数据库系统课程实验报告

实验名称：	数据更新
实验日期：	2022.11.25
实验地点：	四号楼
提交日期：	2022.11.28

学号：	22920202202877
姓名：	陈鑫蕾
专业年级：	数媒 2020 级
学年学期：	2022-2023 学年第一学期

1. 实验目的

- 熟练掌握单条记录和小批量数据插入的方法（INSERT）
- 熟练掌握使用子查询实现数据插入的方法(INSERT INTO…

SUBQUERY)

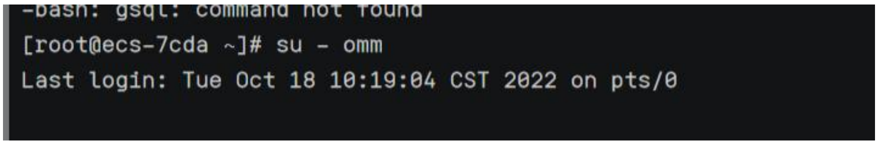
- 熟练掌握数据修改和删除的方法（UPDATE,DELETE,TRUNCATE）

2. 实验内容和步骤

（1）连接到数据库

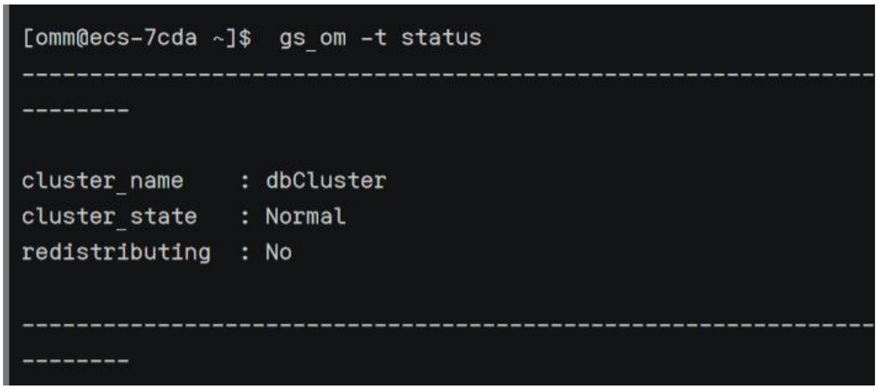
步骤如下:

1.在数据库主节点服务器上，切换至omm操作系统用户环境。



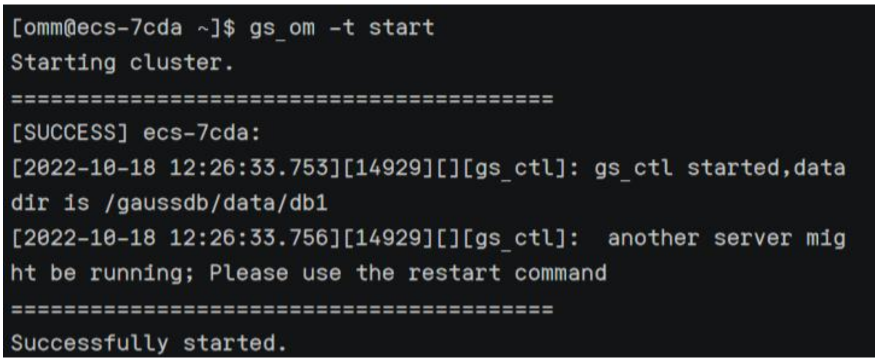
```
-bash: gsql: command not found
[root@ecs-7cda ~]# su - omm
Last login: Tue Oct 18 10:19:04 CST 2022 on pts/0
```

2.查看服务是否启动。



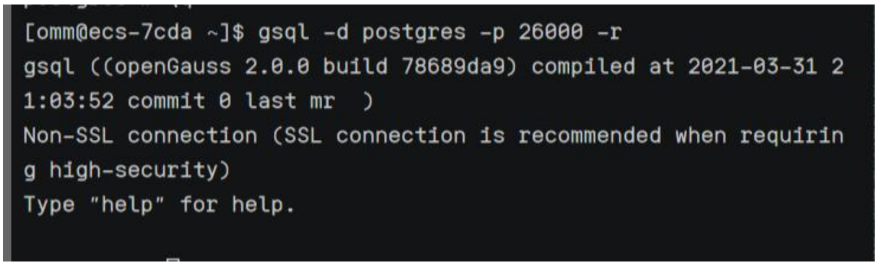
```
[omm@ecs-7cda ~]$  gs_om -t status
-----------------------------------------------------------------
--------

cluster_name    : dbCluster
cluster_state   : Normal
redistributing  : No

-----------------------------------------------------------------
--------
```

3.启动数据库服务



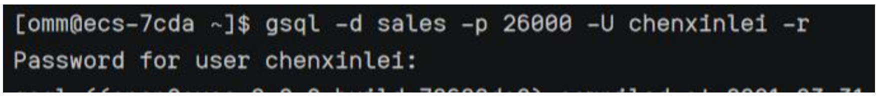
```
[omm@ecs-7cda ~]$ gs_om -t start
Starting cluster.
=========================================
[SUCCESS] ecs-7cda:
[2022-10-18 12:26:33.753][14929][][gs_ctl]: gs_ctl started,data
dir is /gaussdb/data/db1
[2022-10-18 12:26:33.756][14929][][gs_ctl]:  another server mig
ht be running; Please use the restart command
=========================================
Successfully started.
```

4.连接数据库



```
[omm@ecs-7cda ~]$ gsql -d postgres -p 26000 -r
gsql ((openGauss 2.0.0 build 78689da9) compiled at 2021-03-31 2
1:03:52 commit 0 last mr  )
Non-SSL connection (SSL connection is recommended when requirin
g high-security)
Type "help" for help.
```

5. 使用自己创建的用户连接到此数据库



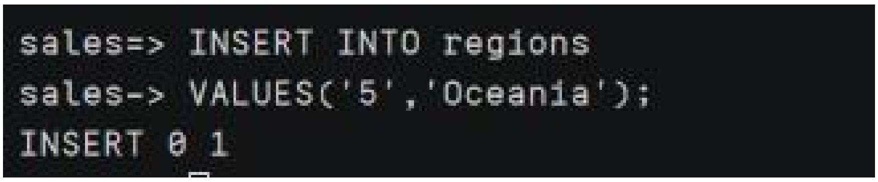
```
[omm@ecs-7cda ~]$ gsql -d sales -p 26000 -U chenxinlei -r
Password for user chenxinlei:
```

（2）数据更新

1.为地区表 regions 新增一条记录：（‘5’,’Oceania’）。

INSERT INTO regions

VALUES('5','Oceania');



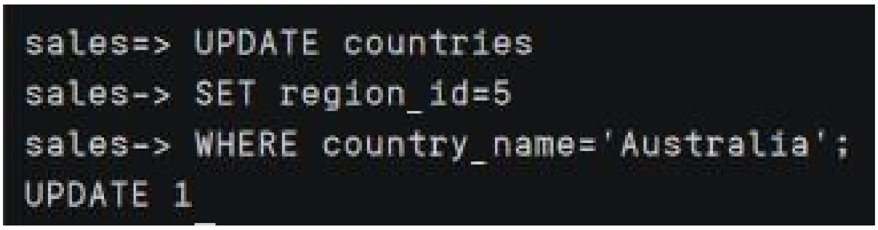
```
sales=> INSERT INTO regions
sales-> VALUES('5','Oceania');
INSERT 0 1
```

2.将 countries 表中的国家名为 Austrialia 的 region_id 改为 5。

UPDATE countries

SET region_id=5

WHERE country_name='Australia';



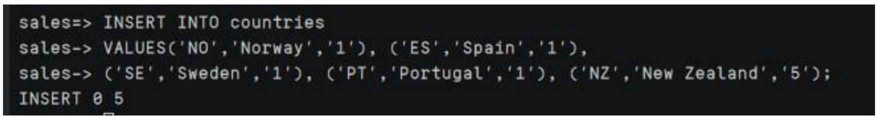
```
sales=> UPDATE countries
sales-> SET region_id=5
sales-> WHERE country_name='Australia';
UPDATE 1
```

3.使用一条批量插入数据语句为 countries 表新增 5 条记录：('NO','Norway','1'), ('ES','Spain','1'), ('SE','Sweden','1'), ('PT','Portugal','1'), ('NZ','New Zealand','5')。

INSERT INTO countries

VALUES('NO','Norway','1'), ('ES','Spain','1'),

('SE','Sweden','1'), ('PT','Portugal','1'), ('NZ','New Zealand','5');



```
sales=> INSERT INTO countries
sales-> VALUES('NO','Norway','1'), ('ES','Spain','1'),
sales-> ('SE','Sweden','1'), ('PT','Portugal','1'), ('NZ','New Zealand','5');
INSERT 0 5
```

4.创建一张名为 Asia_countries(country_id,country_name)的新表，其中字段为 countries 表中的同名字段。

CREATE TABLE Asia_countries

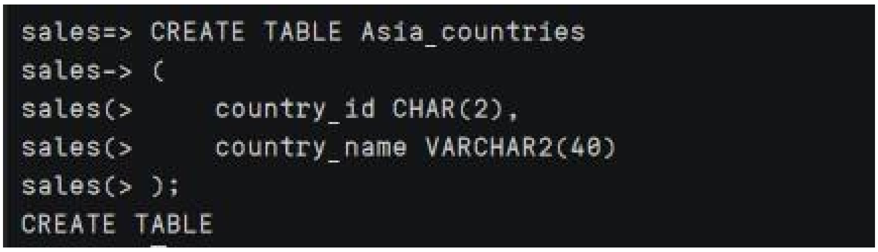
(

    country_id CHAR(2),

    country_name VARCHAR2(40),

    region_id NUMBER

);



```
sales=> CREATE TABLE Asia_countries
sales-> (
sales(>     country_id CHAR(2),
sales(>     country_name VARCHAR2(40)
sales(> );
CREATE TABLE
```

5.将 countries 表中所有亚洲国家的数据插入到该表中。

INSERT INTO Asia_countries

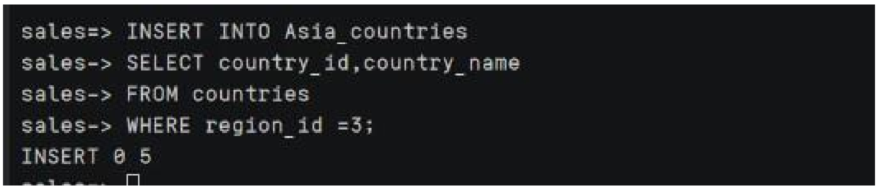
SELECT *

FROM countries

WHERE region_id IN SELECT *

FROM regions

WHERE region_name='Asia';



```
sales=> INSERT INTO Asia_countries
sales-> SELECT country_id,country_name
sales-> FROM countries
sales-> WHERE region_id =3;
INSERT 0 5
```

6.创建一张名为 order_total(order_id,total_price)的视图，该视图存放每个订单号及其总价,其中 total_price 为总价，其值为数量 quantity 与单价 unit_price 乘积之和，order_id，quantity

和 unit_price 为 order_items 表中的同名字段。

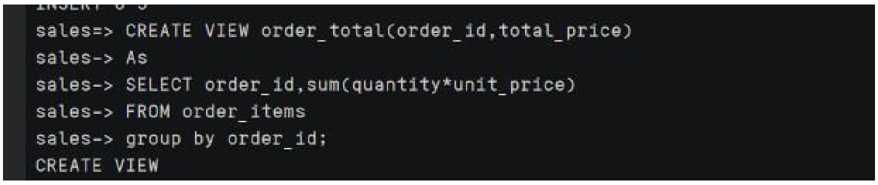
CREATE VIEW order_total(order_id,total_price)

As

SELECT order_id,sum(quantity*unit_price)

FROM order_items

group by order_id;



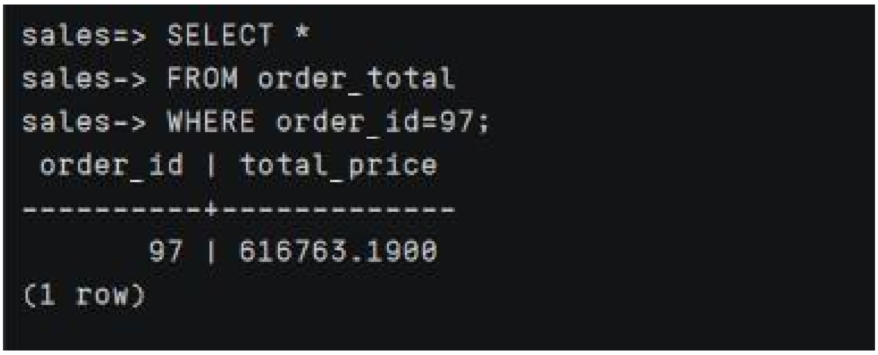
```
sales=> CREATE VIEW order_total(order_id,total_price)
sales-> As
sales-> SELECT order_id,sum(quantity*unit_price)
sales-> FROM order_items
sales-> group by order_id;
CREATE VIEW
```

7.查询 order_total 视图中订单号 order_id 为 97 的总价并记录该结果。

SELECT *

FROM order_total

WHERE order_id=97;



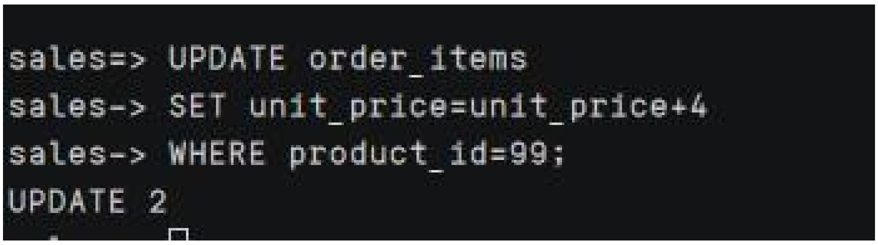
```
sales=> SELECT *
sales-> FROM order_total
sales-> WHERE order_id=97;
 order_id | total_price
----------+-------------
       97 | 616763.1900
(1 row)
```

8.将 order_items 表中 product_id 为 99 的单价 unit_price 增加 4 元。

UPDATE order_items

SET unit_price=unit_price+4

WHERE product_id=99;



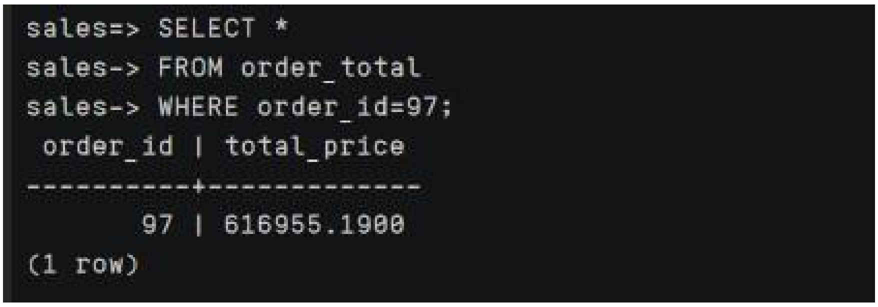
```
sales=> UPDATE order_items
sales-> SET unit_price=unit_price+4
sales-> WHERE product_id=99;
UPDATE 2
```

9. 查询视图 order_total 中订单号 order_id 为 97 的总价，将其与第（7）步的结果进行比较，观察其异同。

SELECT *

FROM order_total

WHERE order_id=97;



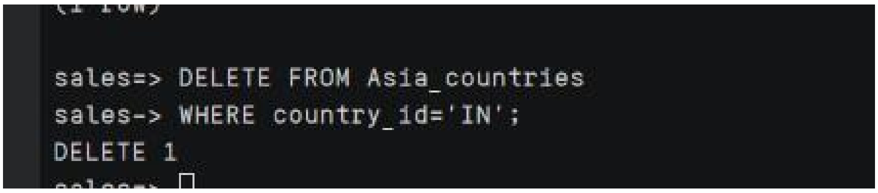
```
sales=> SELECT *
sales-> FROM order_total
sales-> WHERE order_id=97;
 order_id | total_price
----------+-------------
       97 | 616955.1900
(1 row)
```

总价提升，视图会跟着表一起修改

10.使用 delete 命令删除 Asia_countries 表中 country_id 为 IN 的记录。

DELETE FROM Asia_countries

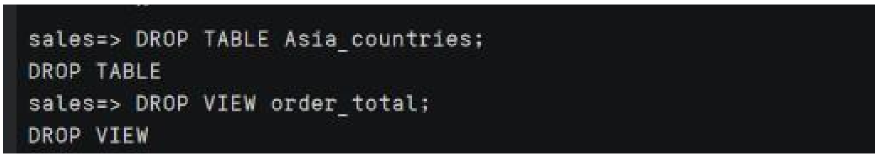
WHERE country_id='IN';



```
(1 row)

sales=> DELETE FROM Asia_countries
sales-> WHERE country_id='IN';
DELETE 1
sales=> 
```

11.使用 truncate 命令清空 Asia_countries 表的所有记录。

truncate Asia_countries;

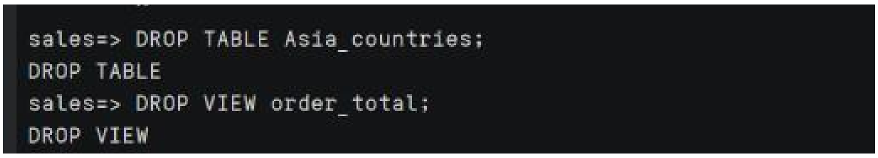


```
sales=> DROP TABLE Asia_countries;
DROP TABLE
sales=> DROP VIEW order_total;
DROP VIEW
```

12.删除 Asia_countries 表和视图 order_total。

DROP TABLE Asia_countries

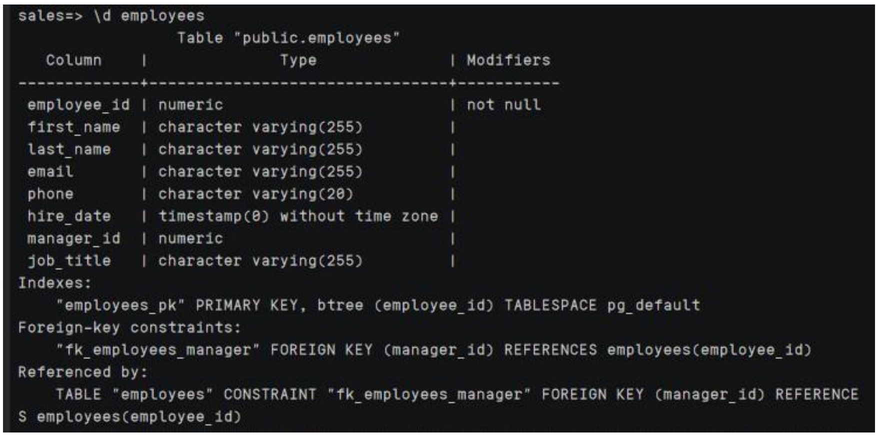
DROP VIEW order_total



```
sales=> DROP TABLE Asia_countries;
DROP TABLE
sales=> DROP VIEW order_total;
DROP VIEW
```

13.使用命令\d employees 查看 employees 表的外码约束语句，包括 on delete cascade 选项。

\d employees



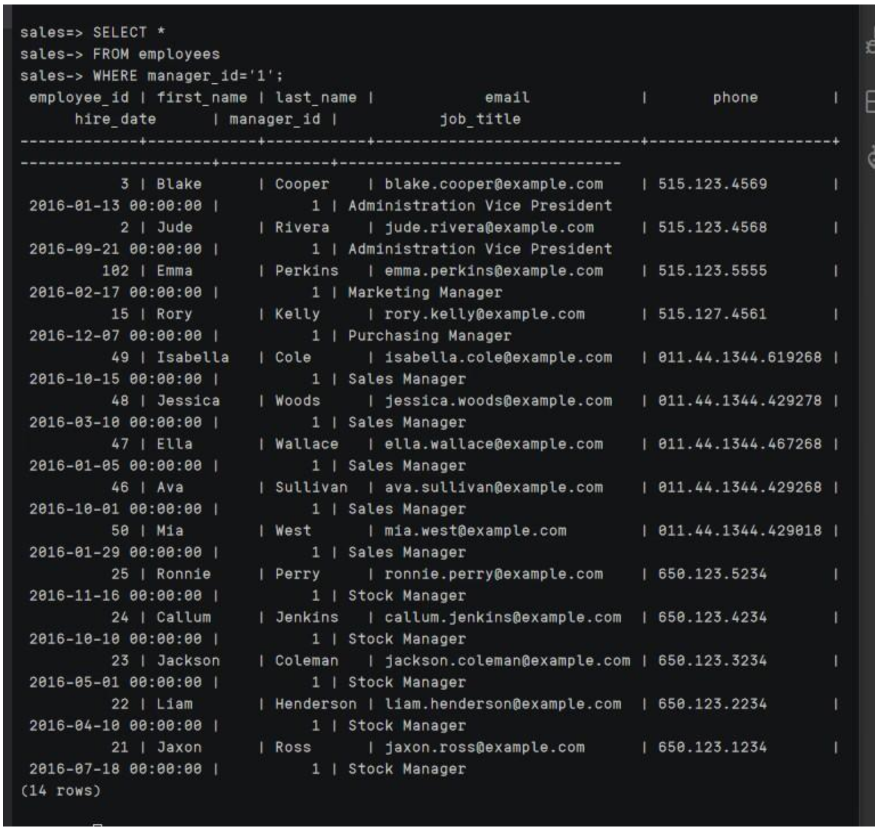
```
sales=> \d employees
                  Table "public.employees"
   Column    |              Type              | Modifiers
-------------+--------------------------------+-----------
 employee_id | numeric                        | not null
 first_name  | character varying(255)         |
 last_name   | character varying(255)         |
 email       | character varying(255)         |
 phone       | character varying(20)          |
 hire_date   | timestamp(0) without time zone |
 manager_id  | numeric                        |
 job_title   | character varying(255)         |
Indexes:
    "employees_pk" PRIMARY KEY, btree (employee_id) TABLESPACE pg_default
Foreign-key constraints:
    "fk_employees_manager" FOREIGN KEY (manager_id) REFERENCES employees(employee_id)
Referenced by:
    TABLE "employees" CONSTRAINT "fk_employees_manager" FOREIGN KEY (manager_id) REFERENCE
S employees(employee_id)
```

14.查询 employees 表中 manager_id 为 1 的记录。

SELECT *

FROM employees

WHERE manager_id='1';



```
sales=> SELECT *
sales-> FROM employees
sales-> WHERE manager_id='1';
 employee_id | first_name | last_name |            email            |       phone        |
      hire_date      | manager_id |          job_title
-------------+------------+-----------+-----------------------------+--------------------+
---------------------+------------+-------------------------------
           3 | Blake      | Cooper    | blake.cooper@example.com    | 515.123.4569       |
 2016-01-13 00:00:00 |          1 | Administration Vice President
           2 | Jude       | Rivera    | jude.rivera@example.com     | 515.123.4568       |
 2016-09-21 00:00:00 |          1 | Administration Vice President
         102 | Emma       | Perkins   | emma.perkins@example.com    | 515.123.5555       |
 2016-02-17 00:00:00 |          1 | Marketing Manager
          15 | Rory       | Kelly     | rory.kelly@example.com      | 515.127.4561       |
 2016-12-07 00:00:00 |          1 | Purchasing Manager
          49 | Isabella   | Cole      | isabella.cole@example.com   | 011.44.1344.619268 |
 2016-10-15 00:00:00 |          1 | Sales Manager
          48 | Jessica    | Woods     | jessica.woods@example.com   | 011.44.1344.429278 |
 2016-03-10 00:00:00 |          1 | Sales Manager
          47 | Ella       | Wallace   | ella.wallace@example.com    | 011.44.1344.467268 |
 2016-01-05 00:00:00 |          1 | Sales Manager
          46 | Ava        | Sullivan  | ava.sullivan@example.com    | 011.44.1344.429268 |
 2016-10-01 00:00:00 |          1 | Sales Manager
          50 | Mia        | West      | mia.west@example.com        | 011.44.1344.429018 |
 2016-01-29 00:00:00 |          1 | Sales Manager
          25 | Ronnie     | Perry     | ronnie.perry@example.com    | 650.123.5234       |
 2016-11-16 00:00:00 |          1 | Stock Manager
          24 | Callum     | Jenkins   | callum.jenkins@example.com  | 650.123.4234       |
 2016-10-10 00:00:00 |          1 | Stock Manager
          23 | Jackson    | Coleman   | jackson.coleman@example.com | 650.123.3234       |
 2016-05-01 00:00:00 |          1 | Stock Manager
          22 | Liam       | Henderson | liam.henderson@example.com  | 650.123.2234       |
 2016-04-10 00:00:00 |          1 | Stock Manager
          21 | Jaxon      | Ross      | jaxon.ross@example.com      | 650.123.1234       |
 2016-07-18 00:00:00 |          1 | Stock Manager
(14 rows)
```

15.修改 employees 表的外码约束，去掉外码约束中的 on delete cascade 选项，但保留原有的外码引用，即 manager_id 引用本表上的 employee_id。（可通过先删后建实现）

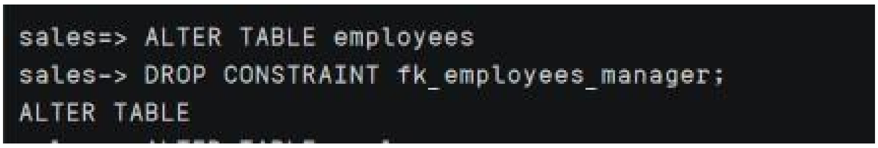
ALTER TABLE employees

DROP CONSTRAINT fk_employees_manager

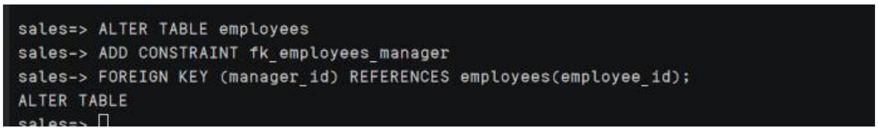
ALTER TABLE employees

ADD CONSTRAINT fk_employees_manager

FOREIGN KEY manager_id REFERENCES employees(manager_id);



```
sales=> ALTER TABLE employees
sales-> DROP CONSTRAINT fk_employees_manager;
ALTER TABLE
```

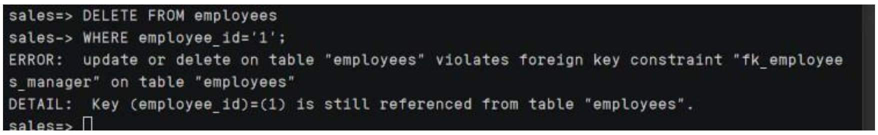


```
sales=> ALTER TABLE employees
sales-> ADD CONSTRAINT fk_employees_manager
sales-> FOREIGN KEY (manager_id) REFERENCES employees(employee_id);
ALTER TABLE
sales=>
```

16.删除 employees 表中 employee_id 为 1 的记录，观察操作结果。

DELETE FROM employees

WHERE employ_id='1';



```
sales=> DELETE FROM employees
sales-> WHERE employee_id='1';
ERROR:  update or delete on table "employees" violates foreign key constraint "fk_employee
s_manager" on table "employees"
DETAIL:  Key (employee_id)=(1) is still referenced from table "employees".
sales=>
```

删除失败，因为表内包含了对该条记录的外键约束

17.修改 employees 表的外码约束，增加 on delete cascade 选项，即回到最初的外码约束状态。

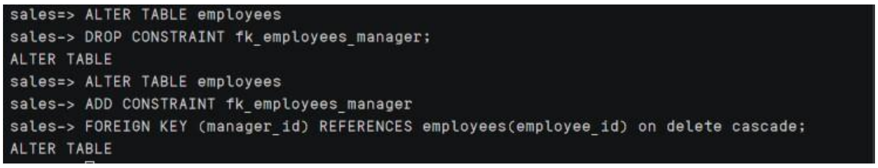
ALTER TABLE employees

DROP CONSTRAINT fk_employees_manager

ALTER TABLE employees

ADD CONSTRAINT fk_employees_manager

FOREIGN KEY manager_id REFERENCES employees(manager_id) on delete cascade;

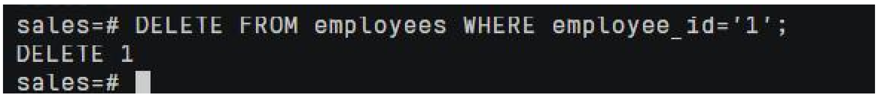


```
sales=> ALTER TABLE employees
sales-> DROP CONSTRAINT fk_employees_manager;
ALTER TABLE
sales=> ALTER TABLE employees
sales-> ADD CONSTRAINT fk_employees_manager
sales-> FOREIGN KEY (manager_id) REFERENCES employees(employee_id) on delete cascade;
ALTER TABLE
```

18.再次执行第（16）步，观察操作结果。

DELETE FROM employees

WHERE employ_id='1';



```
sales=# DELETE FROM employees WHERE employee_id='1';
DELETE 1
sales=#
```

删除成功，将对该条的引用级联删除

思考：

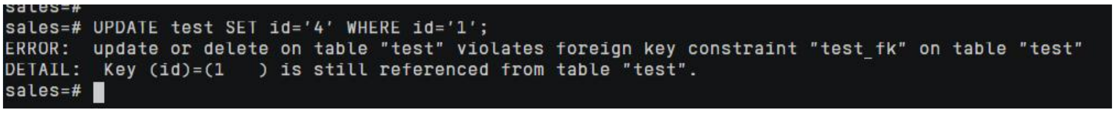
建立测试表（无 ON UPDATE CASCADE）

```
CREATE TABLE test(
    id CHAR(4) PRIMARY KEY,
    father_id CHAR(4),
    CONSTRAINT test_fk FOREIGN KEY(father_id)
    REFERENCES test(id) ON UPDATE CASCADE
);
```

插入数据

```
INSERT INTO test VALUES('1',NULL),('2','1'),('3','1');
```

对 id=1 进行修改

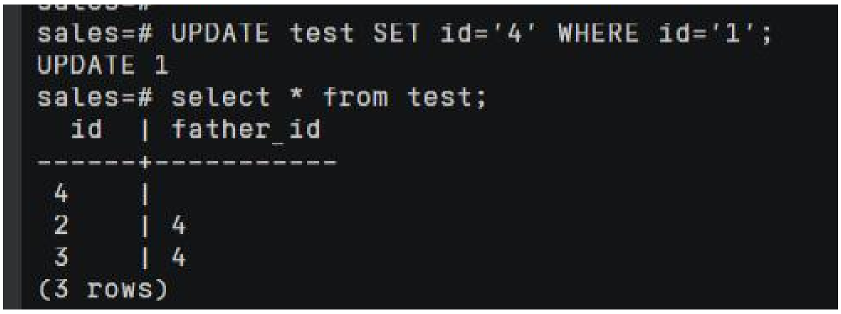


```
sales=#
sales=# UPDATE test SET id='4' WHERE id='1';
ERROR:  update or delete on table "test" violates foreign key constraint "test_fk" on table "test"
DETAIL:  Key (id)=(1   ) is still referenced from table "test".
sales=#
```

无法修改

加上 ON UPDATE CASCADE 后，再对 id=1 修改

```
ALTER TABLE test DROP CONSTRAINT test_fk;
ALTER TABLE test ADD CONSTRAINT test_fk FOREIGN KEY(father_id)
REFERENCES test(id) ON UPDATE CASCADE;
```



```
sales=# UPDATE test SET id='4' WHERE id='1';
UPDATE 1
sales=# select * from test;
  id  | father_id
------+-----------
 4    |
 2    | 4
 3    | 4
(3 rows)
```

3. 实验总结

3.1 完成的工作

建表；

建视图；

插入数据；

修改外码；

级联删除；

3.2 对实验的认识

通过学习，掌握了：

单条记录的插入、

批量记录的插入、

通过 select 语句插入、

删除命令 truncate 和 delete、

级联删除和修改；

3.3 遇到的困难及解决方法

无