

《数据库》

GaussDB(for openGauss)

数据库实例连接及建表数据导入指导手册

目录

前 言	3
实验环境说明	3
1 通过 DAS 连接 GaussDB(for openGauss)	4
1.1 实验介绍	4
1.1.1 关于本实验	4
1.1.2 实验目的	4
1.2 通过 IAM 账户登录到华为云控制台	4
2 通过公网连接 GaussDB(for openGauss)	12
2.1 实验介绍	12
2.1.1 关于本实验	12
2.1.2 实验目的	12
2.2 远程连接 GaussDB(for openGauss)	12
2.2.1 Data Studio 简介	12
2.2.2 Data Studio 安装	13
3 数据库建表及数据导入	18
3.1 实验介绍	18
3.1.1 关于本实验	18
3.1.2 实验目的	18
3.2 *通过 Dbeaver 自动导入数据（自行选用）	18
3.3 DAS/IAM 登录方式下的建表及数据导入	19
3.3.1 数据库建表	19
3.3.2 批量数据导入	19
3.4 公网/IP 登录方式下的建表及数据导入	24
3.4.1 建表/导入	24
4 附录	28
4.1 附录一：安装和配置 JDK	28
4.1.1 下载并安装 JDK	28

4.1.2 配置 JDK 环境变量..... 30

4.2 附录二：创建表语句..... 33

4.2.1 表 tbcell..... 33

4.2.2 表 tbPCIAssignment..... 34

4.2.3 表 tbATUData..... 35

4.2.4 表 tbMROData..... 36

4.2.5 表 tbC2l..... 36

前 言

实验环境说明

本实验环境为华为云环境。

- 1. 本次实验适用于多个用户访问同一数据库 GaussDB(for openGauss)实例。我们已购买了两个 GaussDB(for openGauss)实例，包括一个基础版和一个分布式版本。其中基础版仅支持 DAS 方式连接数据库实例，分布式版本支持 DAS 和公网连接两种方式连接数据库实例（两种连接方式详见后续章节）。
- 2. 华为方已提供云主账号，指导老师/助教已在主账号上创建了以 IAM、IP 方式登陆的子账号，每个子账号分配给一组学生做实验使用。

为了满足数据库原理与实践课程实验需要，建议每套实验环境采用以下配置：

设备名称、型号与版本的对应关系如下：

设备名称	设备型号	软件版本
数据库	GaussDB(for openGauss)8核 64 GB	GaussDB(for openGauss)服务

1 通过 DAS 连接 GaussDB(for openGauss)

1.1 实验介绍

1.1.1 关于本实验

本实验主要描述如何通过华为云数据管理服务（Data Admin Service，简称 DAS），通过预先分配好的 IAM 子账号，来连接华为云 GaussDB(for openGauss)数据库实例。

DAS 是一款专业的简化数据库管理工具，提供优质的可视化操作界面，大幅提高工作效率，让数据管理变得既安全又简单。

DAS 连接数据库，无需使用 IP 地址，易用、安全、高级、智能。

1.1.2 实验目的

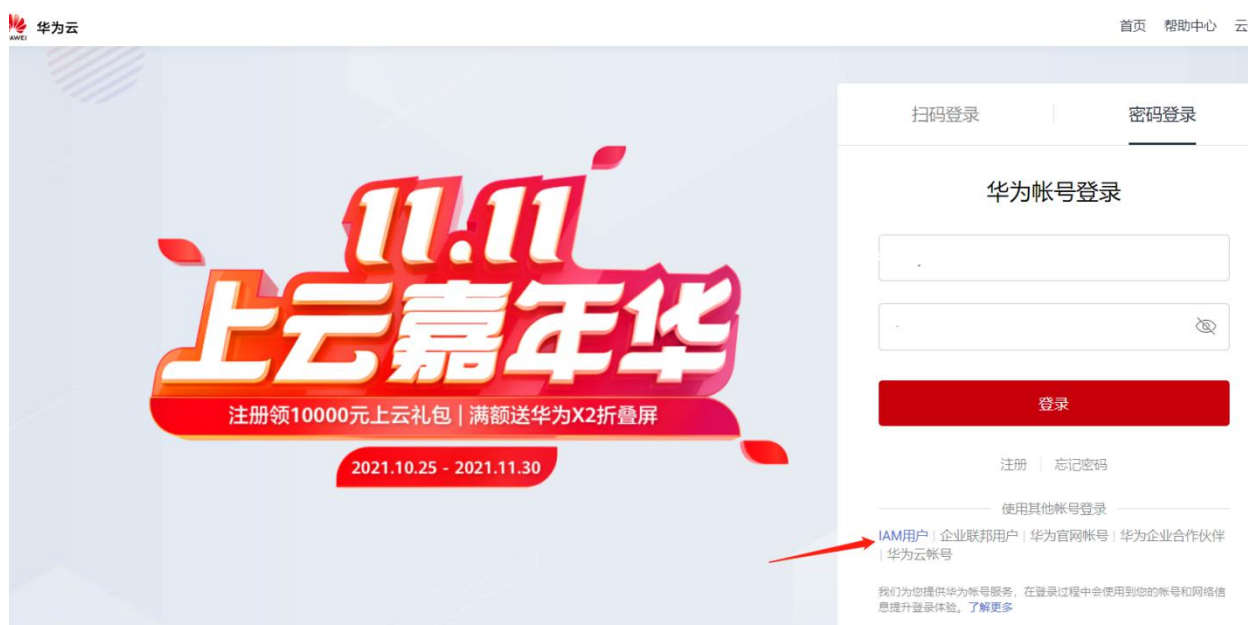
掌握 DAS 连接 GaussDB(for openGauss)数据库实例。

1.2 通过 IAM 账户登录到华为云控制台

说明：统一身份认证（Identity and Access Management，简称 IAM）是华为云提供权限管理的基础服务，可以帮助用户安全地控制华为云服务和资源的访问权限。

步骤 1 登陆华为云官网

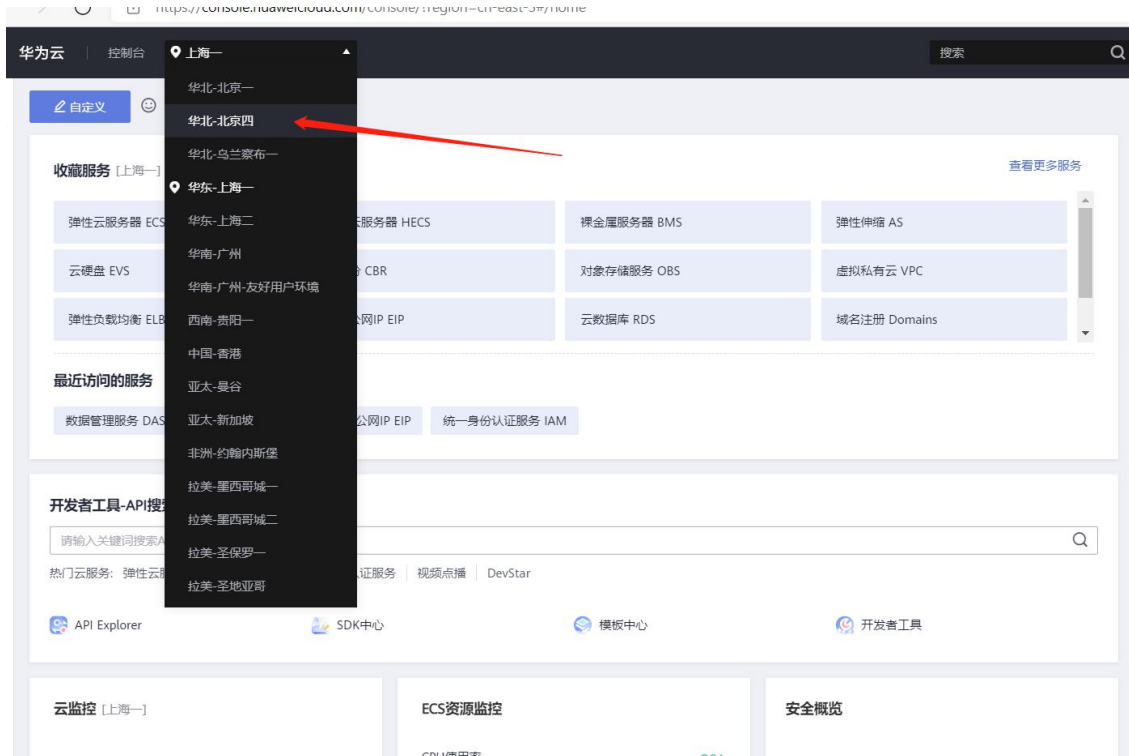
登陆 <https://www.huaweicloud.com/>，进入华为云官网，选择 **IAM 用户登陆**。



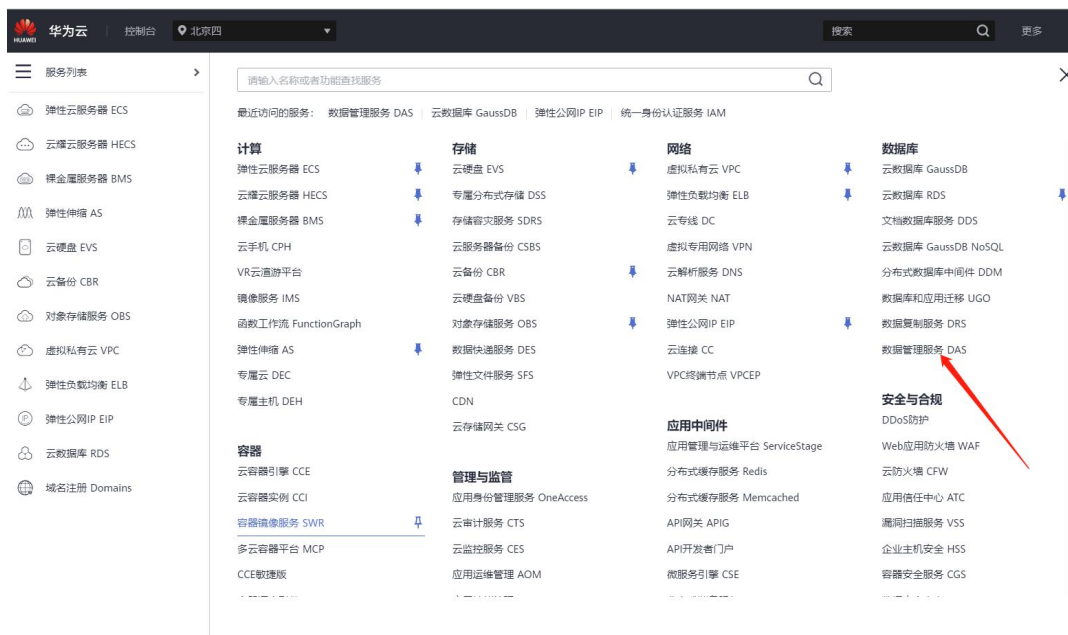
根据自己分配的账号填写主账号、IAM 用户名(这里以 user100 为示例)及密码，登陆。



步骤 2 登陆后进入控制台，选择北京四。



在服务列表，选择数据库中的**数据库管理服务 DAS**。



进入开发工具。



步骤 3 新增数据库实例登录



单击“新增数据库实例登录”，具体如下：

在此设置页面，“数据库引擎”选择 **GaussDB(for openGauss)**，然后在“数据库来源”中就会出现前面提到的 GaussDB(for openGauss)数据库基础版和分布式版实例，接着**选中想要连接的实例**。

设置**数据库名称**（如：finance）、**登录用户名及密码**，然后先**测试下连接**，测试成功后会提示“连接成功”，勾选“**记住密码**”、打开“**定时采集**”，设置完成后单击“**立即新增**”。

新增数据库实例登录

* 数据库引擎

GaussDB(for openGauss)

* 数据库来源

GaussDB

请输入实例名称

Q

C

实例名称	数据库引擎版本	实例类型	状态
<input type="radio"/> gauss-fb76	GaussDB (for openGauss) ...	分布式	● 正常
<input checked="" type="radio"/> gauss-a6f8	GaussDB (for openGauss) ...	主备	● 正常

* 数据库名称

finance

* 登录用户名

user

* 密码

.....

测试连接

☒ 记住密码

同意DAS使用加密方式记住密码 (建议选中, 否则DAS将无法开启元数据采集功能)

描述

定时采集

☒

若不开启, DAS只能实时的从数据库获取结构定义数据, 将会影响数据库实时性能。

SQL执行记录

☒

开启后, 便于查看SQL执行历史记录, 并可再次执行, 无需重复输入。

立即新增

返回

新增完成, 通过单击操作中的“登录”可以进入到相应的数据库实例, 具体如下:

新增数据库实例登录

GaussDB

全部数据库引擎

请输入数据库实例登录地址/实例名称/备注

数据库实例	数据库引擎版本	数据库来源	登录用户名	记住密码	描述	创建时间	共享用...	操作
gauss-a6f8(192.168.0.172,1...	GaussDB (for openGa...	GaussDB	user	是		2021/11/01 10:1...	查看 (0)	登录 修改 删除 智能运维

单击 finance 库右边操作下的“库管理”, 进入库管理页面, 具体如下:

Data Admin Service GaussDB(for openGauss)

SQL操作 库管理 导入/导出 账号管理

首页

实例名称: gauss-a6f8 数据库版本类型: GaussDB(for openGauss) 1.4.1

数据库列表

+ 新建数据库

用户库

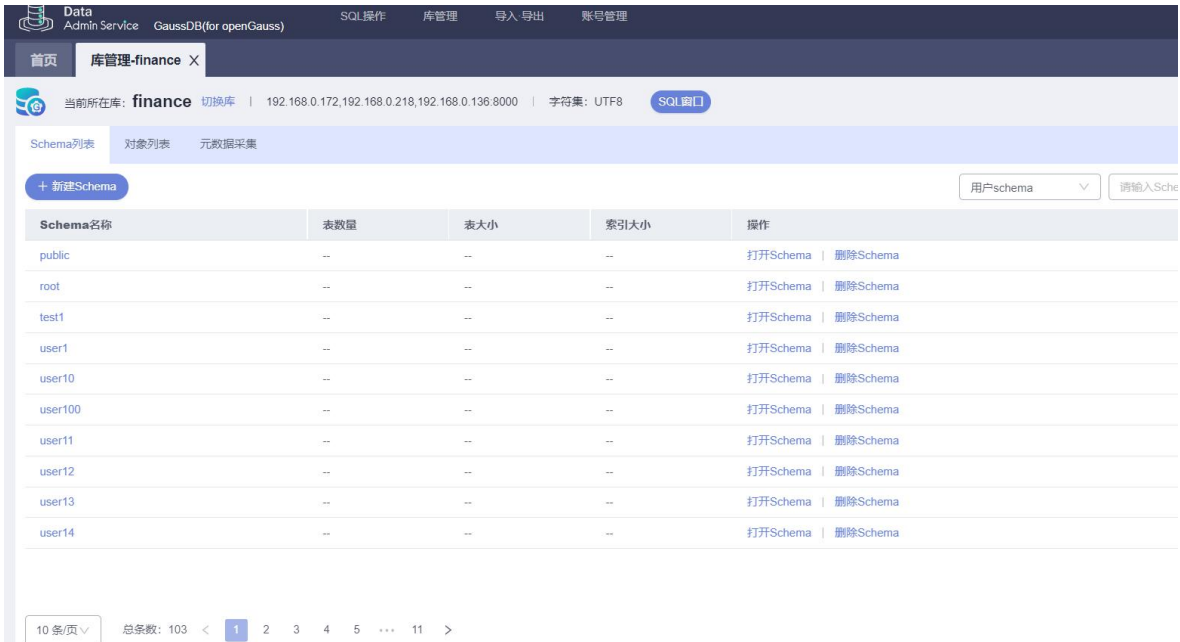
请输入库名进

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

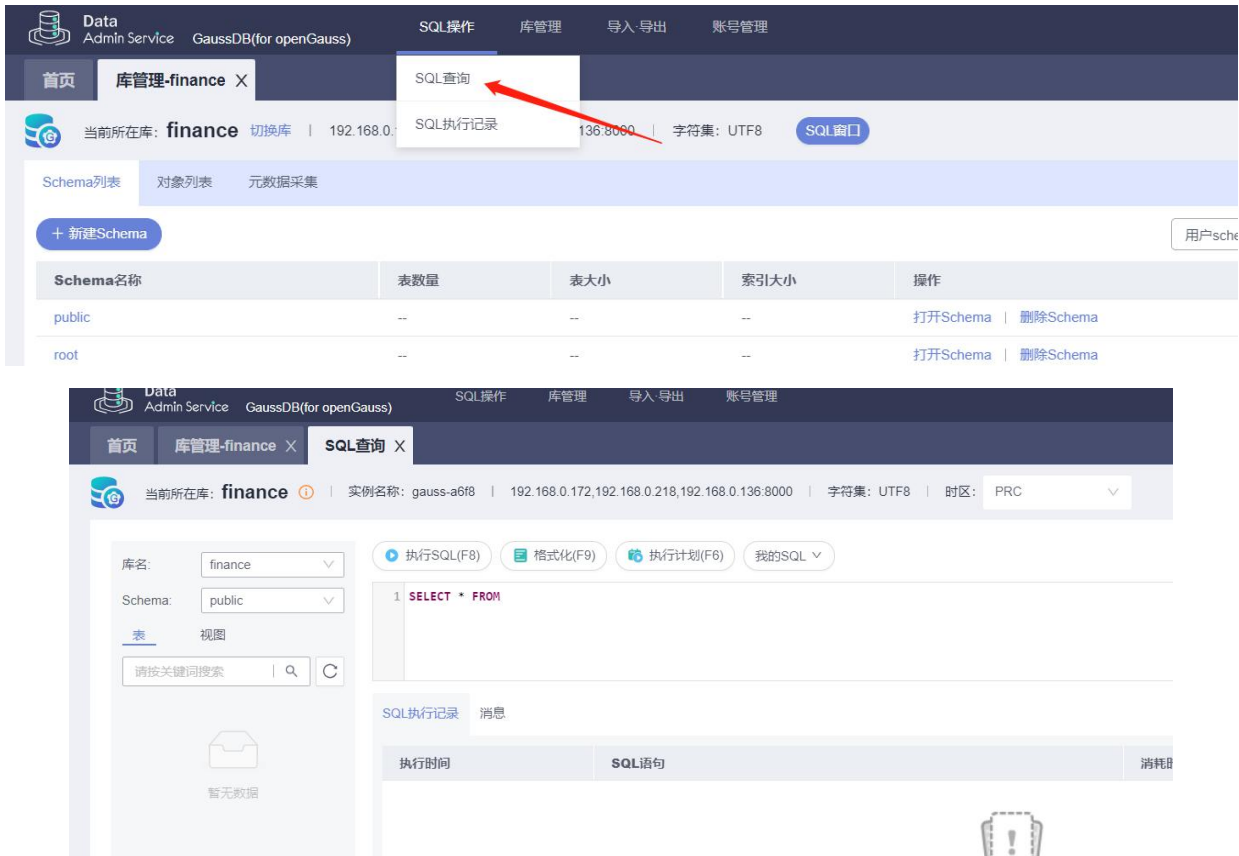
15条/页

总条数: 1

< 1 >



单击页面中的“SQL 窗口”按钮，进入如下页面：

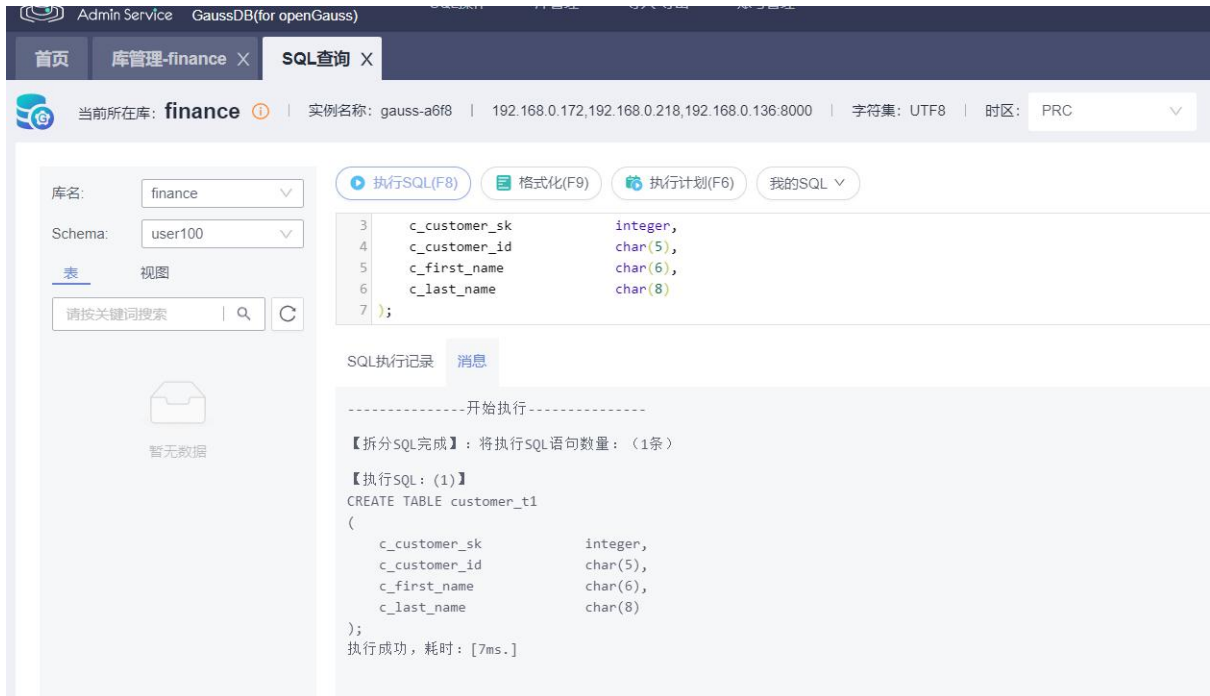


在此请确认，库名为 finance，Schema 为 user*(你所登陆的用户名)。如果不是请单击下拉箭头进行选择。

步骤 4 验证数据库用户权限。

因为用户 user*只有在同名的 Schema(user*)下才有对数据库对象的各项操作权限，其它权限都没有。

比如可以成功创建表。



语句如下：

```
CREATE TABLE customer_t1
(
    c_customer_sk      integer,
    c_customer_id      char(5),
    c_first_name       char(6),
    c_last_name        char(8)
);
```

也可以操作自己 Schema 下的对象，如下：



语句如下：

```
insert into customer_t1 VALUES(1,'cu001','zhang','xiao');
```

```
SELECT * from customer_t1;
```

但是不能创建用户，如下：



语句如下：

```
CREATE USER zhangxiao IDENTIFIED BY 'zhangxiao1@123';
```

也不能跨 Schema 来操作，如在 Schema (user99) 操作，具体如下：



语句如下：

```
SELECT * from customer_t1;
```

```
drop TABLE customer_t1;
```

DAS 方式登陆数据库实例实验结束。

2 通过公网连接 GaussDB(for openGauss)

2.1 实验介绍

2.1.1 关于本实验

本实验主要描述如何使用 Data Studio 通过公网连接 GaussDB(for openGauss)实例。

2.1.2 实验目的

掌握使用 Data Studio 通过公网来连接 GaussDB(for openGauss)数据库实例。

2.2 远程连接 GaussDB(for openGauss)

2.2.1 Data Studio 简介

帮助文档：https://support.huaweicloud.com/tg-dws/DWS_DS_19.html

可以通过 Data Studio 完成远程连接 GaussDB(for openGauss)的连接。

Data Studio 是一个集成开发环境（IDE），帮助数据库开发人员便捷地构建应用程序，以图形化界面形式提供数据库关键特性。

数据库开发人员仅需掌握少量的编程知识，即可使用该工具进行数据库对象操作。Data Studio 提供丰富多样的特性，例如：

- 创建和管理数据库对象
- 执行 SQL 语句/脚本
- 编辑和执行 PL/SQL 语句
- 图形化查看执行计划和开销
- 导出表数据等

创建和管理数据库对象包括：

- 数据库
- 模式

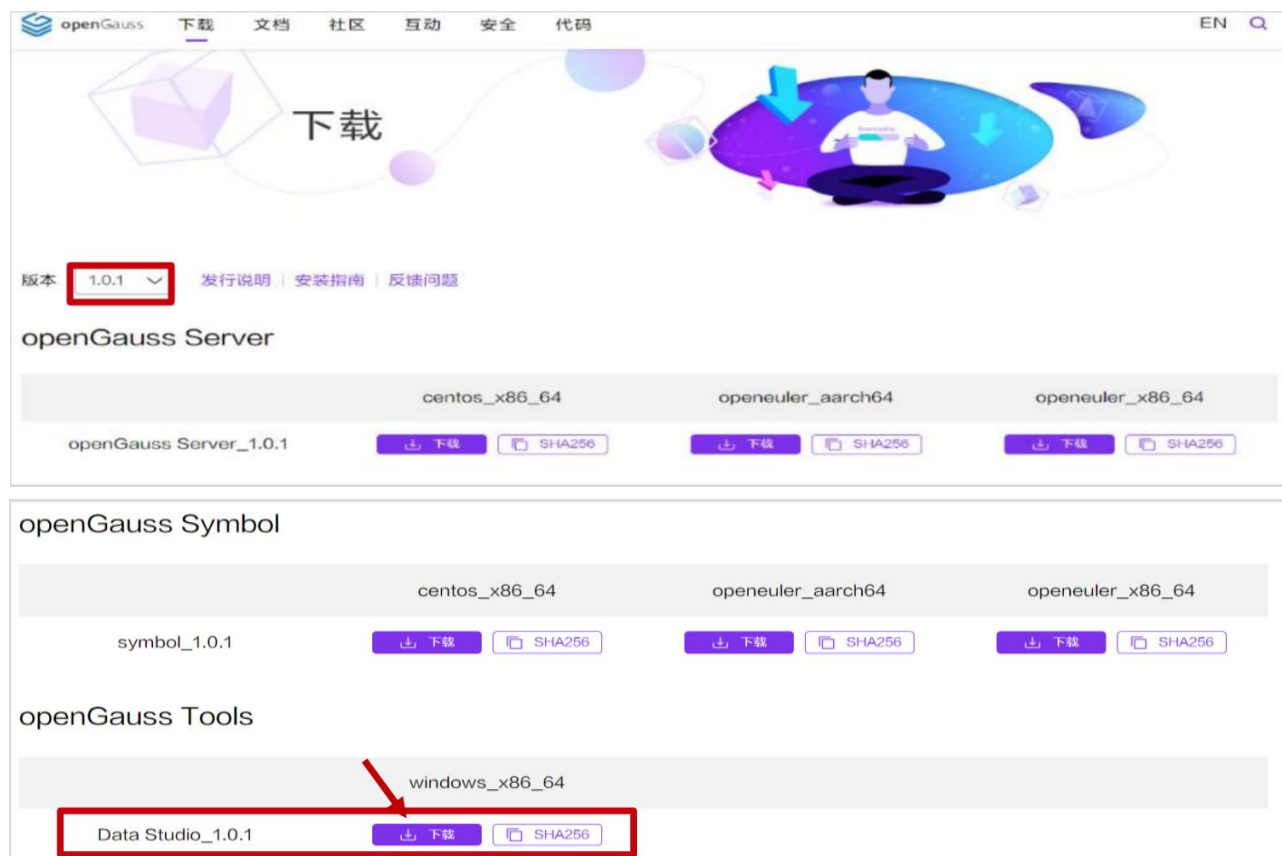
- 函数
- 过程
- 表
- 序列
- 索引
- 视图
- 表空间
- 同义词

Data Studio 还提供 SQL 助手用于在“SQL 终端”和“PL/SQLViewer”中执行各种查询/过程/函数。

2.2.2 Data Studio 安装

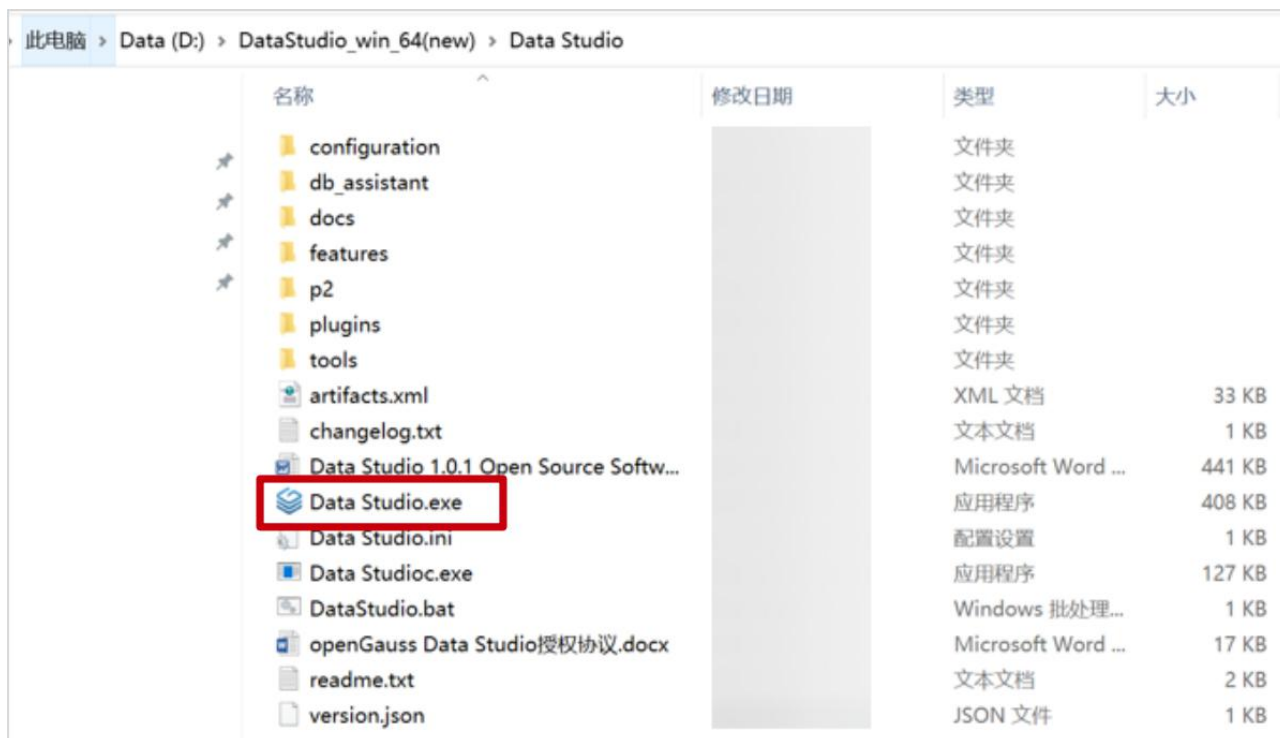
下载工具包：获取参考地址：

https://opengauss.obs.cn-south-1.myhuaweicloud.com/1.0.1/DataStudio_win_64.zip



下载后的文件名为：DataStudio_win_64.zip

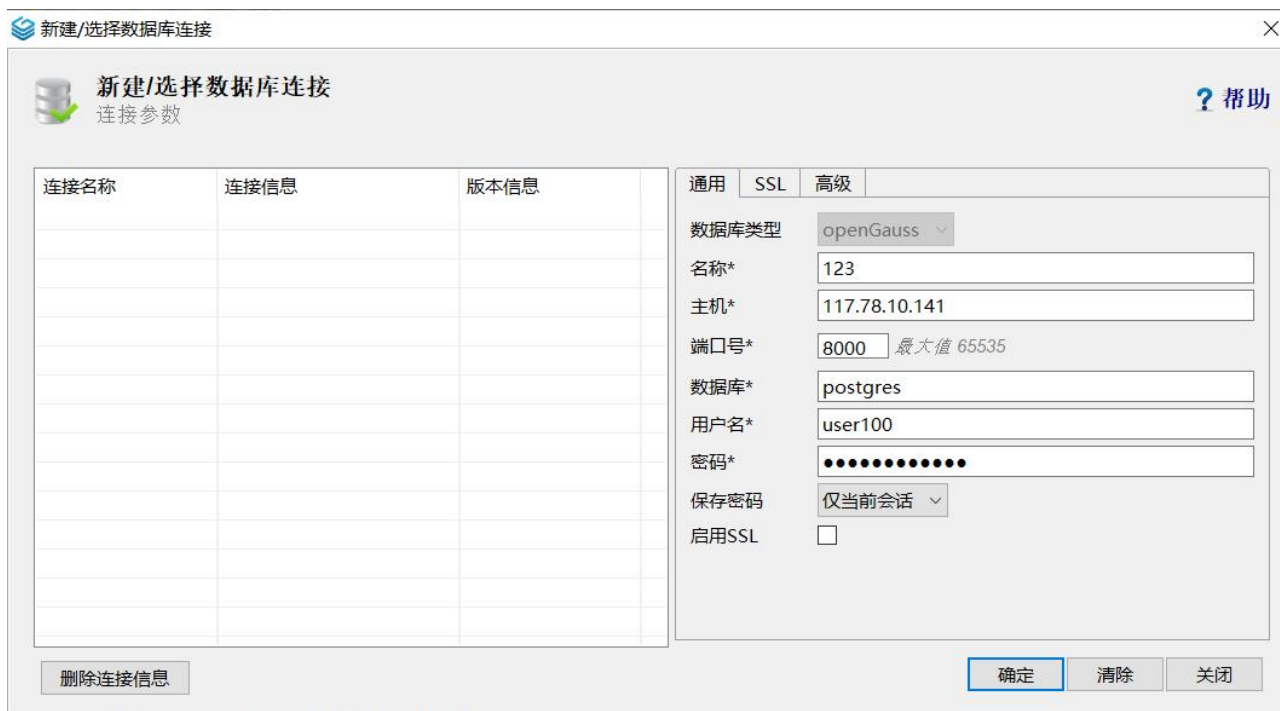
将下载的软件包（DataStudio_win_64.zip）解压到自己指定的位置，比如解压至 D 盘，具体如下：



注：Data Studio 的安装需要电脑安装 JDK，如果电脑没有安装 JDK，请按照附件一的步骤安装和部署 JDK。

定位并双击 Data Studio.exe，启动 Data Studio 客户端，输入相关信息，连接数据库：

注：如果没有弹出窗口，可以在 Data Studio 工具界面上，单击“文件”下的“新建连接”。



名称：自定义

主机：117.78.10.141（绑定数据库服务器的弹性公网 IP）

端口：8000

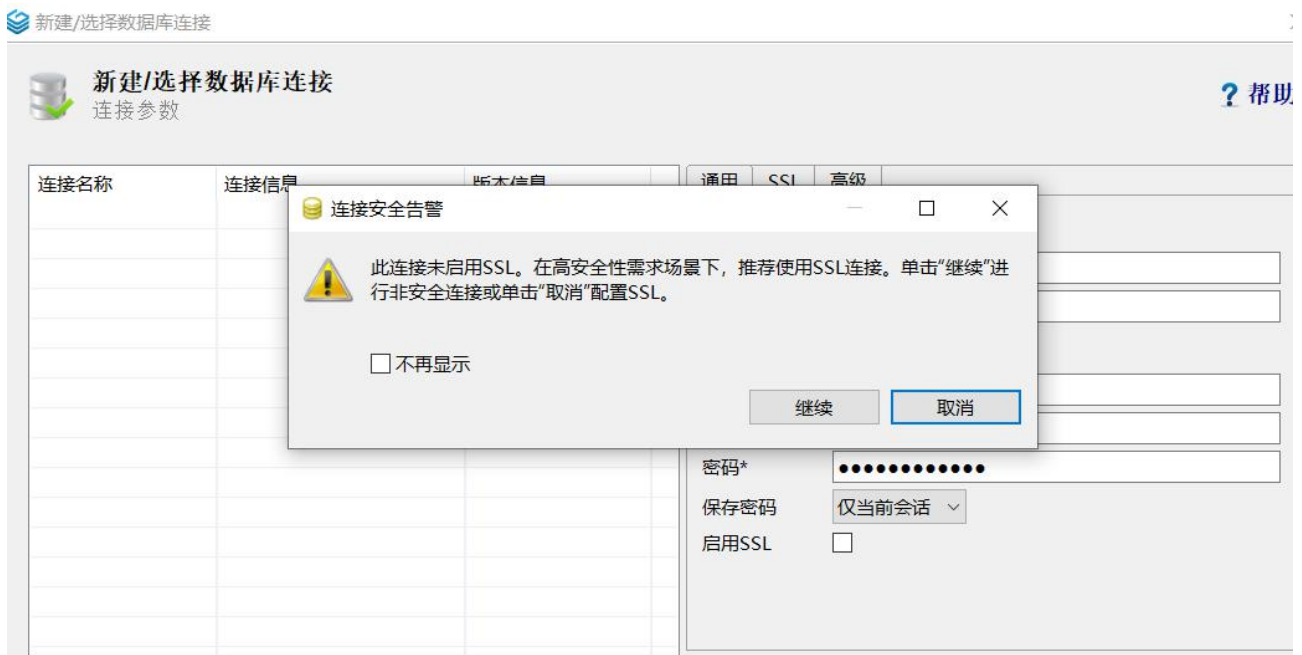
数据库：postgres

用户名：user*（根据自己分配的账号信息填写）

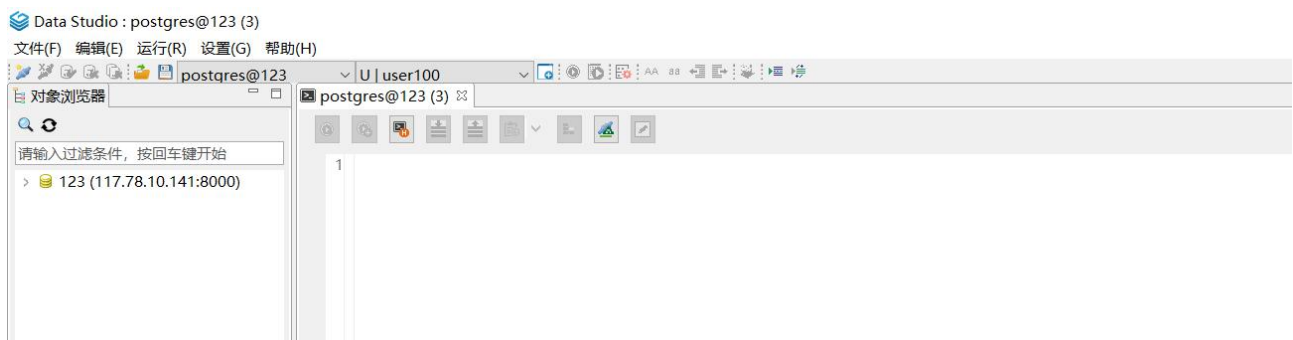
密码：*（根据自己分配的账号信息填写）

启用 SSL：不启用

设置完成后，单击“确定”按钮进行连接，单击“继续”。



连接成功后界面如下：



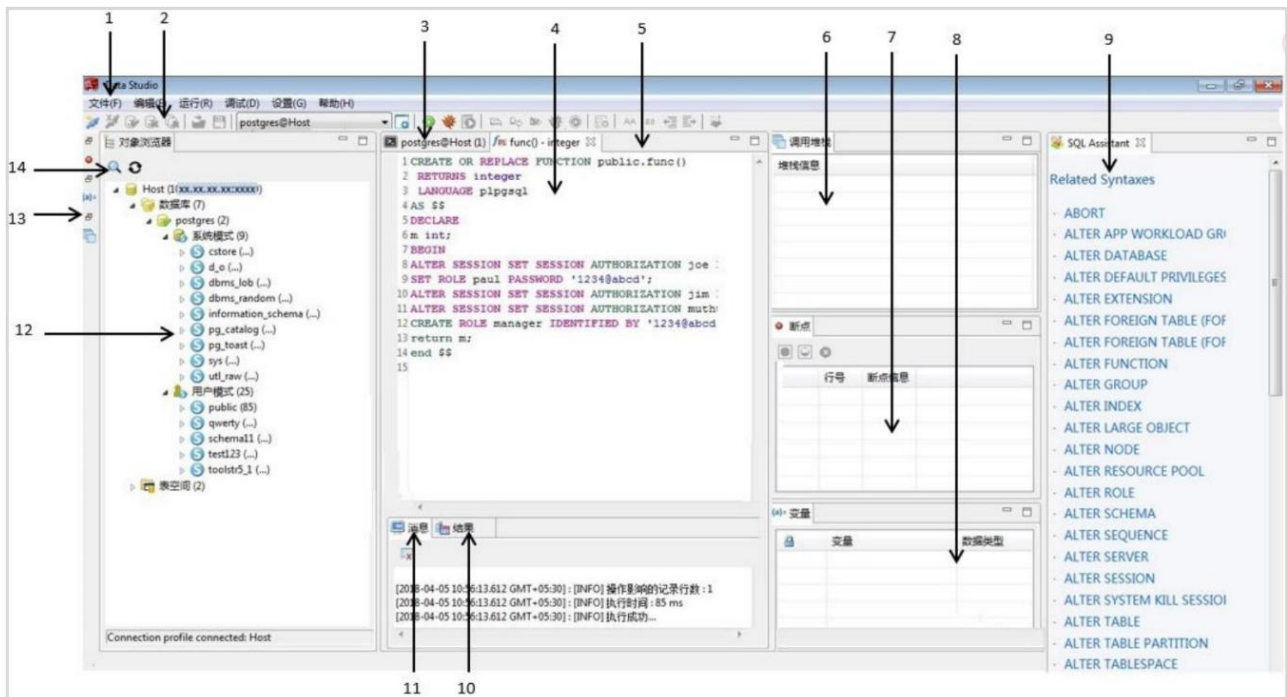
使用 Data Studio 客户端

Data Studio 主界面包括：

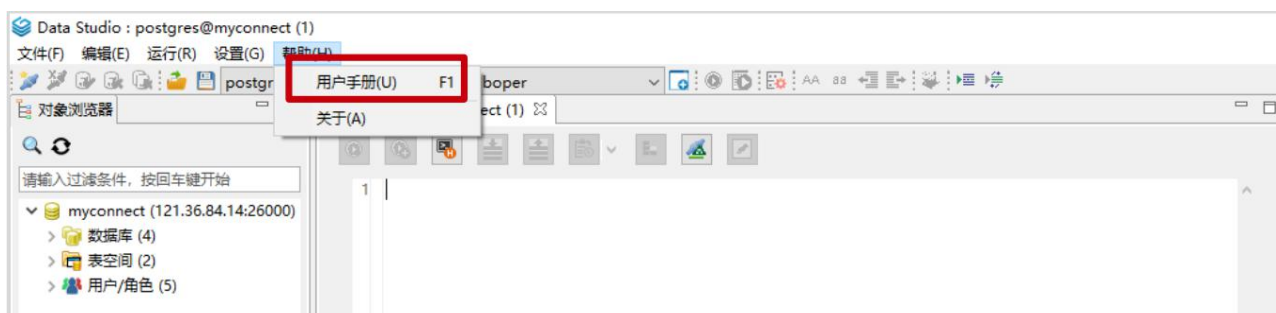
1. 主菜单：提供使用 Data Studio 的基本操作；

2. 工具栏：提供常用操作入口；
3. “SQL 终端”页签：在该窗口，可以执行 SQL 语句和函数/过程；
4. “PL/SQL Viewer”页签：显示函数/过程信息；
5. 编辑区域用于进行编辑操作；
6. “调用堆栈”窗格：显示执行栈；
7. “断点”窗格：显示断点信息；
8. “变量”窗格：显示变量及其变量值；
9. “SQL 助手”页签：显示“SQL 终端”和“PL/SQL Viewer”页签中输入信息的建议或参考；
10. “结果”页签：显示所执行的函数/过程或 SQL 语句的结果；
11. “消息”页签：显示进程输出。显示标准输入、标准输出和标准错误；
12. “对象浏览器”窗格：显示数据库连接的层级树形结构和用户有权访问的相关数据库对象；除公共模式外，所有默认创建的模式均分组在“系统模式”下，用户模式分组在相应数据库的“用户模式”下；
13. “最小化窗口窗格”：用于打开“调用堆栈”和“变量”窗格。该窗格仅在“调用堆栈”、“变量”窗格中的一个或多个窗格最小化时显示。
14. 搜索工具栏：用于在“对象浏览器”窗格中搜索对象。

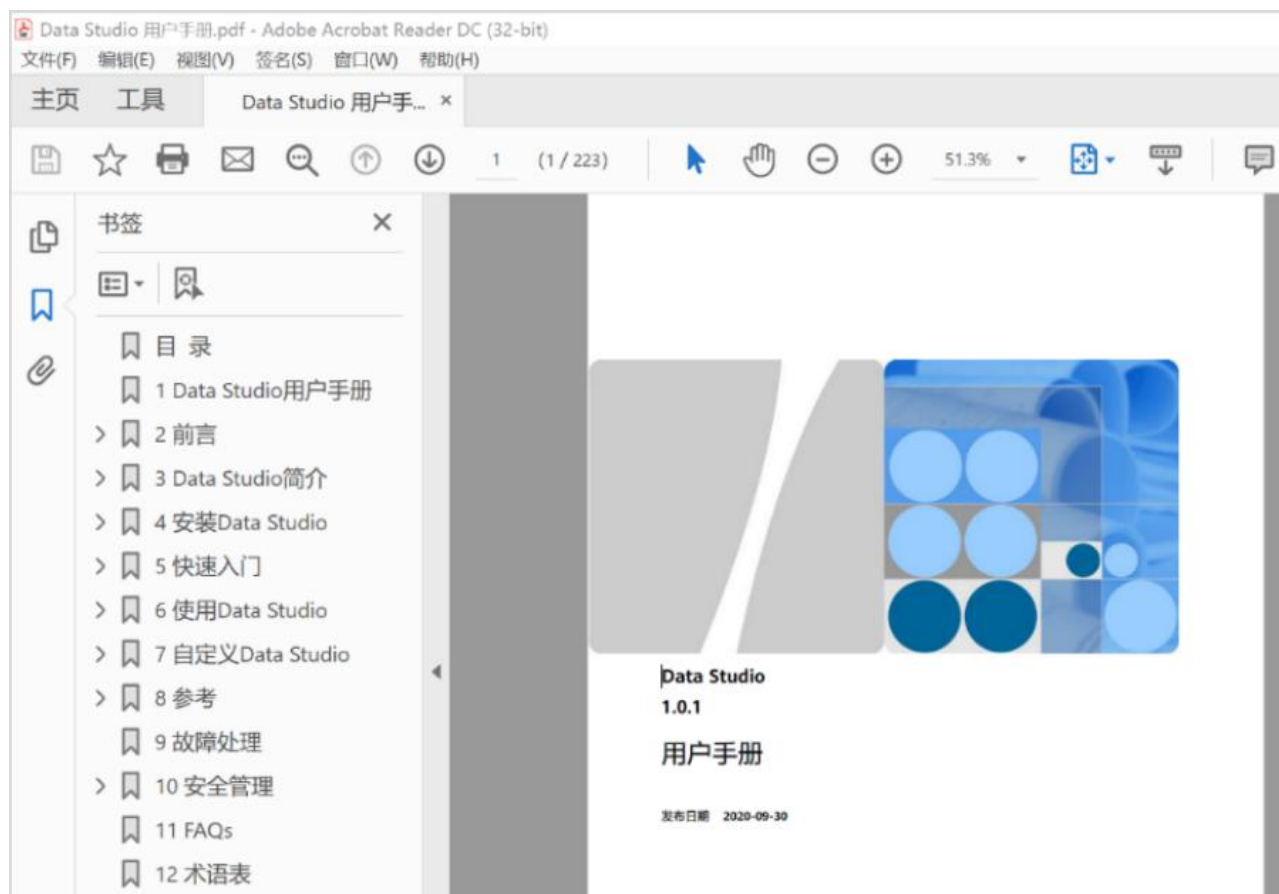
有些项不可见，除非触发特定功能。下图以 openGauss 界面为例说明：



在 Data Studio 主界面的主菜单上单击帮助下的用户手册，具体如下：



单击后即可得到使用手册，如下：



通过公网连接 GaussDB(for openGauss)实验结束。

3

数据库建表及数据导入

3.1 实验介绍

3.1.1 关于本实验

本实验主要描述如何通过：

1) DAS 连接数据库、以 IAM 账号登录数据库，采用 SQL 语句或 Data Studio 图形化界面，创建数据库表；2) 在已经建表基础上，批量导入 tbCell 等全部实验数据；

注：

1、无论是网页端还是 Data Studio，都不支持导入数据的时候自动建列，因此我们将列较为复杂的表的创建语句已给出（预先设置的数据类型均为 char,同学们在实验时可根据实际情况修改），见附录二。列较少的表请同学们自行创建。

2、通过软件 Dbeaver 无需手动创建数据库表，可以直接导入数据，自动匹配数据类型。（*表示实验方法自行选用）

3.1.2 实验目的

掌握在 IAM 登录、IP 登录两种方式下，建表及导入实验数据的方法。

3.2 *通过 Dbeaver 自动导入数据（自行选用）

Dbeaver 是一个跨平台的数据库开发者工具，包括 SQL 编程，数据库管理和分析。它支持任意适配 JDBC 驱动的数据库系统。同时该工具也支持一些非 JDBC 的数据源，如 MongoDB, Cassandra, Redis, DynamoDB 等。

- 该工具提供了许多强大的特性，诸如元数据编辑器、SQL 编辑器、富文本数据编辑器、ERD、数据导入/导出/迁移，SQL 执行计划等
- 该工具基于 eclipse 平台开发
- 适配的数据库有 MySQL/MariaDB, PostgreSQL, Greenplum, Oracle, DB2 LUW, Exasol, SQL Server, Sybase/SAP ASE, SQLite, Firebird, H2, HSQLDB, Derby, Teradata, Vertica, Netezza, Informix 等

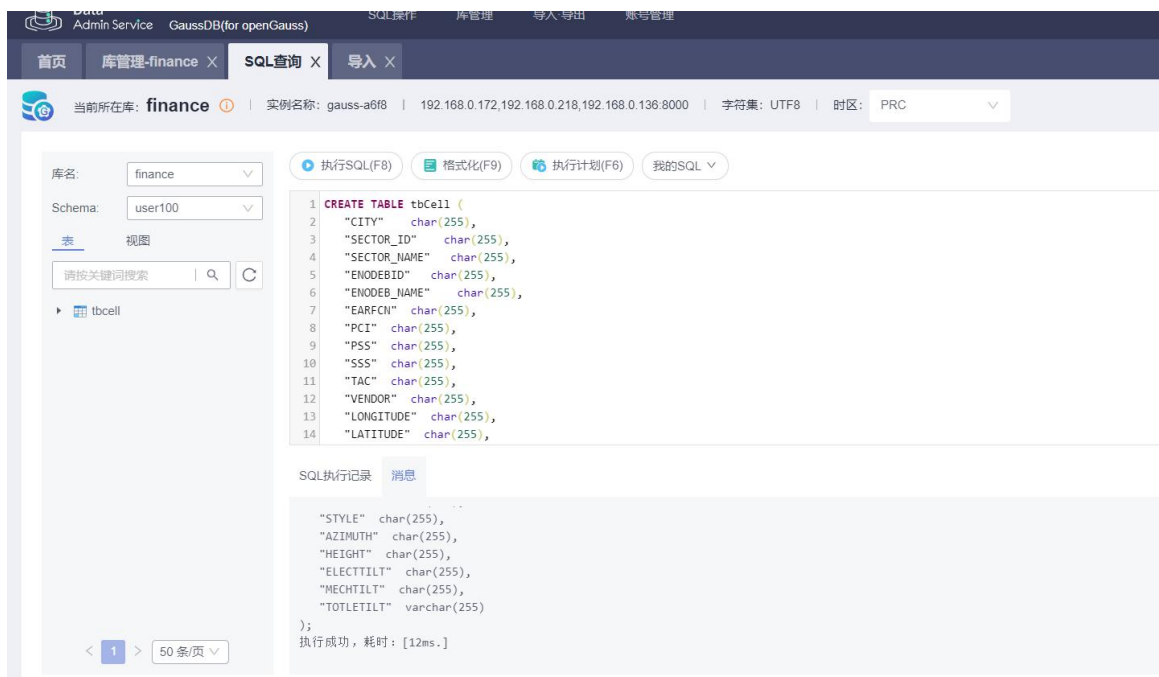
通过 Dbeaver 无须手动输入列，可自动导入数据，匹配数据类型。

此处不再详细讲解用法，附上华为官方教程：
https://opengauss.org/zh/blogs/blogs.html?post/justbk/2020-10-30_dbeaver_for_opengauss/

3.3 DAS/IAM 登录方式下的建表及数据导入

3.3.1 数据库建表

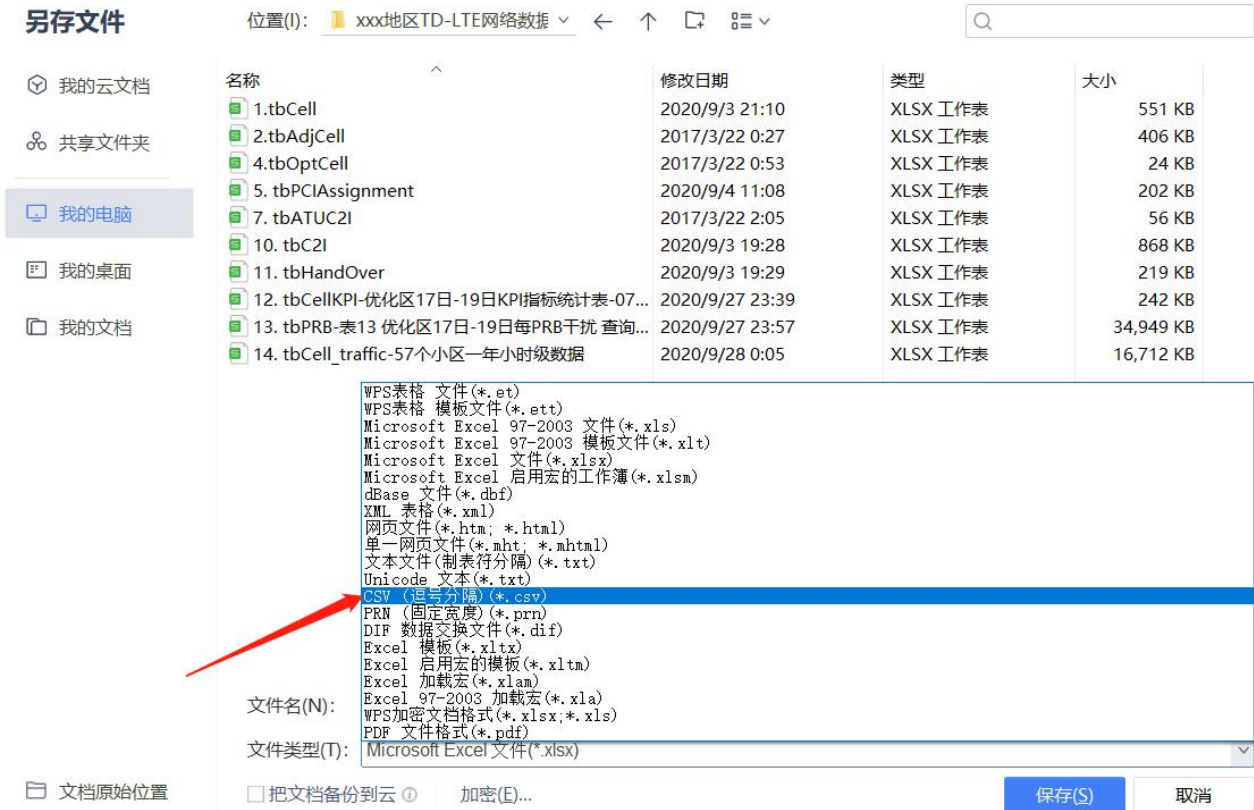
在网页中创建所需表格，执行 SQL 建表语句创建表，以创建表 tbCell 为例：



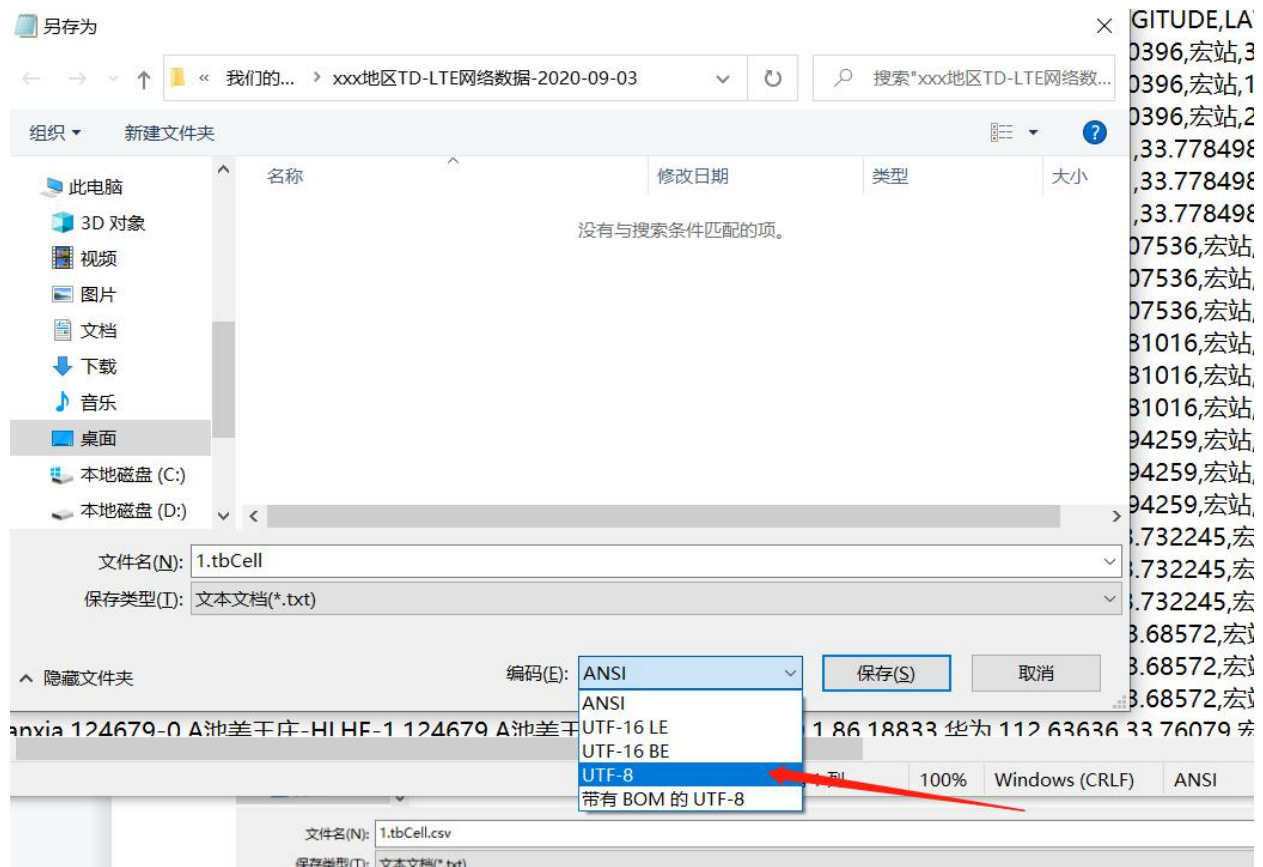
3.3.2 批量数据导入

步骤 1 首先将 excel 格式转换为 csv

用 excel 打开 tbcell 表格，点击文件---另存为，选择另存为 csv 格式

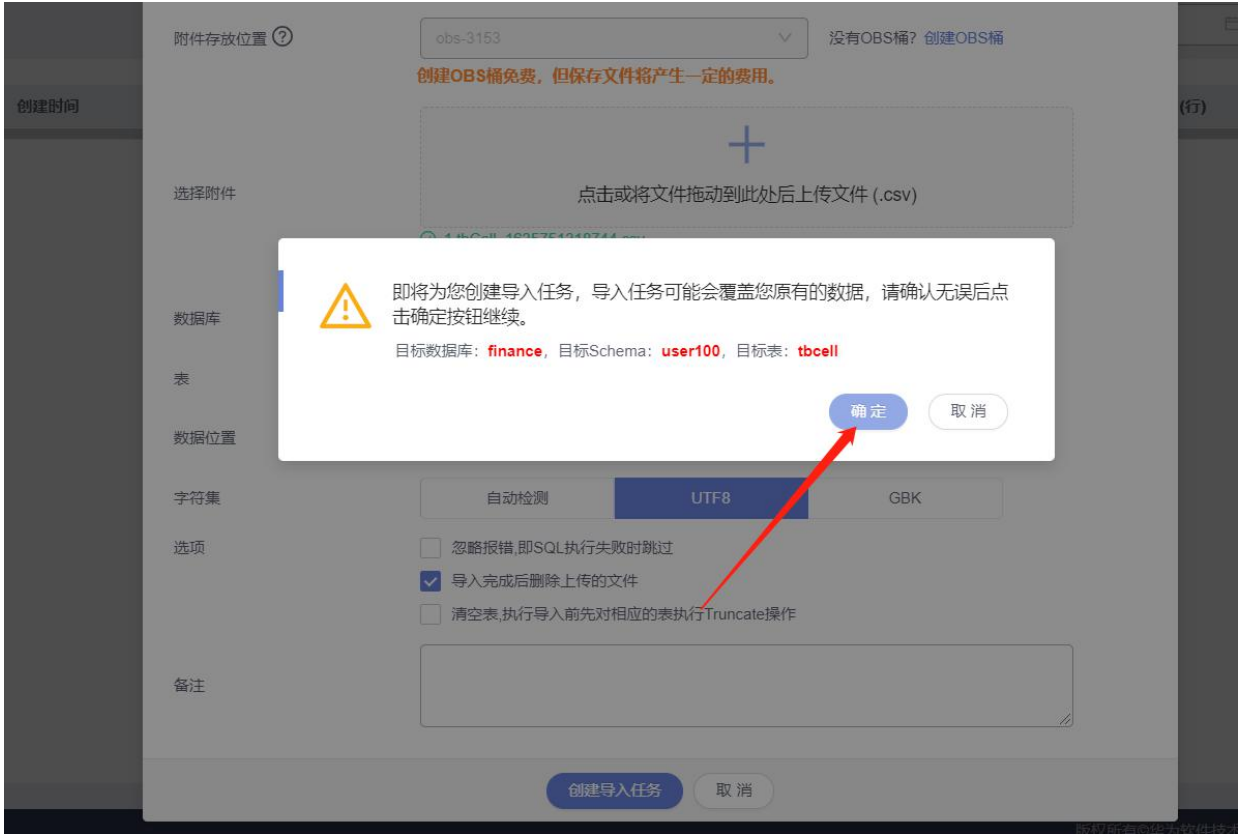


用记事本打开刚刚保存的 csv 文件，点击文件---另存为，选择编码 **UTF-8(没有 BOM)**，点击保存



步骤 2 导入数据





新建任务

导入类型

sqlCSV

文件来源

上传文件从OBS中选择

附件存放位置 ?

obs-3153

没有OBS桶? 创建OBS桶

选择附件

+

点击或将文件拖动到此处后上传文件 (.csv)

1.tbCell_1635751318744.csv

最大不能超过1GB, 且只能上传一个附件

数据库

finance

表

"user100"."tbcell"

数据位置

第1行为属性第1行为数据

字符集

自动检测UTF8GBK

选项

☐ 忽略报错,即SQL执行失败时跳过

☒ 导入完成后删除上传的文件

☐ 清空表,执行导入前先对相应的表执行Truncate操作

备注

导入数据成功

Data Admin Service GaussDB(for openGauss)

SQL操作库管理导入/导出账号管理

user100/user1

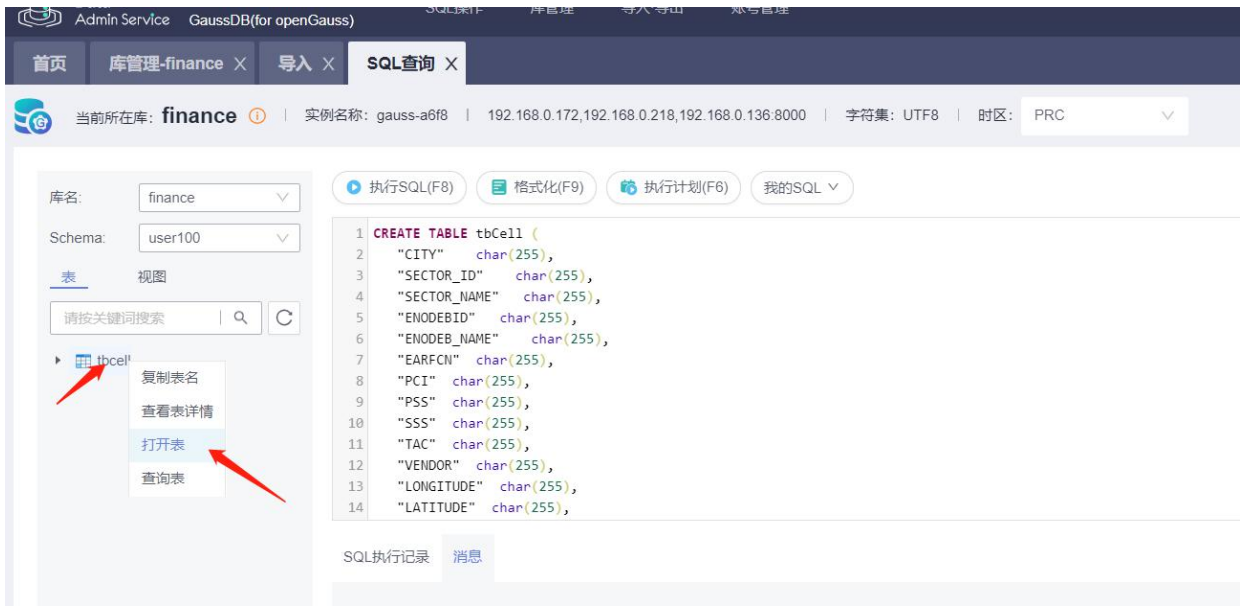
首页库管理-financeSQL查询导入

新建任务删除任务

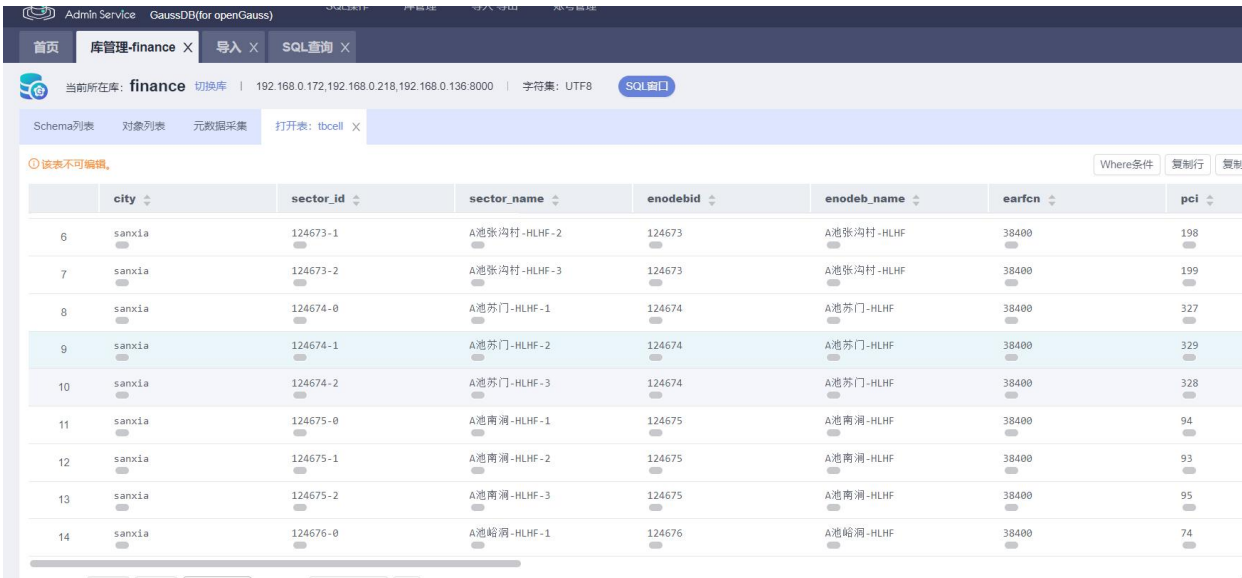
时间范围: 任务创建日期起于任务创建日期止于请输入检索条件

<input type="checkbox"/>	任务ID	创建时间	导入文件类型	库表	文件名	任务状态	执行时间	导入成功(行)	是否忽略报错	进度
<input type="checkbox"/>	69aafd972cf74cd9aaf d972cf75cd9b5	2021-11-01 15:24:08	CSV	目标库: finance, 目标Schema: user100, 目标表: tbcell	1.tbCell_1635751318744.csv	已完成	2秒	5504	否	1

查看表中的数据，右键单击表 tbcell，单击打开表。



可以看到数据导入成功。



至此从本地文件导入到数据库已完成。

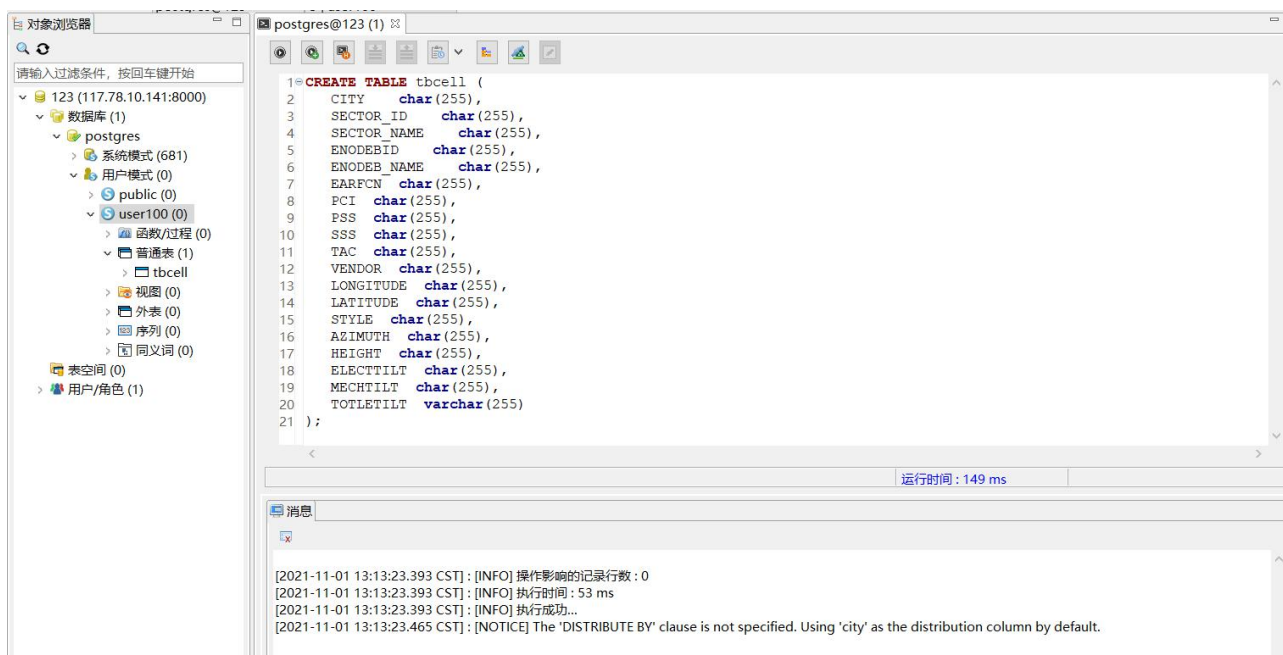
3.4 公网/IP 登录方式下的建表及数据导入

在 IP 登录方式下，数据库用户通过 Data Studio 可以像 DAS/IAM 登录方式下，通过建表和批量数据导入两个步骤，将数据库实例中导入全部实验数据，供后续实验使用。

3.4.1 建表/导入

以 tbCell 表的导入为例，建表/导入过程包括以下各步骤。

步骤 1 在 Data Studio 中新建所需表

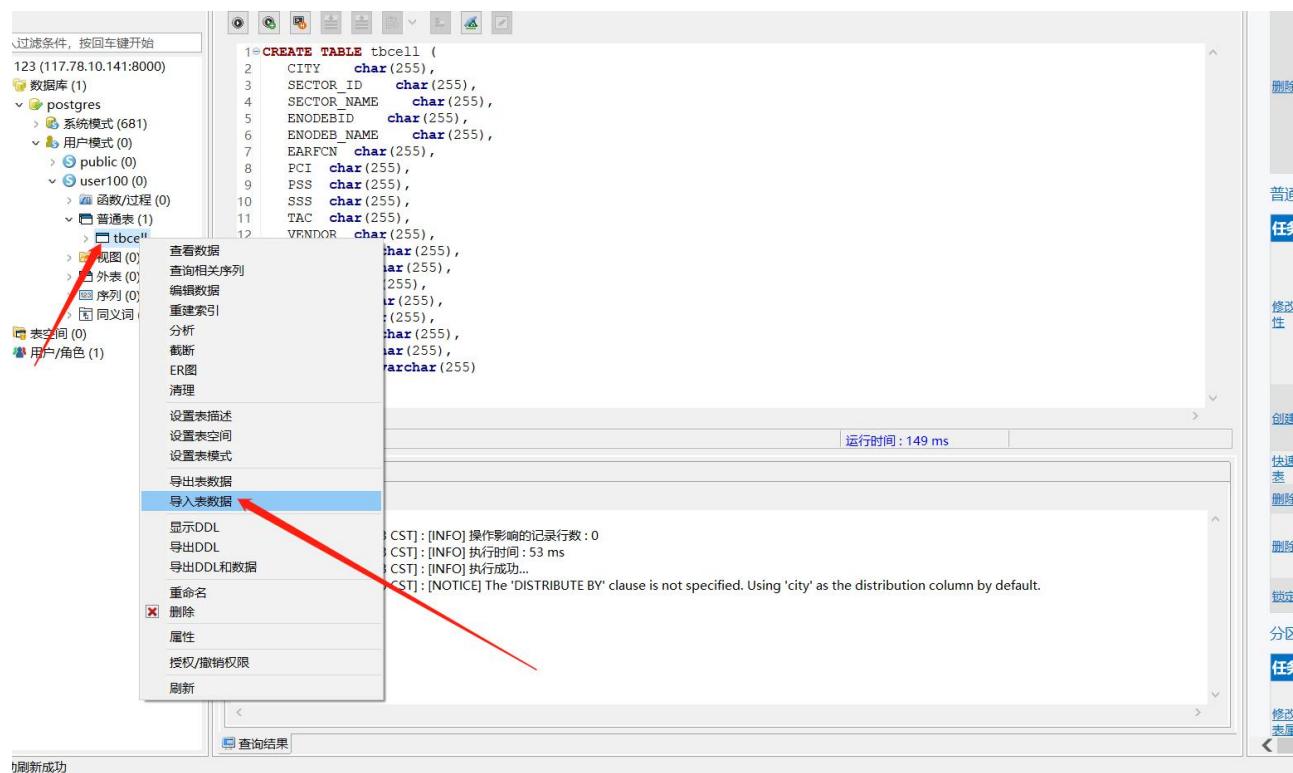


步骤 2 将 excel 格式转换为 csv(同 3.3.2 步骤一)

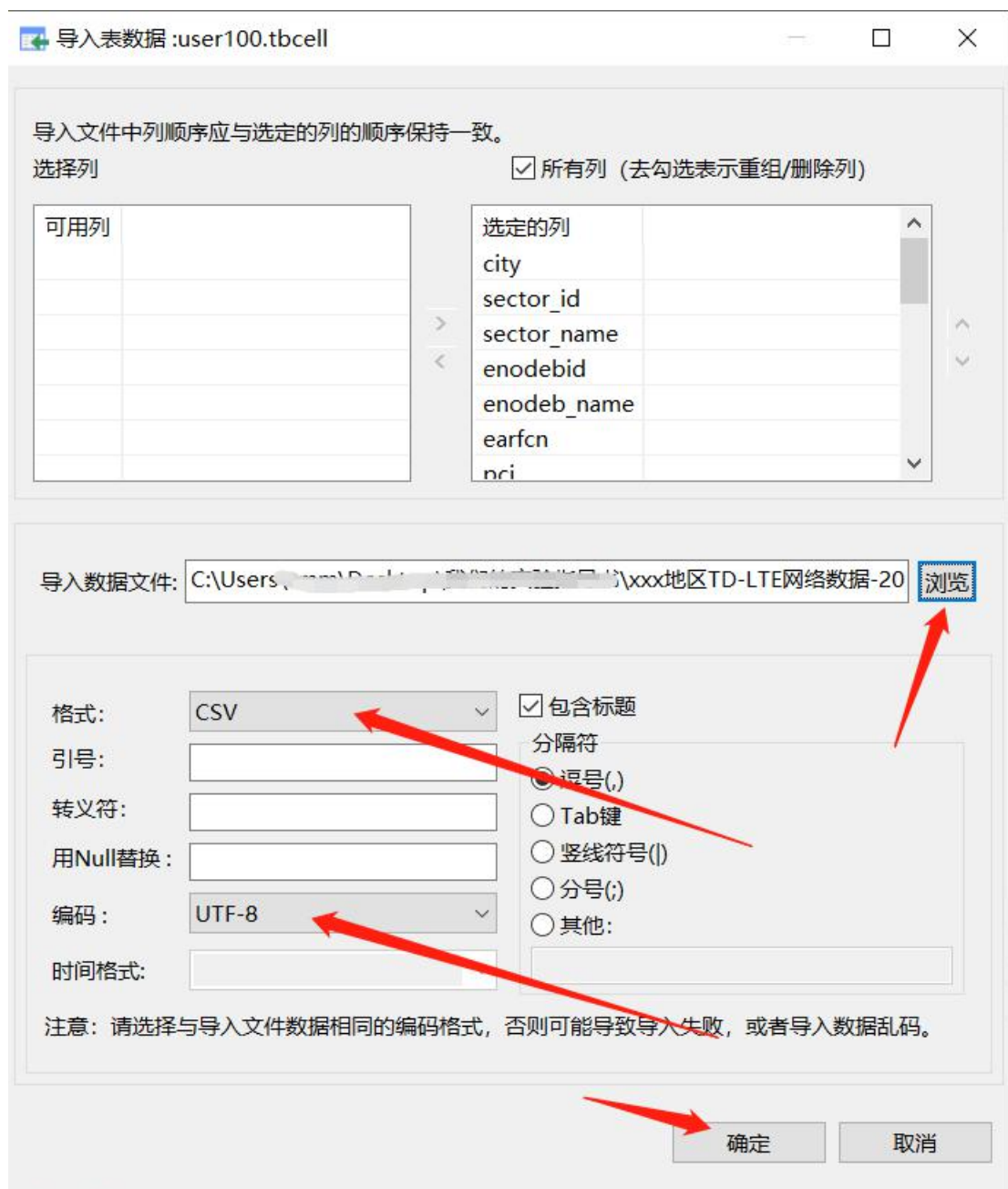
略

步骤 3 导入表数据

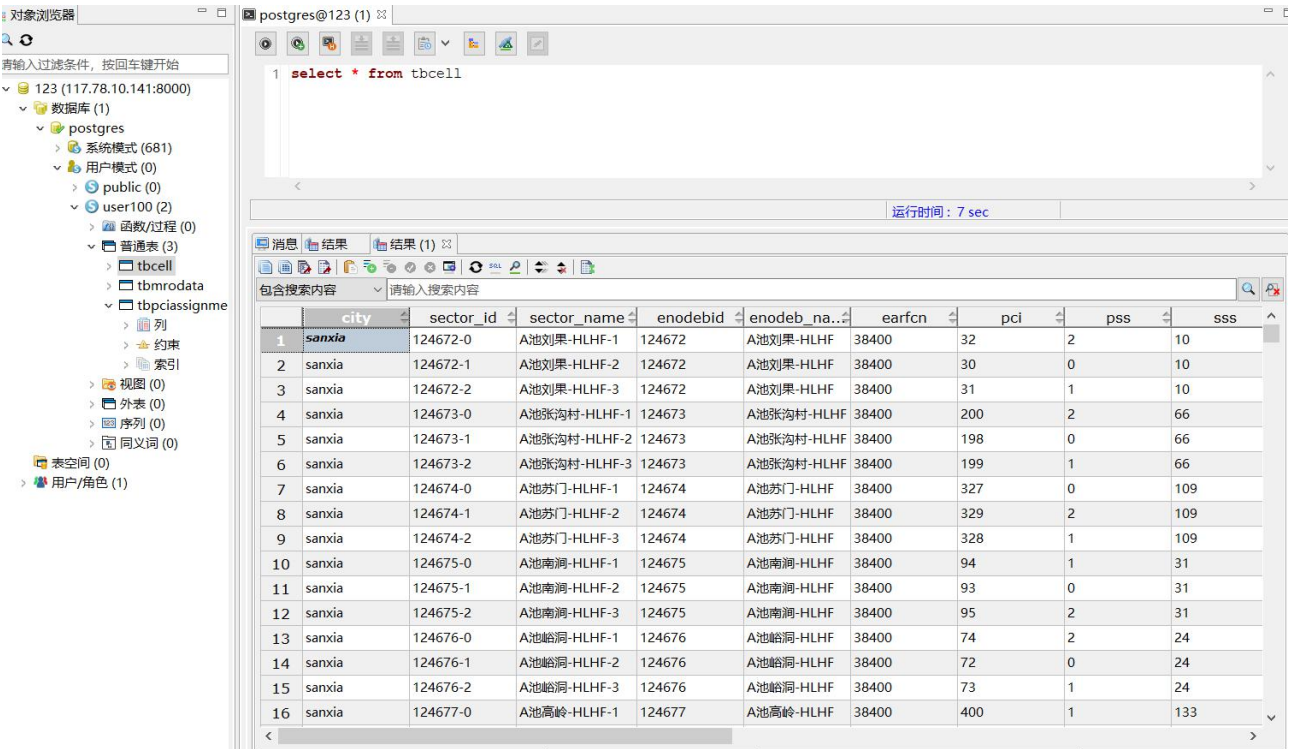
右键单击创建好的表，单击“导入表数据”。



选择相应的表进行导入，注意格式选择。



步骤 4 打开数据库表, 观察导入的数据。



导入表数据成功

4 附录

4.1 附录一：安装和配置 JDK

注：参考博客 <https://blog.csdn.net/u013129932/article/details/114275846>

4.1.1 下载并安装 JDK

步骤 1 下载 JDK

<https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>

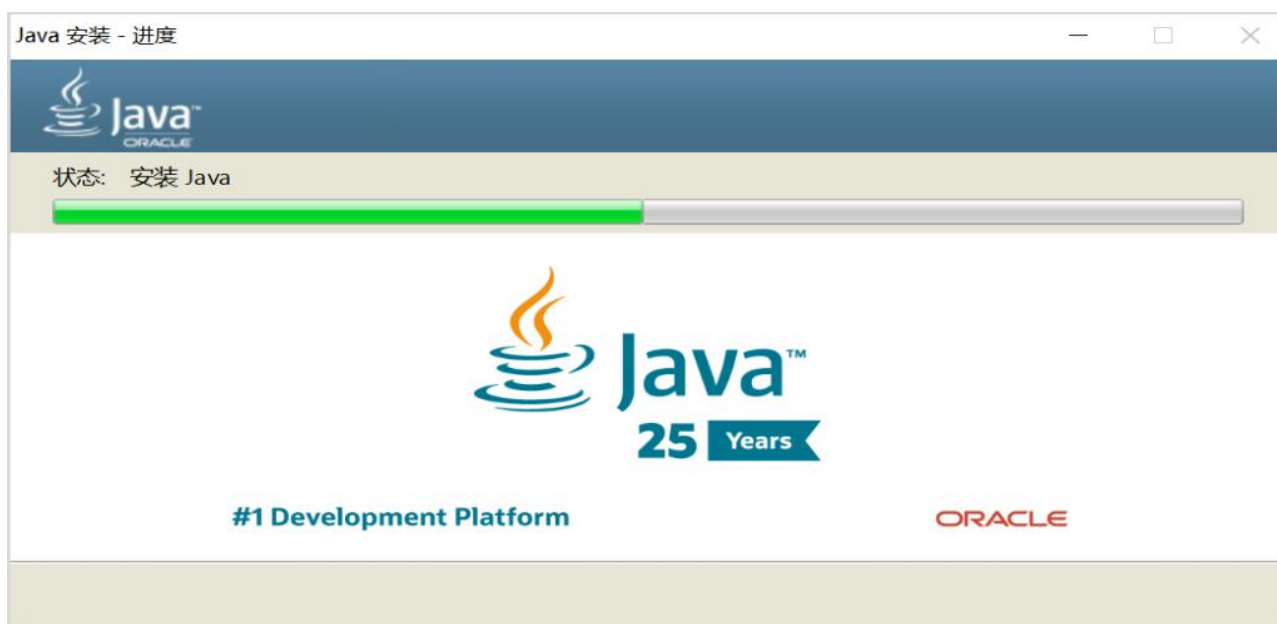
Solaris x64	92.47 MB	jdk-8u261-solaris-x64.tar.gz
Windows x86	154.52 MB	jdk-8u261-windows-i586.exe
Windows x64	166.28 MB	jdk-8u261-windows-x64.exe



jdk-8u261-windows-x64.exe

步骤 2 双击 jdk-8u261-windows-x64.exe 进行安装。

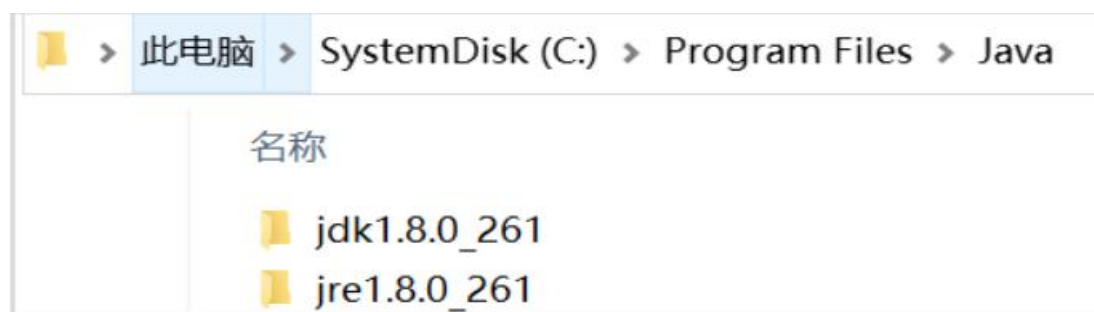
默认设置即可，出现安装进度



如下显示表示安装成功：



步骤 3 查看安装目录。



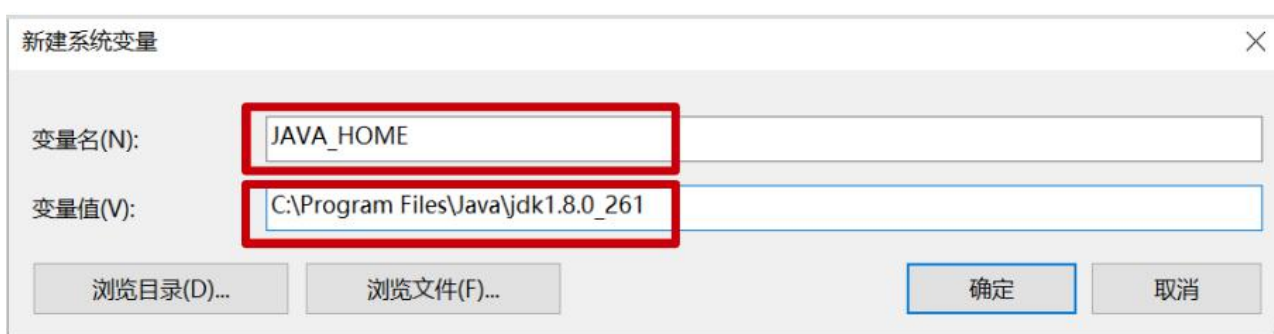
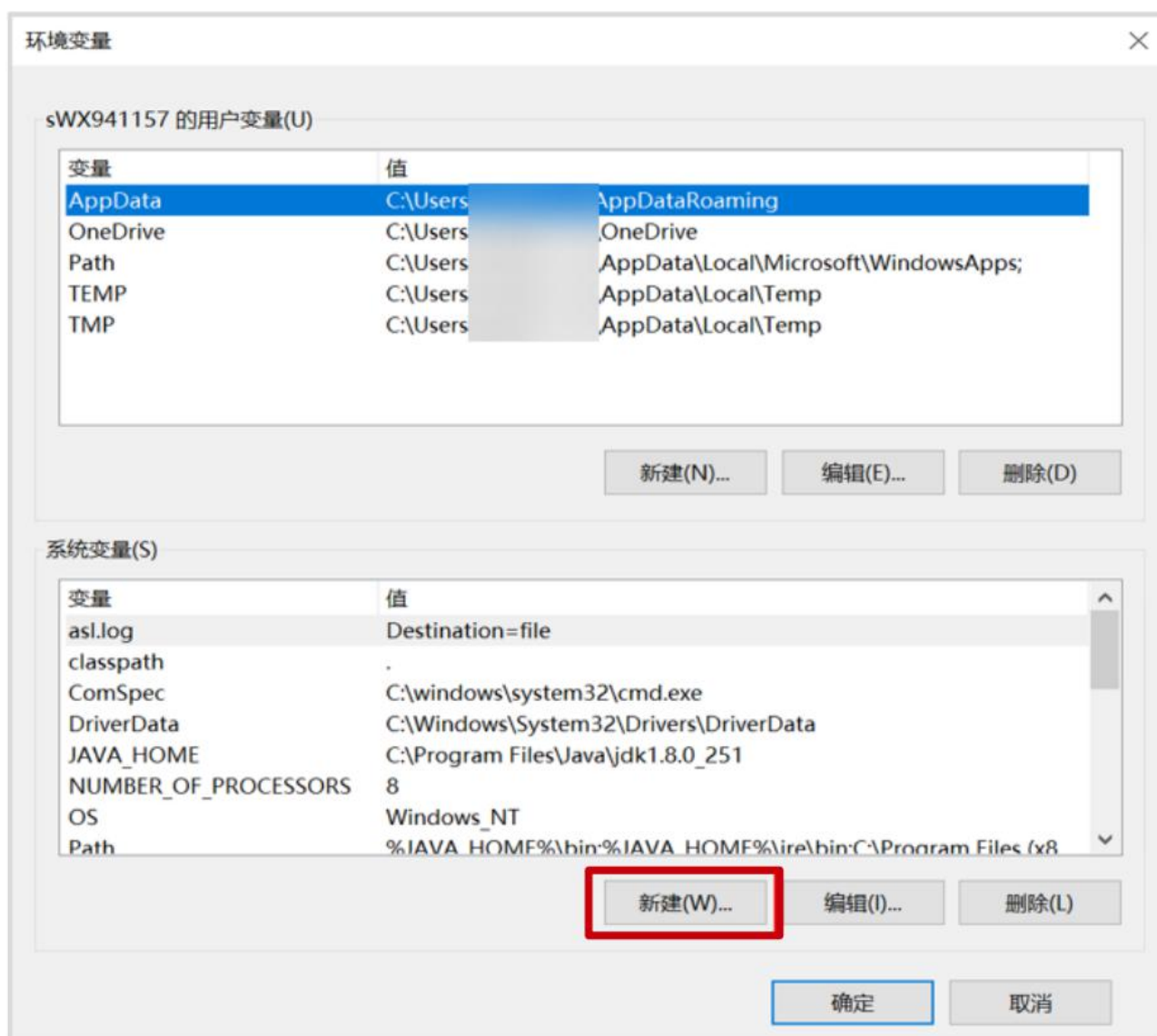
4.1.2 配置 JDK 环境变量

步骤 1 右击“此电脑”选择“属性”，单击“高级系统设置”。



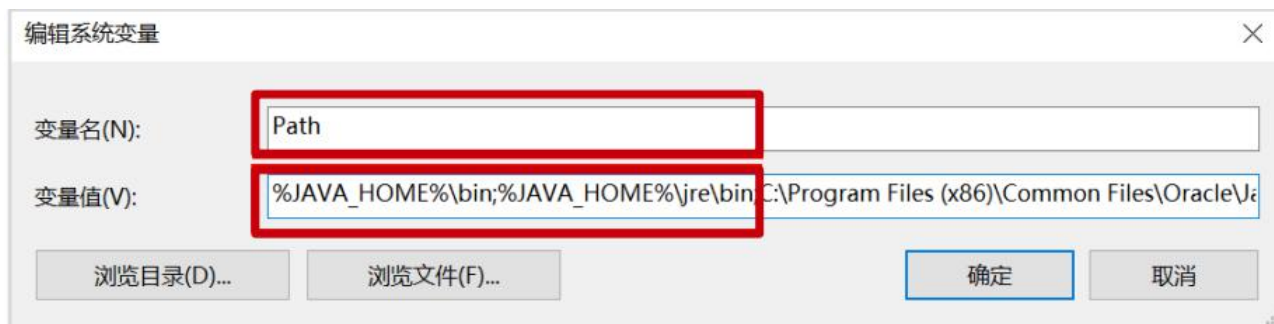


步骤 2 单击“环境变量”，新建系统变量“JAVA_HOME”，输入 JDK 安装目录。



“C:\Program Files\Java\jdk1.8.0_261” 为 JDK 安装目录。

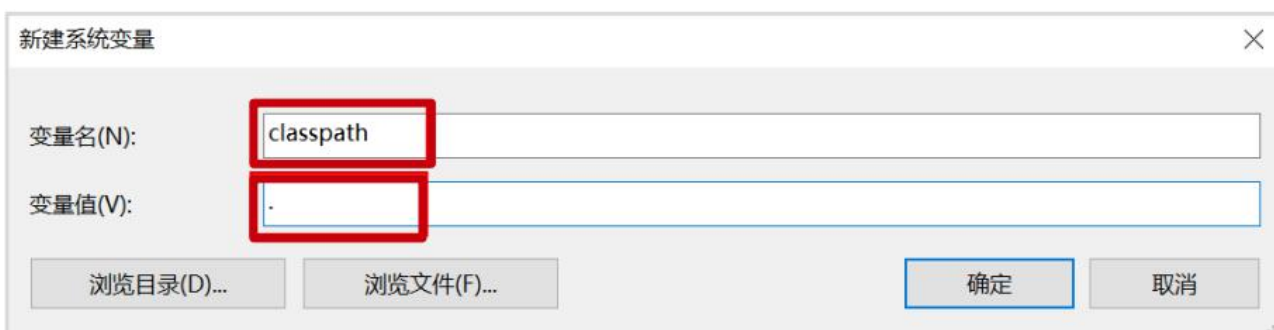
步骤 3 编辑系统变量“path”。



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 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_261"  
Java(TM) SE Runtime Environment (build 1.8.0_261-b12)  
Java HotSpot(TM) 64-Bit Server VM (build 25.261-b12, mixed mode)
```

4.2 附录二：创建表语句

4.2.1 表 tbcell

```
CREATE TABLE tbCell (  
    "CITY" char(255),
```

```
"SECTOR_ID" char(255),  
"SECTOR_NAME" char(255),  
"ENODEBID" char(255),  
"ENODEB_NAME" char(255),  
"EARFCN" char(255),  
"PCI" char(255),  
"PSS" char(255),  
"SSS" char(255),  
"TAC" char(255),  
"VENDOR" char(255),  
"LONGITUDE" char(255),  
"LATITUDE" char(255),  
"STYLE" char(255),  
"AZIMUTH" char(255),  
"HEIGHT" char(255),  
"ELECTTILT" char(255),  
"MECHTILT" char(255),  
"TOTLETILT" varchar(255)  
);
```

4.2.2 表 tbPCIAssignment

```
CREATE TABLE tbpciassignment (  
    "f1" varchar(255) ,  
    "EARFCN" varchar(255) ,  
    "SECTOR_ID" varchar(255) ,  
    "SECTOR_NAME" varchar(255) ,  
    "ENODEB_ID" varchar(255) ,  
    "PCI" varchar(255) ,  
    "PSS" varchar(255) ,  
    "SSS" varchar(255) ,  
    "LONGITUDE" varchar(255) ,  
    "LATITUDE" varchar(255) ,
```

```
"STYLE" varchar(255) ,  
"OPT_DATETIME" varchar(255)  
)
```

4.2.3 表 tbATUData

```
CREATE TABLE tbatudata (  
    "seq" varchar(255) ,  
    "FileName" varchar(255) ,  
    "Time" varchar(255) ,  
    "Longitude" varchar(255) ,  
    "Latitude" varchar(255) ,  
    "CellID" varchar(255) ,  
    "TAC" varchar(255) ,  
    "EARFCN" varchar(255) ,  
    "PCI" varchar(255) ,  
    "RSRP" varchar(255) ,  
    "RS SINR" varchar(255) ,  
    "NCell_ID_1" varchar(255) ,  
    "NCell_EARFCN_1" varchar(255) ,  
    "NCell_PCI_1" varchar(255) ,  
    "NCell_RSRP_1" varchar(255) ,  
    "NCell_ID_2" varchar(255) ,  
    "NCell_EARFCN_2" varchar(255) ,  
    "NCell_PCI_2" varchar(255) ,  
    "NCell_RSRP_2" varchar(255) ,  
    "NCell_ID_3" varchar(255) ,  
    "NCell_EARFCN_3" varchar(255) ,  
    "NCell_PCI_3" varchar(255) ,  
    "NCell_RSRP_3" varchar(255) ,  
    "NCell_ID_4" varchar(255) ,  
    "NCell_EARFCN_4" varchar(255) ,  
    "NCell_PCI_4" varchar(255) ,
```

```
"NCell_RSRP_4" varchar(255) ,  
"NCell_ID_5" varchar(255) ,  
"NCell_EARFCN_5" varchar(255) ,  
"NCell_PCI_5" varchar(255) ,  
"NCell_RSRP_5" varchar(255) ,  
"NCell_ID_6" varchar(255) ,  
"NCell_EARFCN_6" varchar(255) ,  
"NCell_PCI_6" varchar(255) ,  
"NCell_RSRP_6" varchar(255)  
);
```

4.2.4 表 tbMROData

```
CREATE TABLE tbMROData (  
    "TimeStamp" varchar(255) ,  
    "ServingSector" varchar(255) ,  
    "InterferingSector" varchar(255) ,  
    "LteScRSRP" varchar(255) ,  
    "LteNcRSRP" varchar(255) ,  
    "LteNcEarfcn" varchar(255) ,  
    "LteNcPci" varchar(255)  
);
```

4.2.5 表 tbC2I

```
CREATE TABLE tbc2i (  
    "CITY" varchar(255) ,  
    "SCell" varchar(255) ,  
    "NCell" varchar(255) ,  
    "PrC2I9" varchar(255) ,  
    "C2I_Mean" varchar(255) ,  
    "std" varchar(255) ,  
    "SampleCount" varchar(255) ,  
    "WeightedC2I" varchar(255)  
)
```