbye!

4 使用 ODBC 连接数据库

4.1 加载 GaussDB ODBC 驱动包

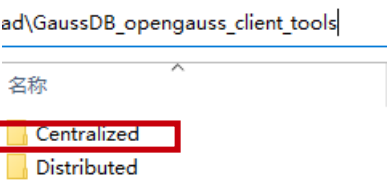
步骤 1 下载 GaussDB Windows 版 ODBC 驱动包。

从如下地址下载 GaussDB_opengauss_client_tools.zip。

https://dbs-download.obs.cn-north-1.myhuaweicloud.com/GaussDB/1660794000209/GaussDB_opengauss_client_tools.zip

将 GaussDB_opengauss_client_tools.zip 包下载到 d:\download 目录下。

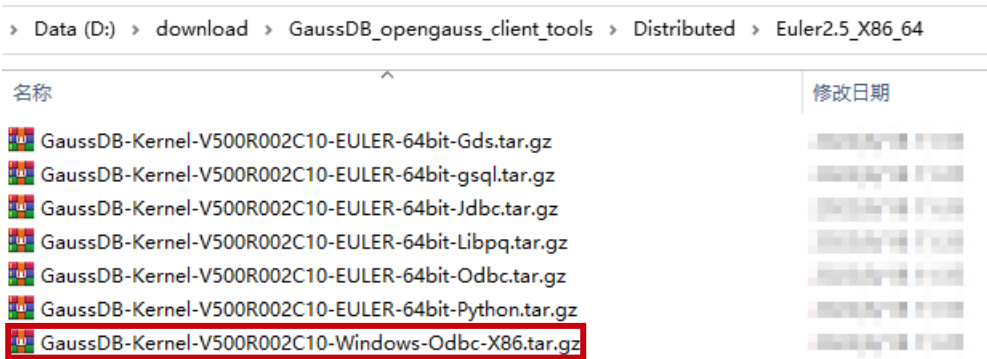
步骤 2 对此压缩包进行解压后会有两个文件目录，对于分布式 GaussDB 实例，请选择 Distributed；对于主备实例，请选择 Centralized。



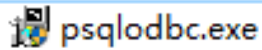
进入文件夹后，具体如下：



选择与云数据库相应的 X86 版驱动版本，双击 Euler2.5_X86_64 进入。

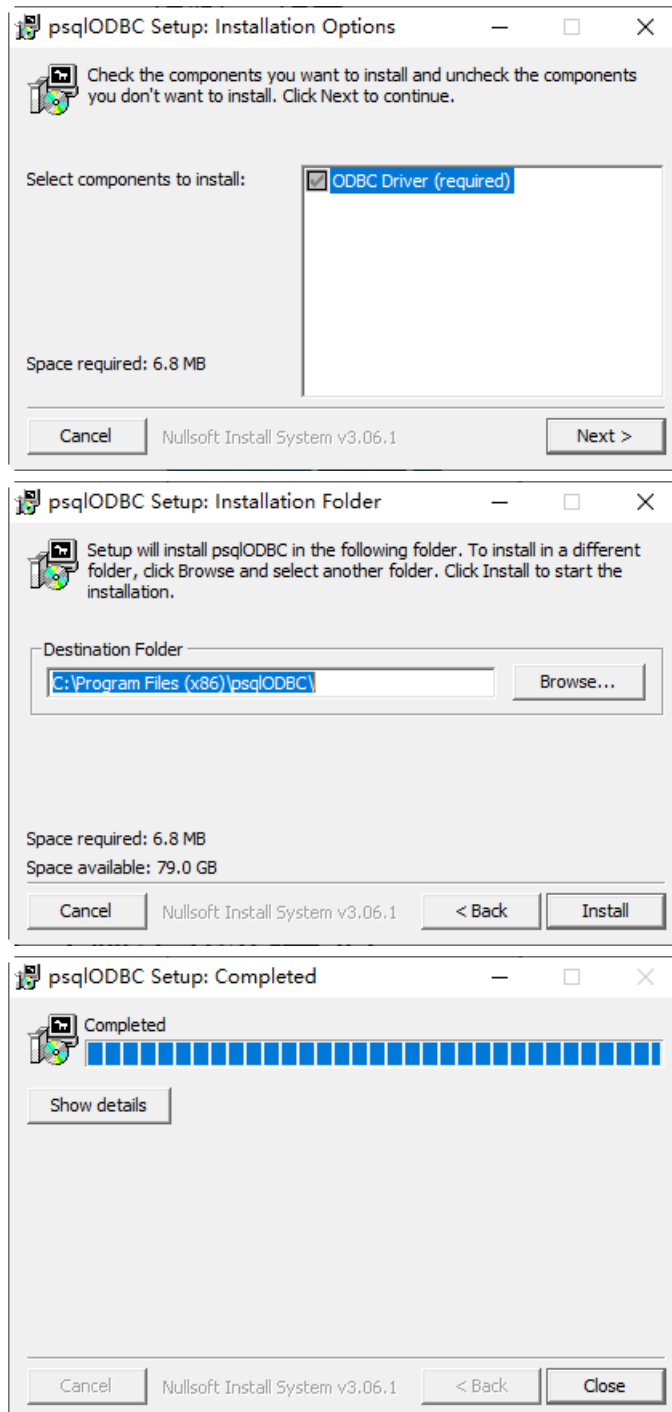


步骤 3 解压 GaussDB-Kernel-V500R002C10-Windows-Odbc-x86.tar.gz，得到 psqlodbc.exe。



步骤 4 双击 psqlodbc.exe 进行驱动安装。

根据安装向导的指引进行安装即可。



4.2 配置数据源

Windows 操作系统自带 ODBC 数据源管理器，无需用户手动安装管理器便可直接进行配置。

步骤 1 打开 ODBC 数据源管理器。

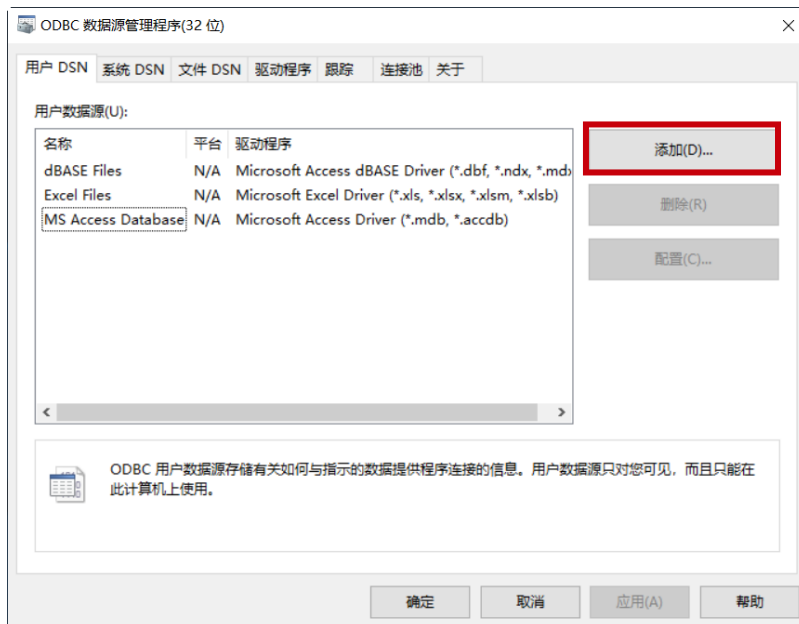
在 Windows 左下角的搜索框中输入 ODBC，即可出现数据源管理器。选择 ODBC Data Sources(32-bit)。

注意：数据库目前仅支持 32 位的 ODBC 驱动管理器，对于 64 位的机器也请选择 32 位驱动管理器。

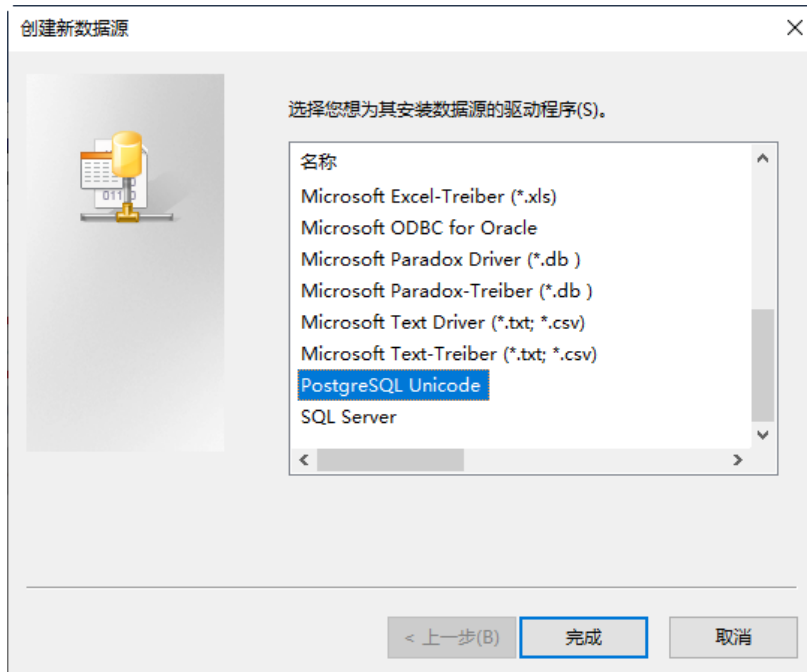


步骤 2 配置数据源。

在打开的驱动管理器上，在“用户 DSN”页签，单击“添加”。



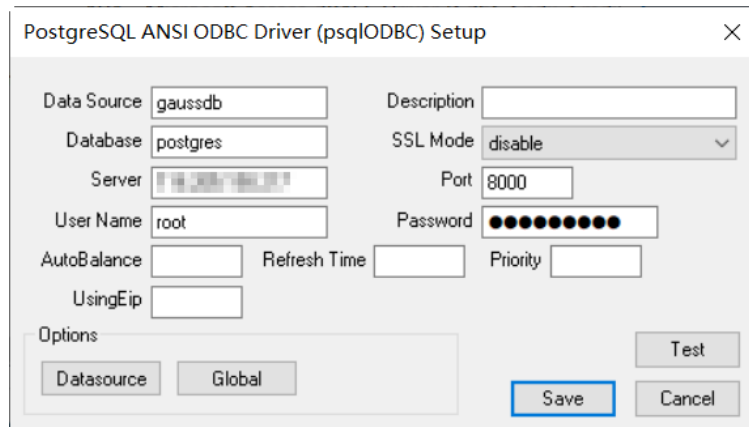
在“创建新数据源”对话框中选择“PostgreSQL Unicode”。



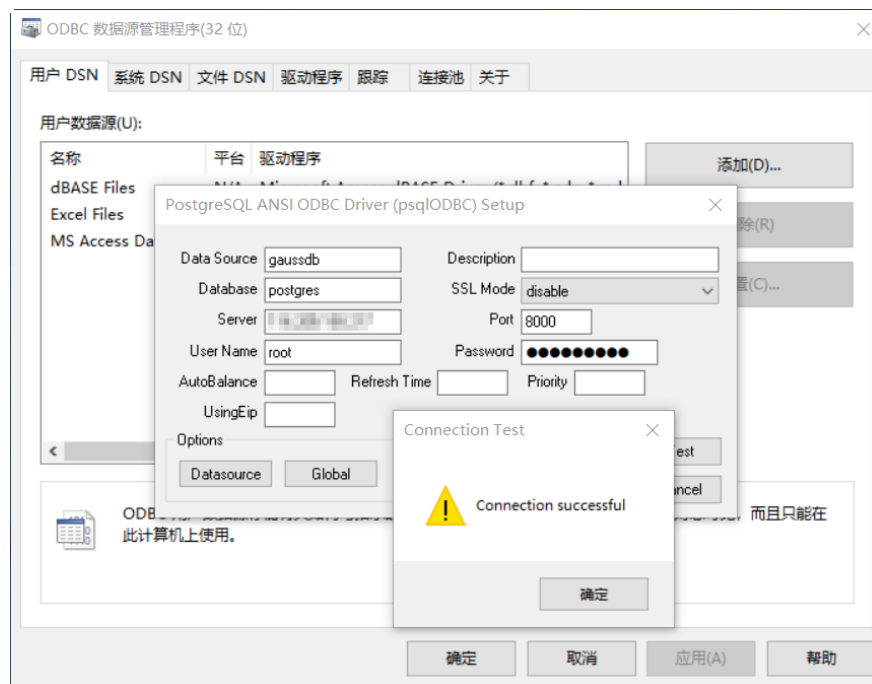
在“PostgreSQL ANSI ODBC Driver (psqlODBC) Setup”对话框中进行配置。

- **Data Source:** 根据喜好设置数据源名称，例如 gaussdb。
- **Database:** 设置为需要连接的数据库名称，例如此处连接的是 GaussDB 实例创建后的初始数据库 postgres。
- **Sever:** 请输入所要连接的数据库实例的 IP 地址，如果是外网连接 GaussDB，请输入实例所绑定的弹性公网 IP 地址。
- **Port:** 请输入购买数据库实例时设置的端口，通常为 8000。

- **User Name/Password:** 请输入连接数据库的用户名和密码。例如购买数据库实例时所创建的 root 用户及其密码。

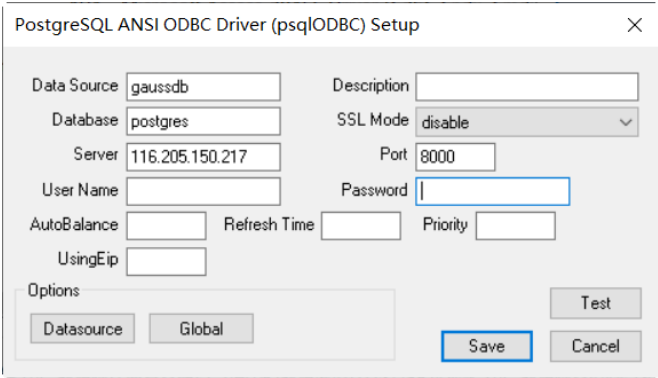


输入完成后，单击“Test”，测试是否可以成功连接数据库。弹出“Connection successful”，表示可以成功连接数据库。



此界面上配置的用户名及密码信息，将会被记录在 Windows 注册表中，再次连接数据库时就不再需要输入认证信息。但是出于安全考虑，建议在单击“Save”按钮保存配置信息前，清空相关敏感信息；在使用 ODBC 的连接 API 时，再传入所需的用户名、密码信息。

其他保持默认，单击“Save”。



4.3 执行示例代码连接并操作数据库

步骤 1 将如下示例代码拷贝到 C/C++ IDE 中。

注意：红色字体请根据实际替换上一节所设置的数据源名称。

```
// 此示例演示如何通过 ODBC 方式获取 GaussDB 中的数据。
// DBtest.c (compile with: libodbc.so)
#include <windows.h>
#include <stdlib.h>
#include <stdio.h>
#include <sqlext.h>
#ifdef WIN32
#endif
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,V_OD_id;
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,
(SQLPOINTER)SQL_AUTOCOMMIT_ON, 0);
    // 5. 连接数据源, 这里的"userName"与"password"分别表示连接数据库的用户名和用户
    密码, 请根据实际情况修改。
    // 如果 odbc.ini 文件中已经配置了用户名密码, 那么这里可以留空 (""); 但是不建议这
    么做, 因为一旦 odbc.ini 权限管理不善, 将导致数据库用户密码泄露。
    V_OD_erg = SQLConnect(V_OD_hdbc, (SQLCHAR*) "gaussdb", SQL_NTS,
        (SQLCHAR*) "userName", SQL_NTS, (SQLCHAR*) "password",
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。
    SQLExecDirect(V_OD_hstmt, "drop table IF EXISTS customer_t1", SQL_NTS);
    SQLExecDirect(V_OD_hstmt, "CREATE TABLE customer_t1(c_customer_sk INTEGER,
c_customer_name VARCHAR(32));", SQL_NTS);
    SQLExecDirect(V_OD_hstmt, "insert into customer_t1 values(25, 'li')", SQL_NTS);
    // 9. 准备执行
    SQLPrepare(V_OD_hstmt, "insert into customer_t1 values(?)", SQL_NTS);
    // 10. 绑定参数
    SQLBindParameter(V_OD_hstmt, 1, SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
        &value, 0, NULL);
    // 11. 执行准备好的语句
    SQLExecute(V_OD_hstmt);
    SQLExecDirect(V_OD_hstmt, "select c_customer_sk from customer_t1", SQL_NTS);
    // 12. 获取结果集某一列的属性
    SQLELEN descType=0;
    SQLColAttribute(V_OD_hstmt, 1, SQL_DESC_TYPE, typename, 100, NULL, (SQLELEN*)&descType);
    printf("SQLColAttribute %s\n", typename);
    // 13. 绑定结果集
    SQLBindCol(V_OD_hstmt, 1, SQL_C_SLONG, (SQLPOINTER)&V_OD_buffer, 150,
        (SQLELEN*)&V_OD_err);
    // 14. 通过 SQLFetch 取结果集中数据

```

```
V_OD_erg=SQLFetch(V_OD_hstmt);  
// 15. 通过 SQLGetData 获取并返回数据。  
while(V_OD_erg != SQL_NO_DATA)  
{  
    SQLGetData(V_OD_hstmt,1,SQL_C_SLONG,(SQLPOINTER)&V_OD_id,0,NULL);  
    printf("SQLGetData ----ID = %d\n",V_OD_id);  
    V_OD_erg=SQLFetch(V_OD_hstmt);  
};  
printf("Done !\n");  
// 16. 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);  
}
```

步骤 2 编译运行上述示例代码，返回如下内容，表示示例代码执行成功。

```
Connected !  
SQLColAttribute  
SQLGetData ----ID = 25  
SQLGetData ----ID = 100  
Done !  
  
-----  
Process exited after 0.6748 seconds with return value 0
```

此次试验结束。