

学号 WA2214014 专业 人工智能 姓名 杨跃浙  
实验日期 06.20 教师签字 成绩

# 实验报告

【实验名称】 实验 5-数据更新和视图管理

## 【实验目的】

- 熟悉数据更新操作的概念与操作类型;
- 熟练掌握 INSERT、UPDATE、DELETE 语句的基本语法;
- 熟练运用 INSERT、UPDATE、DELETE 语句实现数据的插入、修改与删除操作。
- 理解视图的基本概念与作用;
- 熟练掌握创建视图的方法;
- 熟悉通过视图访问基本表的数据的方法。

## 【实验原理】

数据更新

插入数据

修改数据

删除数据

插入数据

两种插入数据方式

1. 插入元组
  2. 插入子查询结果
- 可以一次插入多个元组

## 一、插入元组

语句格式

```
INSERT INTO <表名> [(<属性列 1>[, <属性列 2>…)]
VALUES (<常量 1> [, <常量 2>] … )
```

功能

将新元组插入指定表中

INTO 子句

属性列的顺序可与表定义中的顺序不一致

没有指定属性列

指定部分属性列

VALUES 子句

提供的值必须与 INTO 子句匹配

值的个数

值的类型

## 二、插入子查询结果

语句格式

```
INSERT INTO <表名> [<属性列 1> [, <属性列 2>… ]]
子查询;
```

功能

将子查询结果插入指定表中

INTO 子句(与插入元组类似)

子查询

SELECT 子句目标列必须与 INTO 子句匹配

值的个数

值的类型

RDBMS 在执行插入语句时会检查所插元组是

否破坏表上已定义的完整性规则

实体完整性

参照完整性

用户定义的完整性

NOT NULL 约束

UNIQUE 约束

值域约束

如果有破坏完整性约束条件的，DBMS 拒绝执行插入操作

## 修改数据

语句格式

UPDATE <表名>

    SET <列名>=<表达式>[, <列名>=<表达式>]…

    [WHERE <条件>];

功能

修改指定表中满足 WHERE 子句条件的元组

SET 子句

指定修改方式

要修改的列

修改后取值

WHERE 子句

指定要修改的元组

缺省表示要修改表中的所有元组

三种修改方式

1. 修改某一个元组的值

2. 修改多个元组的值

3. 带子查询的修改语句

RDBMS 在执行修改语句时会检查修改操作

是否破坏表上已定义的完整性规则

实体完整性

主码不允许修改

用户定义的完整性

NOT NULL 约束

UNIQUE 约束

值域约束

如果有破坏完整性约束条件的，DBMS 拒绝执行修改操作

## 删除数据

语句格式

```
DELETE
  FROM    <表名>
  [WHERE <条件>][RESTRICT|CASCADE];
```

功能

删除指定表中满足 WHERE 子句条件的元组

WHERE 子句

指定要删除的元组

缺省表示要删除表中的全部元组，表的定义仍在字典中

RESTRICT|CASCADE (有的 DBMS 没有这个选项)

RESTRICT: 如有其它表中元组参照该元组的主码值，拒绝删除

CASCADE: 如有其它表中元组参照该元组的主码值，级联删除

三种删除方式

1. 删除某一个元组的值

2. 删除多个元组的值

3. 带子查询的删除语句

[例 11] 删除会员“江涛”的信息。

删除会员“江涛”的信息，为保证数据库的数据一致性，必须先删除其订单信息，而删除订单，则必须先删除该订单的明细信息，最后才能删除会员信息。

第一步：删除订单明细信息；

```
DELETE
FROM DETAIL
WHERE OID IN (SELECT OID
                FROM BOOKORDER, MEMBER
               WHERE BOOKORDER.MID = MEMBER.MID
                 AND MEMBER.NAME='江涛');
```

第二步：删除订单信息；

```
DELETE
FROM BOOKORDER
WHERE BOOKORDER.MID IN (SELECT MID
                           FROM MEMBER
                          WHERE MEMBER.NAME='江涛');
```

第三步：删除会员“江涛”的信息；

```
DELETE
FROM MEMBER
WHERE MEMBER.NAME='江涛';
```

如果两个关系通过定义外部码实现参照完整性约束，则删除被参照表（如学生表 STUDENT）中的元组时，参照表（如选课表 SC）中相关元组必须先做删除；  
允许级联删除的 DBMS 中，删除语句后面加“CASCADE”选项。

```
DELETE FROM BOOK
WHERE ISBN='U001' CASCADE;
```

数据更新操作除了用 SQL 语句实现，也可以很方便地在 DBMS 中用对基本表的交互式编辑的方式实现；

右击基本表，在弹出菜单中选择编辑（打开）项，直接在表中编辑；  
编辑过程中注意各种完整性约束条件对元组的限制，不符合约束条件的元组不能保存到基本表中。

## 视图管理

视图简介

定义视图

查询视图

更新视图

视图的作用

## 视图简介

视图的特点

虚表，是从一个或几个基本表（或视图）导出的表

数据库中只存放视图的定义，不存放视图对应的数据

基本表中的数据发生变化，从视图中查询出的数据也随之改变

基于视图的操作

查询

删除

受限更新

定义基于该视图的新视图

## 定义视图

建立视图

删除视图

建立视图

语句格式

CREATE VIEW

<视图名> [<列名> [, <列名>]…]

AS <子查询>

[WITH CHECK OPTION];

组成视图的属性列名：全部省略或全部指定

子查询是否可以含有 ORDER BY 子句和 DISTINCT 短语, 取决于具体系统的实现  
RDBMS 执行 CREATE VIEW 语句时只是把视图定义存入数据字典, 并不执行其中的 SELECT 语句。

在对视图进行查询时, 按视图的定义从基本表中将数据查出。

对 QH\_BOOK2 视图的更新操作:

修改操作: 自动加上 PUB = '清华大学出版社' 的条件

删除操作: 自动加上 PUB = '清华大学出版社' 的条件

插入操作: 自动检查 PUB 属性值是否为'清华大学出版社'

如果不是, 则拒绝该插入操作

如果没有提供 PUB 属性值, 则自动定义 PUB 为'清华大学出版社'

## 删除视图

语句的格式:

```
DROP VIEW <视图名>;
```

该语句从数据字典中删除指定的视图定义

如果该视图上还导出了其他视图, 使用 CASCADE 级联删除语句, 把该视图和由它导出的所有视图一起删除; 不支持 CASCADE 的 DBMS 必须先手动删除导出的视图

删除基表时, 由该基表导出的所有视图定义都必须显式地使用 DROP VIEW 语句删除

## 查询视图

用户角度: 查询视图与查询基本表相同

RDBMS 实现视图查询的方法

视图消解法 (View Resolution)

进行有效性检查

转换成等价的对基本表的查询

执行修正后的查询

视图消解法的局限

有些情况下, 视图消解法不能生成正确查询。

## 更新视图

更新视图的限制：一些视图是不可更新的，因为对这些视图的更新不能唯一地有意义地转换成对相应基本表的更新

例：视图 S\_DETAIL 为不可更新视图。

```
UPDATE S_DETAIL
SET      SQTY=100
WHERE   ISBN=  'Q001' ;
```

这个对视图的更新无法转换成对基本表 DETAIL 的更新。

允许对行列子集视图进行更新

对其他类型视图的更新不同系统有不同限制

## 视图的作用

1. 视图能够简化用户的操作
2. 视图使用户能以多种角度看待同一数据
3. 视图对重构数据库提供了一定程度的逻辑独立性
4. 视图能够对机密数据提供安全保护
5. 适当的利用视图可以更清晰的表达查询

## 【实验内容】

针对供应管理数据库 SPJ，进行各种数据更新（有些操作需要分步进行）：

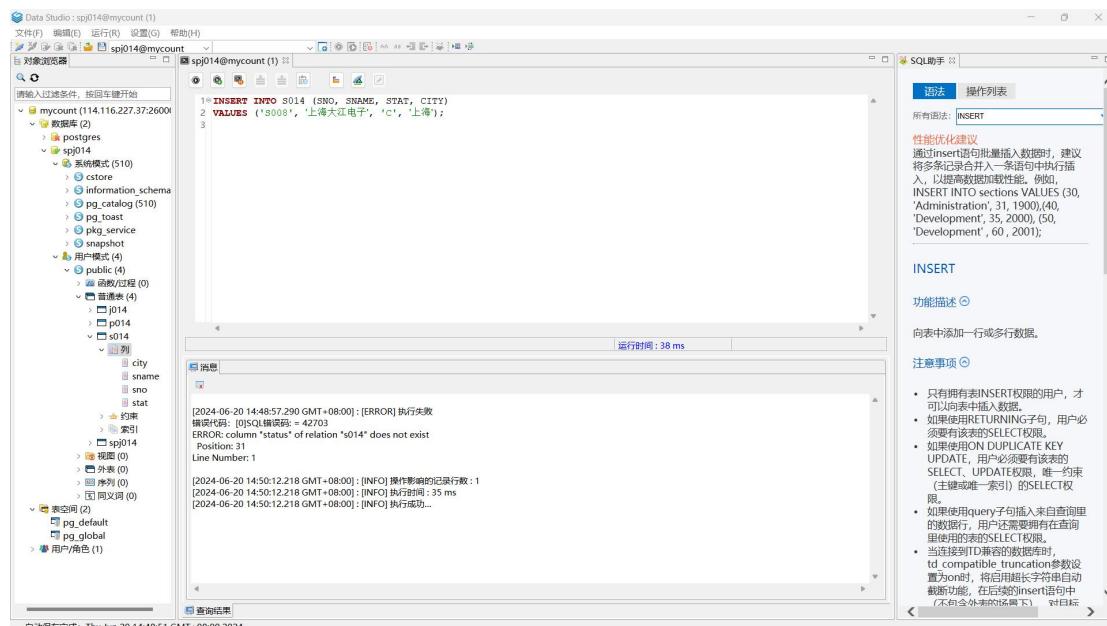
- (1) 插入一个“上海”供应商“上海大江电子”的信息，编号：“S008”，等级“C”；
- (2) 插入一个零件“开关”的信息，编号：“P008”，颜色“红”。重量：40；
- (3) 插入一个“上海”的工程项目“梅陇电子”的信息，编号：“J008”；
- (4) 插入“S008”供应项目“J008”零件“P008”的信息，供应量为500；
- (5) 将零件“P008”的颜色改为“黄”色；
- (6) 将工程“J008”的城市改为“合肥”；

- (7) 将“S008”供应商供应工程“J008”零件“P008”的供应量改为300;
- (8) 修改供应量大于500的供应详情，将供应量再增加100;
- (9) 将“J001”项目的大供应量降为原来的一半;
- (10) 删除项目“J008”的所有供应信息;
- (11) 删除“S008”供应商信息(注意其供应信息);
- (12) 删除零件“P008”的信息(注意其供应信息);
- (13) 删除项目“J008”的信息(注意其供应信息);
- (14) 统计所有供应商供应量的最大、最小和平均值，保存到一张表中(先建表)。

请针对供应管理数据库SPJ，进行各种视图操作：

- (1) 建立“北京”供应商视图BJ\_S;
- (2) 建立“北京”工程视图BJ\_J;
- (3) 建立“红”色零件视图RED\_P;
- (4) 建立“北京”供应商的供应情况视图BJS\_SPJ;
- (5) 建立“北京”供应商供应“北京”工程的供应情况视图BJSJ\_SPJ;
- (6) 建立“北京新天地”供应商的供应情况视图WM\_SPJ;
- (7) 分别删除以上定义的各个视图。

(1)



插入一个“上海”供应商“上海大江电子”的信息，编号：“S008”，等级“C”；

The screenshot shows the Data Studio interface. On the left is the object browser with 's001' selected. In the center is a table named 's001' with the following data:

sno	sname	stat	city
1	天津安贝儿	B	天津
2	北京后明星	A	北京
3	北京新天地	C	北京
4	天津丰泰盛	B	天津
5	上海普丰	C	上海
6	合肥四达	B	合肥
7	上海大江电子	C	上海

On the right, the SQL Assistant pane shows the following code:

```
INSERT INTO sections VALUES (30, 'Administration', 31, 1900),(40, 'Development', 35, 2000),(50, 'Development', 60, 2001);
```

(2)

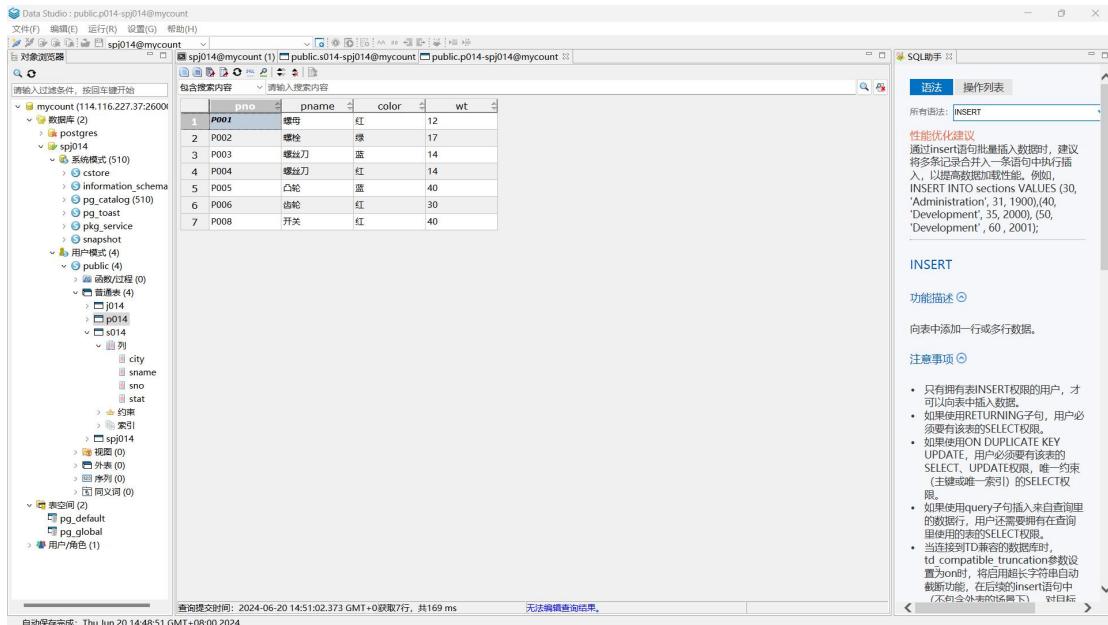
The screenshot shows the Data Studio interface. On the left is the object browser with 's014' selected. In the center, a message box displays the following error:

```
1: INSERT INTO p014 (pmo, pname, color, wt)
2: VALUES ('P008', '开关', '红', 40);
3:

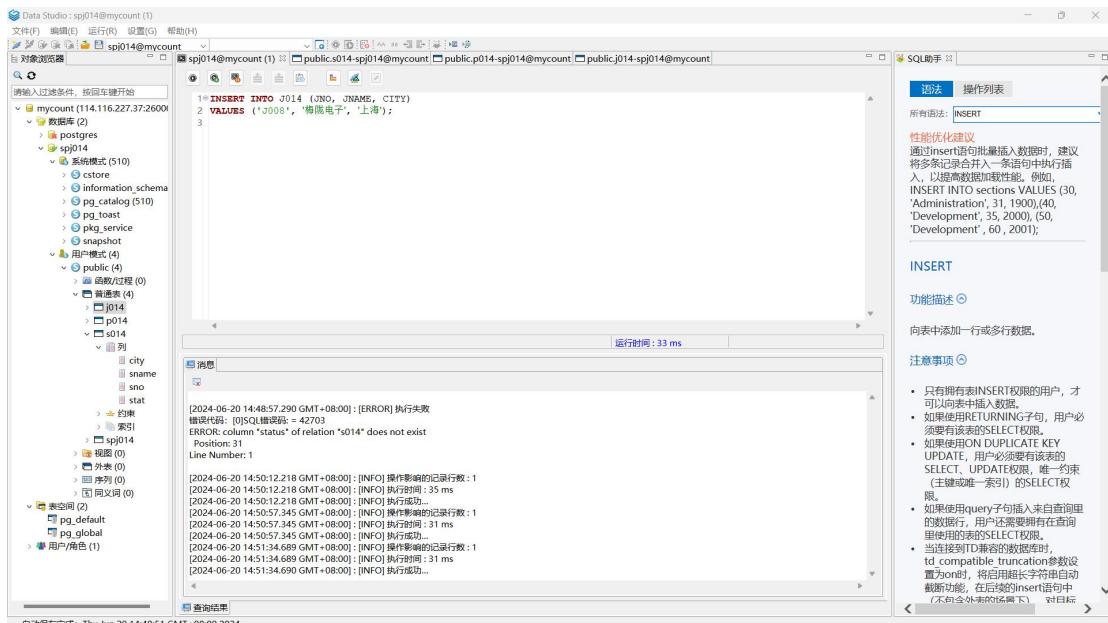
[2024-06-20 14:48:57.290 GMT+08:00] : [ERROR] 执行失败
错误代码: [0SQL] 插入语句 = 42703
ERROR: column "status" of relation "s014" does not exist
Position: 31
Line Number: 1

[2024-06-20 14:48:57.290 GMT+08:00] : [INFO] 操作影响的记录行数: 1
[2024-06-20 14:48:57.290 GMT+08:00] : [INFO] 执行时间: 35 ms
[2024-06-20 14:48:57.290 GMT+08:00] : [INFO] 执行成功...
[2024-06-20 14:48:57.290 GMT+08:00] : [INFO] 操作影响的记录行数: 1
[2024-06-20 14:48:57.290 GMT+08:00] : [INFO] 执行时间: 31 ms
[2024-06-20 14:48:57.290 GMT+08:00] : [INFO] 执行成功...
```

插入一个零件“开关”的信息，编号：“P008”，颜色“红”。重量：40；



(3)



插入一个“上海”的工程项目“梅陇电子”的信息，编号：“J008”；

The screenshot shows the Data Studio interface for the 'mycount' database. On the left, the object browser displays tables like 'jno', 'jname', and 'city'. A table view shows data for these columns. On the right, the SQL editor has 'INSERT' selected in the dropdown. The code input field contains:

```
1= INSERT INTO S014 (SNO, ENO, JNO, QTY)
2 VALUES ('S008', 'E008', 'J008', 500);
```

The status bar at the bottom indicates the query was submitted at 2024-06-20 14:51:40.054 GMT+0 and took 8ms.

(4)

The screenshot shows the Data Studio interface after the insertion attempt. The SQL editor now displays an error message in the results pane:

```
ERROR: column "status" of relation "s014" does not exist
Position: 31
Line Number: 1
```

The status bar at the bottom indicates the query was submitted at 2024-06-20 14:48:51 GMT+08:00 and took 49ms.

插入“S008” 供应项目“J008” 零件“P008”的信息，供应量为 500;

截屏展示了 Data Studio 的界面，显示了对 mycount 数据库的连接。左侧是对象浏览器，右侧是 SQL 助手窗口。

**SQL 助手窗口内容：**

```
INSERT INTO sections VALUES (30, 'Administration', 31, 1900), (40, 'Development', 35, 2000), (50, 'Development', 60, 2001);
```

**注意事项：**

- 只有具有表 INSERT 权限的用户，才可向该表中插入数据。
- 如果使用 RETURNING 子句，用户必须要有该表的 SELECT 权限。
- 如果使用 ON DUPLICATE KEY UPDATE，用户必须要有该表的 SELECT、UPDATE 权限，唯一约束（主键或唯一索引）的 SELECT 权限。
- 如果使用 query 子句插入来自自查询语句的执行行，用户还需要拥有在查询语句使用的表的 SELECT 权限。
- 当连接到 TD 演示数据库时，td\_compatible\_truncation 参数设置为 0 时，将启用粗字符串自动截断功能。在后端的 insert 语句中（不包含外带的扬程下），它将禁用。

(5)

截屏展示了 Data Studio 的界面，显示了对 mycount 数据库的连接。左侧是对象浏览器，右侧是 SQL 助手窗口。

**SQL 助手窗口内容：**

```
UPDATE P014
SET color = '黄'
WHERE pno = 'P008';

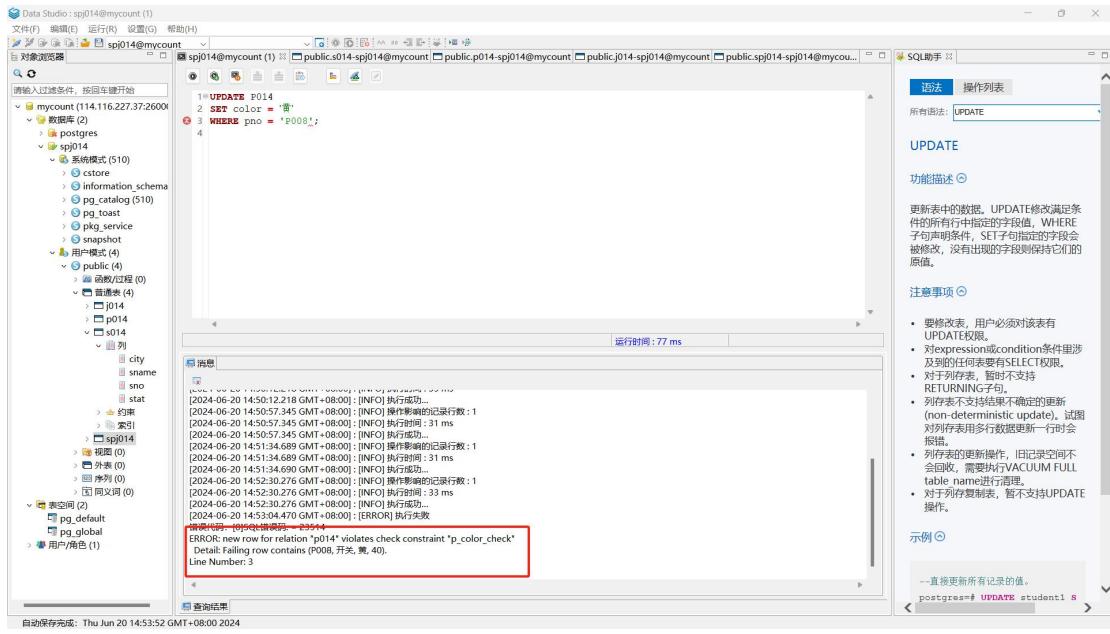
```

**注意事项：**

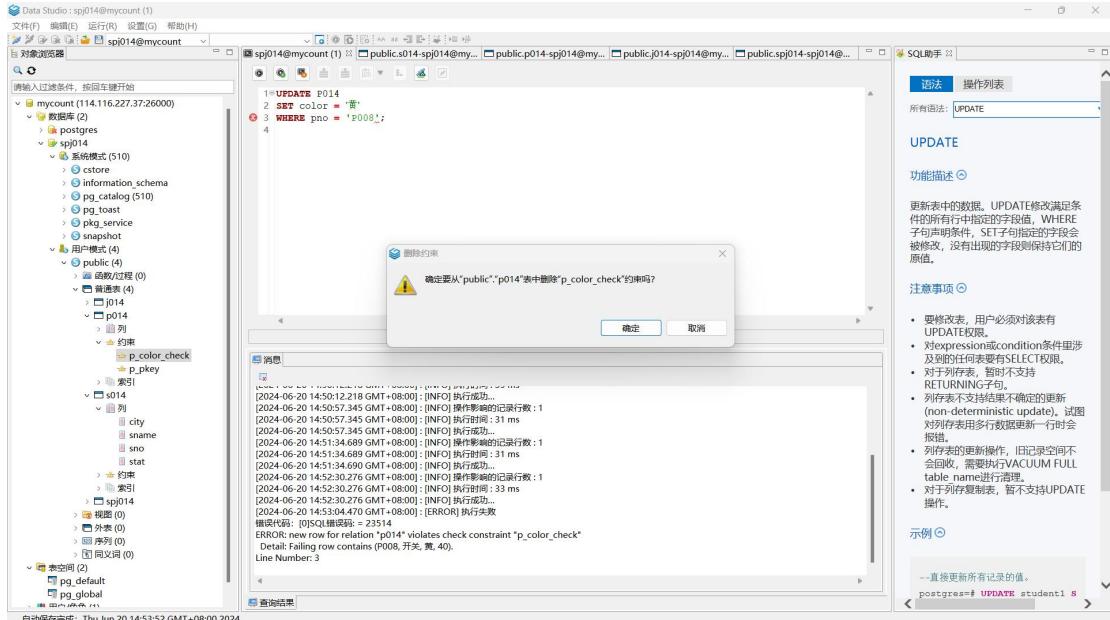
- 要修改表，用户必须对该表有 UPDATE 权限。
- 对 expression 或 condition 条件里涉及的任何表要有 SELECT 权限。
- 对于列存储，暂时不支持 RETURNING 子句。
- 列存表不支持结果不确定的更新 (non-deterministic update)，试图对列存表进行多行数据更新一行时会报错。
- 列存表的更新操作，旧记录空间不会回收，需要执行 VACUUM FULL table\_name 进行清理。
- 对于列存复制表，暂不支持 UPDATE 操作。

**示例：**

```
--直接更新所有记录的值。
postgres=# UPDATE student_s
```



## 需要删除约束条件



将零件“P008”的颜色改为“黄”色;

截图展示了 Data Studio 的界面，显示了数据库 spj014@mycount 中的表 p014。表 p014 包含以下数据：

pno	pname	color	wt
P001	榔母	红	12
P002	螺丝	绿	17
P003	螺丝刀	蓝	14
P004	螺丝刀	红	14
P005	凸轮	蓝	40
P006	齿轮	红	30
P008	开关	黄	40

SQL 助手显示了 UPDATE 语句：

```
UPDATE J014
SET CITY = '合肥'
WHERE JNO = 'J008';
```

查询提交时间：2024-06-20 14:54:49.228 GM 获取7行，共176 ms

(6)

截图展示了 Data Studio 的界面，显示了数据库 spj014@mycount 中的表 p014。SQL 助手显示了以下错误信息：

```
UPDATE J014
SET CITY = '合肥'
WHERE JNO = 'J008';
4
[2024-06-20 14:54:54.600 GMT+08:00] [INFO] 执行成功。
[2024-06-20 14:52:30.276 GMT+08:00] [INFO] 检查受影响的记录行数 : 1
[2024-06-20 14:52:30.276 GMT+08:00] [INFO] 执行时间 : 33 ms
[2024-06-20 14:52:30.276 GMT+08:00] [INFO] 执行成功。
[2024-06-20 14:53:04.470 GMT+08:00] [ERROR] 执行失败。
错误代码: [0]SQL 错误码 = 23514
错误消息: "DETAIL: 不能插入行。违反了约束条件“p_color_check”。
违反约束: public.p014_color_check"
Details: Failing row contains (P008, 开关, 黄, 40)。
Line Number: 3
```

[2024-06-20 14:54:57.414 GMT+08:00] [INFO] 操作影响的记录行数 : 1
[2024-06-20 14:54:57.414 GMT+08:00] [INFO] 执行时间 : 31 ms
[2024-06-20 14:55:05.532 GMT+08:00] [INFO] 检查受影响的记录行数 : 1
[2024-06-20 14:55:05.532 GMT+08:00] [INFO] 执行时间 : 29 ms
[2024-06-20 14:55:05.532 GMT+08:00] [INFO] 执行成功...

将工程“J008”的城市改为“合肥”；

mycount (114.116.227.37:26000)

jno	jname	city
J001	北京三通	北京
J002	长春一汽	长春
J003	新安弹簧厂	天津
J004	临江造船厂	天津
J005	唐山机车厂	唐山
J006	新新光达电厂	常州
J007	铭泰半导体厂	南京
J008	海联电子	合肥

所有语法: UPDATE

- ALTER DATABASE
- ALTER DATA SOURCE
- ALTER DEFAULT PRIVILEGES
- ALTER DIRECTORY
- ALTER FUNCTION
- ALTER GROUP
- ALTER INDEX
- ALTER LARGE OBJECT
- ALTER ROLE
- ALTER ROW LEVEL SECURITY POLICY
- ALTER SCHEMA
- ALTER SEQUENCE
- ALTER SESSION
- ALTER SYNONYM
- ALTER SYSTEM KILL SESSION
- ALTER TABLE
- ALTER TABLE PARTITION
- ALTER TABLESPACE
- ALTER TEXT SEARCH CONFIGURATIC
- ALTER TEXT SEARCH DICTIONARY
- ALTER TRIGGER
- ALTER TYPE
- ALTER USER
- ALTER VIEW
- ANALYZE
- ANALYSE
- BEGIN
- CALL
- CHECKPOINT
- CLOSE

(7)

```

1: UPDATE mycount
2: SET QTY = 300
3: WHERE SNO = 'S008' AND PNO = 'P008' AND JNO = 'J008';
4:

```

运行时间: 38 ms

消息

[2024-06-20 14:54:37.414 GMT+08:00] [INFO] 执行成功...
[2024-06-20 14:54:37.470 GMT+08:00] [ERROR] 执行失败
错误代码: 0
SQL语句行数 = 23514
ERROR: new row violates check constraint 'p\_color\_check'
Detail: Failing row contains (P008,开关,黄,40).
Line Number: 3

[2024-06-20 14:54:37.414 GMT+08:00] [INFO] 操作影响的记录行数: 1
[2024-06-20 14:54:37.414 GMT+08:00] [INFO] 执行时间: 31 ms
[2024-06-20 14:54:37.414 GMT+08:00] [INFO] 执行成功...
[2024-06-20 14:55:05.532 GMT+08:00] [INFO] 操作影响的记录行数: 1
[2024-06-20 14:55:05.532 GMT+08:00] [INFO] 执行时间: 29 ms
[2024-06-20 14:56:19.157 GMT+08:00] [INFO] 操作影响的记录行数: 1
[2024-06-20 14:56:19.157 GMT+08:00] [INFO] 执行时间: 29 ms
[2024-06-20 14:56:19.157 GMT+08:00] [INFO] 执行成功...
[2024-06-20 14:56:19.157 GMT+08:00] [INFO] 执行成功...

查询结果

将“S008”供应商供应工程“J008”零件“P008”的供应量改为300;

mycount (114.116.227.37:26000)

sno	pno	jno	qty
1	S001	P001	100
2	S001	P001	100
3	S001	P001	700
4	S001	P002	100
5	S002	P003	400
6	S002	P003	200
7	S002	P003	500
8	S002	P003	400
9	S002	P005	100
10	S002	P005	100
11	S003	P001	200
12	S003	P003	200
13	S004	P005	100
14	S004	P006	300
15	S004	P006	200
16	S005	P002	100
17	S005	P003	200
18	S005	P006	200
19	S005	P006	500
20	S008	P008	300

(8)

```

1: UPDATE S0014
2: SET QTY = QTY + 100
3: WHERE QTY > 500;
4:

```

运行时间: 32 ms

消息

ERROR: new row for relation "p014" violates check constraint "p\_color\_check"  
Detail: Failing row contains (P008,开关,黄,40).  
Line Number: 3

[2024-06-20 14:54:37.414 GMT+08:00]: [INFO] 操作影响的记录行数: 1  
[2024-06-20 14:54:37.414 GMT+08:00]: [INFO] 执行时间: 31 ms  
[2024-06-20 14:54:37.414 GMT+08:00]: [INFO] 操作影响的记录行数: 1  
[2024-06-20 14:55:05.532 GMT+08:00]: [INFO] 操作影响的记录行数: 1  
[2024-06-20 14:55:05.532 GMT+08:00]: [INFO] 执行时间: 29 ms  
[2024-06-20 14:56:19.157 GMT+08:00]: [INFO] 操作影响的记录行数: 1  
[2024-06-20 14:56:19.157 GMT+08:00]: [INFO] 执行时间: 29 ms  
[2024-06-20 14:57:41.965 GMT+08:00]: [INFO] 操作影响的记录行数: 1  
[2024-06-20 14:57:41.965 GMT+08:00]: [INFO] 执行时间: 31 ms  
[2024-06-20 14:57:41.965 GMT+08:00]: [INFO] 执行成功...

修改供应量大于 500 的供应详情，将供应量再增加 100；

截屏展示了 Data Studio 的界面，显示了数据库 mycount 中的表 spj014。在 SQL 助手窗口中执行了以下 UPDATE 语句：

```
UPDATE SPJ014
SET QTY = QTY / 2
WHERE JNO = 'J001' AND QTY = (SELECT MAX(QTY) FROM SPJ014 WHERE JNO = 'J001');
```

语句成功执行，返回了 20 行受影响的记录。右侧的操作列表显示了可用的 SQL 命令。

sno	pno	jno	qty	
1	S001	P001	J001	200
2	S001	P001	J003	100
3	S001	P002	J002	100
4	S002	P003	J001	400
5	S002	P003	J002	200
6	S002	P003	J004	500
7	S002	P003	J005	400
8	S002	P005	J001	400
9	S002	P005	J002	100
10	S003	P001	J001	200
11	S003	P003	J001	200
12	S004	P005	J001	100
13	S004	P006	J003	300
14	S004	P006	J004	200
15	S005	P002	J004	100
16	S005	P003	J001	200
17	S005	P006	J002	200
18	S005	P006	J004	500
19	S008	P008	J008	300
20	S001	P001	J004	800

(9)

截屏展示了 Data Studio 的界面，显示了数据库 mycount 中的表 spj014。在 SQL 助手窗口中执行了以下 UPDATE 语句：

```
UPDATE SPJ014
SET QTY = QTY / 2
WHERE JNO = 'J001' AND QTY = (SELECT MAX(QTY) FROM SPJ014 WHERE JNO = 'J001');
```

语句成功执行，返回了 20 行受影响的记录。右侧的操作列表显示了可用的 SQL 命令。

将“J001”项目的最大供应量降为原来的一半；

The screenshot shows a database interface with a sidebar containing schema and table structures. A main window displays a table with columns: sno, pno, jno, and qty. The data consists of 20 rows. The last two rows, where jno is 200, are highlighted with a red box.

	sno	pno	jno	qty
1	S001	P001	J001	200
2	S001	P001	J003	100
3	S001	P002	J002	100
4	S002	P003	J002	200
5	S002	P003	J004	500
6	S002	P003	J005	400
7	S002	P005	J002	100
8	S003	P001	J001	200
9	S003	P003	J001	200
10	S004	P005	J001	100
11	S004	P006	J003	300
12	S004	P006	J004	200
13	S005	P002	J004	100
14	S005	P003	J001	200
15	S005	P006	J002	200
16	S005	P006	J004	500
17	S008	P008	J008	300
18	S001	P001	J004	800
19	S002	P003	J001	200
20	S002	P005	J001	200

(10)

The screenshot shows a database interface with a sidebar containing schema and table structures. A main window displays a SQL editor with the following code:

```
1 DELETE FROM SPJ014
2 WHERE JNO = 'J008';
```

The message panel shows the log of the executed command:

```
2024-06-20 14:58:49.318 GM获取20行, 共177 ms
无法编辑查询结果。
运行时间: 32 ms
```

删除项目“J008”的所有供应信息;

mycount (114.116.227.37:26000)

sno	pno	jno	qty
1 S001	P001	J001	200
2 S001	P001	J003	100
3 S001	P002	J002	100
4 S002	P003	J002	200
5 S002	P003	J004	500
6 S002	P003	J005	400
7 S002	P005	J002	100
8 S003	P001	J001	200
9 S003	P003	J001	200
10 S004	P005	J001	100
11 S004	P006	J003	300
12 S004	P006	J004	200
13 S005	P002	J004	100
14 S005	P003	J001	200
15 S005	P006	J002	200
16 S005	P006	J004	500
17 S001	P001	J004	800
18 S002	P003	J001	200
19 S002	P005	J001	200

(11)

```

1: DELETE FROM s014
2: WHERE sno = 'S008';
3:
4: DELETE FROM s014
5: WHERE sno = 'S008';
6:

```

运行时间: 64 ms

消息

```

2024-06-20 14:56:56.9157 GMT+08:00 [INFO] 执行成功。
2024-06-20 14:57:41.3652 GMT+08:00 [INFO] 操作影响的记录行数: 1
2024-06-20 14:57:41.9652 GMT+08:00 [INFO] 执行时间: 31 ms
2024-06-20 14:57:41.9652 GMT+08:00 [INFO] 执行成功。
2024-06-20 14:58:36.9442 GMT+08:00 [INFO] 操作影响的记录行数: 2
2024-06-20 14:58:36.9442 GMT+08:00 [INFO] 执行时间: 35 ms
2024-06-20 14:58:36.9442 GMT+08:00 [INFO] 执行成功。
2024-06-20 14:58:36.9442 GMT+08:00 [INFO] 操作影响的记录行数: 1
2024-06-20 14:58:36.9442 GMT+08:00 [INFO] 执行时间: 28 ms
2024-06-20 15:02:06.1792 GMT+08:00 [INFO] 执行成功。
2024-06-20 15:05:09.0192 GMT+08:00 [INFO] 操作影响的记录行数: 0
2024-06-20 15:05:09.0192 GMT+08:00 [INFO] 执行时间: 33 ms
2024-06-20 15:05:09.0192 GMT+08:00 [INFO] 执行成功。
2024-06-20 15:05:09.0512 GMT+08:00 [INFO] 操作影响的记录行数: 1
2024-06-20 15:05:09.0512 GMT+08:00 [INFO] 执行时间: 29 ms
2024-06-20 15:05:09.0512 GMT+08:00 [INFO] 执行成功...

```

删除“S008”供应商信息（注意其供应信息）；

Two screenshots of the MySQL Data Studio interface showing the results of a query on the 'mycount' database.

**Screenshot 1:** The screenshot shows the results of a query on the 'mycount' database. The table 'jno' has 19 rows of data. The columns are sno, pno, jno, and qty.

	sno	pno	jno	qty
1	S001	P001	J001	200
2	S001	P001	J003	100
3	S001	P002	J002	100
4	S002	P003	J002	200
5	S002	P003	J004	500
6	S002	P003	J005	400
7	S002	P005	J002	100
8	S003	P001	J001	200
9	S003	P003	J001	200
10	S004	P005	J001	100
11	S004	P006	J003	300
12	S004	P006	J004	200
13	S005	P002	J004	100
14	S005	P003	J001	200
15	S005	P006	J002	200
16	S005	P006	J004	500
17	S001	P001	J004	800
18	S002	P003	J001	200
19	S002	P005	J001	200

**Screenshot 2:** The screenshot shows the results of a query on the 'mycount' database. The table 'city' has 6 rows of data. The columns are sno, sname, stat, and city.

	sno	sname	stat	city
1	S001	天津宝贝儿	B	天津
2	S002	北京丽丽	A	北京
3	S003	北京新天地	C	北京
4	S004	天津丰泰盛	B	天津
5	S005	上海普丰	C	上海
6	S006	合肥四达	B	合肥

(12)

```

1 DELETE FROM SP014
2 WHERE pno = 'P008';
3
4 DELETE FROM P014
5 WHERE pno = 'P008';
6

```

运行时间: 69 ms

消息

```

[2024-06-20 14:58:36.944 GMT+08:00]: [INFO] 执行成功...
[2024-06-20 15:02:06.179 GMT+08:00]: [INFO] 操作影响的记录行数: 1
[2024-06-20 15:02:06.179 GMT+08:00]: [INFO] 执行成功...
[2024-06-20 15:05:09.019 GMT+08:00]: [INFO] 操作影响的记录行数: 0
[2024-06-20 15:05:09.019 GMT+08:00]: [INFO] 执行成功...
[2024-06-20 15:05:09.019 GMT+08:00]: [INFO] 执行成功...
[2024-06-20 15:05:09.051 GMT+08:00]: [INFO] 操作影响的记录行数: 1
[2024-06-20 15:05:09.051 GMT+08:00]: [INFO] 执行成功...
[2024-06-20 15:05:50.028 GMT+08:00]: [INFO] 操作影响的记录行数: 0
[2024-06-20 15:05:50.028 GMT+08:00]: [INFO] 执行时间: 30 ms
[2024-06-20 15:05:50.030 GMT+08:00]: [INFO] 执行成功...
[2024-06-20 15:05:50.060 GMT+08:00]: [INFO] 操作影响的记录行数: 1
[2024-06-20 15:05:50.060 GMT+08:00]: [INFO] 执行时间: 32 ms
[2024-06-20 15:05:50.060 GMT+08:00]: [INFO] 执行成功...

```

查询结果

删除零件“P008”的信息（注意其供应信息）；

sno	pno	jno	qty
1	S001	P001	200
2	S001	P001	100
3	S001	P002	100
4	S002	P003	200
5	S002	P003	500
6	S002	P003	400
7	S002	P005	100
8	S003	P001	200
9	S003	P003	200
10	S004	P005	100
11	S004	P006	300
12	S004	P006	200
13	S005	P002	100
14	S005	P003	200
15	S005	P006	200
16	S005	P006	500
17	S001	P001	800
18	S002	P003	200
19	S002	P005	200

查询提交时间: 2024-06-20 15:06:02.754 GM 获取19行, 共152 ms 无法编辑查询结果。

The screenshot shows the Data Studio interface with the following details:

- 左侧树形结构 (Object Browser):** 显示了数据库对象，包括 mycount、spj014、p014 等。
- 中间表格 (Table):** 显示了表 p014 的数据，包含列 pno, pname, color, wt。数据如下：

pno	pname	color	wt
P001	榔母	红	12
P002	螺栓	绿	17
P003	螺丝刀	蓝	14
P004	螺丝刀	红	14
P005	凸轮	蓝	40
P006	齿轮	红	30

- 右侧 SQL 助手 (SQL Assistant):** 显示了所有语法列表，当前选中的是 DELETE 语句。
- 底部状态栏:** 指示查询提交时间为 2024-06-20 15:06:13.804 GM，共耗时 163 ms。

(13)

The screenshot shows the Data Studio interface with the following details:

- 左侧树形结构 (Object Browser):** 显示了数据库对象，包括 mycount、spj014、p014 等。
- 中间 SQL 编辑器 (SQL Editor):** 显示了以下 DELETE 语句：

```

1: DELETE FROM p014
2: WHERE JNO = 'J008';
3:
4: DELETE FROM J014
5: WHERE JNO = 'J008';
6:

```

- 右侧 SQL 助手 (SQL Assistant):** 显示了执行结果，运行时间为 74 ms，并输出了系统日志，显示了删除操作的详细信息。
- 底部状态栏:** 指示自动保存完成于 2024-06-20 15:18:52 GMT+08:00。

删除项目“J008”的信息（注意其供应信息）；

Two screenshots of the Data Studio interface showing database structures and data.

**Screenshot 1:** Database structure for 'mycount' (114.116.227.37:26000). The tree view shows the schema 'mycount' containing databases (2), roles (2), and tables (2). One table, 'jname', is displayed in a grid:

	jno	jname	city
1	J001	北京三玻	北京
2	J002	长春一汽	长春
3	J003	新安弹簧厂	天津
4	J004	临江造船厂	天津
5	J005	唐山机车厂	唐山
6	J006	新新无线电厂	常州
7	J007	长飞半导体厂	南京

**Screenshot 2:** Database structure for 'spj014' (114.116.227.37:26000). The tree view shows the schema 'spj014' containing databases (2), roles (2), and tables (2). One table, 'sno', is displayed in a grid:

	sno	pno	jno	qty
1	S001	P001	J001	200
2	S001	P001	J003	100
3	S001	P002	J002	100
4	S002	P003	J002	200
5	S002	P003	J004	500
6	S002	P003	J005	400
7	S002	P005	J002	100
8	S003	P001	J001	200
9	S003	P003	J001	200
10	S004	P005	J001	100
11	S004	P006	J003	300
12	S004	P006	J004	200
13	S005	P002	J004	100
14	S005	P003	J001	200
15	S005	P006	J002	200
16	S005	P006	J004	500
17	S001	P001	J004	800
18	S002	P003	J001	200
19	S002	P005	J001	200

(14)

```

CREATE TABLE SupplierStats (
    MaxQty INT,
    MinQty INT,
    AvgQty DECIMAL(10,2)
);

INSERT INTO SupplierStats (MaxQty, MinQty, AvgQty)
SELECT MAX(QTY), MIN(QTY), AVG(QTY)
FROM SPJ014;

```

自动保存完成: Thu Jun 20 15:18:52 GMT+08:00 2024

统计所有供应商供应量的最大、最小和平均值，保存到一张表中（先建表）。

	maxqty	minqty	avgqty
1	800	100	252.63

查询提交时间: 2024-06-20 15:21:50.854 GM 获取1行, 共168 ms 无法编辑查询结果。

(1)

The screenshot shows the Data Studio interface with the following details:

- 左侧树形结构:** 显示了数据库 schema 和表的树状目录。
- 中间 SQL 编辑器:**

```

CREATE VIEW BJ_S AS
SELECT *
FROM S014
WHERE CITY = '北京';
    
```
- 右侧操作列表:** 列出了许多 PostgreSQL 的 DDL 指令，如 ALTER DATABASE, ALTER TABLE 等。
- 下方消息窗口:** 显示了执行语句的日志信息。
- 底部状态栏:** 自动保存时间为 Thu Jun 20 15:23:52 GMT+08:00 2024。

建立“北京”供应商视图 BJ\_S；

The screenshot shows the Data Studio interface with the following details:

- 左侧树形结构:** 显示了数据库 schema 和表的树状目录。
- 中间结果集:**

SNO	sname	stat	city
S002	北京启明星	A	北京
S003	北京新天地	C	北京
- 右侧操作列表:** 显示了 CREATE VIEW 的操作。
- 下方功能描述:** 提供了关于视图创建的说明文字。
- 示例:** 提供了一个使用 spcname=pg\_default 创建视图的示例。
- 语法格式:** 显示了 CREATE VIEW 的语法格式。
- 参数说明:** 显示了视图参数的说明。
- 底部状态栏:** 查询提交时间为 2024-06-20 15:26:03.711 GM，共耗时 182 ms。

(2)

```

CREATE VIEW BJ_J AS
SELECT *
FROM J014
WHERE CITY = '北京';

```

运行时间: 67 ms

消息

```

[2024-06-20 15:18:52.947 GMT+08:00]: [INFO] 执行成功...
[2024-06-20 15:18:52.947 GMT+08:00]: [INFO] 操作影响的记录行数: 1
[2024-06-20 15:18:52.980 GMT+08:00]: [INFO] 执行成功...耗时: 33 ms
[2024-06-20 15:18:52.980 GMT+08:00]: [INFO] 执行成功...
[2024-06-20 15:21:24.448 GMT+08:00]: [INFO] 操作影响的记录行数: 0
[2024-06-20 15:21:24.448 GMT+08:00]: [INFO] 执行时间: 43 ms
[2024-06-20 15:21:24.448 GMT+08:00]: [INFO] 执行成功...
[2024-06-20 15:21:24.448 GMT+08:00]: [INFO] 执行成功...
[2024-06-20 15:26:37.718 GMT+08:00]: [INFO] 操作影响的记录行数: 1
[2024-06-20 15:26:37.718 GMT+08:00]: [INFO] 执行时间: 28 ms
[2024-06-20 15:25:12.478 GMT+08:00]: [INFO] 执行成功...
[2024-06-20 15:25:12.478 GMT+08:00]: [INFO] 执行成功...
[2024-06-20 15:25:42.236 GMT+08:00]: [INFO] 执行时间: 37 ms
[2024-06-20 15:25:42.236 GMT+08:00]: [INFO] 执行成功...
[2024-06-20 15:26:37.718 GMT+08:00]: [INFO] 操作影响的记录行数: 0
[2024-06-20 15:26:37.718 GMT+08:00]: [INFO] 执行时间: 32 ms
[2024-06-20 15:26:37.718 GMT+08:00]: [INFO] 执行成功...

```

查询结果

建立“北京”工程视图 BJ\_J；

```

CREATE VIEW BJ_J AS
SELECT *
FROM J014
WHERE CITY = '北京';

```

运行时间: 186 ms

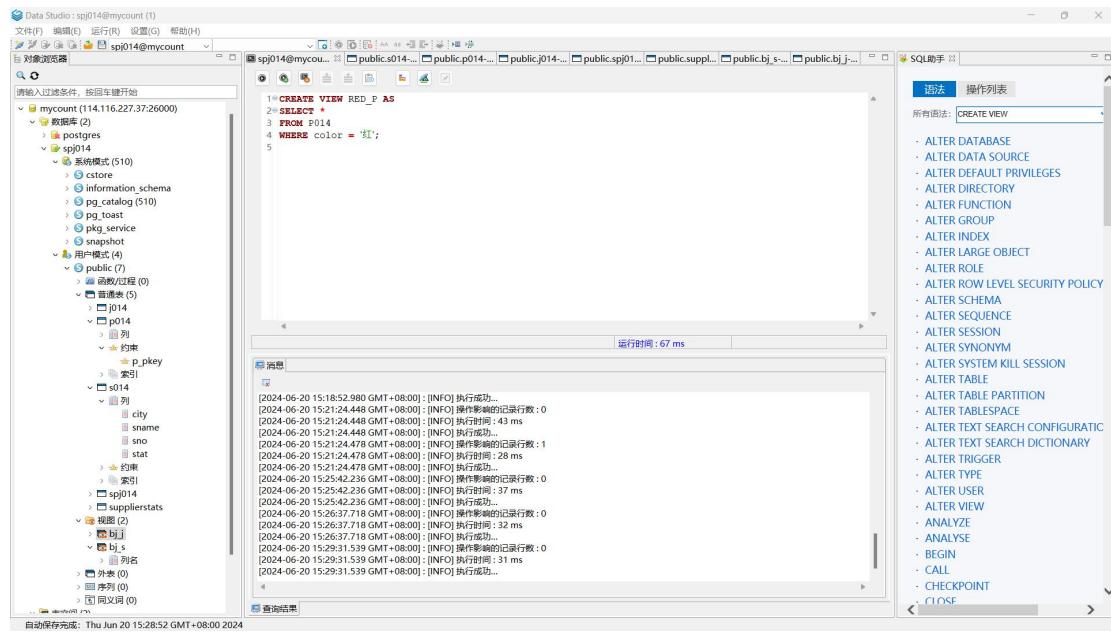
消息

```

[2024-06-20 15:26:43.432 GM获取1行, 共186 ms]
无法编辑查询结果。

```

(3)



建立“红”色零件视图

RED\_P;

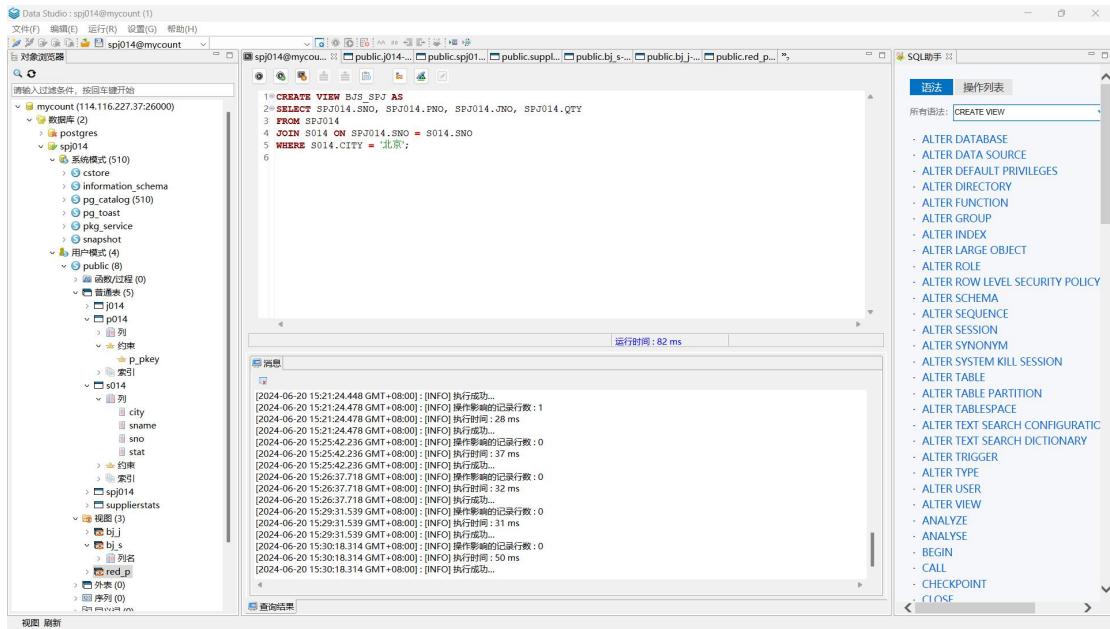
```

CREATE VIEW RED_P AS
SELECT *
FROM P014
WHERE color = '红';

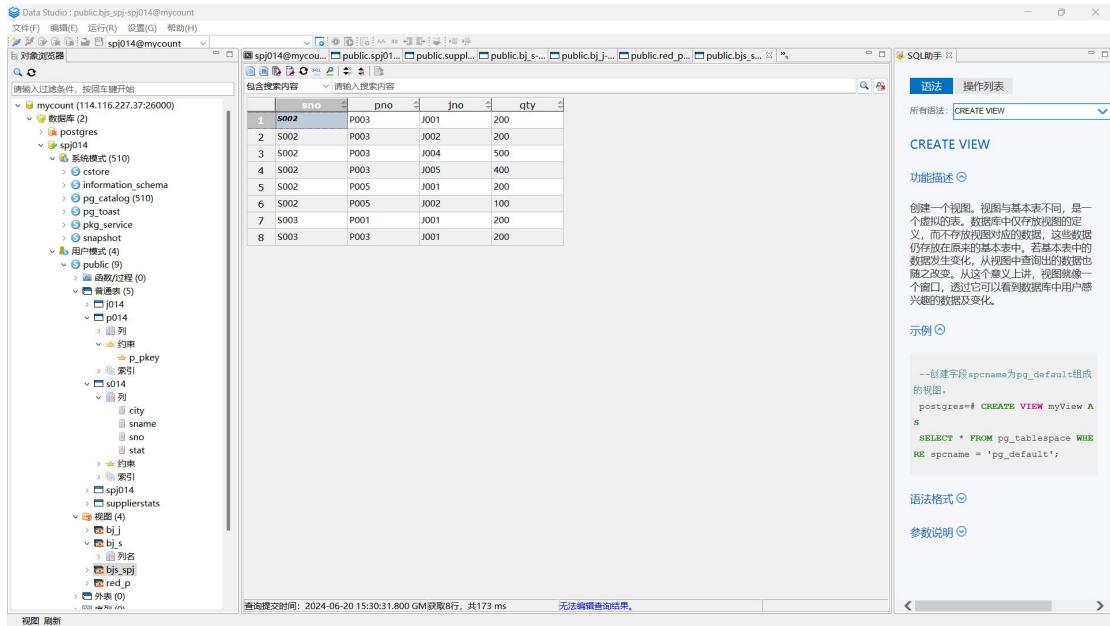
```

pno	pname	color	wt
P001	螺母	红	12
P004	螺丝刀	红	14
P006	齿轮	红	30

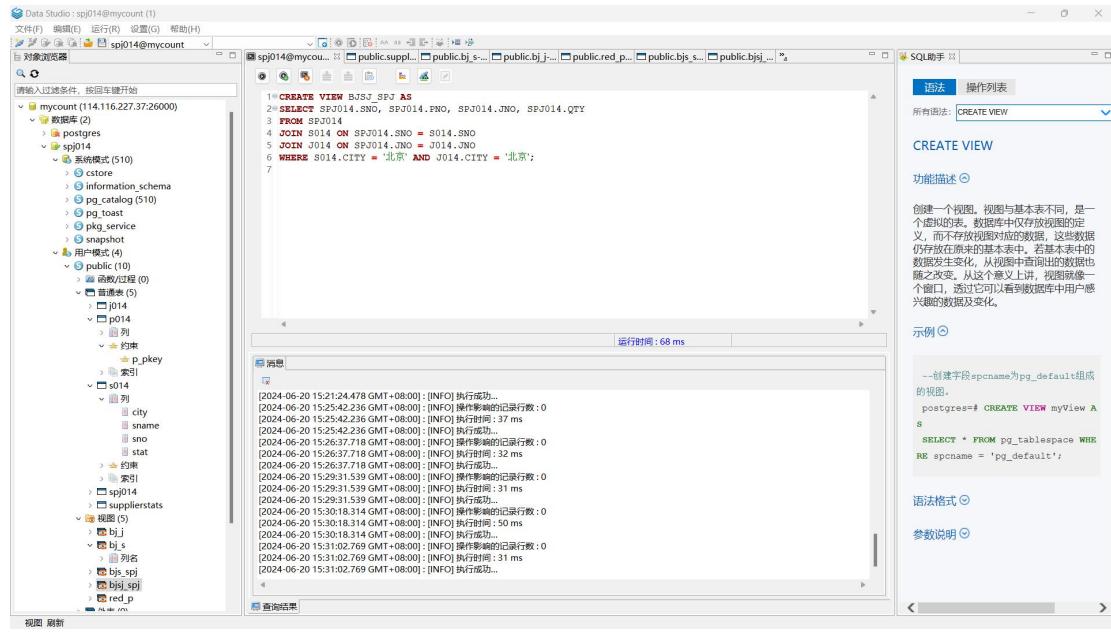
(4)



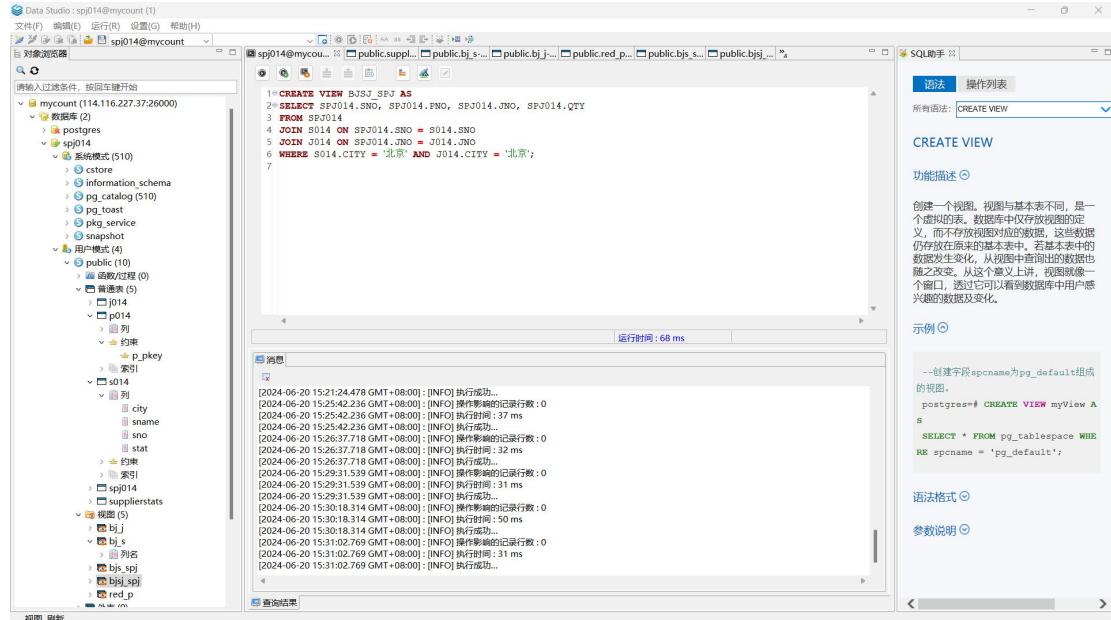
## 建立“北京”供应商的供应情况视图 BJS\_SPJ;



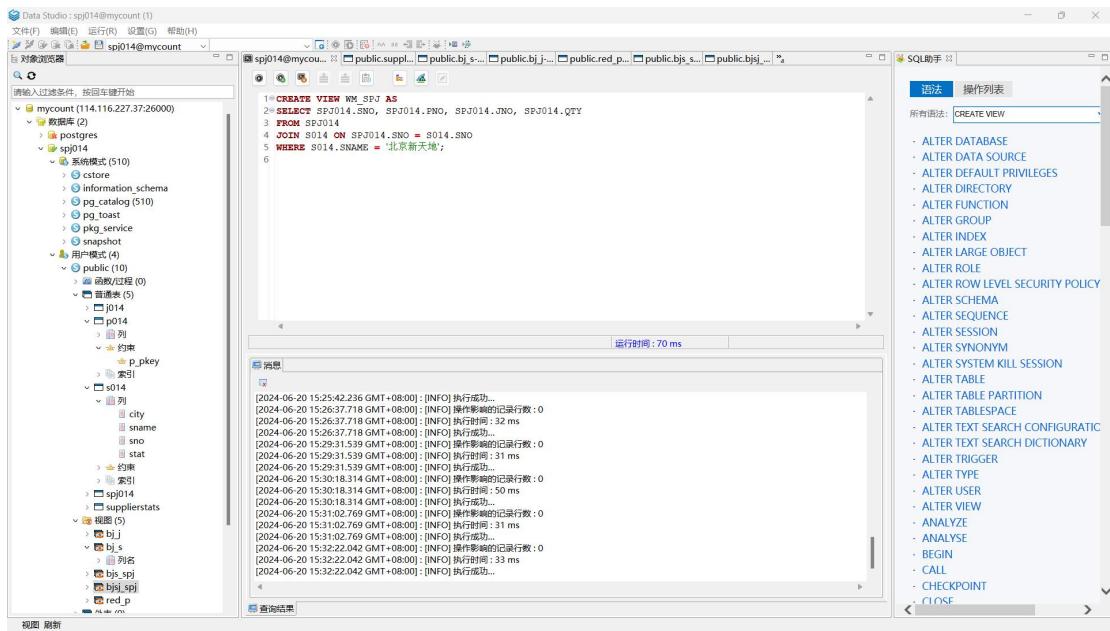
(5)



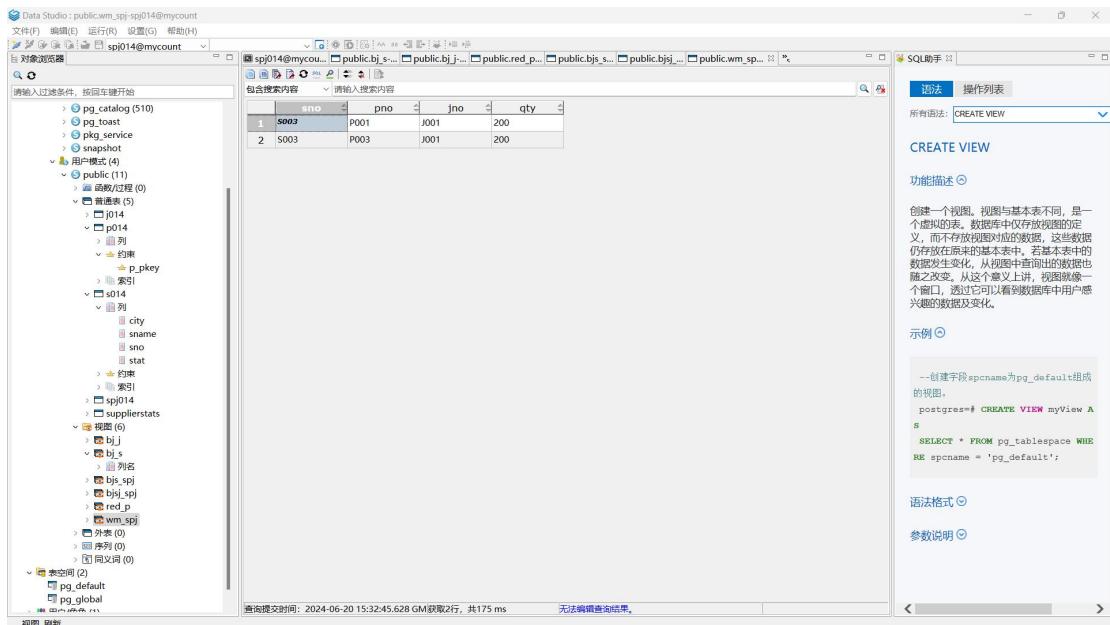
建立“北京”供应商供应“北京”工程的供应情况视图 BJSJ\_SPJ；



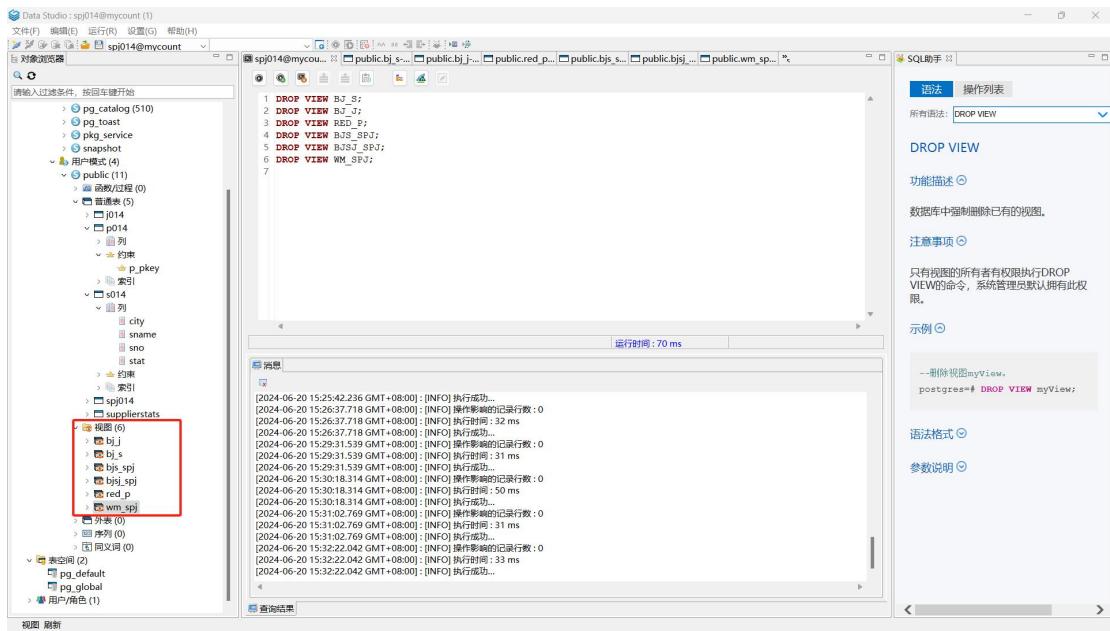
(6)



建立“北京新天地”供应商的供应情况视图 WM\_SPJ；

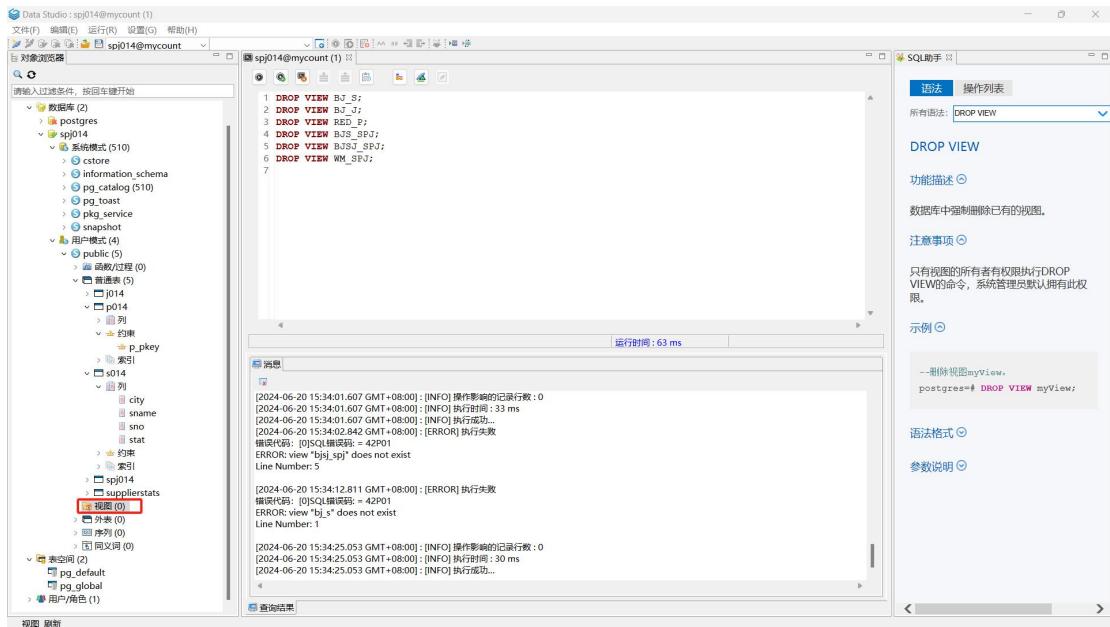


(7)



删除前要先关闭

分别删除以上定义的各个视图。



## 【小结或讨论】

通过本次实验，我们系统地学习并实践了数据更新操作和视图管理操作。在数据更新部分，我们掌握了 INSERT、UPDATE、DELETE 语句的基本语法及其应用。这些操作使我们能够灵活地插入、修改和删除数据，并确保数据的完整性和一致性。通过实际操作，我们熟悉了插入元组和插入子查询结果的两种方式，

体会到不同插入方式的灵活性和应用场景。此外，我们学习了如何在数据库中修改数据并确保不会破坏既有的完整性规则，进一步增强了对 SQL 语句中 SET 和 WHERE 子句的理解和运用。

在视图管理部分，我们了解了视图的基本概念和作用，并掌握了创建和删除视图的方法。视图作为一种虚表，不仅能够简化用户操作、以多种角度查看数据，还能对机密数据提供安全保护。通过创建各种视图，如北京供应商视图、红色零件视图、以及特定供应商和工程项目的视图，我们体会到视图在实际应用中的便利性和重要性。同时，通过对视图的查询和更新操作，我们深入理解了视图在数据操作中的局限性和适用范围。

实验的操作步骤详细涵盖了从插入新的供应商、零件、工程项目信息，到修改和删除现有数据，再到统计供应量的最大、最小和平均值的全过程。这些操作不仅加深了我们对 SQL 语句的理解和熟练程度，也帮助我们更好地掌握了数据库管理的基本技能。尤其是视图的创建和管理，使我们能够灵活地从不同角度查看和操作数据，提高了数据处理的效率和安全性。

在本次实验中，通过实际操作，我不仅巩固了 SQL 语句的使用技巧，还进一步理解了数据库管理中的重要概念和操作方式。实验内容紧密结合实际应用，具有很强的实践意义，为今后更复杂的数据库操作和管理打下了坚实的基础。这次实验不仅提高了我的数据处理能力，也增强了我对数据库系统整体运作的理解和掌控力。