西安交通大学实验报告

课程名称：数据库基础与应用

实验名称：数据的完整性

学院：物理学院

实验日期：2023.4.5

班级：物试 2201

姓名：汪洋

学号：2223711803

1 实验目的

1. 了解索引的概念，理解索引的意义，能够创建索引。
2. 理解数据完整性的概念和实现机制，掌握不同的完整性的设置方法和作用。
3. 了解触发器的概念，理解触发器的组成，掌握触发器的使用方法。

2 实验任务

1. 创建索引。
2. 验证参照完整性。
3. 创建触发器，记录操作日志。
4. 创建触发器，保证数据一致性。

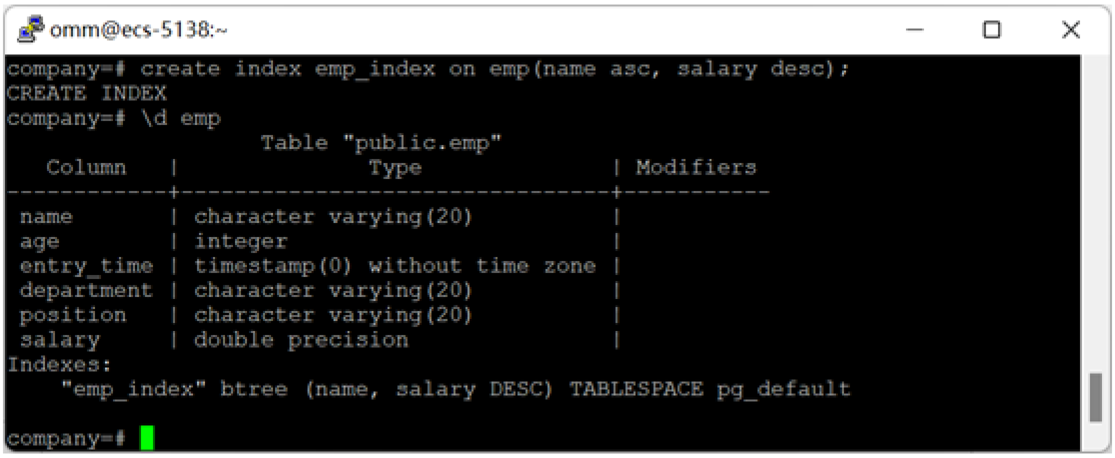
3 实验环境

1. 华为 ECS，openEuler OS
2. DB: openGauss
3. 远程连接:putty

4 实验步骤与结果

1. 创建索引。
   创建一个数据表，再为其创建索引，将创建语句和结果截图写入实验报告。

```sql
create database company owner omm;
create table emp(name varchar(20), age int, entry_time date, department varchar(20),
    position varchar(20), salary float);
insert into emp values('Andy', 24, '2022-7-1', 'software', 'manager', 50000);
insert into emp values('Bear', 28, '2021-2-9', 'finance', 'clerk', 10000);
insert into emp values('Tom', 33, '2015-5-18', 'law', 'clerk', 8000);
insert into emp values('Jerry', 20, '2023-2-1', 'technology', 'manager', 80000);

create index emp_index on emp(name asc, salary desc);
```

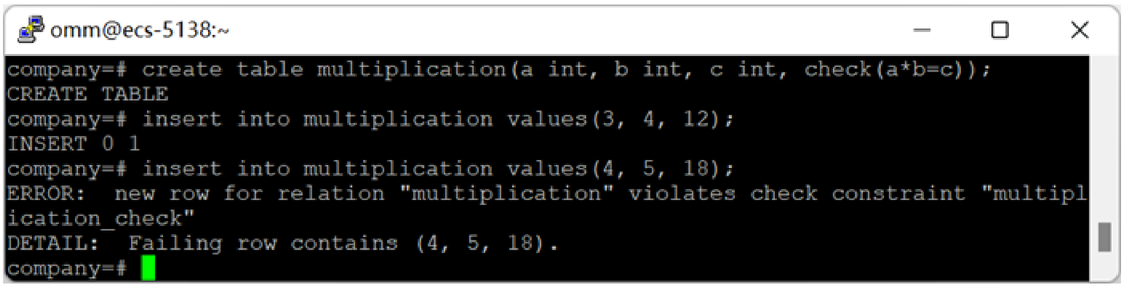


```
company=# create index emp_index on emp(name asc, salary desc);
CREATE INDEX
company=# \d emp
                   Table "public.emp"
   Column    |              Type              | Modifiers
-------------+--------------------------------+-----------
 name        | character varying(20)          |
 age         | integer                        |
 entry_time  | timestamp(0) without time zone |
 department  | character varying(20)          |
 position    | character varying(20)          |
 salary      | double precision               |
Indexes:
    "emp_index" btree (name, salary DESC) TABLESPACE pg_default

company=# 
```

2. 设计数据表，验证 CHECK 约束。

   设计数据表，其中有 CHECK 约束，输入符合和不符合约束的数据，观察得到的结果，将数据库的创建语句、数据插入语句和结果的截图写入实验报告。

```
create table multiplication(a int, b int, c int, check(a*b=c));
insert into multiplication values(3, 4, 12);
insert into multiplication values(4, 5, 18);
```

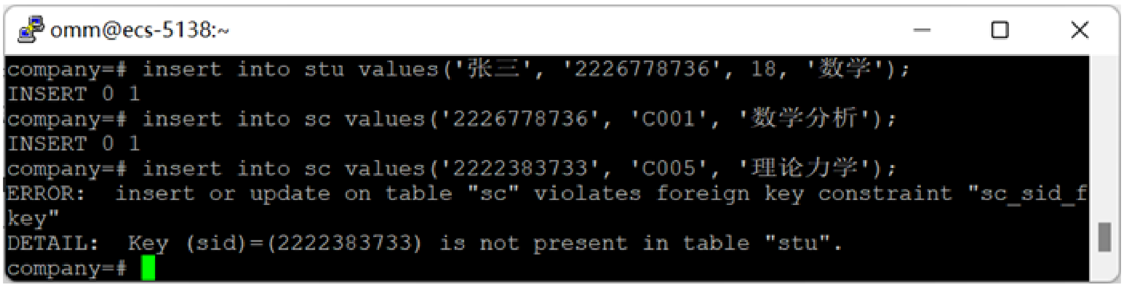


```
company=# create table multiplication(a int, b int, c int, check(a*b=c));
CREATE TABLE
company=# insert into multiplication values(3, 4, 12);
INSERT 0 1
company=# insert into multiplication values(4, 5, 18);
ERROR:  new row for relation "multiplication" violates check constraint "multipl
ication_check"
DETAIL:  Failing row contains (4, 5, 18).
company=# 
```

3. 验证参照完整性。

   (1) 设计两个数据表，建立参照完整性，输入符合和不符合参照完整性的数据，观察得到的结果，将数据库的创建语句、数据插入语句和结果截图写入实验报告。

```
create table stu(sname varchar(20), sID char(10), age int, major varchar(20), primary
    key (sID));
create table sc(sID char(10), cID varchar(10), cname varchar(20), foreign key (sID)
    references stu);
insert into stu values('张三', '2226778736', 18, '数学');
insert into sc values('2226778736', 'C001', '数学分析');
insert into sc values('2222383733', 'C005', '理论力学');
```

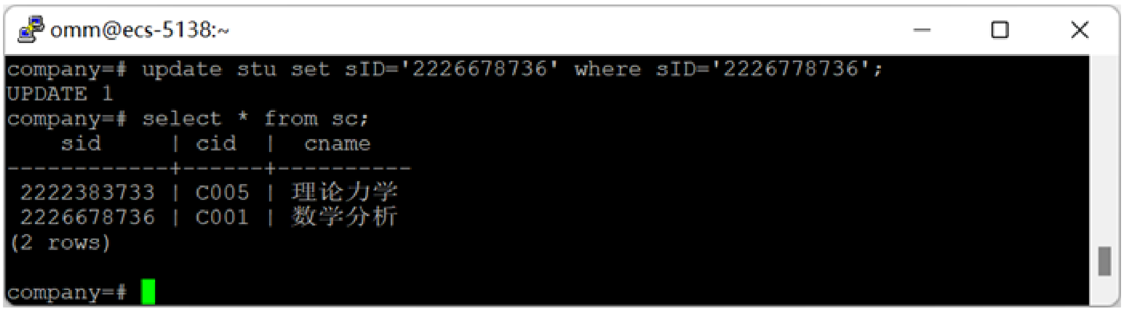


```
company=# insert into stu values('张三', '2226778736', 18, '数学');
INSERT 0 1
company=# insert into sc values('2226778736', 'C001', '数学分析');
INSERT 0 1
company=# insert into sc values('2222383733', 'C005', '理论力学');
ERROR:  insert or update on table "sc" violates foreign key constraint "sc_sid_f
key"
DETAIL:  Key (sid)=(2222383733) is not present in table "stu".
company=# 
```

   (2) 设计两个数据表，建立参照完整性，并设置级联更新和级联删除，修改和删除主表中主键的值，使其符合和不符合参照完整性，观察从表的变化，将语句和结果截图写入实验报告。修改从表的外键值，使其符合和不符合参照完整性，观察结果，将语句和结果截图写入实验报告。

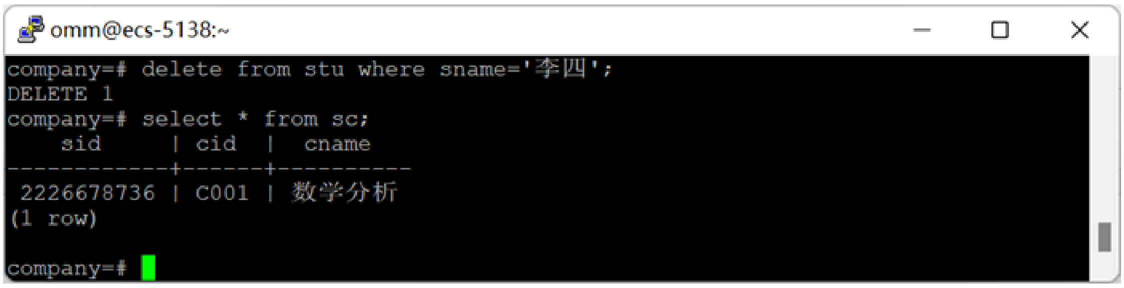
```
drop table sc;
drop table stu;
create table stu(sname varchar(20), sID char(10), age int, major varchar(20), primary
    key (sID));
create table sc(sID char(10), cID varchar(10), cname varchar(20), foreign key (sID)
    references stu on update cascade on delete cascade);
```

```
insert into stu values('张三', '2226778736', 18, '数学');
insert into stu values('李四', '2222383733', 18, '物理');
insert into sc values('2226778736', 'C001', '数学分析');
insert into sc values('2222383733', 'C005', '理论力学');

update stu set sID='2226678736' where sID='2226778736';
select * from sc;
```



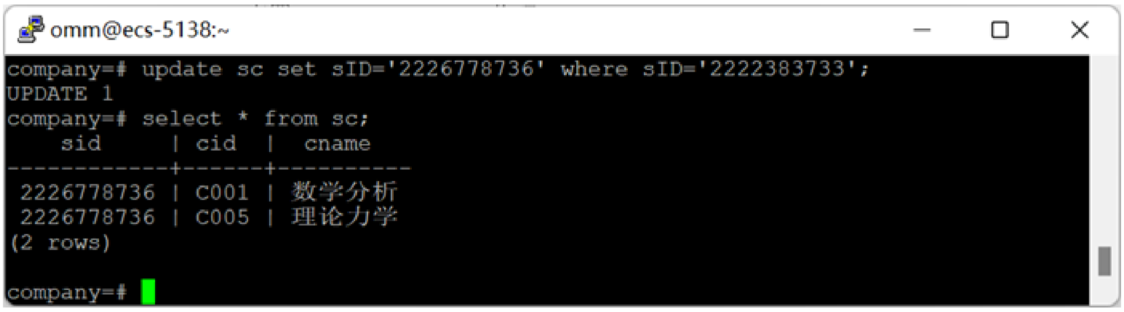
```
company=# update stu set sID='2226678736' where sID='2226778736';
UPDATE 1
company=# select * from sc;
    sid     | cid  |  cname
------------+------+----------
 2222383733 | C005 | 理论力学
 2226678736 | C001 | 数学分析
(2 rows)

company=#
```

```
delete from stu where sname='李四';
select * from sc;
```



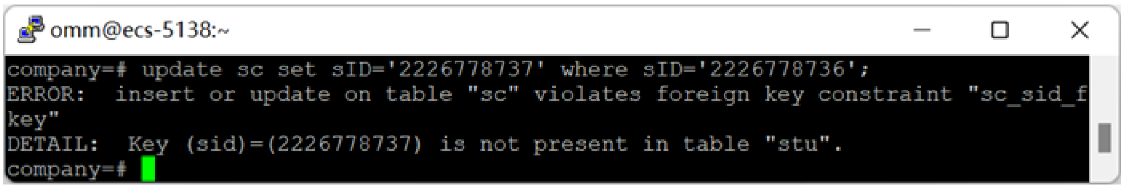
```
company=# delete from stu where sname='李四';
DELETE 1
company=# select * from sc;
    sid     | cid  |  cname
------------+------+----------
 2226678736 | C001 | 数学分析
(1 row)

company=#
```

```
drop table sc;
drop table stu;
create table stu(sname varchar(20), sID char(10), age int, major varchar(20), primary key (sID));
create table sc(sID char(10), cID varchar(10), cname varchar(20), foreign key (sID) references stu on update cascade on delete cascade);
insert into stu values('张三', '2226778736', 18, '数学');
insert into stu values('李四', '2222383733', 18, '物理');
insert into sc values('2226778736', 'C001', '数学分析');
insert into sc values('2222383733', 'C005', '理论力学');

update sc set sID='2226778736' where sID='2222383733';
select * from sc;
```



```
company=# update sc set sID='2226778736' where sID='2222383733';
UPDATE 1
company=# select * from sc;
    sid     | cid  |  cname
------------+------+----------
 2226778736 | C001 | 数学分析
 2226778736 | C005 | 理论力学
(2 rows)

company=#
```

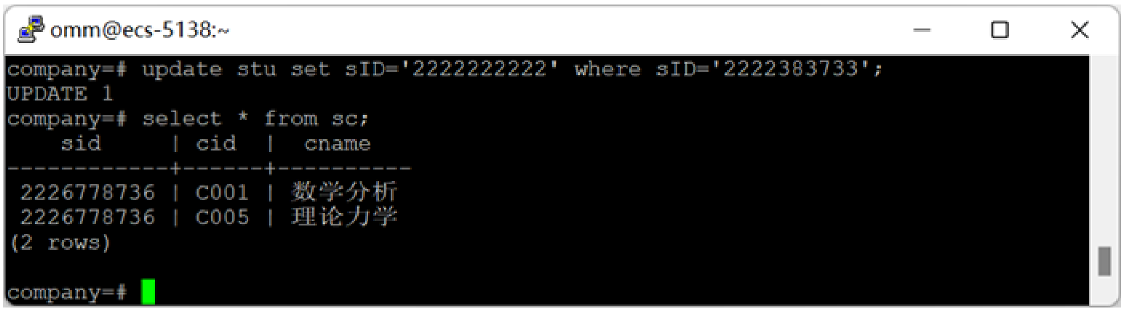
```
update sc set sID='2226778737' where sID='2226778736';
```



```
company=# update sc set sID='2226778737' where sID='2226778736';
ERROR:  insert or update on table "sc" violates foreign key constraint "sc_sid_f
key"
DETAIL:  Key (sid)=(2226778737) is not present in table "stu".
company=#
```

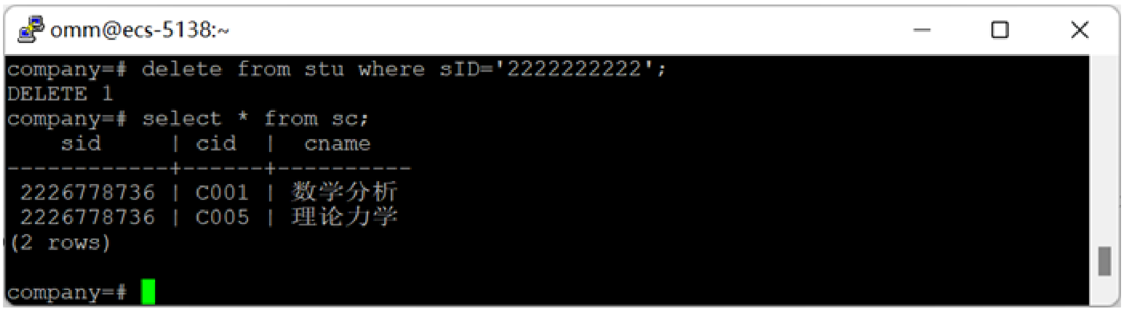
(3) 设计两个数据表，建立参照完整性约束，从表中的外键设置默认值，将违反参照完整性的违约操作设置为 “SET DEFAULT”，修改和删除主表中主键的值，使其符合和不符合参照完整性，观察从表的变化，将语句和结果截图写入实验报告。

```sql
drop table sc;
drop table stu;
create table stu(sname varchar(20), sID char(10), age int, major varchar(20), primary key (sID));
create table sc(sID char(10) default '2226778736', cID varchar(10), cname varchar(20), foreign key (sID) references stu on update set default on delete set default);
insert into stu values('张三', '2226778736', 18, '数学');
insert into stu values('李四', '2222383733', 18, '物理');
insert into sc values('2226778736', 'C001', '数学分析');
insert into sc values('2222383733', 'C005', '理论力学');

update stu set sID='2222222222' where sID='2222383733';
select * from sc;
```



```
company=# update stu set sID='2222222222' where sID='2222383733';
UPDATE 1
company=# select * from sc;
    sid     | cid  |   cname
------------+------+----------
 2226778736 | C001 | 数学分析
 2226778736 | C005 | 理论力学
(2 rows)

company=#
```

```sql
delete from stu where sID='2222222222';
select * from sc;
```



```
company=# delete from stu where sID='2222222222';
DELETE 1
company=# select * from sc;
    sid     | cid  |   cname
------------+------+----------
 2226778736 | C001 | 数学分析
 2226778736 | C005 | 理论力学
(2 rows)

company=#
```

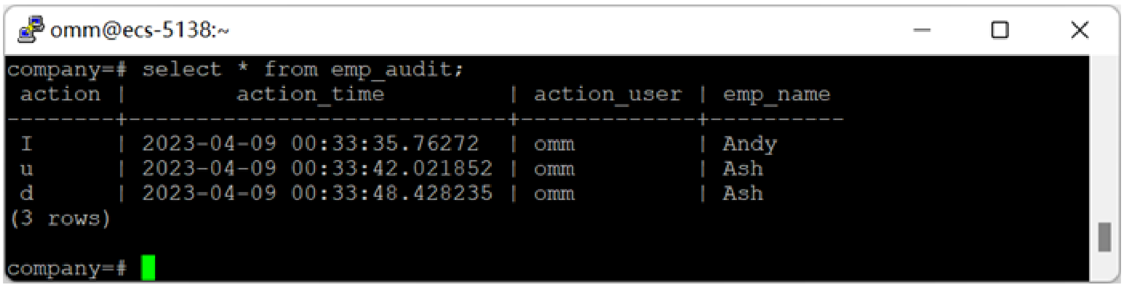
4. 创建触发器，记录操作日志。

创建职员表 emp 和审计日志表 emp_audit，在职员表上创建触发器以在审计日志表中记录用户所做的操作。在 emp 上做相应操作，验证触发器的功能，最后删除相应的对象。在实验报告中展示操作过程和结果。

审计日志表应包括操作类型（INSERT、DELETE、UPDATE）、时间、操作员和职员姓名。

```sql
drop table if exists emp;
create table emp(id serial primary key, name varchar(20));
create table emp_audit(action char(1), action_time timestamp, action_user varchar(20), emp_name varchar(20));

create or replace function emp_audit_funcion()
returns trigger
language plpgsql
as $$
begin
if (tg_op='INSERT') then
insert into emp_audit values('I', now(), user, new.name);
return new;
elsif (tg_op='UPDATE') then
insert into emp_audit values('u', now(), user, new.name);
```

```
return new;
elsif (tg_op='DELETE') then
insert into emp_audit values('d', now(), user, old.name);
return old;
end if;
return null;
end;
$$;

create trigger emp_audit_trigger after insert or update or delete on emp for each row
execute procedure emp_audit_funcion();

insert into emp values('123456', 'Andy');
update emp set name = 'Ash' where name = 'Andy';
delete from emp where name='Ash';
select * from emp_audit;
```



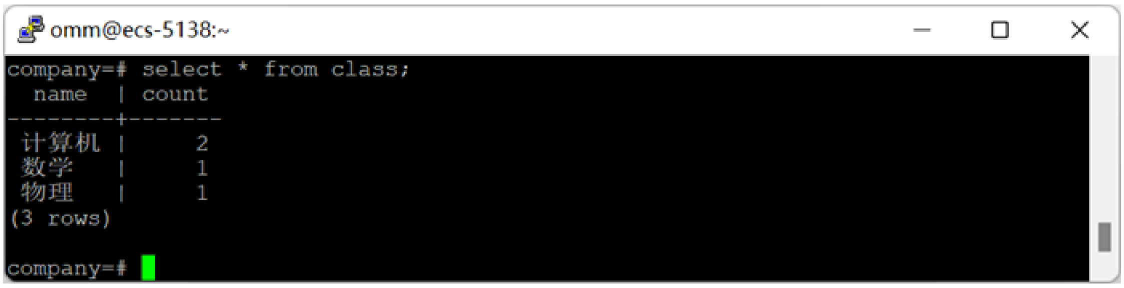
```
company=# select * from emp_audit;
 action |          action_time          | action_user | emp_name
--------+-------------------------------+-------------+----------
 I      | 2023-04-09 00:33:35.76272  | omm         | Andy
 u      | 2023-04-09 00:33:42.021852 | omm         | Ash
 d      | 2023-04-09 00:33:48.428235 | omm         | Ash
(3 rows)

company=#
```

```
drop trigger emp_audit_trigger on emp;
drop function emp_audit_function();
drop table emp;
drop table emp_audit;
```

5. 创建触发器，保证数据一致性。创建学生表和班级表，班级表中记录班级名称和人数，在学生表上创建触发器，实现学生表的添加、修改和删除时相应的班级表中的班级人数随之改变。在学生表上进行相应操作，验证触发器的功能，最后删除相应的对象。
   在实验报告中展示操作过程和结果。

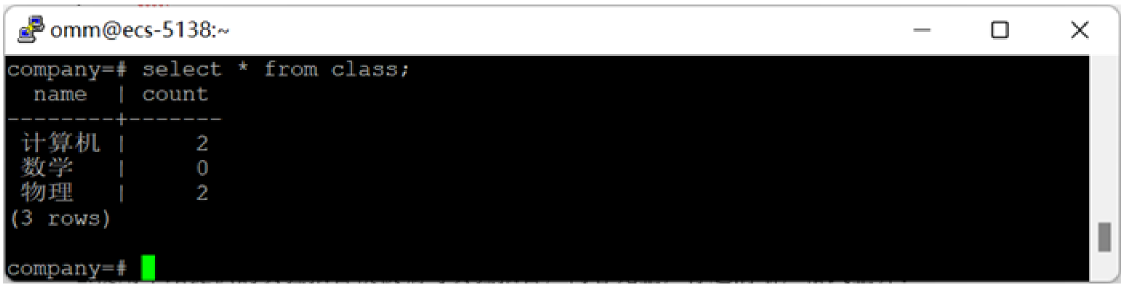
```
create table stu(sID varchar(10) primary key, name varchar(20), class varchar(20), age int);
create table class(name varchar(20), count int);

create or replace function change_class_count()
returns trigger
language plpgsql
as $$
begin
if (tg_op='INSERT') then
  if (exists(select * from class where name=new.class)) then
update class set count=count+1 where name=new.class;
return new;
else
insert into class values(new.class, 1);
return new;
end if;
elseif (tg_op='UPDATE') then
  if (new.class!=old.class) then
update class set count=count-1 where name=old.class;
if (exists(select * from class where name=new.class)) then
update class set count=count+1 where name=new.class;
return new;
else
insert into class values(new.class,1);
return new;
end if;
end if;
```

```
elseif (tg_op='DELETE') then
update class set count=count-1 where name=old.class;
return old;
end if;
return null;
end;
$$;

create trigger change_class_count_trigger after insert or update or delete on stu
for each row
execute procedure change_class_count();

insert into stu values('001', '张三', '计算机', 18);
insert into stu values('002', '李四', '计算机', 18);
insert into stu values('003', '王五', '数学', 18);
insert into stu values('004', '赵六', '物理', 18);
select * from class;
```



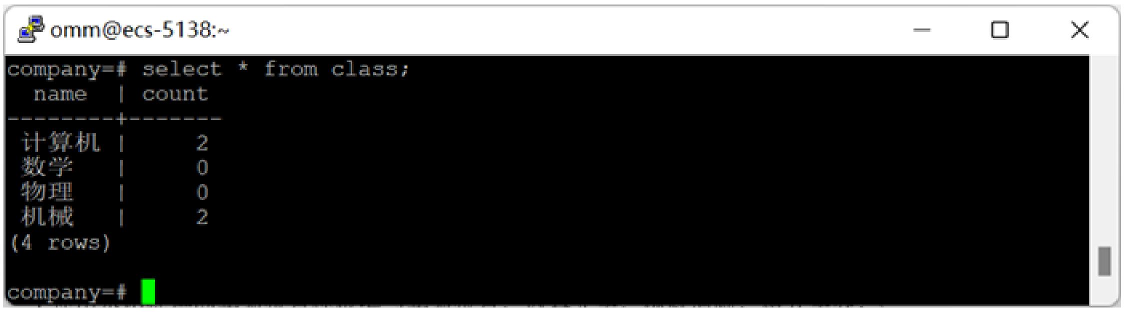
```
omm@ecs-5138:~
company=# select * from class;
  name   | count
---------+-------
 计算机 |     2
 数学   |     1
 物理   |     1
(3 rows)

company=#
```

```
update stu set class='物理' where class='数学';
select * from class;
```



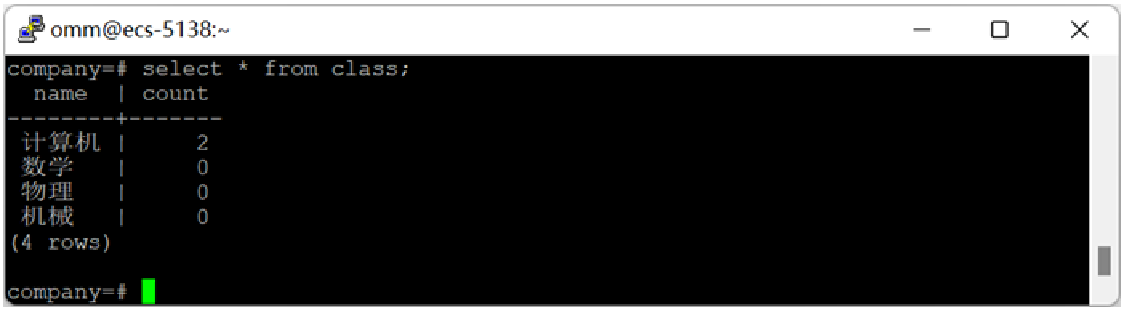
```
omm@ecs-5138:~
company=# select * from class;
  name   | count
---------+-------
 计算机 |     2
 数学   |     0
 物理   |     2
(3 rows)

company=#
```

```
update stu set class='机械' where class='物理';
select * from class;
```



```
omm@ecs-5138:~
company=# select * from class;
  name   | count
---------+-------
 计算机 |     2
 数学   |     0
 物理   |     0
 机械   |     2
(4 rows)

company=#
```

```
delete from stu where class='机械' ;
select * from class;
```



```
company=# select * from class;
  name   | count
--------+-------
 计算机 |     2
 数学   |     0
 物理   |     0
 机械   |     0
(4 rows)

company=# 
```

```
drop trigger change_class_count_trigger on stu;
drop function change_class_count();
drop table stu;
drop table class;
```

5 实验总结

略。