**《数据库系统原理》**

**课程实验报告**



**学院：计算机（国家示范性软件）学院**

**班级：** 2021211304

**姓名：** **张梓良 杨晨 苗雨**

**学号：**2021212484 2021212171 2021212492

目录

[一、实验目的 2](#_Toc154746429)

[二、实验平台及环境 2](#_Toc154746430)

[三、实验内容及要求 2](#_Toc154746431)

[1实验内容 2](#_Toc154746432)

[2实验要求 2](#_Toc154746433)

[四、实验步骤 3](#_Toc154746434)

[1环境配置 3](#_Toc154746435)

[2.编写并执行Java程序访问数据库 4](#_Toc154746436)

[五、实验总结 7](#_Toc154746437)

# 一、实验目的

1. 通过编写数据库应用程序，培养数据库应用程序开发能力。
2. 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。
3. 熟悉动态ESQL，使用数据接口访问数据库

# 二、实验平台及环境

1. 实验平台：
   1. GaussDB(for openGauss)云数据库
   2. ip：116.205.151.208
   3. 数据库：finance
2. 实验环境：IntelliJ IDEA 2022.3.2
3. 编程语言：Java 19.0.2

# 三、实验内容及要求

## 1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。

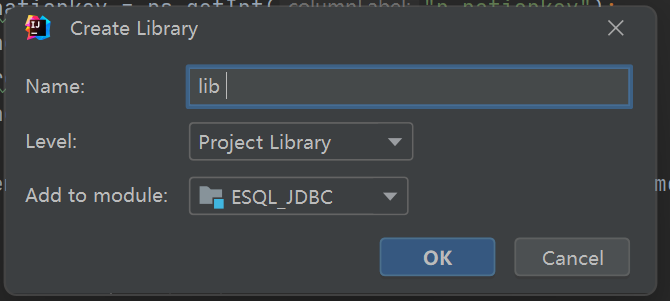
## 2实验要求

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。
   1. 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。

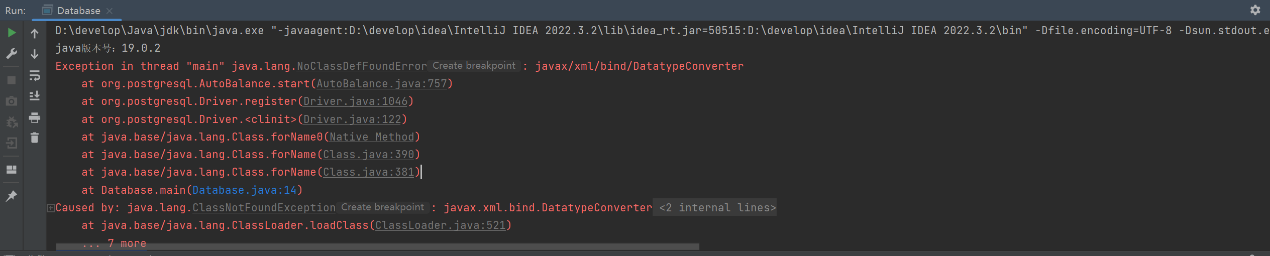
# 四、实验步骤

## 1环境配置

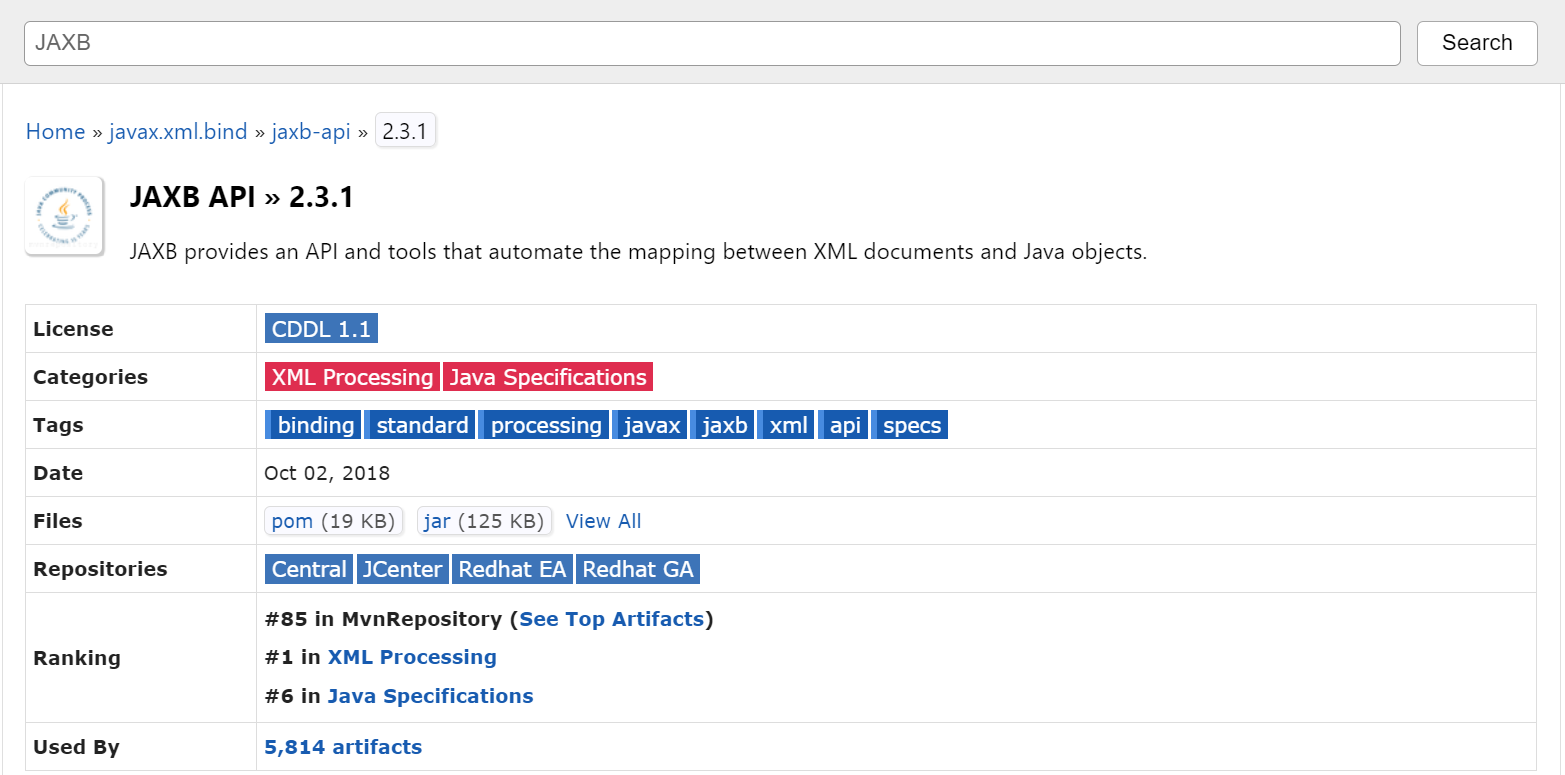
（1）选用应用编程接口JDBC，根据实验指导书下载驱动包解压，得到gsjdbc4.jar，放到项目中的自建文件夹lib下，并将该文件夹添加为Project library.



（2）由于高版本Java EE不自带JAXB包，需要自行下载并添加到lib中，否则运行实例程序会有如下报错:



JAXB包安装：网址<https://mvnrepository.com/>，搜索JAXB，选择jar File直接下载，之后放入建好的Lib文件夹即可。



## 2.编写并执行Java程序访问数据库

（1）数据库连接URL及用户名和密码

static final String *DB\_URL* = "jdbc:postgresql://116.205.151.208:8000/finance?ApplicationName=app1";  
static final String *USER* = "user202308";

（2）注册 JDBC 驱动

Class.*forName*(*JDBC\_DRIVER*);

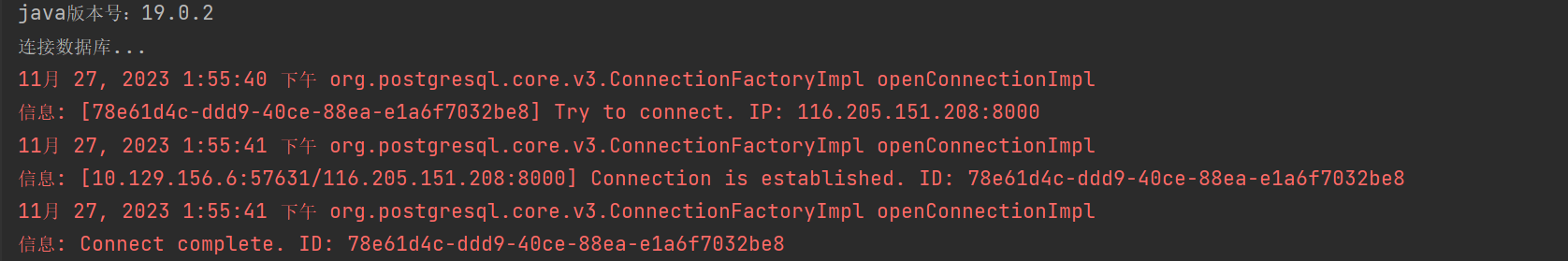
（3）连接数据库

conn = DriverManager.*getConnection*(*DB\_URL*,*USER*,*PASS*);

（4）实例化statement对象

stmt = conn.createStatement();

执行结果：成功连接数据库并且实例化对象：



（5）查询

Sql语句：select n\_nationkey ,n\_name, n\_regionkey, n\_comment from nation

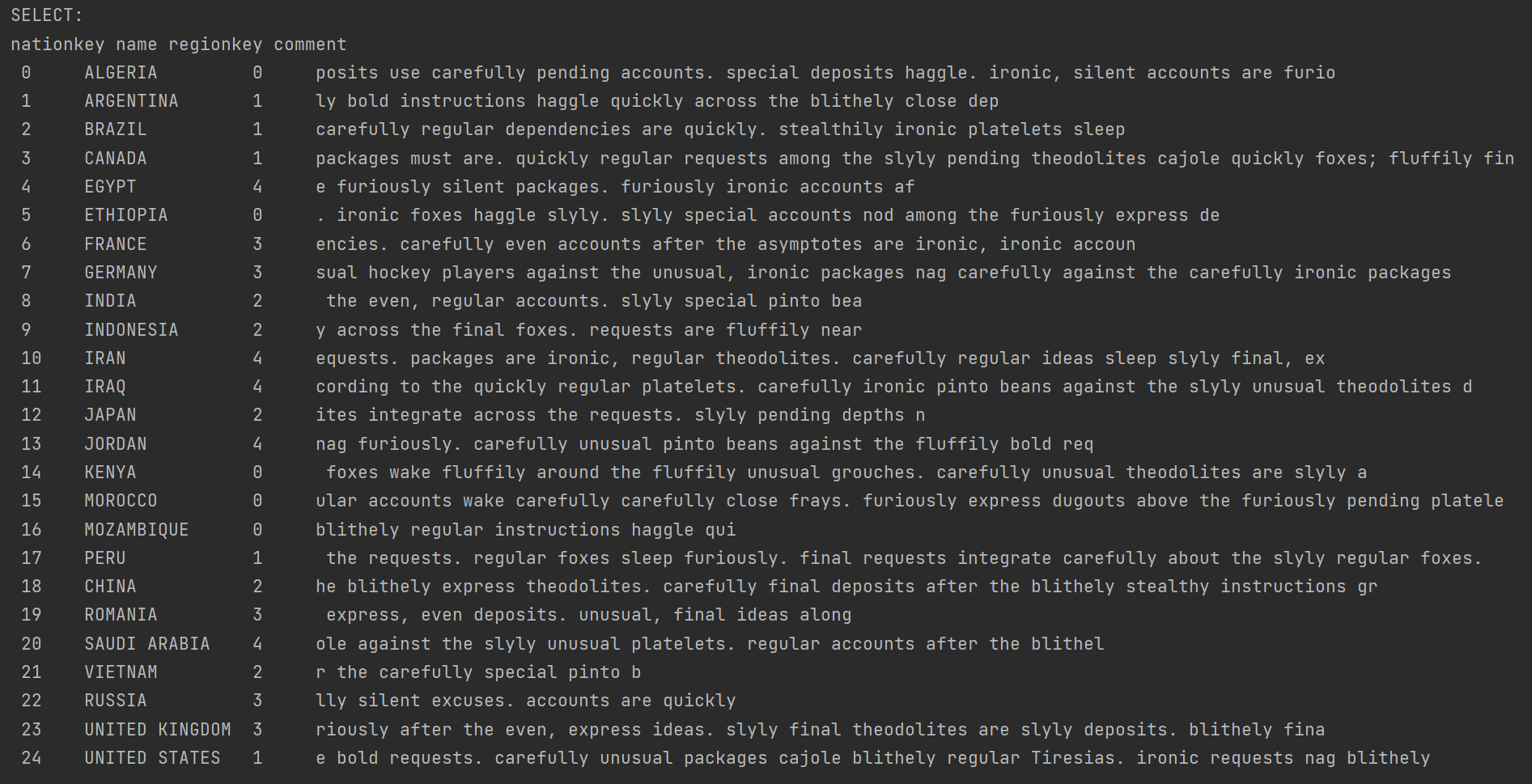
查询nation表中每条记录的n\_nationkey ,n\_name, n\_regionkey, n\_comment属性信息。

String sql\_select;  
sql\_select = "select n\_nationkey ,n\_name, n\_regionkey, n\_comment from nation";  
ResultSet rs = stmt.executeQuery(sql\_select);

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);  
}

执行结果：



（6）增加

Sql语句：insert into nation(n\_nationkey ,n\_name, n\_regionkey, n\_comment) values(25, 'South Korea' , 3, null)

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nation表中

String sql\_add;  
sql\_add = "insert into nation(n\_nationkey ,n\_name, n\_regionkey, n\_comment) values(25, 'South Korea' , 3, null)";  
stmt.executeUpdate(sql\_add);

使用select语句根据n\_nationkey = 25查询记录并显示结果，判断增加是否成功

sql\_select = "select n\_nationkey ,n\_name, n\_regionkey, n\_comment from nation where n\_nationkey = 25";  
rs = stmt.executeQuery(sql\_select);

查询结果:



能查询到增加的记录（25， South Korea, 3, null），插入成功。

（7）更新

Sql语句：update nation set n\_regionkey = 2 where n\_nationkey = 25

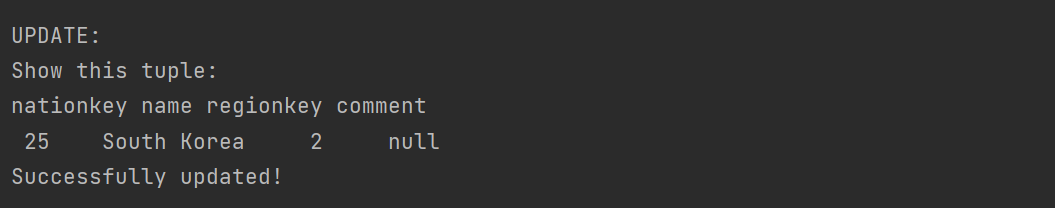
将插入的数据中的 n\_regionkey 更新为 2。

String sql\_update;  
sql\_update = "update nation set n\_regionkey = 2 where n\_nationkey = 25";  
stmt.executeUpdate(sql\_update);

使用select语句根据n\_nationkey = 25查询记录并显示结果，判断更新是否成功

sql\_select = "select n\_nationkey ,n\_name, n\_regionkey, n\_comment from nation where n\_nationkey = 25";  
rs = stmt.executeQuery(sql\_select);

查询结果:



查询到插入的记录现为（25， South Korea, 2 null），regionkey属性更新成功。

（8）删除

Sql语句：delete from nation where n\_nationkey = 25

删除插入的信息。

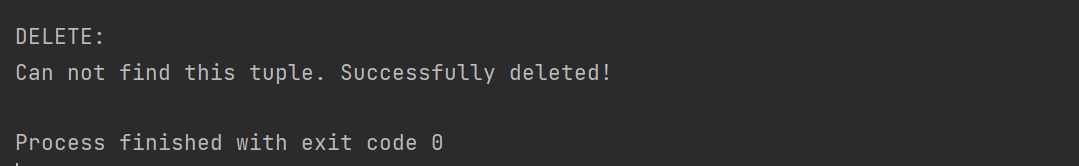
String sql\_delete;  
sql\_delete = "delete from nation where n\_nationkey = 25";  
stmt.executeUpdate(sql\_delete);

使用select语句根据n\_nationkey = 25查询记录，依据是否找到记录判断删除是否成功。

sql\_select = "select n\_nationkey ,n\_name, n\_regionkey, n\_comment from nation where n\_nationkey = 25";  
rs = stmt.executeQuery(sql\_select);

if (rs.next()) {  
 // 如果找到记录，说明删除操作未成功  
 System.*out*.println("Tuple still exists. Deletion failed!");  
} else {  
 // 如果未找到记录，说明删除成功  
 System.*out*.println("Can not find this tuple. Successfully deleted!");  
}

查询结果:



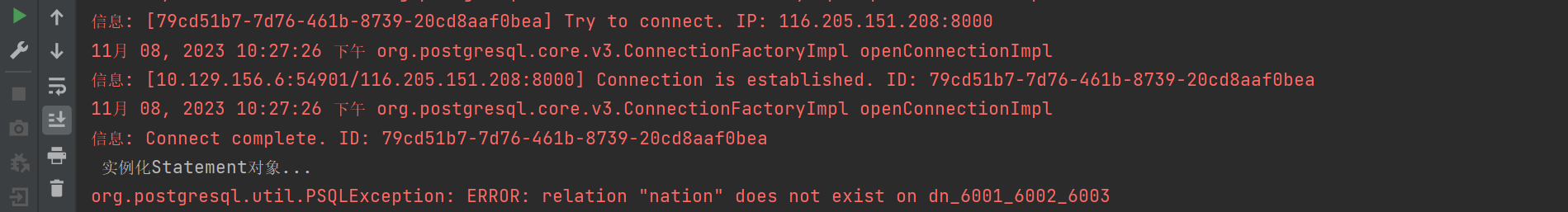
未找到相关记录，删除成功。

# 五、实验总结

本次实验中，我们了解了动态 SQL 与数据库应用编程接口的原理，掌握了配置方法和接口调用方式。采用JDBL接口编写了简单的数据库应用程序实现了对TPC-H 数据库的访问，执行了行查找、增加、 删除、更新等操作。在实验过程中，我们遇到了一些问题，通过不断寻因并尝试，这些问题都得到了解决：

1.高版本Java EE不自带JAXB包，需要自行下载，查阅相关博客后掌握了自行下载Java相关包并添加到依赖库的方法。

2.根据指导书配置url里的ip和数据库名称（旧，postgres数据库），没有使用本次实验所指定的finance数据库，导致报错“relation "nation" does not exist on dn-6001-6002-6003”



3.在编写Java程序测试执行sql语句时，向数据库加入nationkey（主码）为25的记录之后并没有删除，引发在重新运行时发生报错” duplicate key value violates unique constraint "nation\_pkey”，增加的记录与表中已有的记录主码重复，违反了唯一性约束。删除后恢复正常

