电子信息学院

实验报告书

课程名： 大型数据库

题 目： 实验九 触发器(二)

实验类别 【验证】

姓 名： 张飒

评语：

实验态度：认真（ ） 一般（ √ ） 较差（ ）

实验结果：正确（ ） 部分正确（√ ）错（ ）

实验理论： 掌握（ √ ） 熟悉（ ） 了解（ ） 生疏（ ）

操作技能：较强（ ） 一般（ √ ） 较差（ ）

实验报告： 较好（ ） 一般（ √ ） 较差（ ）

成绩： 指导教师： 林志杰

批阅时间：2019 年 11月1日

评语：

实验态度：认真（ ） 一般（ ） 较差（ ）

实验结果：正确（ ） 部分正确（ ）错（ ）

实验理论：掌握（ ） 熟悉（ ） 了解（ ） 生疏（ ）

操作技能：较强（ ） 一般（ ） 较差（ ）

实验报告：较好（ ） 一般（ ） 较差（ ）

成绩： 指导教师：

批阅时间： 年 月 日

评语：

实验态度：认真（ ） 一般（ ） 较差（ ）

实验结果：正确（ ） 部分正确（ ）错（ ）

实验理论：掌握（ ） 熟悉（ ） 了解（ ） 生疏（ ）

操作技能：较强（ ） 一般（ ） 较差（ ）

实验报告：较好（ ） 一般（ ） 较差（ ）

成绩： 指导教师：

批阅时间： 年 月 日

1．实验目的

（1）掌握oracle 11g的SQL语言中的触发器的相关操作。

（2）熟练oracle创建数据库的过程，能够对触发器进行深入理解和掌握。

（3）掌握oracle的中触发器的存在的意义和作用。

（4）掌握SQL语言中操作触发器的基本技能。

2．实验目标：

在本次实验中，学生应该能够利用所学触发器的基本理论知识，进行触发器的创建、更新、删除和修改触发器的基本操作，能够在实际应用中熟练运用触发器。

3. 触发器简介

触发器的定义就是说某个条件成立的时候，触发器里面所定义的语句就会被自动的执行。因此触发器不需要人为的去调用，也不能调用。

然后，触发器的触发条件其实在你定义的时候就已经设定好了。

这里面需要说明一下，触发器可以分为语句级触发器和行级触发器。

简单的说就是语句级的触发器可以在某些语句执行前或执行后被触发。而行级触发器则是在定义的了触发的表中的行数据改变时就会被触发一次。

具体说明：

1、 在一个表中定义的语句级的触发器，当这个表被删除时，程序就会自动执行触发器里面定义的操作过程。这个就是删除表的操作就是触发器执行的条件了。

2、 在一个表中定义了行级的触发器，那当这个表中一行数据发生变化的时候，比如删除了一行记录，那触发器也会被自动执行了。

4、触发器语法

触发器的语法：

**create [or replace] tigger 触发器名 触发时间 触发事件**

**on 表名**

**[for each row]**

**begin**

**pl/sql语句**

**end**

其中：

**触发器名**：触发器对象的名称。由于触发器是数据库自动执行的，因此该名称只是一个名称，没有实质的用途。

**触发时间**：指明触发器何时执行，该值可取：

before：表示在数据库动作之前触发器执行;

after：表示在数据库动作之后触发器执行。

**触发事件**：指明哪些数据库动作会触发此触发器：

insert：数据库插入会触发此触发器;

update：数据库修改会触发此触发器;

delete：数据库删除会触发此触发器。

**表 名**：数据库触发器所在的表。

**for each row**：对表的每一行触发器执行一次。

如果没有这一选项，则只对整个表执行一次。

触发器能实现如下功能：

功能：

1、 允许/限制对表的修改

2、 自动生成派生列，比如自增字段

3、 强制数据一致性

4、 提供审计和日志记录

5、 防止无效的事务处理

6、 启用复杂的业务逻辑

5．实验内容与步骤

1)、按照给出的提示，创建触发器，用来记录表的删除数据；

--创建表

CREATE TABLE employee(

id VARCHAR2(4) NOT NULL,

name VARCHAR2(15) NOT NULL,

age NUMBER(2) NOT NULL,

sex CHAR NOT NULL

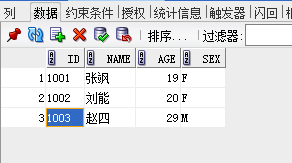
);

--插入数据(插入三行元组，包含你本人的学号和名字)

INSERT INTO "SYSTEM"."EMPLOYEE" (ID, NAME, AGE, SEX) VALUES ('1001', '张飒', '19', 'F');

INSERT INTO "SYSTEM"."EMPLOYEE" (ID, NAME, AGE, SEX) VALUES ('1002', '刘能', '20', 'M');

INSERT INTO "SYSTEM"."EMPLOYEE" (ID, NAME, AGE, SEX) VALUES ('1003', '赵四', '29', 'M');



--创建记录表(包含已插入的数据记录)

create table emp\_log(

type varchar2(20),

logdate date,

emp\_id varchar2(10),

emp\_name varchar2(15),

emp\_age number,

emp\_sex char(5)

);

--创建触发器

create or replace

trigger emp\_insert\_trigger

before insert or delete or update

on employee

for each row

begin

if inserting THEN

insert into emp\_log values('insert',sysdate,:new.id, :new.name, :new.age, :new.sex);

elsif updating THEN

insert into emp\_log values('update',sysdate,:new.id, :new.name, :new.age, :new.sex);

elsif deleting then

insert into emp\_log values('delete',sysdate,:old.id, :old.name, :old.age, :old.sex);

end if;

end;

--进行测试，写出测试语句的代码和执行结果；

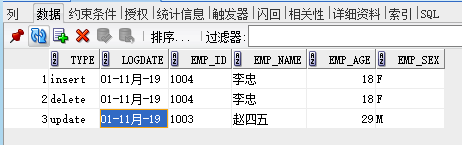
测试语句：

INSERT INTO "SYSTEM"."EMPLOYEE" (ID, NAME, AGE, SEX) VALUES ('1004', '李忠', '18', 'F')

UPDATE "SYSTEM"."EMPLOYEE" SET NAME = '赵四五' WHERE id=1003

DELECT "SYSTEM"."EMPLOYEE" where id=1004

执行结果：



创建以下三张表并插入数据如下：

创建S、SC和C三张表及添加表命令如下：

Create Table S

(Sno CHAR(5) NOT NULL,

Sname VARCHAR(20),

Sage SMALLINT CHECK(Sage>=15 AND Sage<=45),

Ssex CHAR(2),

Sdept CHAR(10),constraint pr\_sno primary key(sno));

Create Table C

(Cno CHAR(2) NOT NULL,

Cname VARCHAR(20) NOT NULL,

Cpno CHAR(2),

Ccredit SMALLINT,constraint pr\_Cno primary key(Cno));

Create Table SC(

Sno CHAR(5) NOT NULL,

Cno CHAR(2) NOT NULL,

Grade SMALLINT,constraint pr\_Cno primary key(Sno,Cno));

INSERT INTO Student VALUES(‘18001’,’钱横’,18,’男’,’CS’);

INSERT INTO Student VALUES (‘18002’,’王林’,18,’男’,’CS’);

INSERT INTO Student VALUES (‘18003’,’李民’,20,’男’,’IS’);

INSERT INTO Student VALUES (‘18004’,’赵蕊’,18,’女’,’MA’);

INSERT INTO Student VALUES (‘18005’,’姬彧’,19,’女’,’MA’);

INSERT INTO Student VALUES (‘18006’,’姬彧卿’,18,’女’,’CS’);

INSERT INTO Student VALUES (‘08007’,’姬彧瑾’,18,’女’,’CS’);

INSERT INTO Course VALUES (‘1’,’数据库系统’,’5’,4);

INSERT INTO Course VALUES (‘2’,’数学分析’,null,2);

INSERT INTO Course VALUES (‘3’,’信息系统导论’,’1’,3);

INSERT INTO Course VALUES (‘4’,’操作系统原理’,’6’,3);

INSERT INTO Course VALUES (‘5’,’数据结构’,’7’,4);

INSERT INTO Course VALUES (‘6’,’数据处理基础’,null,4);

INSERT INTO Course VALUES (‘7’,’C语言’,’6’,3);

INSERT INTO SC VALUES (‘18001’,’1’,87);

INSERT INTO SC VALUES (‘18001’,’3’,90);

INSERT INTO SC VALUES (‘18001’,’2’,67);

INSERT INTO SC VALUES (‘18002’,’2’,95);

INSERT INTO SC VALUES (‘18002’,’3’,88);

INSERT INTO SC VALUES (‘18004’,’4’,52);

INSERT INTO SC VALUES (‘18003’,’5’,80);

INSERT INTO SC VALUES (‘18001’,’5’,83);

完成下列各题：

1. 创建触发器，当对选课表进行插入，更新，删除的时候，对表格变更的数据行进行备份；

create table c\_log(

type varchar2(20),

logdate date,

c\_cno varchar2(10),

c\_cname varchar2(15),

c\_pno number,

c\_credit smallInt

);

B) 创建触发器，当更改course课程表中课程号的时候，将选课表的中的信息同时进行更改来保证数据库中数据的一致性；

create or replace

trigger c\_insert\_trigger

before insert or delete or update

on c

for each row

begin

if inserting THEN

insert into c\_log values('insert',sysdate,:new.cno, :new.name, :new.pno, :new.credit);

elsif updating THEN

insert into c\_log values('update',sysdate,:new.cno, :new.name, :new.pno, :new.crefit);

elsif deleting then

insert into c\_log values('delete',sysdate,:old.cno, :old.name, :old.pno, :old.credit);

end if;

end;

C) 创建触发器，当更改课程学分不在1~~4的范围时候，将修改记录存入course\_log记录表中，要求记录修改课程的课程号，课程名，修改日期，和修改学分的原值和修改值。

CREATE OR REPLACE TRIGGER course\_credit\_update\_log

AFTER UPDATE

ON course

FOR EACH ROW

BEGIN

IF :new.ccredit NOT BETWEEN 1 AND 4 THEN

INSERT INTO course\_log VALUES(cno,cname,sysdate,:old.ccredit,:new.ccredit);

END IF;

END;

4.结果分析与实验体会