**华东师范大学软件学院上机实践报告**

|  |  |  |
| --- | --- | --- |
| **课程名称**：数据库应用 | **年级**：14级 | **上机实践成绩**： |
| **指导教师**：张召 | **姓名**：王承 |  |
| **上机实践名称**：Transact-SQL实践 | **学号**：10142510221 | **上机实践日期**：16-04-12，16-04-26 |
| **上机实践编号**：No.2 | **组号**：1-221 | **上机实践时间**：4学时 |

***说明：一、二、三部分手写在学院发的试验报告纸上，第四部分开始可以打印***。

**一、目的**

1．掌握 Transact-SQL的基本语法和基本控制结构

2．能够编写简单的触发器和存储过程

**二、内容与设计思想**

1．触发器

（1）有表

emp(empno,ename,salary,deptno)

dept(deptno,dname,count\_emp)

要求实时，动态记录每个部门的人数

设计思路：建立以emp表中数据插入和删除为触发条件的触发器

（2）emp(empno,ename,salary,deptno)

要求记录每个员工的薪水历史记录，包括日期和涨幅

设计思路：建立一个表rec用于记录emp中薪水变化的涨幅和日期，建立以emp中salary数据更新为触发条件的触发器。

2．存储过程

（1）要求完成一存储过程，计算任意两个数的加、减、乘、除

该存储过程有四个参数，分别是两个操作数，一个操作符和一个返回值。

设计思路：存储过程中以操作符的不同作为if的判断条件。

（2）有一张表emp(empno,ename,salary,deptno)，写一存储过程计算每个部门的平均工资。

设计思路：运用avg() 和 group by 求各部门的平均工资。

**三、使用环境**

OS X 10.11.4

MySQL 5.7.11

**四、调试过程**

**1.1**

代码

drop trigger if exists t1;

drop trigger if exists t2;

delimiter //

create trigger t1 after insert on emp

for each row

begin

update dept set count\_emp = count\_emp + 1 where dept.deptno = new.deptno;

end//

create trigger t2 after delete on emp

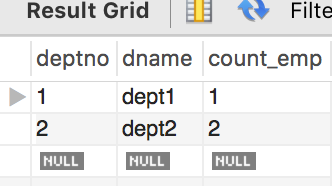
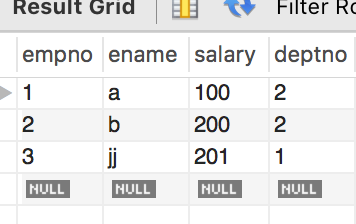
for each row

begin

update dept set count\_emp = count\_emp - 1 where dept.deptno = old.deptno;

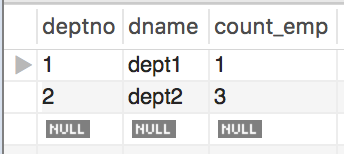
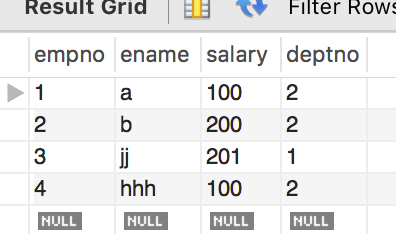
end//

delimiter ;

emp表最初状态 dept表最初状态

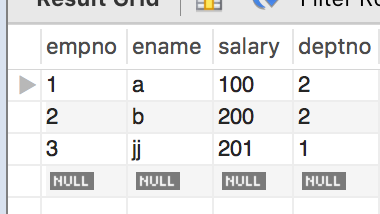
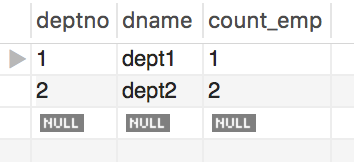
执行

INSERT INTO `test`.`emp` (`empno`, `ename`, `salary`, `deptno`) VALUES ('4', 'hhh', '100', ‘2');

emp表 dept表

执行

delete from emp where empno = 4;

emp表 dept表

1.2

代码

drop trigger if exists record;

delimiter //

create trigger record after update on emp

for each row

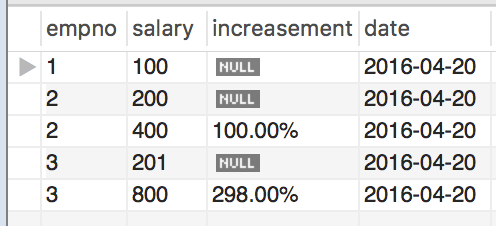
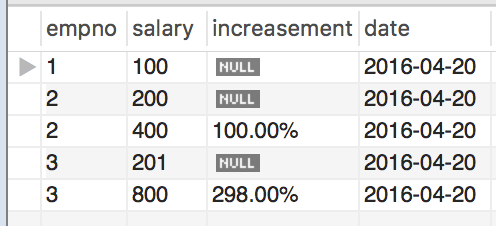
begin

insert into rec(r\_empno, r\_salary, r\_increase, r\_date)

values((select empno from emp where salary = new.salary), new.salary, (new.salary - old.salary)/old.salary, curdate());

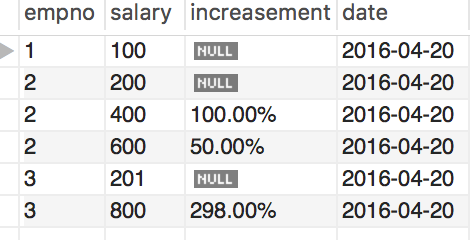
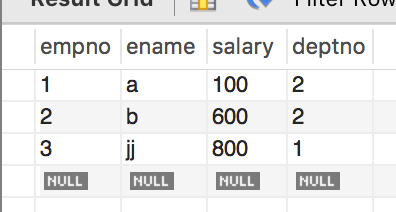
end //

delimiter ;

用于记录的表rec的初始数据 emp表初始数据

执行

UPDATE `test`.`emp` SET `salary`='600' WHERE `empno`='2';

emp表数据 rec表数据

rec表的查询语句

SELECT r\_empno as empno, r\_salary as salary, concat(truncate(r\_increase \* 100,2),'%') as 'increasement', r\_date as date FROM test.rec order by empno;

rec表建表语句

create table rec (

r\_id int,

r\_empno int,

r\_salary int,

r\_increase float,

r\_date date,

primary key (r\_id),

foreign key (r\_empno) references emp(empno)

);

2.1

代码

drop procedure if exists cal;

delimiter //

create procedure cal(IN a float, IN ch char(1), IN b float, OUT c float)

begin

if ch = '+' then set c = a+b; end if;

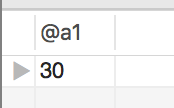
if ch = '-' then set c = a-b; end if;

if ch = '\*' then set c = a\*b; end if;

if ch = '/' then set c = a/b; end if;

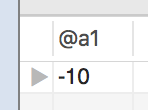
end//

delimiter ;

执行

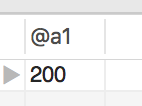
set @a1 = 0;

call cal(10,'+',20,@a1);

select @a1;

set @a1 = 0;

call cal(10,'-',20,@a1);

select @a1;

set @a1 = 0;

call cal(10,'\*',20,@a1);

select @a1;



set @a1 = 0;

call cal(10,'/',20,@a1);

select @a1;

2.2

代码

drop procedure if exists avgSal;

delimiter //

create procedure avgSal()

begin

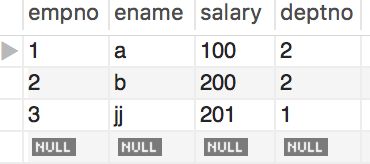
select deptno,avg(salary) from emp group by deptno;

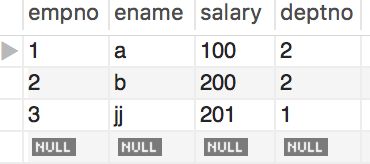
end//

delimiter ;

call avgsal();

执行

emp表状态

查询结果

**五、总结**

由于mysql与sqlserver的语法存在略微的差异，在写触发器和存储过程的过程中出现了一些bug，在及时查阅相关资料并加以探索之后已经得到解决。通过这次实验进一步熟悉触发器和存储过程的用法和编写方法，更加熟练地掌握了数据库的知识技能。

**六、附录**