数据库系统课程实验报告

实验名称：	实验五 – 数据更新
实验日期：	2023/4/22
实验地点：	文宣楼 A402
提交日期：	2023/4/22

学号：	22920212204392
姓名：	黄勖
专业年级：	软工 2021 级
学年学期：	2022-2023 学年第二学期

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）

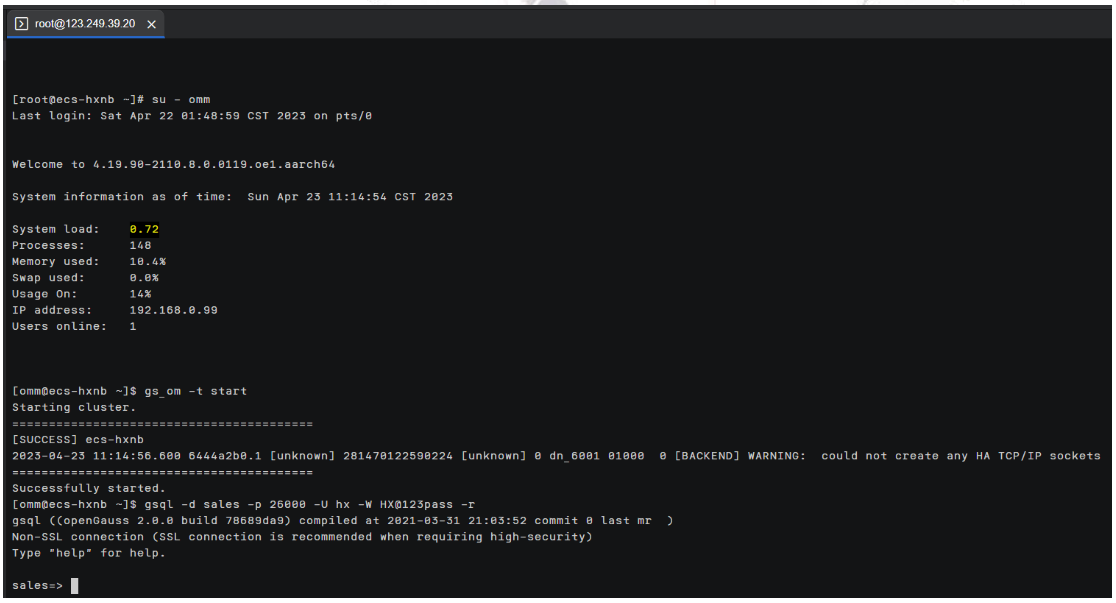
2.实验内容和步骤

（0）登录 ECS 服务器，以 omm 操作系统管理员身份登录数据库，使用 gsql 连接到数据库。

su - omm

gs_om -t start

gsql -d sales -p 26000 -U hx -W HX@123pass -r

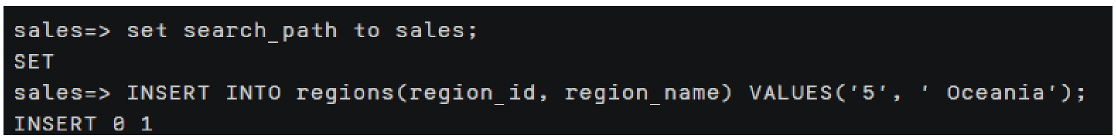


root@123.249.39.20

```
[root@ecs-hxnb ~]# su - omm
Last login: Sat Apr 22 01:48:59 CST 2023 on pts/0


Welcome to 4.19.90-2110.8.0.0119.oe1.aarch64

System information as of time:  Sun Apr 23 11:14:54 CST 2023

System load:    0.72
Processes:      148
Memory used:    10.4%
Swap used:      0.0%
Usage On:       14%
IP address:     192.168.0.99
Users online:   1



[omm@ecs-hxnb ~]$ gs_om -t start
Starting cluster.
=========================================
[SUCCESS] ecs-hxnb
2023-04-23 11:14:56.600 6444a2b0.1 [unknown] 281470122590224 [unknown] 0 dn_6001 01000  0 [BACKEND] WARNING:  could not create any HA TCP/IP sockets
=========================================
Successfully started.
[omm@ecs-hxnb ~]$ gsql -d sales -p 26000 -U hx -W HX@123pass -r
gsql ((openGauss 2.0.0 build 78689da9) compiled at 2021-03-31 21:03:52 commit 0 last mr  )
Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.

sales=> 
```

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

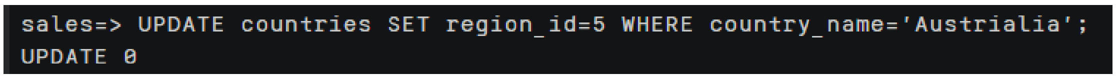
INSERT INTO regions(region_id, region_name) VALUES('5', ' Oceania');



```
sales=> set search_path to sales;
SET
sales=> INSERT INTO regions(region_id, region_name) VALUES('5', ' Oceania');
INSERT 0 1
```

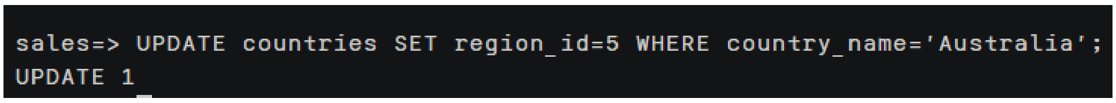
（2）将 countries 表中的国家名为 Austrialia 的 region_id 改为 5。

UPDATE countries SET region_id=5 WHERE country_name='Austrialia';



```
sales=> UPDATE countries SET region_id=5 WHERE country_name='Austrialia';
UPDATE 0
```

改为'Australia'可以更新一条数据



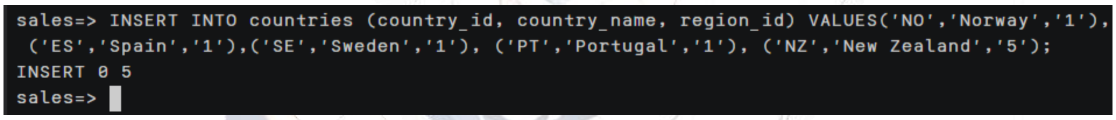
```
sales=> UPDATE countries SET region_id=5 WHERE country_name='Australia';
UPDATE 1
```

（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 (country_id, country_name, region_id)

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

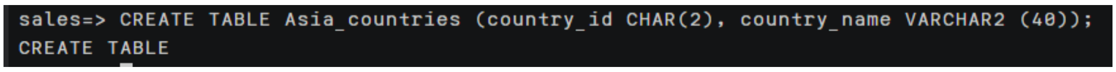
('PT','Portugal','1'), ('NZ','New Zealand','5');



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

（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。

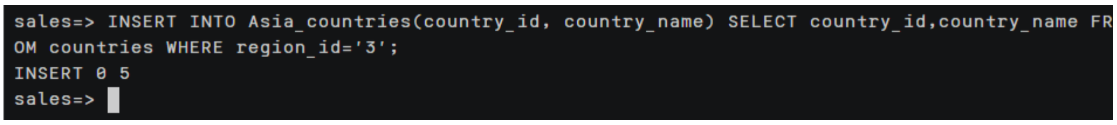
CREATE TABLE Asia_countries (country_id CHAR(2), country_name VARCHAR2 (40));



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

（5）将 countries 表中所有亚洲国家的数据插入到该表中。（要求使用插入子查询结果的方法实现）

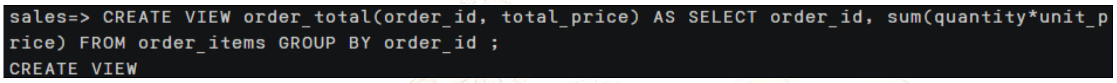
INSERT INTO Asia_countries(country_id, country_name) SELECT country_id,country_name FROM countries WHERE region_id='3';



```
sales=> INSERT INTO Asia_countries(country_id, country_name) SELECT country_id,country_name FR
OM countries WHERE region_id='3';
INSERT 0 5
sales=>
```

（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。

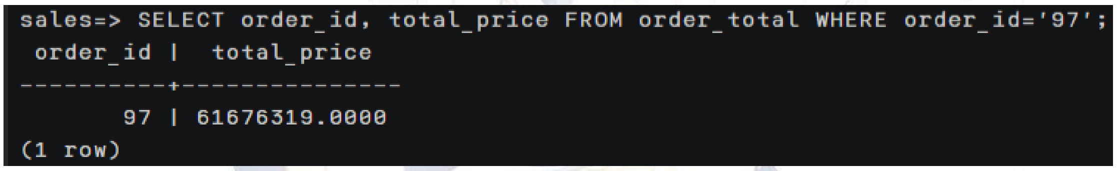
CREATE VIEW order_total(order_id, total_price) AS SELECT order_id, sum(quantity*unit_price) FROM order_items GROUP BY order_id ;



```
sales=> CREATE VIEW order_total(order_id, total_price) AS SELECT order_id, sum(quantity*unit_p
rice) FROM order_items 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。

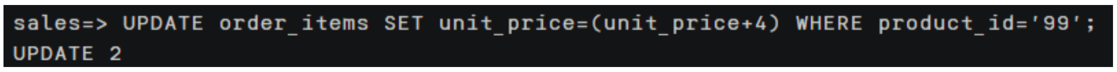
SELECT order_id, total_price FROM order_total WHERE order_id='97';



```
sales=> SELECT order_id, total_price FROM order_total WHERE order_id='97';
 order_id |  total_price
----------+---------------
       97 | 61676319.0000
(1 row)
```

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

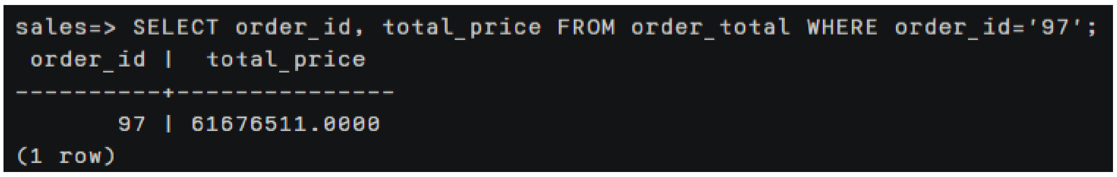
UPDATE order_items SET unit_price=(unit_price+4) WHERE product_id='99';



```
sales=> UPDATE order_items SET unit_price=(unit_price+4) 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。

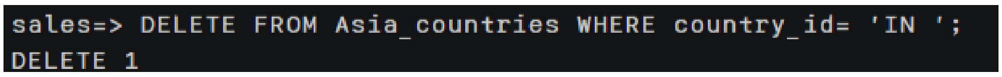
SELECT order_id, total_price FROM order_total WHERE order_id='97';



```
sales=> SELECT order_id, total_price FROM order_total WHERE order_id='97';
 order_id |  total_price
----------+---------------
       97 | 61676511.0000
(1 row)
```

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

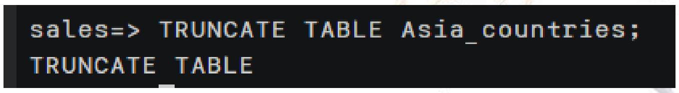
DELETE FROM Asia_countries WHERE country_id= 'IN ';



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

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

TRUNCATE TABLE Asia_countries;

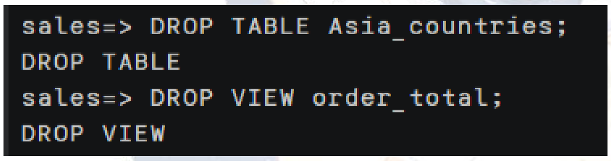


```
sales=> TRUNCATE TABLE Asia_countries;
TRUNCATE TABLE
```

（12）删除 Asia_countries 表和视图 order_total。

DROP TABLE Asia_countries;

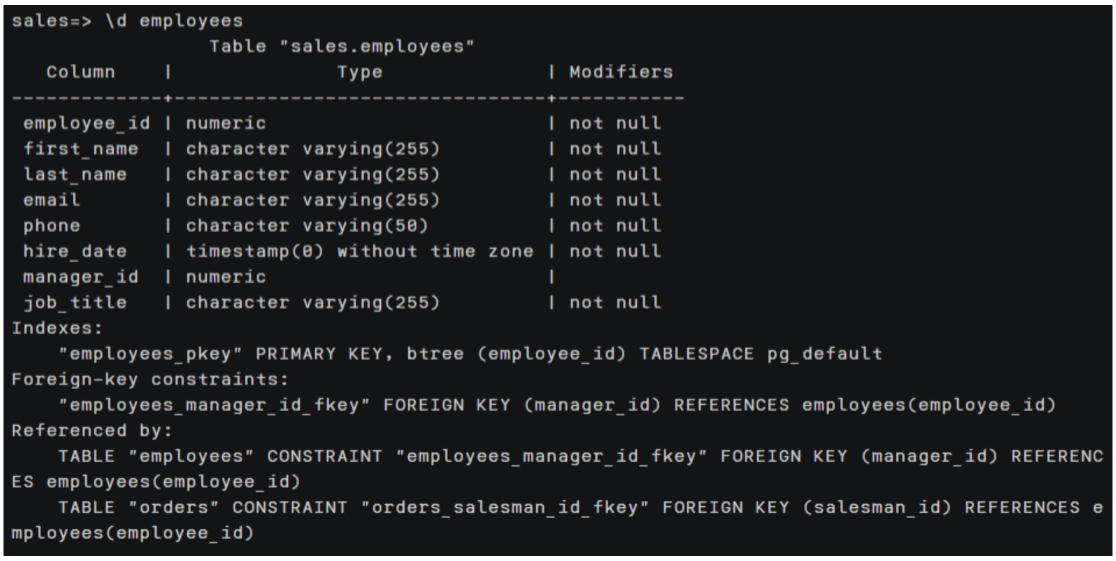
DROP VIEW order_total;



```
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。

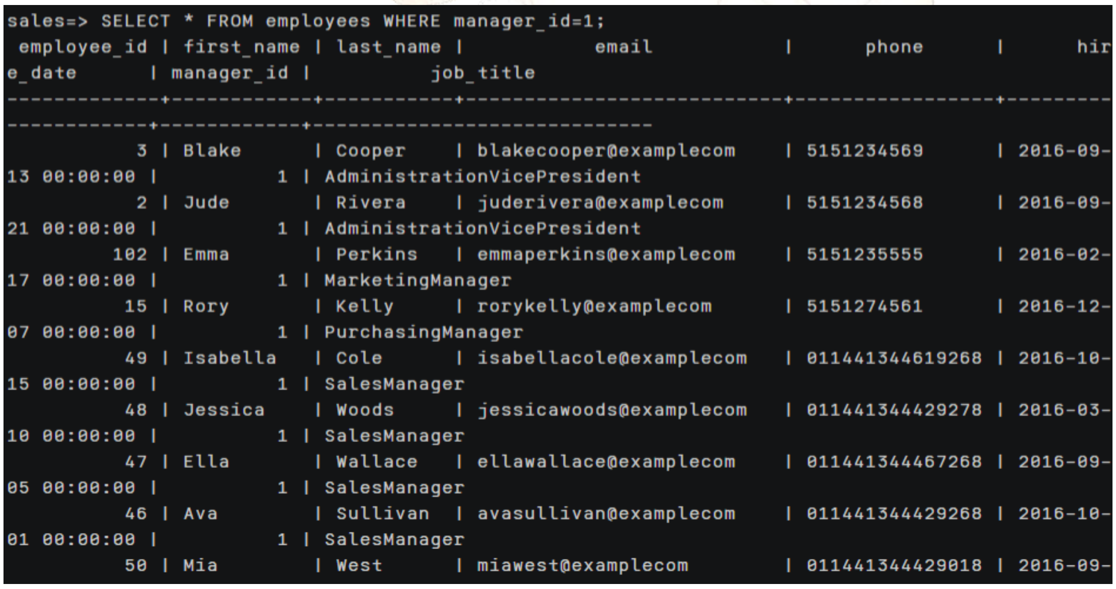
\d employees



```
sales=> \d employees
                 Table "sales.employees"
   Column    |            Type             | Modifiers
-------------+-----------------------------+-----------
 employee_id | numeric                     | not null
 first_name  | character varying(255)      | not null
 last_name   | character varying(255)      | not null
 email       | character varying(255)      | not null
 phone       | character varying(50)       | not null
 hire_date   | timestamp(0) without time zone | not null
 manager_id  | numeric                     |
 job_title   | character varying(255)      | not null
Indexes:
    "employees_pkey" PRIMARY KEY, btree (employee_id) TABLESPACE pg_default
Foreign-key constraints:
    "employees_manager_id_fkey" FOREIGN KEY (manager_id) REFERENCES employees(employee_id)
Referenced by:
    TABLE "employees" CONSTRAINT "employees_manager_id_fkey" FOREIGN KEY (manager_id) REFERENC
ES employees(employee_id)
    TABLE "orders" CONSTRAINT "orders_salesman_id_fkey" FOREIGN KEY (salesman_id) REFERENCES e
mployees(employee_id)
```

（14）查询 employees 表中 manager_id 为 1 的记录。

SELECT * FROM employees WHERE manager_id=1;

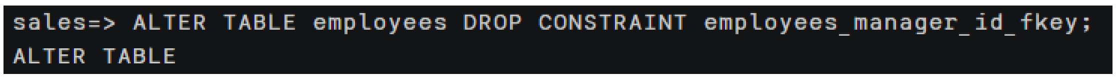


```
sales=> SELECT * FROM employees WHERE manager_id=1;
 employee_id | first_name | last_name |           email            |      phone       |      hir
e_date       | manager_id |          job_title
-------------+------------+-----------+----------------------------+------------------+---------
------------+------------+----------------------------
           3 | Blake      | Cooper    | blakecooper@examplecom     | 5151234569       | 2016-09-
13 00:00:00 |          1 | AdministrationVicePresident
           2 | Jude       | Rivera    | juderivera@examplecom      | 5151234568       | 2016-09-
21 00:00:00 |          1 | AdministrationVicePresident
         102 | Emma       | Perkins   | emmaperkins@examplecom     | 5151235555       | 2016-02-
17 00:00:00 |          1 | MarketingManager
          15 | Rory       | Kelly     | rorykelly@examplecom       | 5151274561       | 2016-12-
07 00:00:00 |          1 | PurchasingManager
          49 | Isabella   | Cole      | isabellacole@examplecom    | 011441344619268 | 2016-10-
15 00:00:00 |          1 | SalesManager
          48 | Jessica    | Woods     | jessicawoods@examplecom    | 011441344429278 | 2016-03-
10 00:00:00 |          1 | SalesManager
          47 | Ella       | Wallace   | ellawallace@examplecom     | 011441344467268 | 2016-09-
05 00:00:00 |          1 | SalesManager
          46 | Ava        | Sullivan  | avasullivan@examplecom     | 011441344429268 | 2016-10-
01 00:00:00 |          1 | SalesManager
          50 | Mia        | West      | miawest@examplecom         | 011441344429018 | 2016-09-
```

（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

employees_manager_id_fkey;

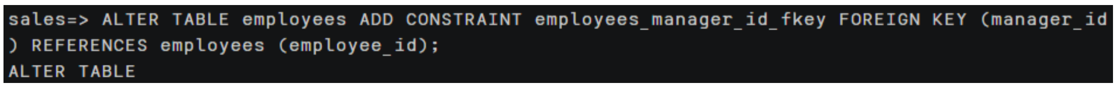


```
sales=> ALTER TABLE employees DROP CONSTRAINT employees_manager_id_fkey;
ALTER TABLE
```

ALTER TABLE employees ADD CONSTRAINT

employees_manager_id_fkey FOREIGN KEY (manager_id)

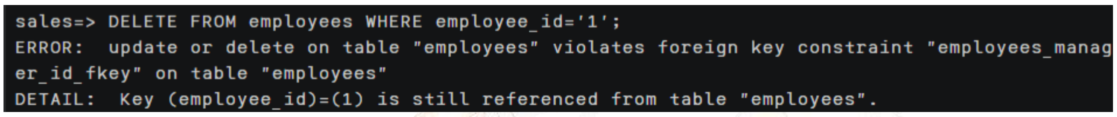
REFERENCES employees (employee_id);



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

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

DELETE FROM employees WHERE employee_id='1';



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

（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。

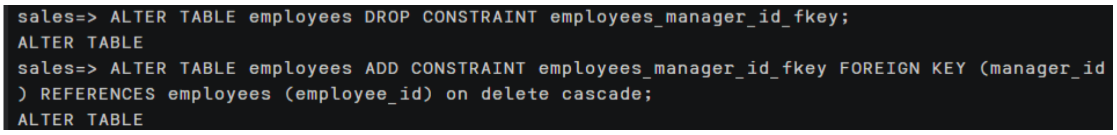
ALTER TABLE employees DROP CONSTRAINT

employees_manager_id_fkey;

ALTER TABLE employees ADD CONSTRAINT

employees_manager_id_fkey FOREIGN KEY (manager_id)

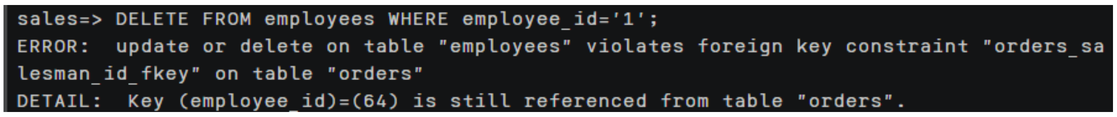
REFERENCES employees (employee_id) on delete cascade;



```
sales=> ALTER TABLE employees DROP CONSTRAINT employees_manager_id_fkey;
ALTER TABLE
sales=> ALTER TABLE employees ADD CONSTRAINT employees_manager_id_fkey 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。

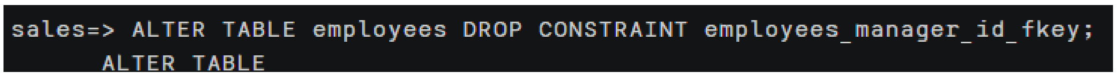
DELETE FROM employees WHERE employee_id='1';



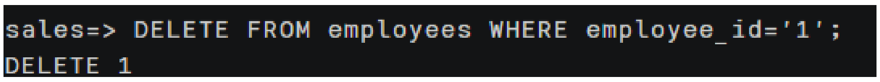
```
sales=> DELETE FROM employees WHERE employee_id='1';
ERROR:  update or delete on table "employees" violates foreign key constraint "orders_sa
lesman_id_fkey" on table "orders"
DETAIL:  Key (employee_id)=(64) is still referenced from table "orders".
```

换了一个错误，仍然不能删除。

但完全删除外码限制之后就可以删除：



```
sales=> ALTER TABLE employees DROP CONSTRAINT employees_manager_id_fkey;
      ALTER TABLE
```



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

3.实验总结

3.1 完成的工作

设计正确的 SQL 语句并完成了所有数据更新修改等操作。

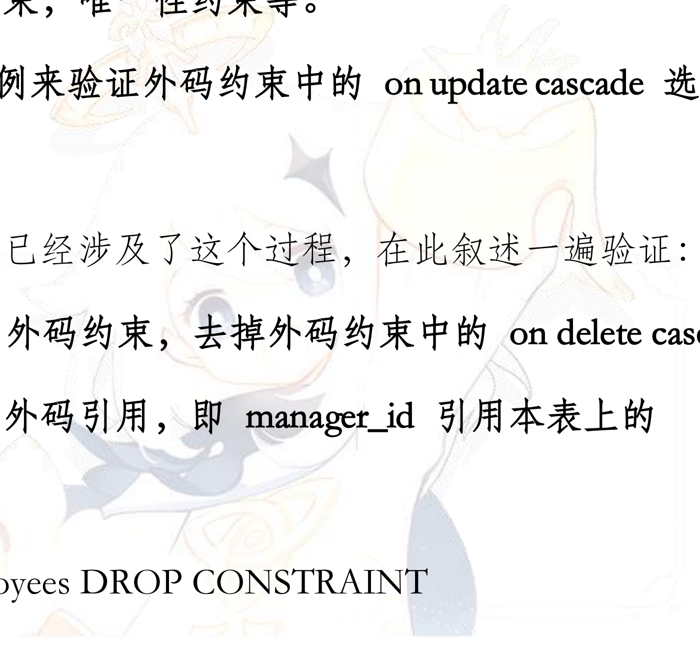
3.2 对实验的认识

（1）当更新数据失败时，一个主要原因可能是因为违反了完整性约束，如主外码约束，唯一性约束等。

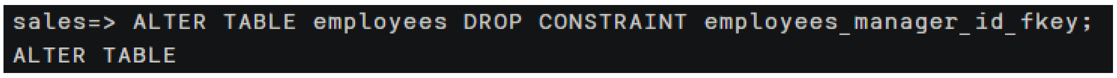
问题：请设计实例来验证外码约束中的 on update cascade 选项的作用。

在前面的过程中已经涉及了这个过程，在此叙述一遍验证：

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。

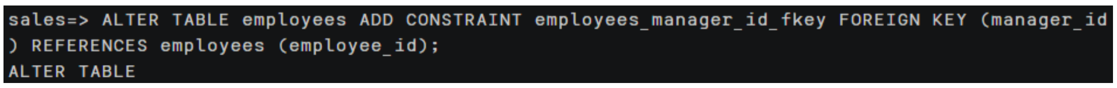


ALTER TABLE employees DROP CONSTRAINT employees_manager_id_fkey;



```
sales=> ALTER TABLE employees DROP CONSTRAINT employees_manager_id_fkey;
ALTER TABLE
```

ALTER TABLE employees ADD CONSTRAINT employees_manager_id_fkey FOREIGN KEY (manager_id) REFERENCES employees (employee_id);



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

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

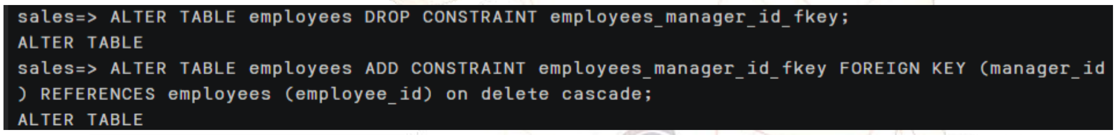
DELETE FROM employees WHERE employee_id='1';

这时无法删除

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 employees_manager_id_fkey;

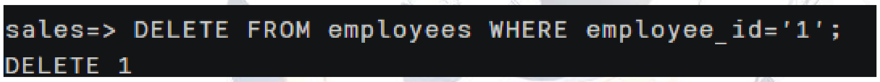
ALTER TABLE employees ADD CONSTRAINT employees_manager_id_fkey FOREIGN KEY (manager_id) REFERENCES employees (employee_id) **on delete cascade**;



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

再次执行，观察操作结果。

DELETE FROM employees WHERE employee_id='1';



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

此时应当可以删除。

（2）收获

这一次实验反复使用数据库查询、更新数据语言，在加深理解的同时，我提高了运用数据库代码实现自己相关需求的能力，未来在组织数据库语言方面我会更加熟练。

3.3 遇到的困难及解决方法

无。