

学号 WA2214014 专业 人工智能 姓名 杨跃浙
实验日期 05.23 教师签字 成绩

实验报告

【实验名称】 实验 1-数据定义

【实验目的】

1. 学习并掌握 SQL 数据库定义功能，掌握基本表、索引的概念和作用，熟悉 SQL Server 2008 | openGauss 的数据类型；
2. 熟悉在 SQL Server Management Studio | Data Studio 中利用交互式向导创建和管理数据库、基本表、索引的方法；
3. 熟悉在 SQL Server Management Studio | Data Studio 中利用 SQL 语句创建和管理数据库、基本表、索引的方法；
4. 了解数据库的修改和删除方法。

【实验原理】

1.用 SQL 语句创建数据库

使用 Data Studio 登陆数据库。

创建数据库 SPJ:

```
CREATE DATABASE SPJ ENCODING 'UTF8' template = template0;
```

2.1.用 SQL 语句修改数据库

将数据库名 SPJ 改为“SPJ001”。

```
ALTER DATABASE SPJ RENAME TO SPJ001;
```

使用 Data Studio 登陆数据库。

2.2.用交互式向导修改数据库（方法二）：

右击“数据库名称”（如：SPJ）

点击“重命名”可以修改数据库名称

3.1.用 SQL 语句删除数据库

删除“SPJ001”数据库。

DROP DATABASE SPJ001;

3.2.用交互式向导删除数据库（方法二）：

右击“数据库名称”

点击“删除”可以删除数据库

4.用 SQL 语句创建基本表

点击“新建查询”

输入 SQL 语句，语法如下：

CREATE TABLE <表名>

 (<列名> <数据类型>[<列级完整性约束条件>]

 [, <列名> <数据类型>[<列级完整性约束条件>]] …

 [, <表级完整性约束条件>]) ;

点击“√”分析调试语句，点击“! 执行”，创建基本表

5.用 SQL 语句修改基本表

使用 ALTER TABLE 语句可以修改基本表。

语法

ALTER TABLE <表名>

[ADD <新列名> <数据类型>[完整性约束]]

[DROP <完整性约束名>]

[ALTER COLUMN<列名> <数据类型>]

6.1.用 SQL 语句删除基本表

[例] 将 SPJ 表删除

DROP TABLE SPJ;

注意：使用 DROP TABLE 语句删除的是基本表本身，会将基本表的定义和表中的数据一起删除，表上建立的索引、视图、触发器等一般也将被删除。

如果只想删除基本表中的数据而保留基本表的定义，则不能使用 DROP TABLE 语句，而必须使用后面数据更新功能中介绍的 DELETE 语句。

6.2.用交互式向导删除基本表（方法二）：

右击“表”的名称（如：S）

点击“删除”，可以删除基本表

注：如果该表上中有些列被其他表引用，则需要先将引用的表删除。

7.0.索引简介

建立索引的目的：加快查询操作的速度

谁可以建立索引

DBA 或表的属主 DBO（即建立表的人）

DBMS 一般会建立有以下约束的属性列上的索引

PRIMARY KEY

UNIQUE

谁维护索引

DBMS 自动完成

使用索引

DBMS 自动选择是否使用索引以及使用哪些索引（查询优化）

可以在基本表的一个或多个属性列上建立索引，作为存取路径

RDBMS 中索引一般采用 B+树、HASH 索引来实现

B+树索引具有动态平衡的优点

HASH 索引具有查找速度快的特点

采用 B+树，还是 HASH 索引，则由具体的 RDBMS 来决定

索引是关系数据库的内部实现技术，属于内模式的范畴

CREATE INDEX 语句定义索引时，可以定义索引是唯一索引、非唯一索引或聚簇索引

聚簇索引：元组按照索引键值的顺序存储

7.用 SQL 语句创建索引

语法

```
CREATE [ UNIQUE ] [ CLUSTERED | NONCLUSTERED ] INDEX <index_name>  
ON <object> (<column> [ ASC | DESC ] [ ,...n ] )  
[ WHERE <filter_predicate> ]  
[openGauss] CREATE [ UNIQUE ] INDEX <index_name>  
ON <object> (<column> [ ASC | DESC ] [ ,...n ] )  
[ WHERE <filter_predicate> ]
```

8.用 SQL 语句修改索引

```
ALTER INDEX <index_name> [UNUSABLE|REBUILD|RENAME TO];
```

各选项含义为：

UNUSABLE：禁用索引；

REORGANIZE：重新组织索引；

RENAME TO：修改索引名。

9.用 SQL 语句删除索引

当不再需要某个索引时，可以用过 DROP INDEX 语句将该索引删除，语法格式如下：

```
DROP INDEX <table_name>.<index_name>
```

或

```
DROP INDEX <index_name> ;
```

【实验内容】

为供应链管理数据库 SPJ 完成以下数据定义工作：

1.使用 Data Studio 交互式向导创建供应链管理数据库并命名为“SPJ”；

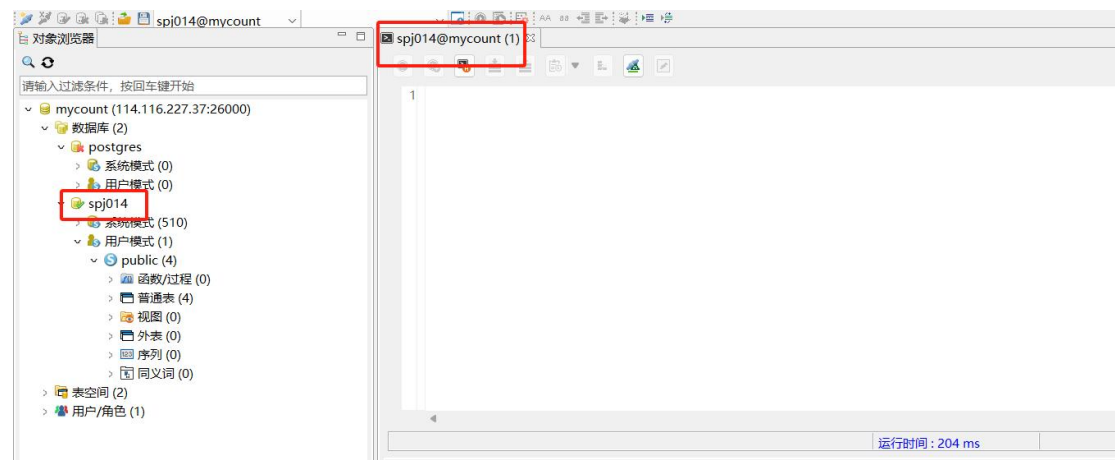
- 2.使用 Data Studio 或 SQL 语句创建供应商表 S、零件表 P;
- 3.使用 SQL 语句创建工程项目表 J 和供应表 SPJ;
- 4.在供应商表 S 中增加经理属性列: MANAGER, 字符型, 宽度 8 位;
- 5.删除供应商表 S 中的经理属性列 MANAGER;
- 6.在零件表 P 中增加一属性列: 生产厂家 FACTORY, 字符型, 宽度 20 位;
- 7.删除零件表 P 中的零件生产厂家属性列 FACTORY;
- 8.为供应商表 S 的供应商名属性 SNAME、城市属性 CITY 分别创建索引;
- 9.为项目表 J 的项目名属性 JNAME、城市属性 CITY 分别创建索引;
- 10.为供应表 SPJ 在供应量属性 QTY 上创建索引。

要求:

每个数据库和表名最后应加上实验者个人学号的末三位 (如 SPJ001, S001 等)

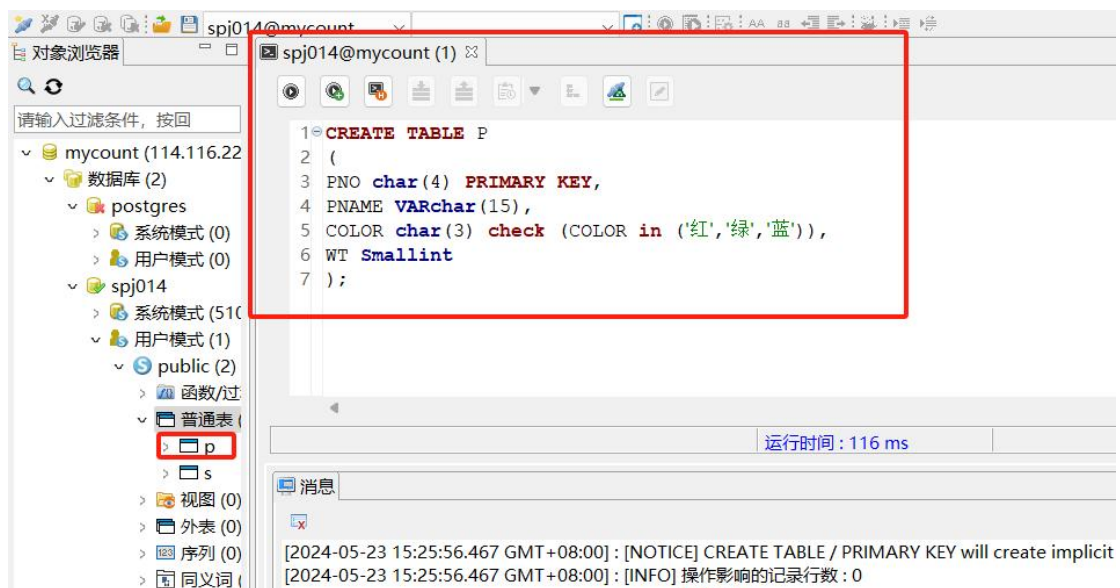
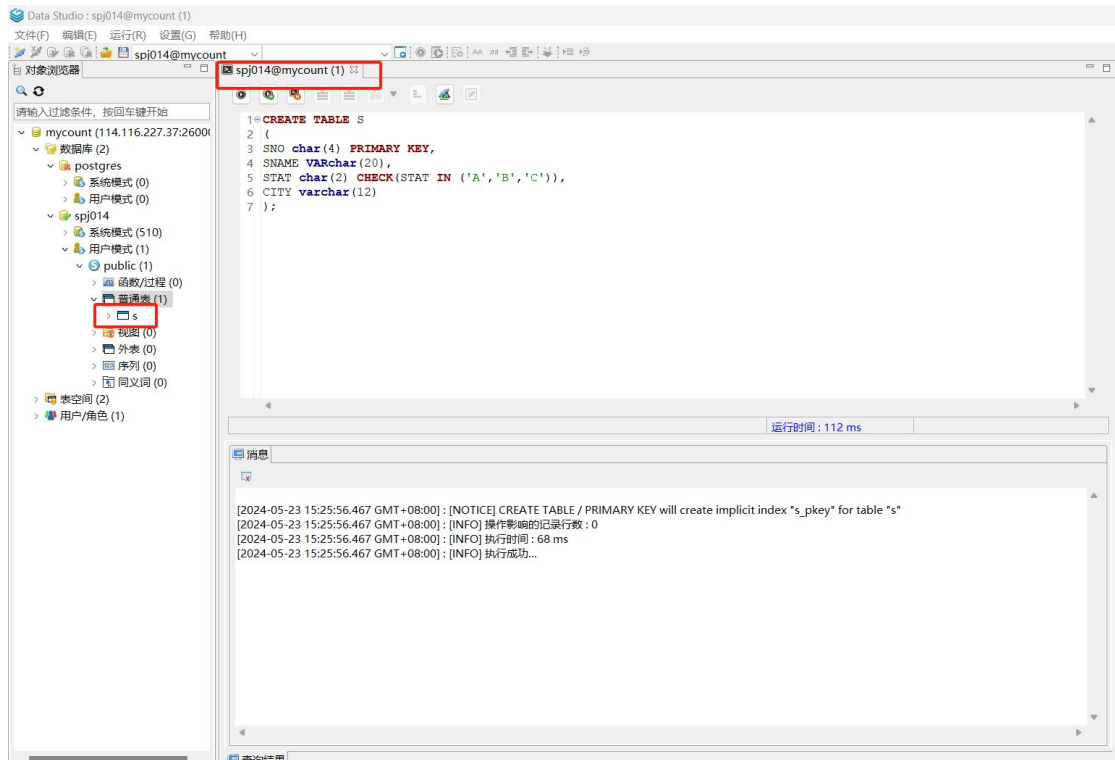
实验报告文档中实验步骤部分每小题题目之后给出 SQL 语句和运行结果的截图, 主要截图中要展示出数据库名或表名。

1.



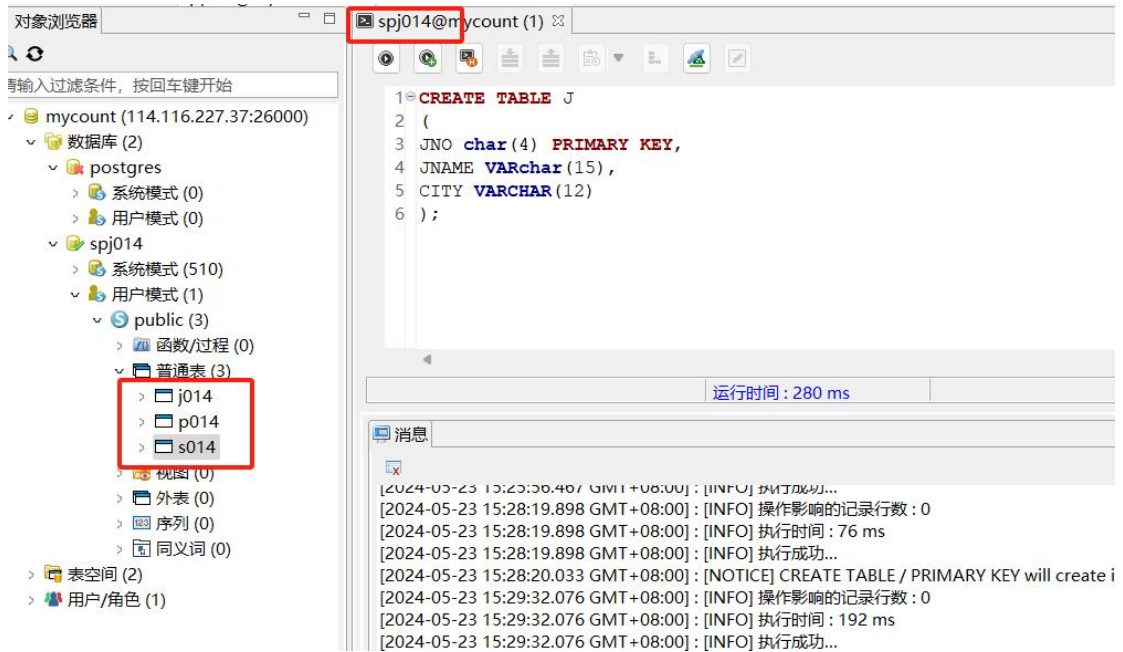
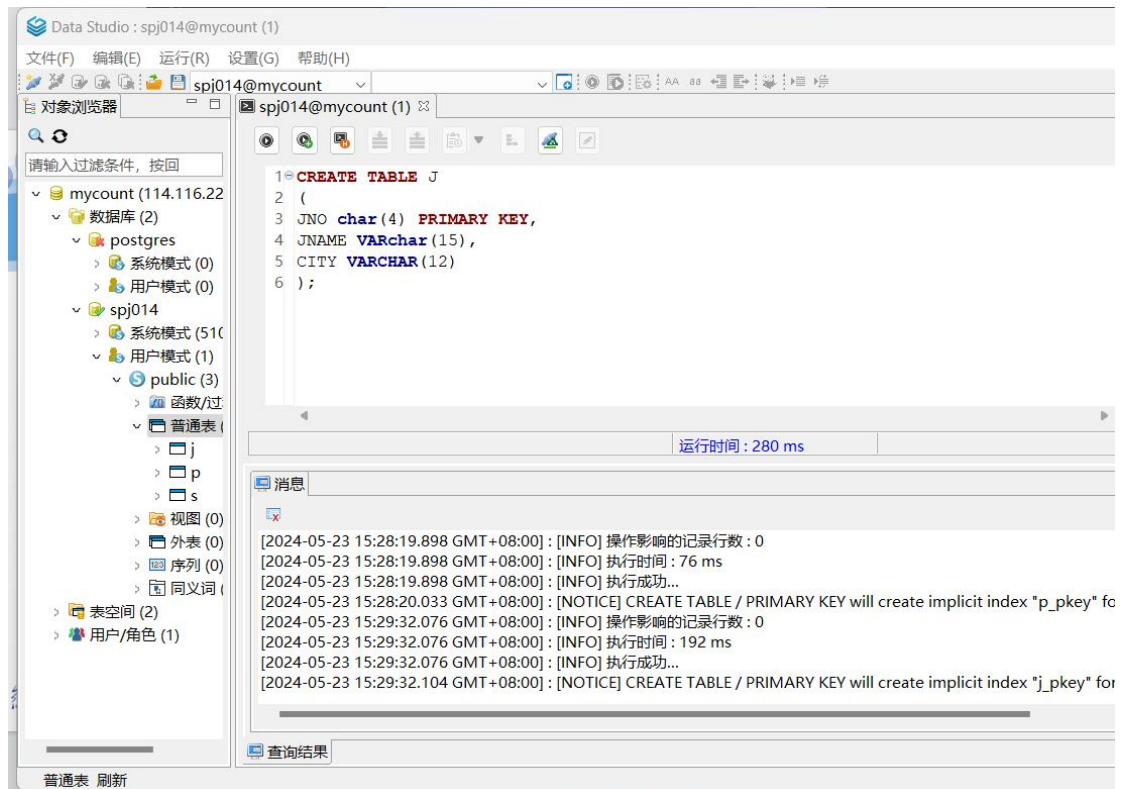
数据库创建成功, 命名为 spj014

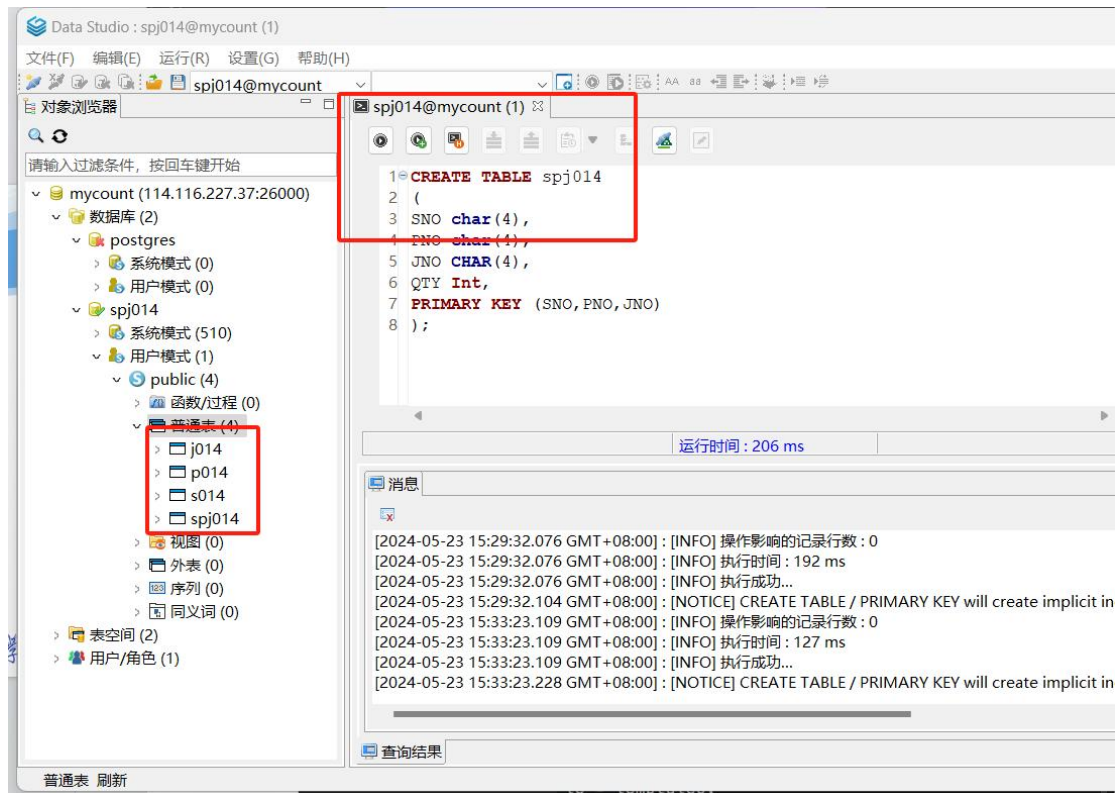
2.



使用 SQL 语句创建 S,P 表成功

3.



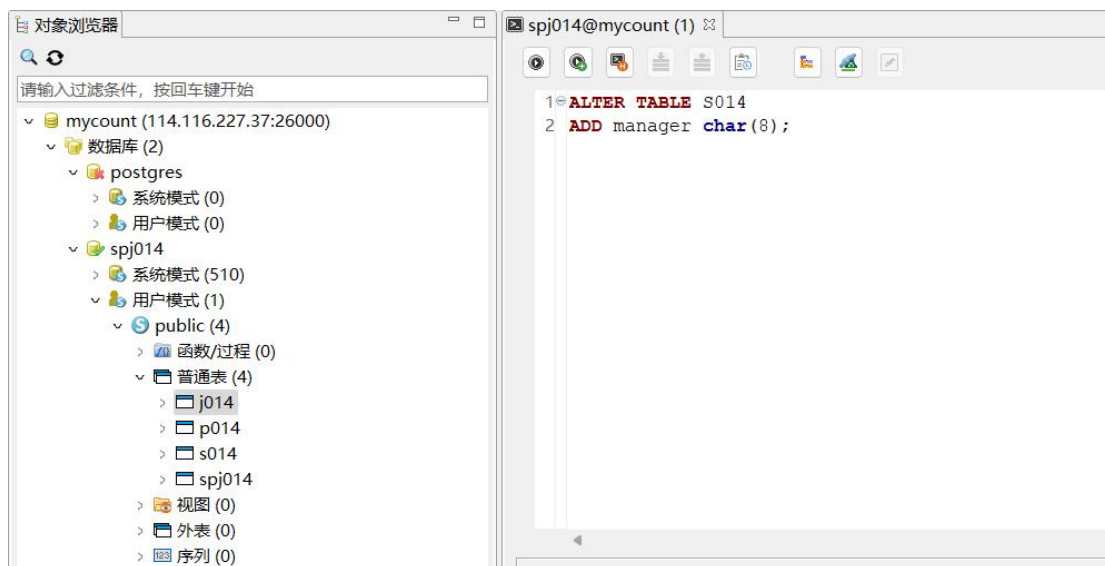


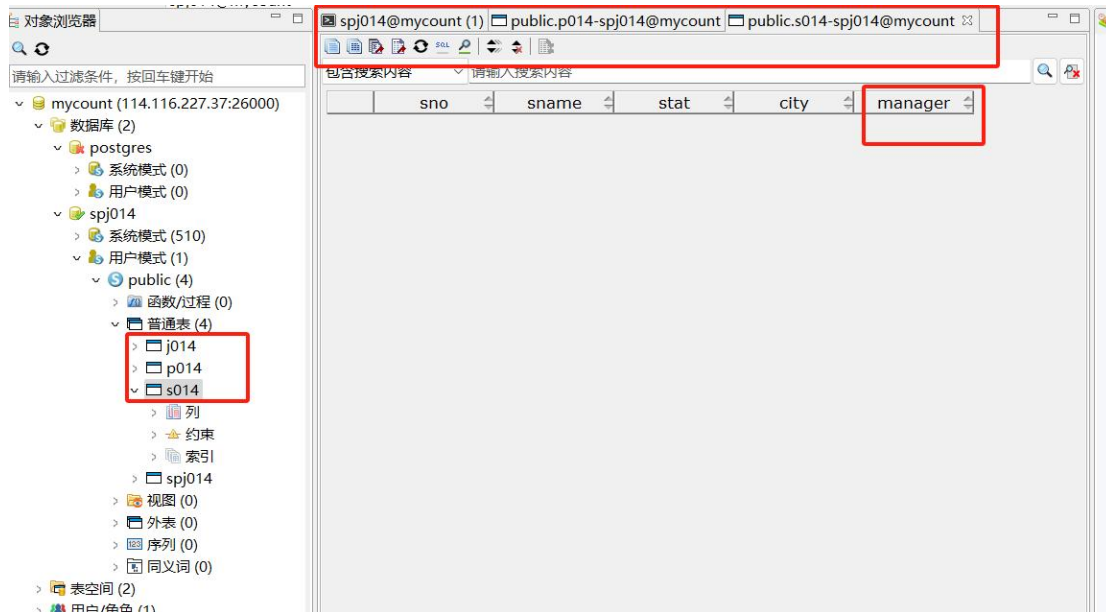
使用 SQL 语句创建 J 表

用交互式向导将 S,P,J 表重命名为 S014, P014, J014

使用 SQL 语句创建 SPJ014 表

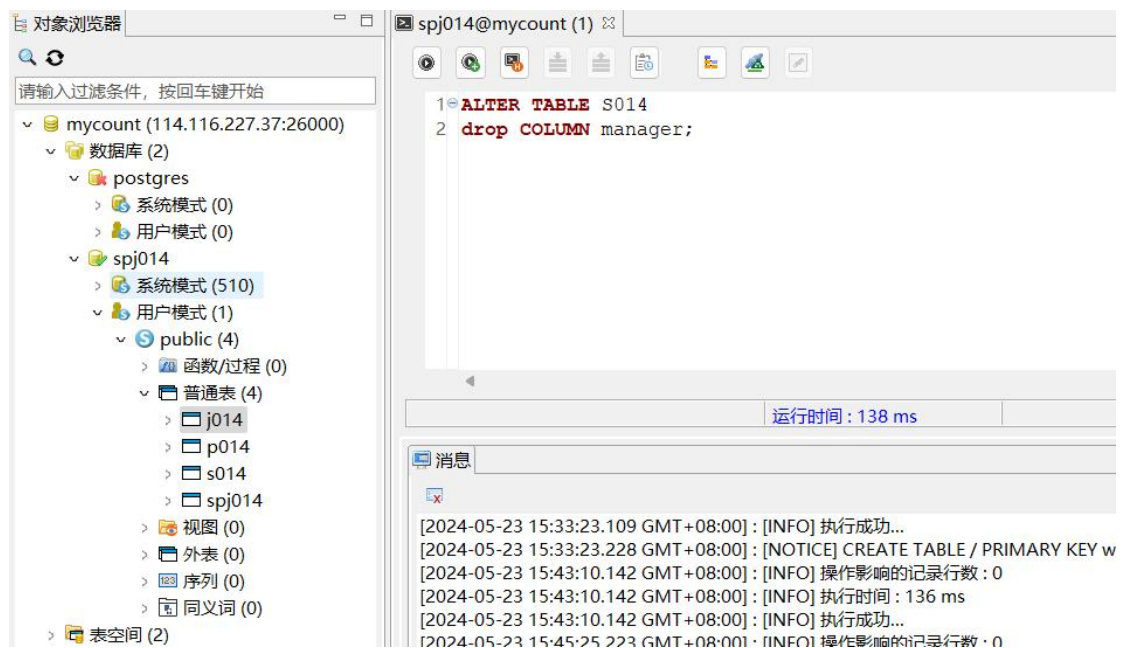
4.

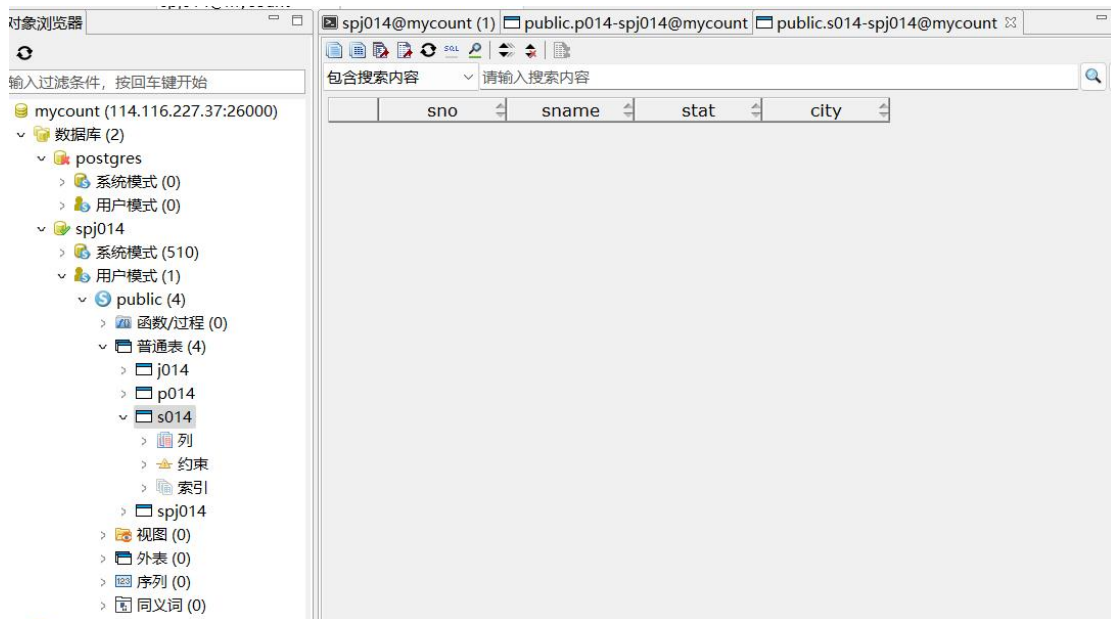




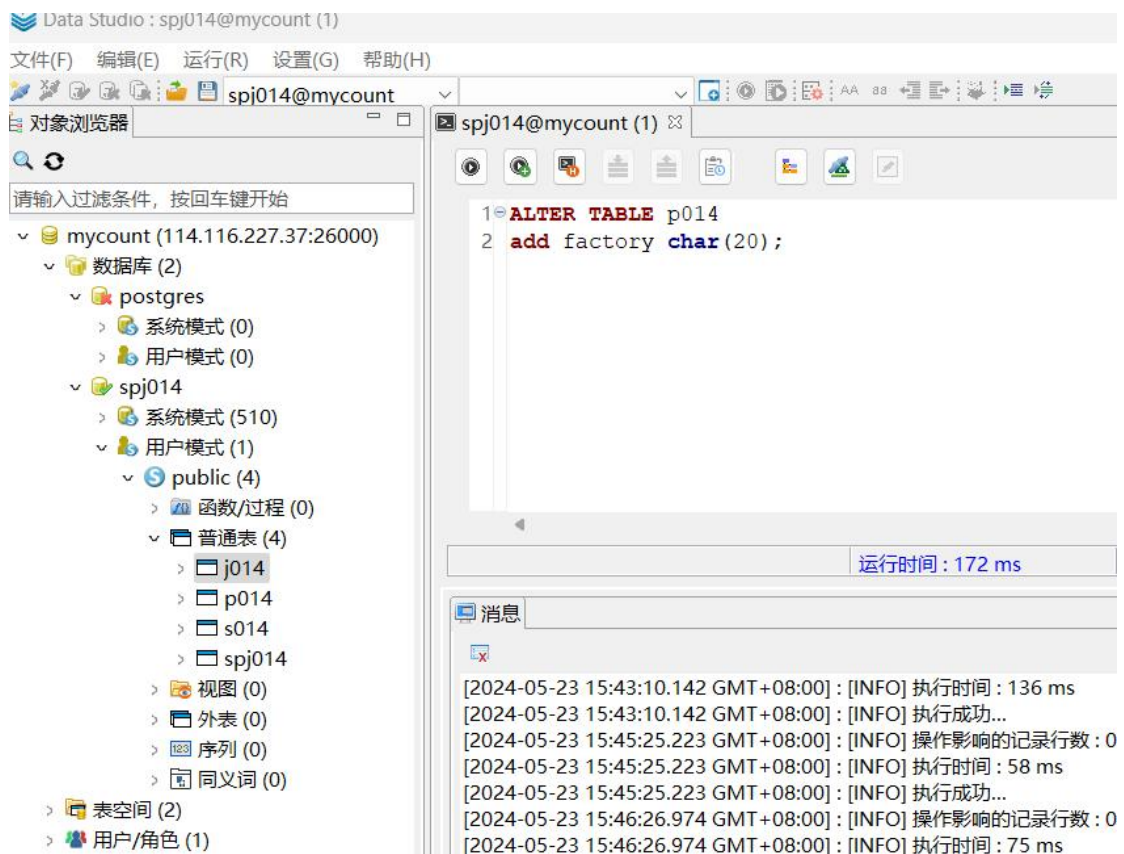
使用 SQL 语句在供应商表 S 中增加经理属性列

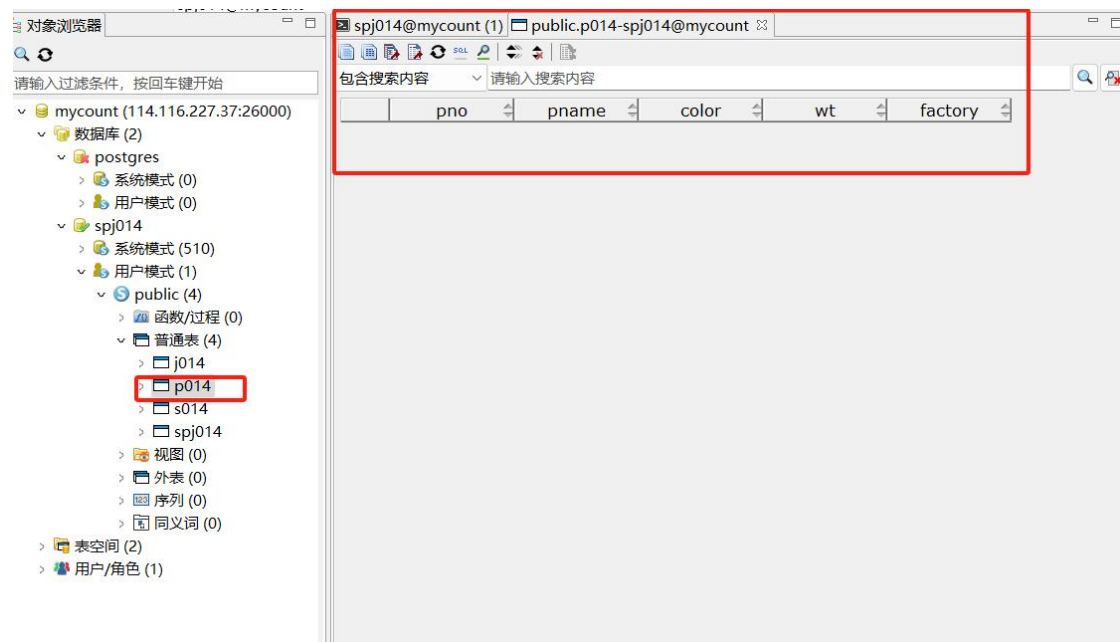
5.





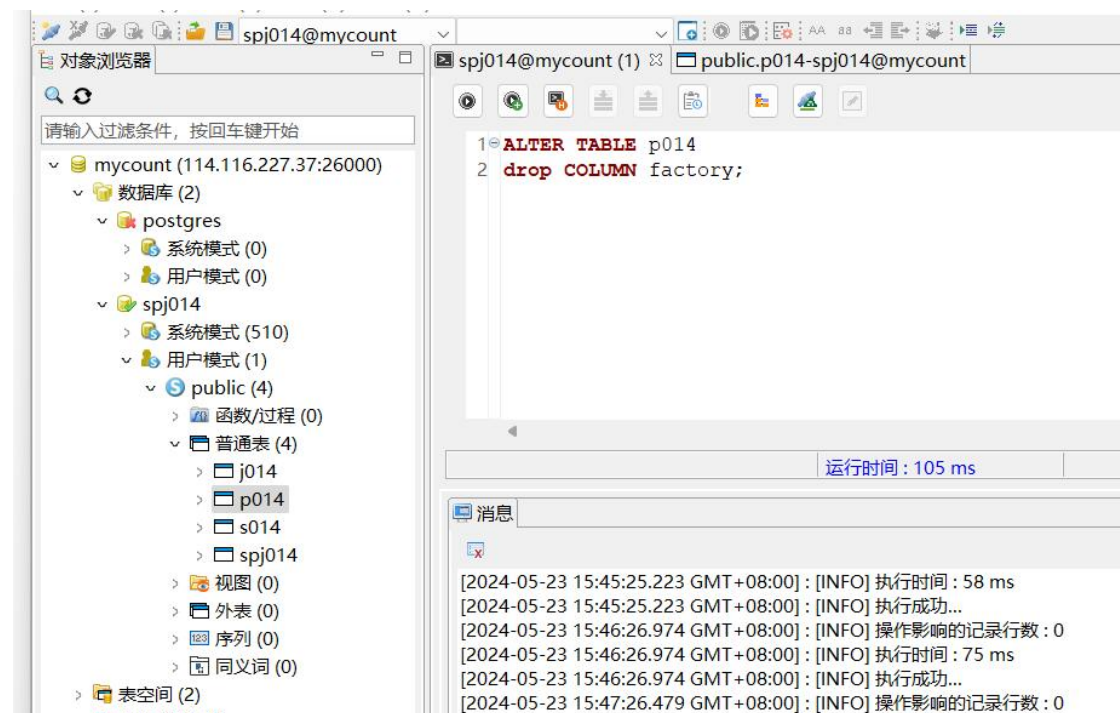
使用 SQL 删除供应商表 S 中的经理属性列 MANAGER

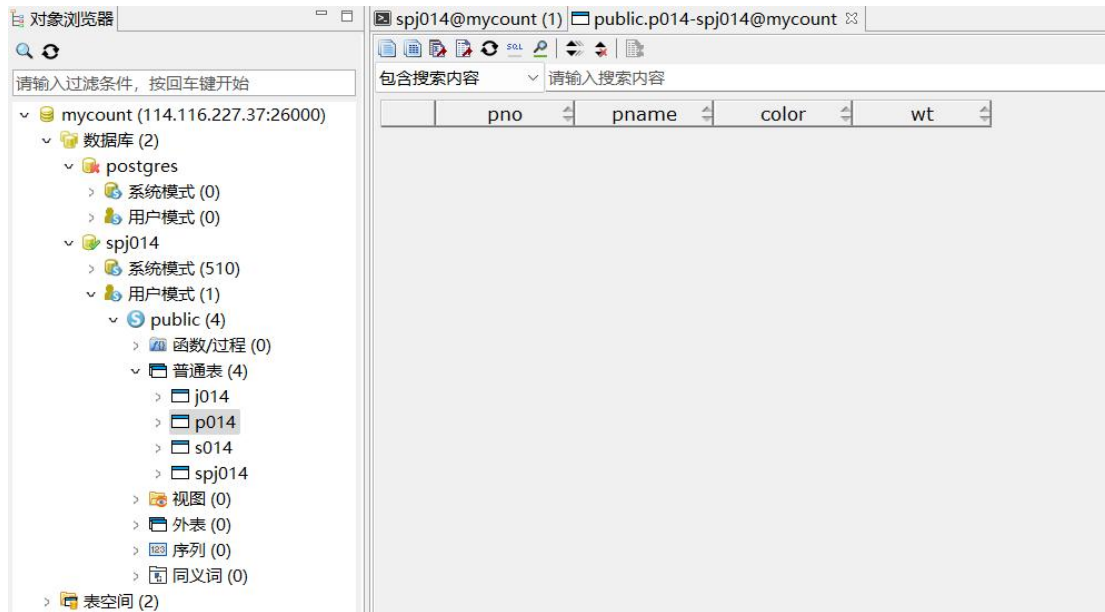




使用 SQL 在零件表 P 中增加一属性列：生产厂家 FACTORY

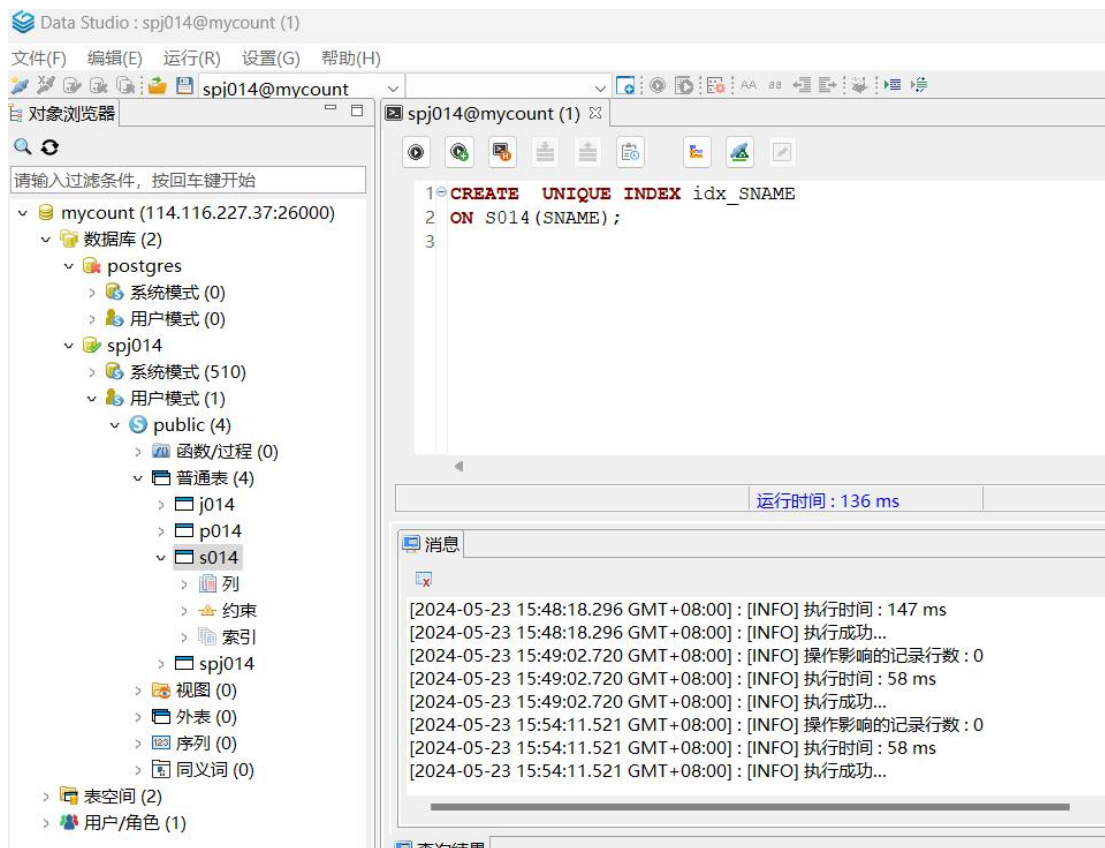
7.

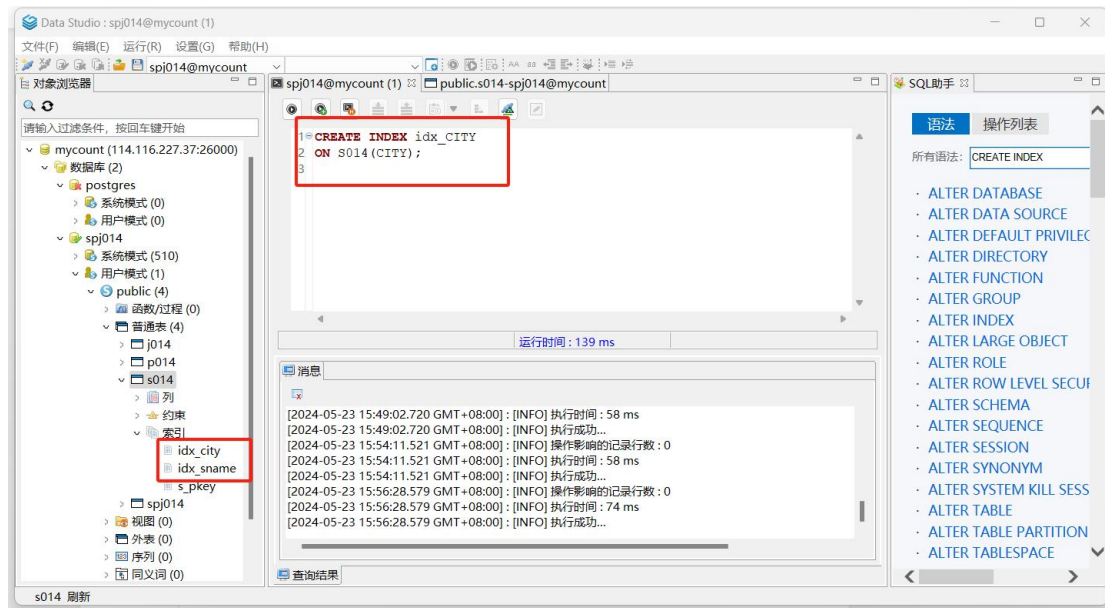




使用 SQL 删除零件表 P 中的零件生产厂家属性列 FACTORY

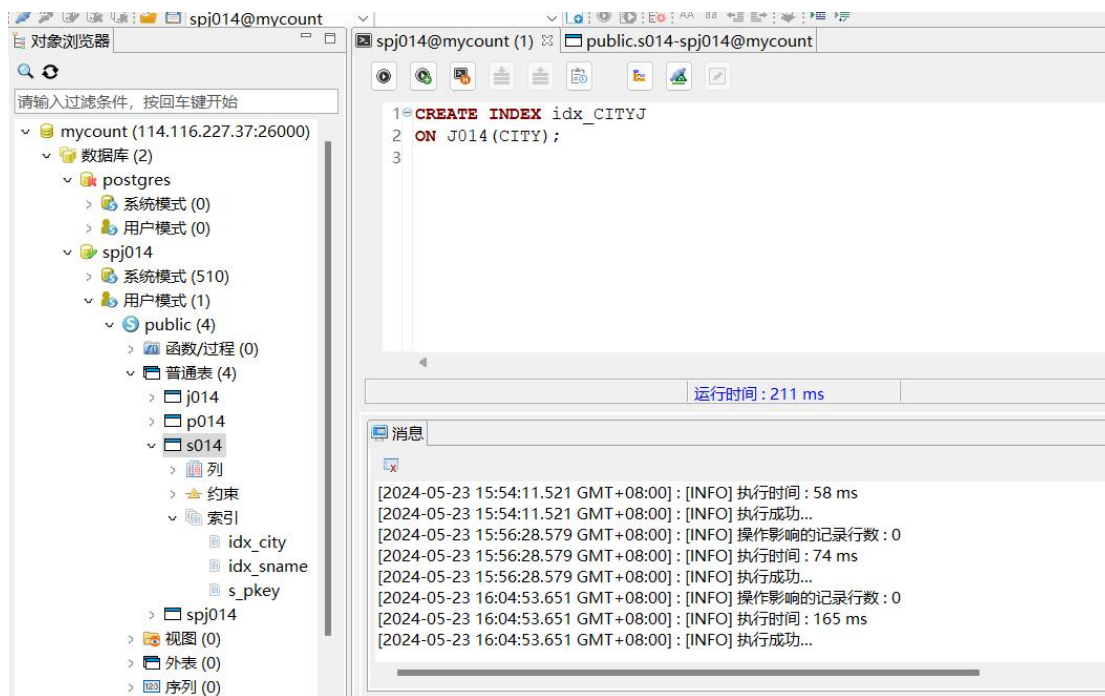
8.





使用 SQL 为供应商表 S 的供应商名属性 SNAME、城市属性 CITY 分别创建索引

9.



The image displays two screenshots of a PostgreSQL IDE interface, likely pgAdmin, showing the execution of SQL commands to create unique indexes.

Top Screenshot:

- Object Explorer (Left):** Shows the database structure. The 'public' schema under the 'spj014' user is expanded, showing tables 'j014', 'p014', and 's014'. The 'idx_city' index is highlighted.
- SQL Editor (Right):** Contains the SQL command:

```
1 CREATE UNIQUE INDEX idx_JNAME
2 ON J014(JNAME);
3
```
- Execution Results (Bottom):** Shows the execution time as 470 ms and a message log with timestamps and status messages (e.g., [INFO] 执行时间: 74 ms, [INFO] 执行成功...).

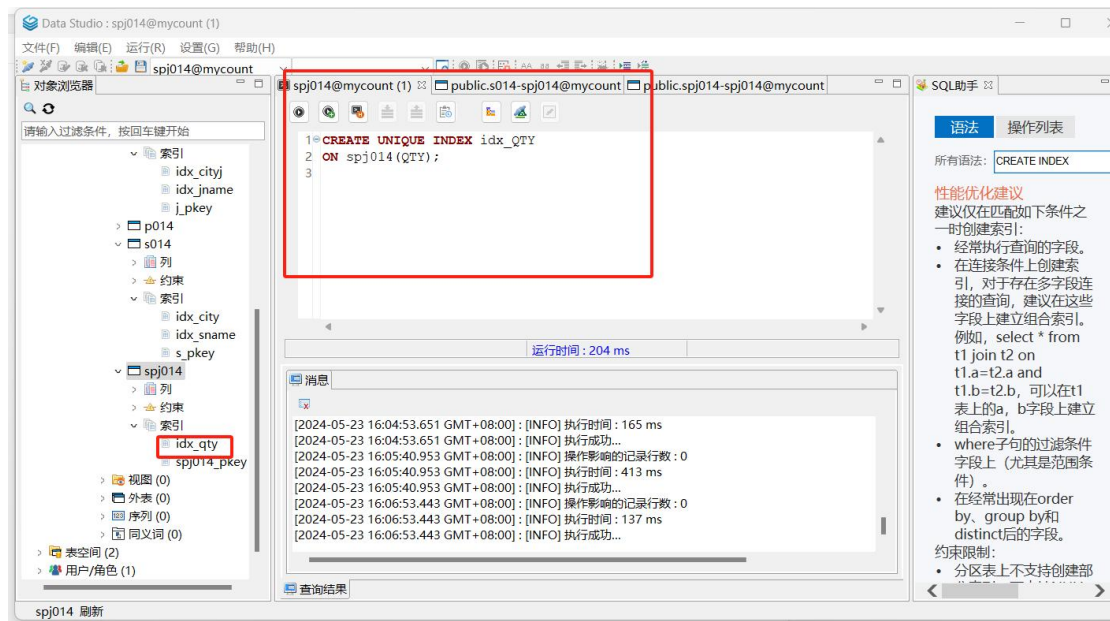
Bottom Screenshot:

- Object Explorer (Left):** Shows the database structure. The 'public' schema under the 'spj014' user is expanded, showing tables 'j014', 'p014', and 's014'. The 'idx_jname' index is highlighted.
- SQL Editor (Right):** Contains the SQL command:

```
1 CREATE UNIQUE INDEX idx_JNAME
2 ON J014(JNAME);
3
```
- Execution Results (Bottom):** Shows the execution time as 470 ms and a message log with timestamps and status messages (e.g., [INFO] 执行时间: 74 ms, [INFO] 执行成功...).

使用 SQL 为项目表 J 的项目名属性 JNAME、城市属性 CITY 分别创建索引

10.



使用 SQL 为供应表 SPJ 在供应量属性 QTY 上创建索引

【小结或讨论】

本次实验主要目的是学习并掌握 SQL 数据库的定义功能，包括创建和管理数据库、表、索引等。通过这次实验，我深入了解了 openGauss 的操作环境，并掌握了在 Data Studio 中使用交互式向导和 SQL 语句来操作数据库的方法。实验中，我创建了名为“SPJ014”的数据库，并对其进行了重命名、修改和删除操作。接着，我创建了供应商表、零件表、工程项目表和供应表，并对这些表进行了增加和删除列的操作。此外，我还为几个关键属性创建了索引，以提高查询效率。

在实验过程中，我学习了如何根据不同的需求选择和创建最适合的索引类型，这对提高数据库性能至关重要。此外，选择合适的数据类型对于优化存储空间和提高查询效率也非常重要。例如，为供应商表中的经理属性选择了字符型数据类型，并指定了宽度，这有助于减少存储空间的浪费。精确和高效使用 SQL 命令对有效管理数据库对象同样重要，实验中我通过正确使用 CREATE、ALTER 和 DROP 等 SQL 命令进行了有效的数据库管理。

通过这次实验，我不仅掌握了数据库定义的基本操作，也对如何在实际应用中优化和管理数据库有了更深的理解，这将对我的未来学习和工作产生积极的影响。