《数据库系统及其应用实践》课程实验报告

实验二：SQL 练习

学号：********** 姓名：*** 完成日期：2023 年 3 月 21 日

学习和掌握 SQL 的基本语法，能够编写 SQL 语句完成指定的查询

实验要求

1、按照实验内容，依次完成每个实验步骤；

2、操作实验步骤时，需要理解该操作步骤的目的，预判操作的结果；当操作结果与预判不符时，及时向任课教师和助教咨询；

3、在实验报告中依次记录**主要操作步骤**的内容和结果（返回的消息或截图）；

4、对实验中遇到的问题、解决方案及收获进行总结；

5、确保实验报告整洁、美观（注意字体、字号、对齐、截图大小和分页等；）

实验过程记录

步骤 1

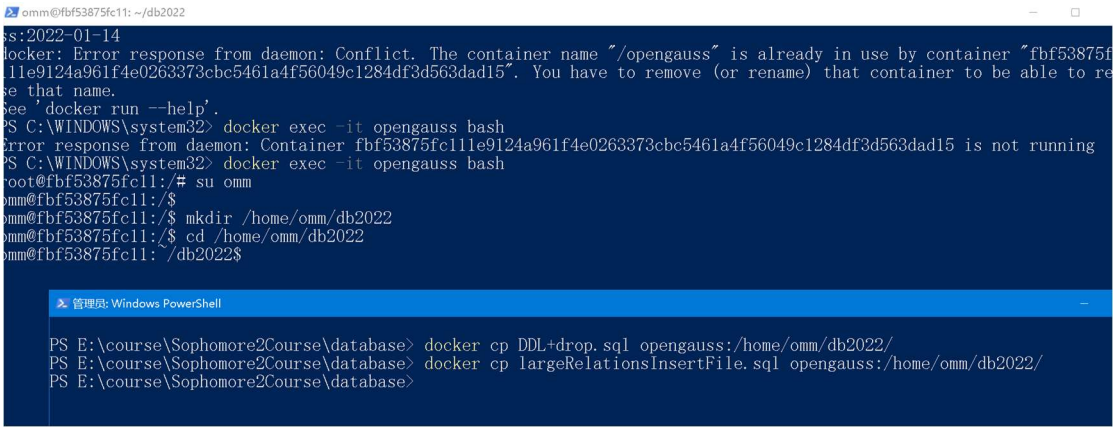
操作内容：

1. 下载实验 2 中的两个 SQL 脚本文件 `DDL+drop.sql` 和 `largeRelationsInsertFile.sql`；
2. 打开一个 PowerShell 窗口，启动在实验 1 中创建的 openGauss 数据库实例容器；

docker run --name opengauss --privileged=true -d -e GS_PASSWORD=Password@123 -p 15432:5432 mygauss:2022-01-14

3. 执行 `docker exec -it opengauss bash`，在容器内部启动一个 Bash Shell 进程，并连接到当前 PowerShell 窗口；
4. 执行 `su omm`，切换到 omm 用户；
5. 执行 `mkdir /home/omm/db2022`，创建文件夹；
6. 执行 `cd /home/omm/db2022`，设置当前工作目录；
7. 打开另一个 PowerShell 窗口，在其中执行 `docker cp .\Downloads\DDL+drop.sql opengauss:/home/omm/db2022/`和 `docker cp .\Downloads\largeRelationsInsertFile.sql opengauss:/home/omm/db2022/`，将下载的 SQL 脚本文件复制到容器中的工作目录下；

操作结果：

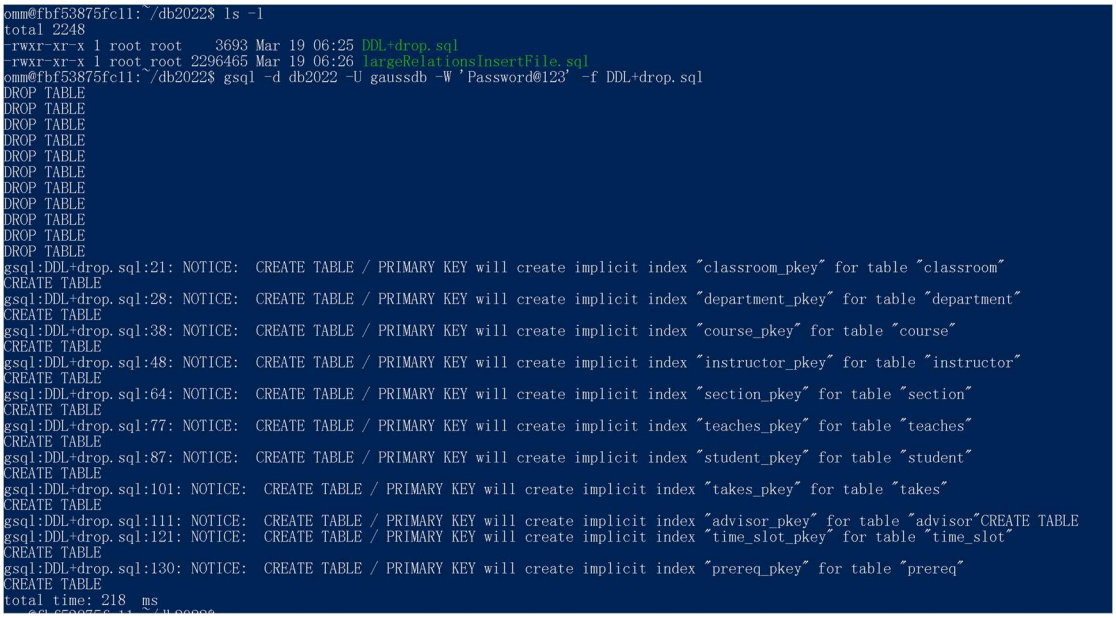


步骤 2

操作内容：

8. 关闭第 2 个 PowerShell 窗口，切换回第 1 个 PowerShell 窗口，执行 `ls -l`，确认 SQL 脚本文件已经被复制到当前工作目录中；
9. 执行 `gsql -d db2022 -U gaussdb -W 'Password@123' -f DDL+drop.sql`，使用客户端工具 `gsql` 通过脚本文件 `DDL+drop.sql` 在数据库 db2022 中创建数据表；

操作结果：



步骤 3

操作内容：

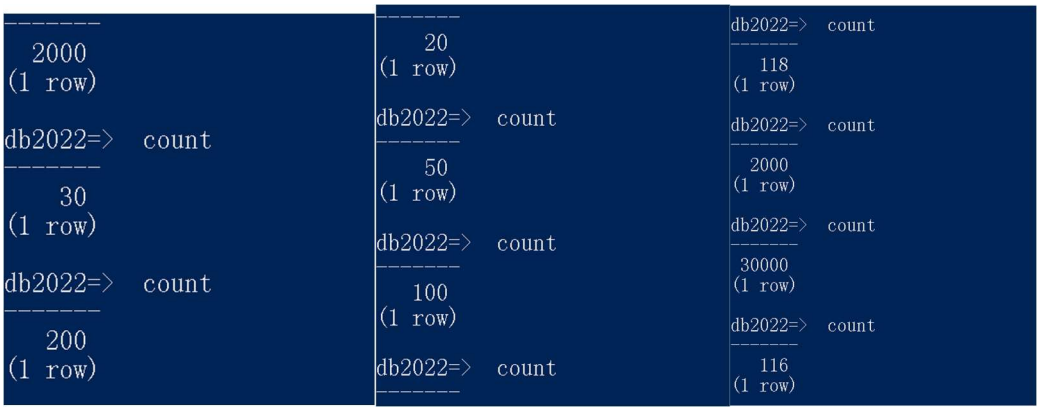
10. 执行 `gsql -d db2022 -U gaussdb -W 'Password@123' -f largeRelationsInsertFile.sql`，使用客户端工具 `gsql` 通过脚本文件 `largeRelationsInsertFile.sql` 向数据表中插入数据；

11. 执行 `gsql -d db2022 -U gaussdb -W 'Password@123'`，使用客户端工具 `gsql` 连接到数据库，然后执行下列 SQL 语句确认每张数据表中的记录数；
12. 执行`\q`，退出客户端工具 gsql，断开与数据库的连接；

```
select count(*) from advisor;

select count(*) from classroom;

select count(*) from course;

select count(*) from department;

select count(*) from instructor;

select count(*) from prereq;

select count(*) from section;

select count(*) from student;

select count(*) from takes;

select count(*) from teaches;

select count(*) from time_slot;
```

操作结果：

```
total time: 197271  ms
omm@fbf53875fc11:~/db2022$ gsql -d db2022 -U gaussdb -W 'Password@123'
gsql ((openGauss 2.1.0 build 590b0f8e) compiled at 2021-09-30 14:29:04 commit 0 last mr  )
Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.

db2022=> select count(*) from advisor;
select count(*) from classroom;
select count(*) from course;
select count(*) from department;
select count(*) from instructor;
select count(*) from prereq;
select count(*) from section;
select count(*) from student;
select count(*) from takes;
select count(*) from teaches;
select count(*) from time_slot; count
```



SQL 练习

步骤 4

操作内容：

`gsql -d db2022 -U gaussdb -W 'Password@123' -c 'select * from department;' -o department.csv` 使用客户端工具 `gsql` 执行单条 SQL 语句，并将查询结果保存到文件 `department.csv` 中；

操作结果：

```
omm@fbf53875fc11:~/db2022$ gsql -d db2022 -U gaussdb -W 'Password@123' -c 'select * from department;' -o department.csv
omm@fbf53875fc11:~/db2022$ ls
DDL+drop.sql  department.csv  largeRelationsInsertFile.sql
omm@fbf53875fc11:~/db2022$ ls -l
total 2252
-rwxr-xr-x 1 root root    3693 Mar 19 06:25 DDL+drop.sql
-rw-rw-r-- 1 omm  omm      871 Mar 19 13:35 department.csv
-rwxr-xr-x 1 root root 2296465 Mar 19 06:26 largeRelationsInsertFile.sql
omm@fbf53875fc11:~/db2022$ cat depatment.csv
cat: depatment.csv: No such file or directory
omm@fbf53875fc11:~/db2022$ cat department.csy
cat: department.csy: No such file or directory
omm@fbf53875fc11:~/db2022$ cat department.csv
  dept_name   |  building   |  budget
--------------+-------------+-----------
 Civil Eng.   | Chandler    | 255041.46
 Biology      | Candlestick | 647610.55
 History      | Taylor      | 699140.86
 Physics      | Wrigley     | 942162.76
 Marketing    | Lambeau     | 210627.58
 Pol. Sci.    | Whitman     | 573745.09
 English      | Palmer      | 611042.66
 Accounting   | Saucon      | 441840.92
 Comp. Sci.   | Lamberton   | 106378.69
 Languages    | Linderman   | 601283.60
 Finance      | Candlestick | 866831.75
 Geology      | Palmer      | 406557.93
 Cybernetics  | Mercer      | 794541.46
 Astronomy    | Taylor      | 617253.94
 Athletics    | Bronfman    | 734550.70
 Statistics   | Taylor      | 395051.74
 Psychology   | Thompson    | 848175.04
 Math         | Brodhead    | 777605.11
 Elec. Eng.   | Main        | 276527.61
 Mech. Eng.   | Rauch       | 520350.65
(20 rows)
```

步骤 4

操作内容：

针对下列查询编写 SQL 语句并执行，记录查询结果（保存成文件或截图）

```
/*13*/

select name

from instructor

where dept_name='Biology';

/*14*/

select title

from course

where dept_name='Comp. Sci.' and credits=3;

/*15*/

select takes.ID,course.course_id,course.title

from course,takes

where takes.ID=2561 and takes.course_id=course.course_id;

/*16*/

select takes.ID,sum(credits)

from course,takes

where takes.course_id=course.course_id and takes.ID='2561'

group by takes.ID;
```

```
/*17*/

select ID,tot_cred

from student;

/*18*/

select distinct name

from course,takes,student

where takes.ID=student.ID and takes.course_id=course.course_id and
course.dept_name='Comp. Sci.';

/*19*/

select distinct instructor.ID

from instructor,teaches

where not exists

(

select *

from teaches

where instructor.ID=ID

);

/*20*/
```

```
select distinct instructor.ID, instructor.name
from instructor,teaches
where not exists
(
select *
from teaches
where instructor.ID=ID
);
/*21*/
create database movie;
create table actors (
   AID varchar(20),
   name varchar(50),
   primary key (AID)
);
create table movies (
   MID varchar(20),
```

```
   title varchar(50),
   primary key (MID)
);
create table actor_role (
   MID varchar(20),
   AID varchar(20),
   rolename varchar(30),
   primary key (MID,AID,rolename),
   foreign key (MID) references movies(MID),
   foreign key (AID) references actors(AID)
);
/*21*/
delete from actor_role;
delete from actors;
delete from movies;
insert into actors values ('01','Charlie Chaplin');
insert into movies values ('M1','City Lights');
```

```
insert into actor_role values ('M1','01','Tramp');

insert into actors values ('02','Stephen Chow');

insert into movies values ('M2','Kung Fu Hustle');

insert into actor_role values ('M2','02','Kung Fu Hustle');

insert into actors values ('03','Jay Chou');

insert into movies values ('M3','Initial D');

insert into actor_role values ('M3','03','Fujiwara Takumi');

insert into actors values ('10215101449','Zhang Guoshuai');

/*23*/

select MID, title, count(rolename)

from actor_role

natural join movies

natural join actors

where name = 'Charlie Chaplin'

group by MID,title;



/*24*/select name from actors
```

```
where name not in (
 select distinct name
 from actor_role natural join actors);
/*25*/
select * from (
        (select name, title
      from actors, movies, actor_role
      where actors.AID = actor_role.AID and movies.MID =
actor_role.MID)
        union
        (select name as name, null as title from
      (select name from actors
        where name not in
        (select distinct name
      from actor_role natural join actors)
        ) as alias1)
        ) as alias2;
/*26*/
```

```
select max(enrollment), min(enrollment)
   from (select sec_id, semester, year, count(distinct id) as enrollment
     from takes
   group by sec_id, semester, year) as alias1
/*26*/
select max(enrollment) as max_enrollment,min(enrollment) as
min_enrollment
from
(
     select count(distinct takes.ID) as enrollment
     from takes
     group by takes.sec_id,takes.course_id,takes.semester,takes.year
) AS C
/*27*/
with C(course_id,sec_id,semester,year,enrollment) as
(
     select course_id,sec_id,semester,YEAR,COUNT(id)
     from takes
```

```
    group by course_id,sec_id,semester,year
)
select C.course_id,C.sec_id,C.semester,C.year,enrollment
from C
where C.enrollment>=all(select enrollment from C)
group by C.course_id,C.sec_id,C.semester,C.year,enrollment
/*28*/
select student.ID,student.name,count(distinct course_id) as
course_num,count(sec_id) as sec_num
from student left outer join takes on student.ID=takes.ID
group by student.ID,student.name
/*29*/
select course_id
from course
where title like'CS-1%'
/*30*/
    select distinct ID, name from (
    select * from teaches natural join instructor)
```

```
    as T where not exists (
    select cs_course.course_id from (
    select course_id from course
    where course_id like 'CS-1%')
    as cs_course where cs_course.course_id not in (
    select course_id from (
    select * from teaches natural join instructor)
    as S where S.name = T.name));
```

(2)

```
    with S(course_id) as (
  select distinct course_id
  from teaches natural join instructor
  where course_id like 'CS-1%')
  select distinct ID, name from (
    select * from teaches natural join instructor) as T
  where ((select count(course_id) from S)=(
     select count(distinct course_id)
```

```
        from teaches natural join instructor
        where name = T.name and course_id like 'CS-1%'
        ));
/*31*/
insert
into student(ID,name,dept_name)
select ID,name,dept_name
from instructor
where not exists
(
    select *
    from student
    where instructor.ID=ID
)
/*32*/
delete
from student
```

```
where student.ID in
(
    select student.ID
    from instructor
    where student.ID=ID
)
/*33*/
update student as S set tot_cred = (
        select sum(credits)
         from takes natural join course
         where S.ID = takes.ID and takes.grade is not null);
34，update instructor
set salary = 10000 * (
         select COUNT(*)
         from teaches
         where teaches.ID = instructor.ID
    )
```

```
35，

create view failed_takes as

select * from takes

where grade='F';

CREATE VIEW

select *

from failed_takes;

36，

select ID, name, 'F' as grade

from student

where 1<(

      select count(*)

      from failed_takes

      where failed_takes.ID=student.ID)

order by ID;

37，  create table grade_point
```

```
	(grade varchar(2),
	point numeric(2,1),
	primary key (grade));
insert into grade_point values
	('A+', 4.3), ('A ', 4.0), ('A-', 3.7), ('B+', 3.3), ('B ', 3.0), ('B-', 2.7), ('C+',
2.3), ('C ', 2.0), ('C-', 1.5), ('D ', 1.3), ('D-', 1.0), ('F', 0);
select * from grade_point;
create or replace function gradeToPoint(in g varchar(2))
returns numeric(2,1) as $$
	declare
		res numeric(2,1);
	begin
		select point into res
		from grade_point
		where grade=g;
		return res;
	end;
 $$ LANGUAGE plpgsql;
```

```
select ID, sum(gradeToPoint(coalesce(grade, 'F')))/(select count(*) as
GPA from takes as S where S.ID=T.ID)
from takes as T
group by ID;
38，with conflicted(building, room_number) as
    (select building, room_number
    from classroom as C
    where 1<(
        select count(*) from section as S
        where S.building=C.building and
S.room_number=C.room_number))
select building, room_number, course_id, sec_id, semester, year
from section natural join conflicted
order by building, room_number;
39，create view faculty as
     select ID, name, dept_name
    from instructor;
CREATE VIEW
```

```
select * from faculty;
```

40，

```
    create view faculty as
     select ID, name, dept_name
     from instructor;
```

```
CREATE VIEW
 select * from faculty;
```

41，不能直接插入，必须声明有 instead of 的触发器，指定 insert 时要干什么。对 CSinstructors 的前三条插入均失败，第一条不满足 salary 范围限制，第二条 0 重复，第三条违反 instructor 对 department 的外键引用。

第四条成功，但其实不满足视图的 where 子句，需要自己在 rule 中修改。

第五条成功。

```
insert into faculty values (0, 'new', 'Comp. Sci.');
create or replace rule r1_faculty_insert as
    on insert to faculty do instead
    insert into instructor values(new.ID, new.name, new.dept_name)
    returning instructor.ID, instructor.name, instructor.dept_name;
insert into faculty values (0, 'new', 'Comp. Sci.');
```

```
select * from faculty order by ID;

create or replace rule r1_CSinstructors_insert as

    on insert to CSinstructors do instead

    insert into instructor values(new.ID, new.name, new.dept_name,
new.salary);

insert into CSinstructors values

    (0, 'new', 'Unknown Dept', 100);

insert into CSinstructors values

    (0, 'new', 'Unknown Dept', 100000);

insert into CSinstructors values

    (1, 'new', 'Unknown Dept', 100000);

insert into CSinstructors values

    (1, 'new', 'Biology', 100000);

insert into CSinstructors values

    (2, 'new', 'Comp. Sci.', 100000);

select * from CSinstructors order by ID;

42，  create user user1 with password 'Password123';

grant select on student to user1;
```

```
gsql -d db2022 -U user1 -W 'Password123' -r

select * from student;

select * from faculty;

43,

grant select on faculty to public;

select * from faculty;
```

存在的问题及解决方案

未出现问题

实验小结

有些语句掌握不熟练，如 order by 语句