数据库系统课程实验报告

实验名称:	实验九：使用 JDBC 连接 openGauss 数据库
实验日期:	2023/6/2
实验地点:	文宣楼 A402
提交日期:	2023/6/3

学号:	22920212204392
姓名:	黄勖
专业年级:	软工 2021 级
学年学期:	2022-2023 学年第二学期

1.实验目的

- 掌握使用 JDBC 连接 openGauss 数据库的方法

2.实验内容和步骤

（1） 配置好服务器和连接

（2） 准备工作

1.修改数据库的 pg_hba.conf 文件

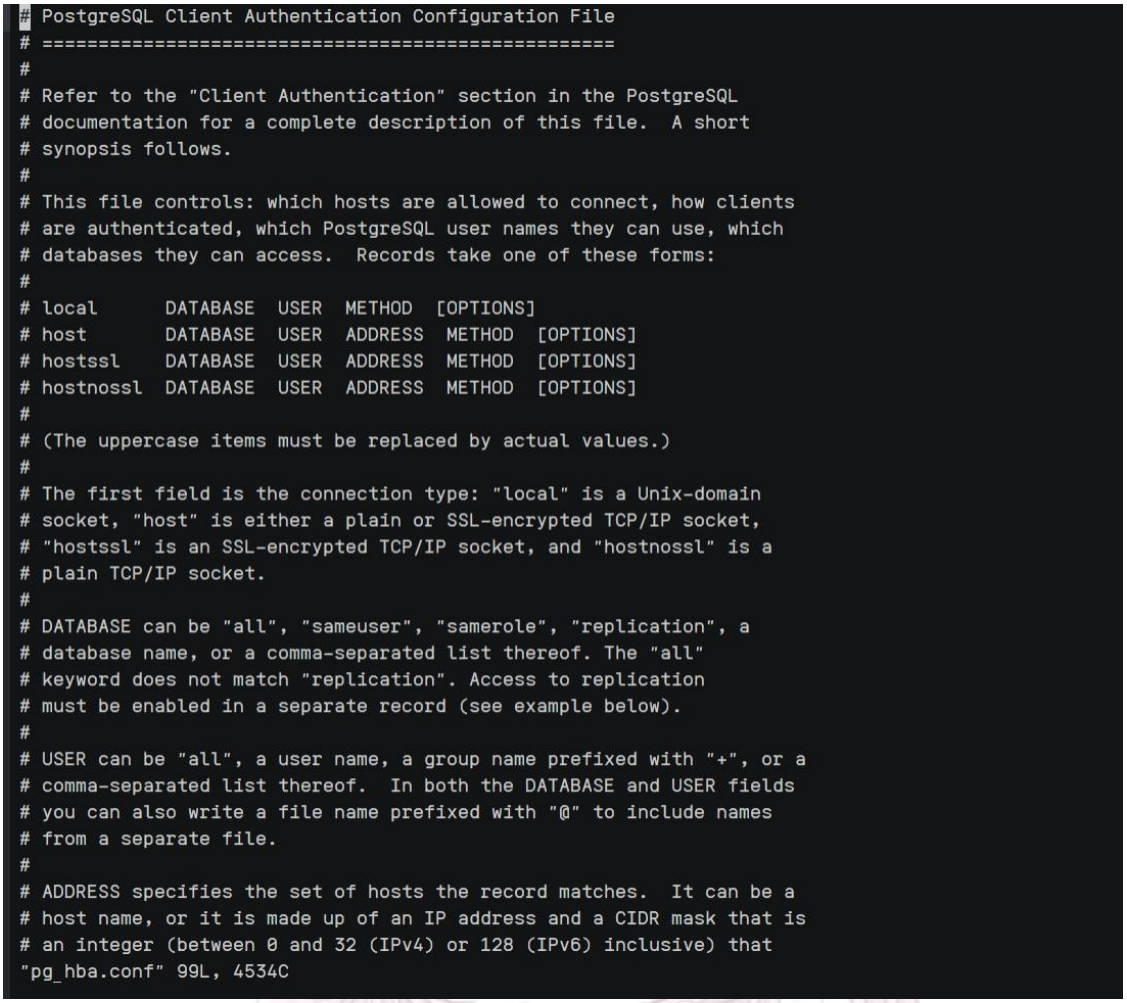
在 GS_HOME 中查找 pg_hba.conf 文件，本实验中数据库 GS_HOME设置的为/gaussdb/data/db1，实际操作中 GS_HOME 地址可以查看安装时的配置文件：<PARAM name="dataNode1" value="/gaussdb/data/db1"/>

cd /gaussdb/data/db1

vi pg_hba.conf

```
[root@ecs-c06a ~]# cd/gaussdb/data/db1
-bash: cd/gaussdb/data/db1: No such file or directory
[root@ecs-c06a ~]# cd /gaussdb/data/db1
[root@ecs-c06a db1]# vi pg_hba.conf
```

结果如下



输入“:90”找到对应位置，然后输入“i”切换到 INSERT 模式，将以下内容添加进 pg_hba.conf 文件，添加后按下“ECS”键，退出 INSERT 模式，输入“:wq”后回车保存。

IPv4 local connections:

host all all 127.0.0.1/32trust

host all all 192.168.0.19/32 trust

host all all 0.0.0.0/0 sha256

IPv6 local connections:

host all all ::1/128trust

```
# 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    192.168.0.247/32    trust
# IPv6 local connections:
host    all             all             ::1/128                 trust
```

使用 omm 用户登陆，使用 gs_ctl 将策略生效

su - omm

gs_ctl reload -D /gaussdb/data/db1/

```
[omm@ecs-c06a ~]$ gs_ctl reload -D /gaussdb/data/db1/
[2023-06-02 16:51:51.241][4463][][gs_ctl]: gs_ctl reload ,datadir is /gaussdb/data/db1
server signaled
```

2. 登陆数据库授权退出

使用 omm 用户登陆数据库给dbuser 用户授权，并退出数据库

gsql -d postgres -p 26000 -r

```
[omm@ecs-c06a ~]$ gs_om -t start
Starting cluster.
=========================================
[SUCCESS] ecs-c06a
2023-06-02 16:50:12.386 6479ad44.1 [unknown] 281462104129552 [unknown] 0 dn_6001 01000  0 [BACKEND] WARNING:  could not c
reate any HA TCP/IP sockets
=========================================
Successfully started.
[omm@ecs-c06a ~]$ gsql -d postgres -p 26000 -r
gsql ((openGauss 2.0.0 build 78689da9) compiled at 2021-03-31 21:03:52 commit 0 last mr  )
Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.

postgres=# \q
```

创建一个角色 dbuser

CREATE ROLE dbuser IDENTIFIED BY 'Bigdata@123';

alter role dbuser createrole createdb;

```
postgres=# alter role dbuser createrole createdb;
ALTER ROLE
```

\q

```
postgres=# \q
[omm@ecs-c06a ~]$
```

3.修改数据库监听地址

在GS_HOME中，本实验中数据库GS_HOME设置的为

/gaussdb/data/db1

```
[omm@ecs-c06a ~]$ cd /gaussdb/data/db1
[omm@ecs-c06a db1]$ vi 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
#------------------------------------------------------------------------------
"postgresql.conf" 870L, 38475C
```

输入“:60”找到对应位置，然后输入“i”切换到 INSERT 模式，将listen_addresses 的值修改成为*，修改后按下“ECS”键，退出INSERT模式，输入“:wq”后回车保存。

```
listen_addresses = '*'          # what IP address(es) to listen on;
                                # comma-separated list of addresses;
                                # defaults to 'localhost'; use '*' for all
                                # (change requires restart)
```

修改完成后重启数据库生效（-D 后面的数据库默认路径，需要根据实际情况进行修改）

gs_ctl restart -D /gaussdb/data/db1/

```
[omm@ecs-c06a db1]$ gs_ctl restart -D /gaussdb/data/db1/
```

4. 下载 Java 驱动包导入工具

```
2023-06-02 17:01:43.492 6479aff7.1 [unknown] 281468440543248 [unknown] 0 dn_6001 00000  0 [BACKEND] LOG:  loaded library
"security_plugin"
2023-06-02 17:01:43.492 6479aff7.1 [unknown] 281468440543248 [unknown] 0 dn_6001 01000  0 [BACKEND] WARNING:  could not c
reate any HA TCP/IP sockets
2023-06-02 17:01:43.496 6479aff7.1 [unknown] 281468440543248 [unknown] 0 dn_6001 01000  0 [BACKEND] WARNING:  No explicit
 IP is configured for listen_addresses GUC.
2023-06-02 17:01:43.496 6479aff7.1 [unknown] 281468440543248 [unknown] 0 dn_6001 00000  0 [BACKEND] LOG:  InitNuma numaNo
deNum: 1 numa_distribute_mode: none inheritThreadPool: 0.
2023-06-02 17:01:43.496 6479aff7.1 [unknown] 281468440543248 [unknown] 0 dn_6001 00000  0 [BACKEND] LOG:  reserved memory
 for backend threads is: 340 MB
2023-06-02 17:01:43.496 6479aff7.1 [unknown] 281468440543248 [unknown] 0 dn_6001 00000  0 [BACKEND] LOG:  reserved memory
 for WAL buffers is: 320 MB
2023-06-02 17:01:43.497 6479aff7.1 [unknown] 281468440543248 [unknown] 0 dn_6001 00000  0 [BACKEND] LOG:  Set max backend
 reserve memory is: 660 MB, max dynamic memory is: 1518 MB
2023-06-02 17:01:43.497 6479aff7.1 [unknown] 281468440543248 [unknown] 0 dn_6001 00000  0 [BACKEND] LOG:  shared memory 1
901 Mbytes, memory context 2178 Mbytes, max process memory 4096 Mbytes
2023-06-02 17:01:43.516 6479aff7.1 [unknown] 281468440543248 [unknown] 0 dn_6001 00000  0 [CACHE] LOG:  set data cache  s
ize(12582912)
2023-06-02 17:01:43.517 6479aff7.1 [unknown] 281468440543248 [unknown] 0 dn_6001 00000  0 [CACHE] LOG:  set metadata cach
e  size(4194304)
2023-06-02 17:01:43.596 6479aff7.1 [unknown] 281468440543248 [unknown] 0 dn_6001 00000  0 [BACKEND] LOG:  gaussdb: fsync
file "/gaussdb/data/db1/gaussdb.state.temp" success
2023-06-02 17:01:43.596 6479aff7.1 [unknown] 281468440543248 [unknown] 0 dn_6001 00000  0 [BACKEND] LOG:  create gaussdb
state file success: db state(STARTING_STATE), server mode(Normal)
2023-06-02 17:01:43.658 6479aff7.1 [unknown] 281468440543248 [unknown] 0 dn_6001 00000  0 [BACKEND] LOG:  max_safe_fds =
977, usable_fds = 1000, already_open = 13
The core dump path is an invalid directory
2023-06-02 17:01:43.660 6479aff7.1 [unknown] 281468440543248 [unknown] 0 dn_6001 00000  0 [BACKEND] LOG:  the configure f
ile /opt/gaussdb/app/etc/gscgroup_omm.cfg doesn't exist or the size of configure file has changed. Please create it by ro
ot user!
2023-06-02 17:01:43.660 6479aff7.1 [unknown] 281468440543248 [unknown] 0 dn_6001 00000  0 [BACKEND] LOG:  Failed to parse
 cgroup config file.

[2023-06-02 17:01:44.385][4530][][gs_ctl]:  done
[2023-06-02 17:01:44.385][4530][][gs_ctl]: server started (/gaussdb/data/db1)
[omm@ecs-c06a db1]$
```

下载 Java 连接 openGauss 的驱动包，并将其导入对应的使用工具

5. 创建测试数据库demo

首先授予dbuser 登录权限

```
postgres=# ALTER ROLE dbuser LOGIN;
ALTER ROLE
```

退出后重新登录，使用 gsql 工具登陆数据库，并输入dbuser 密码

gsql -d postgres -p 26000 -U dbuser -r

```
[omm@ecs-c06a db1]$ gsql -d postgres -p 26000 -U dbuser -r
Password for user dbuser:
gsql ((openGauss 2.0.0 build 78689da9) compiled at 2021-03-31 21:03:52 commit 0 last mr  )
Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.

postgres=>
```

创建数据库demo

create database demo ENCODING 'UTF8' template = template0;

```
postgres=> create database demo ENCODING 'UTF8' template = template0;
CREATE DATABASE
```

切换到demo 数据库，并输入dbuser 密码

\connect demo;

```
postgres=> \connect demo;
Password for user dbuser:
Non-SSL connection (SSL connection is recommended when requiring high-security)
You are now connected to database "demo" as user "dbuser".
demo=>
```

6.创建 schema

创建名为demo 的 schema，并设置 demo 为当前的schema

CREATE SCHEMA demo;

```
demo=> CREATE SCHEMA demo;
CREATE SCHEMA
```

将默认搜索路径设为 demo

SET search_path TO demo;

```
demo=> SET search_path TO demo;
SET
```

7.创建测试表 websites

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';

```
demo=> CREATE TABLE websites (
demo(>   id int NOT NULL,
demo(>   name char(20) NOT NULL DEFAULT '',
demo(>   url varchar(255) NOT NULL DEFAULT '',
demo(>   PRIMARY KEY (id)
demo(> );
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "websites_pkey" for table "websites"
CREATE TABLE
demo=> COMMENT ON COLUMN websites.name IS '站点名称';
COMMENT
demo=>
```

8.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/');

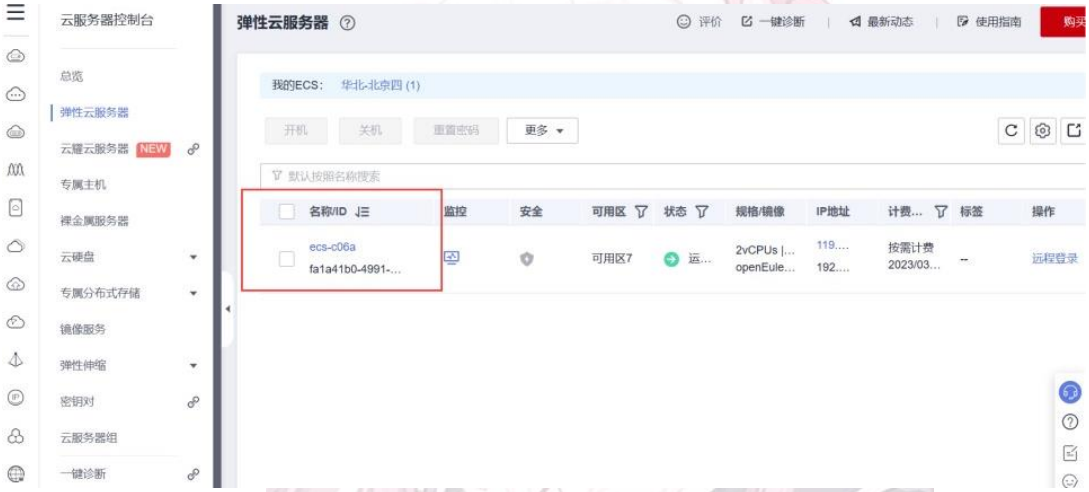
```
COMMENT
demo=> INSERT INTO websites VALUES
demo-> ('1', 'openGauss', 'https://opengauss.org/zh/'),
demo-> ('2', '华为云', 'https://www.huaweicloud.com/'),
demo-> ('3', 'openEuler', 'https://openeuler.org/zh/'),
demo-> ('4', '华为support中心', 'https://support.huaweicloud.com/');
INSERT 0 4
demo=>
```

9.退出数据库

```
demo=> \q
[omm@ecs-c06a db1]$
```

（3）确定 26000 端口是否开放

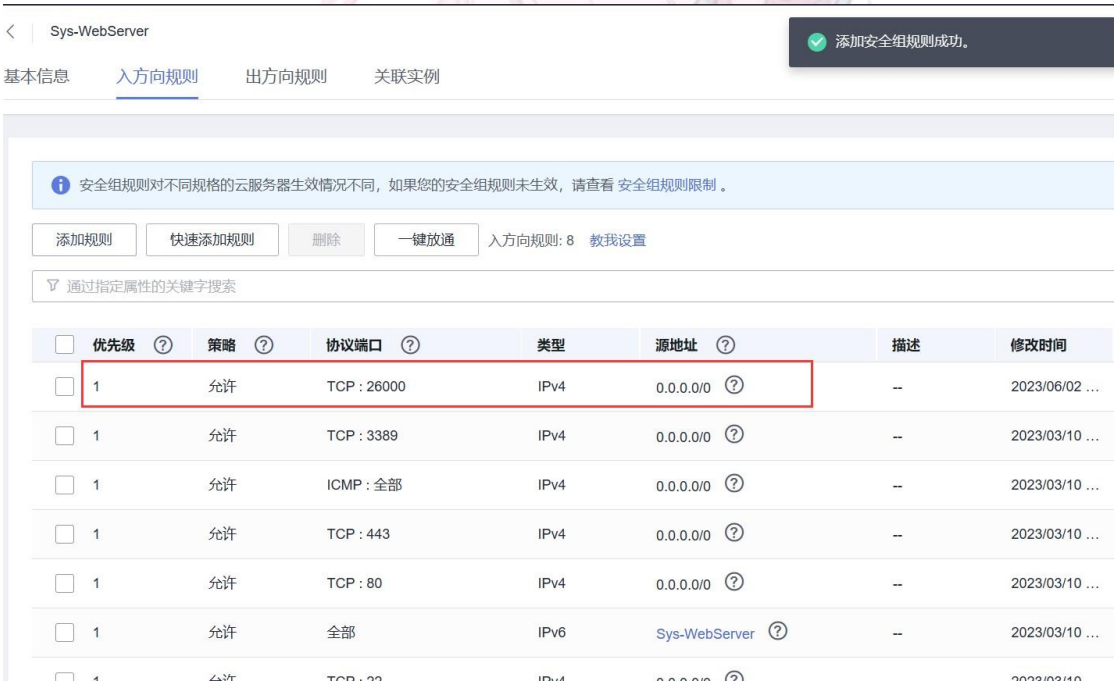
打开华为云首页，登录后进入“控制台”，点击“弹性云服务器 ECS”进入 ECS 列表



在云服务器控制台找到安装数据库主机的 ECS，点击查看基本信息，找到安全组

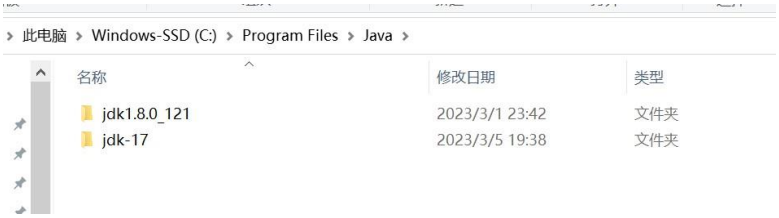
点击进入安全组，选择“入方向规则”并“添加规则”，进行 26000 端口设置

确定后，可以看到入网规则多了“TCP:26000”，如下图：



（4） 下载并安装 JDK

由于之前已经安装并且配置好了，在此不多做说明



（5） 配置 JDK 环境变量

之前已经配置过环境变量，查看如下

变量	值
classpath	.;C:\Program Files\Java\jdk-17\lib
JAVA_HOME	C:\Program Files\Java\jdk-17

（6） 连接 openGauss 并执行java 代码

1. 使用 Java 程序连接数据库并进行查询

在 postgresql.jar 文件所在的文件夹中创建 OpenGaussDemo.java 文件，文件内容如下

OpenGaussDemo.java

```
import java.sql.*;
0 个用法
public class OpenGaussDemo {

    0 个用法
    static final String JDBC_DRIVER = "org.postgresql.Driver";
    0 个用法
    static final String DB_URL = "jdbc:postgresql://119.3.216.100:26000/demo?ApplicationName=app1";
    // 数据库的用户名与密码，需要根据自己的设置
    0 个用法
    static final String USER = "dbuser";
    0 个用法
    static final String PASS = "Bigdata@123";
    0 个用法
    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 demo.websites";
            ResultSet rs = stmt.executeQuery(sql);
```

```
OpenGaussDemo.java
            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();
            try{
                if(conn!=null) conn.close();
            }catch(SQLException se){
                se.printStackTrace();
            }
        }
        System.out.println("Goodbye!");
    }
}
```

2.编译后执行

在安装Java 的本机，打开 cmd 对Java 程序编译后执行

在 cmd 中进入实验九目录，先对 Java 程序进行编译（进入Java 程序的目录）

javac -encoding utf-8 -cp postgresql.jar OpenGaussDemo.java

运行

java postgresql.jar OpenGaussDemo

```
E:\QQ文件>java -cp .;postgresql.jar OpenGaussDemo
连接数据库...
6月 02, 2023 6:14:56 下午 org.postgresql.core.v3.ConnectionFactoryImpl openConnectionImpl
信息: [fa5bddce-42d8-4854-b7ec-0124a51b6b1f] Try to connect. IP: 119.3.216.100:26000
6月 02, 2023 6:15:00 下午 org.postgresql.core.v3.ConnectionFactoryImpl openConnectionImpl
信息: [10.32.42.245:60333/119.3.216.100:26000] Connection is established. ID: fa5bddce-42d8-4854-b7ec-0124a51b6b1f
6月 02, 2023 6:15:01 下午 org.postgresql.core.v3.ConnectionFactoryImpl openConnectionImpl
信息: Connect complete. ID: fa5bddce-42d8-4854-b7ec-0124a51b6b1f
 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!
```

完成连接和操作工作，至此完成实验。

3.实验总结

3.1 完成的工作

本次实验完成了在 openGauss 中创建数据库、表，并使用 jdbc 连接到新创建的数据库，并对 java 程序进行编译运行，输出了连接到的站点的相应信息。

3.2 对实验的认识

（1）简述使用 jdbc 连接到 openGauss 数据库的主要步骤。

答：

1.准备 jdbc 驱动程序

JDBC 驱动程序是连接数据库的重要组成部分。需要从 openGauss 官网下载对应版本的 JDBC 驱动程序（本次实验中已经给出），并将其添加到 CLASSPATH 环境变量中，或者在执行 Java 命令时使用 -cp 参数手动指定类路径。

2.加载 JDBC 驱动程序

在使用 JDBC 连接数据库之前，需要先加载对应的驱动程序。通过 Class.forName() 方法，可以动态地加载驱动程序，并注册到 DriverManager 中。例如：

Class.forName("org.postgresql.Driver");

3. 打开数据库连接

使用 DriverManager.getConnection() 方法，可以打开一个与数据库的连接。该方法返回一个 Connection 对象，该对象用于后续的数据库访问操作。例如：

```
String url = "jdbc:postgresql://localhost:5432/testdb";
String user = "username"; String password = "password";
Connection conn = DriverManager.getConnection(url, user, password);
```

其中，url 参数指定了数据库的 URL，user 和 password 分别是连接数据库所需的用户名和密码。

4.创建 Statement 或者 PreparedStatement 对象

Statement 和 PreparedStatement 对象是执行 SQL 查询和更新语句的核心类。Statement 对象用于执行静态 SQL 语句，而 PreparedStatement 则用于执行动态 SQL 语句。例如：

```
Statement st = conn.createStatement();
PreparedStatement ps = conn.prepareStatement("SELECT * FROM user WHERE id = ?");
```

5.执行 SQL 查询或更新语句

使用 Statement 或 PreparedStatement 对象，可以执行 SQL 查询和更新语句。例如：

```
ResultSet rs = st.eXecuteQuery("SELECT * FROM user");
int rows = ps.eXecuteUpdate();
```

6. 处理查询结果

如果执行的是查询语句，需要处理返回的结果集。可以使用 ResultSet 对象来遍历结果集，如：

```
while (rs.neXt()) {
    int id = rs.getInt("id");
```

```
        String name = rs.getString("name");
        // 处理查询结果
    }
```

7. 关闭数据库连接

在完成所有的数据库操作后，需要关闭与数据库的连接，以释放占用的资源。可以使用 Connection、Statement 和 ResultSet 对象的close() 方法来关闭对应的资源。例如：

```
rs.close();
st.close();
conn.close();
```

3.3 遇到的困难及解决方法

在理解 openGauss 数据连接 jdbc 的过程中，我花费了很多时间。不过，通过查阅参考文档和老师提供的链接，我最终对其有了初步的认识和理解。这帮助我解决了相关问题。