

中国石油大学

(北京)

CHINA UNIVERSITY OF
PETROLEUM

数据库原理上机报告

实验三 数据库的查询和数据库的安全性及完整性

院系名称： 信息学院

专业名称： 计算机

学生学号： 2019011777

学生姓名： 刘康来

完成日期 2021 年 11 月 3 日

《数据库原理》上机报告

报告名称	数据库的查询和数据库的安全性及完整性			粘贴 照片
姓名	刘康来	班级	计算机 19-3	
成绩				

1. 上机实践目的与准备知识（简介，300 字以内）

1.实验目的

- (1) 掌握复杂查询的表示；
- (2) 掌握 GROUP BY、HAVING 子句的作用和使用方法；
- (3) 掌握创建用户和对用户授权操作。
- (4) 掌握实体完整性、参照完整性、自定义完整性的设置
- (5) 掌握触发器的定义，初步了解 T-SQL 的变量定义、数据类型等

2. 实验准备

- (1) 了解子查询的表示方法；
- (2) 了解连接查询的表示方法；
- (3) 了解 SELECT 语句 GROUP BY、HAVING 子句的作用和使用方法；
- (4) 了解创建用户和对用户授权操作。
- (5) 了解完整性的定义。
- (6) 了解触发器的定义。

2. 主要实践内容与具体操作步骤（实践内容完成情况要有描述，如执行的 SQL 命令等，有运行结果截图，图大小以保证文字清晰为准）

(1) 查询男性雇员和女性雇员的人数；

```
select
case
    when sex = 1 then concat(cast(count(Name) as varchar(10)),
' (男性)')
    when sex = 0 then concat(cast(count(Name) as varchar(10)),
' (女性)')
```

```

end as '人数'
from Employees
group by sex;

-- scalar
select distinct case
    when Sex = 1 then '男性'
    when Sex = 0 then '女性'
end as sex, (select count(*)
    from Employees as y -- from Employees natural join
Departments; wrong!
    where x.Sex = y.sex) as number
from Employees as x;

```

	人数
1	11(女性)
2	4(男性)

	sex	number
1	女性	11
2	男性	4

(2) 查询员工人数超过 2 人的部门名称和员工的人数;

```

select DepartmentName, count(Name) as '员工人数'
from Employees natural join Departments
group by DepartmentName
having count(Name) > 2;

```

	DepartmentName	员工人数
1	研发部	4

(3) 查询研发部员工的姓名和出生日期（只要年月日），要求各列标题用中文表示；

```
select Name as '姓名', Birthday as '出生日期'
from Employees natural join Departments
where DepartmentName = '研发部';
```

	姓名	出生日期
1	a	2000-01-01
2	b	2000-02-01
3	d	2000-04-01
4	g	2000-07-01

(4) 根据员工号定义用户，完成以下任务：

请用 SQL 的 GRANT 和 REVOKE 语句(加上视图机制)完成以下授权定义或存取控制功能：

(4.1) 用户王明对三个表有 SELECT 权限、INSERT 和 DELETE 权限。

(4.2) 用户刘星对 Employees 表有 SELECT 权限,对 Address 字段具有更新权限。

(4.3) 用户张新具有修改这三个表的结构权限。

(4.4) 用户杨兰具有从每个部门员工中 SELECT 最高工资、最低工资、平均工资的权限,她不能查看每个人的工资。

```
create user '王明';
grant select, insert, delete on YGGL.* to '王明';
create user '刘星';
grant select on Employees to '刘星';
grant update on Employees.Address to '刘星';
grant update(Address) on Employees to '刘星';
create user '张新';
grant select, insert, delete, update on YGGL.* to '张新';
-- grant alter table on YGGL.* to '张新';
create user '杨兰';
create view Em_come as
    select DepartmentName, max(Income), min(Income), avg(Income)
    from Employees natural join Departments natural join Salary
    group by DepartmentName;
grant select on Em_come to '杨兰';
```

(5)在数据库 YGGL 中创建表 ES_U (Eno,OldIncome,NewIncome)：

- 主码命名为 pk1，主码为 (eno)；
- 定义约束条件 ck1，NewIncome 的值不能为空；
- 定义约束条件 ck2，要求 OldIncome 和 NewIncome 值范围都是 [2000,20000]。

```
create table ES_U(  
    Eno varchar(10) primary key,  
    OldIncome int check (OldIncome between 2000 and 20000),  
    NewIncome int not null check (NewIncome between 2000 and  
20000)  
);
```

(6)修改 employees 表，性别的数据类型改为 char (2)，取值只能是‘男’、‘女’、‘F’、‘M’这四个值

```
alter table Employees modify Sex char(2);  
  
-- change the values at first.  
update Employees  
set Sex = case  
    when Sex = '1' then 'M'  
    else 'F'  
end;  
  
alter table Employees  
add check (Sex = '男' or Sex = '女' or Sex = 'F' or Sex =  
'M');
```

(7)设计多个违反上述完整性的数据，分别用 insert 向 ES_U、employees 表插入。

```
insert into ES_U values('a', 0, 0);  
insert into Employees values('100', 'a', '小学', 20000101, 0, 1,  
'0', 0, 0);
```

```
Y66L> insert into ES_U values('a', 0, 0)
[2021-11-03 08:41:16] [23000][4025] CONSTRAINT 'ES_U.OldIncome' failed for 'Y66L`.`ES_U`
```

(8) 在 Salary 表上定义触发器，当更新数据后，如果 Income 值变大，就在 ES_U 表中插入一条记录。设计插入一条合法数据，触发触发器。

```
create trigger raise_salary before update on ES_U
for each row
begin
    if NEW.NewIncome > OLD.NewIncome then
        set NEW.OldIncome = OLD.NewIncome;
    end if;
end;

insert into ES_U values ('a', null, 5000);

update ES_U
set NewIncome = '6000'
where Eno = 'a';
```

	Eno	OldIncome	NewIncome
1	a	5000	6000

(9) 创建表 employees_log，只有员工人数一个属性，向表中插入值为 employees 表的总人数。在 employees 表上定义触发器：当插入或删除一个员工时，就在 employees_log 表中修改员工人数。插入一个员工记录进行测试。

```
insert into Employees_log value((
    select count(*)
    from Employees
));

-- Unsafe query: 'Update' statement without 'where' updates all
table rows at once
create trigger add_num_employees after delete on Employees
for each row
begin
```

```

        update Employees_log
            set Num_employees = (select count(*) from Employees)
            where 1;
    end;

create trigger minus_num_employees after insert on Employees
for each row
begin
    update Employees_log
        set Num_employees = (select count(*) from Employees)
        where 1;
    end;

delete from Employees
where EmployeeID = '1';
insert into Employees values('1', 'b', '高中', 20000201, 'M', 2,
'1', 0, 0);

select *
from Employees_log;

```

	Num_employees
1	15

	Num_employees
1	14

(10) 自行设计一个针对 YGGL 某个表的触发器。

```

-- function must have return...
create procedure abort()
begin
    rollback;
end;

create trigger check_department before delete on Departments
for each row
begin
    if OLD.DepartmentID in (select DepartmentID from Employees)
then

```

```

-- how to terminate the operation?
-- signal sqlstate '45000'; -- 45000 or 45001 ?
call abort();
end if;
end;

delete from Departments
where DepartmentID = 0;

where DepartmentID = 0
[2021-11-03 08:48:09] [HY000][1422] Explicit or implicit commit is not allowed in stored function or trigger

```

3. 总结与问题分析（100 字以上）

遇到的问题：

如何删除约束，查看约束？

alter table Employees add check (Sex = '男' or Sex = '女' or Sex = 'F' or Sex = 'M');

alter table Employees add constraint name_sex check (Sex = '男' or Sex = '女' or Sex = 'F' or Sex = 'M');

一般是命名约束，再删除，可还是报错，用了以下方法解决：

```
alter table Employees drop constraint CONSTRAINT_1;
```

如何在触发器中取消操作？

在 sql 标准中，可用 rollback 解决问题，但 mariadb 数据库报了安全问题。

1. signal sqlstate '45000'; -- 45000 or 45001 ? 这句命令解决。。。

2. 写了一个 procedure，里面可以用 roolback。尝试去写 function，发现必须要有返回值，其实忽略即可。

总结：易用性固然重要，但更重要的是安全，完整性；遇到问题，还是要 Google 或查使用手册，谁知道设计者怎么想的。

