insert into emp (empno, ename) values(1000,'xiaoming');

create or replace trigger t1

after insert

on emp

for each row

begin

dbms\_output.put\_line('congratulations, insert success.');

end;

**应用场景一：数据库的备份和同步功能**

**例如：建立一个备份职员表，当更新职员工资的时候，备份表也同步更新**

**scott sal=3000;**

**update emp set sal=3800 where ename='SCOTT';**

**只有在行级触发器才会产生NEW OLD 两个记录变量**

记录变量new 存储SCOTT更新后的整行信息 new.sal

记录变量old 存储SCOTT更新前的整行信息 old.sal

create or replace trigger t1

after update

on emp

for each row

begin -- 执行体里用new old 要在new old 前面加:

update emp\_bak set sal= :new.sal where ename=:old.ename;

end;

create or replace trigger t1

after update

on emp

for each row

begin -- 执行体里用new old 要在new old 前面加:

update emp\_bak set sal= :new.sal, job=:new.job where ename=:old.ename;

end;

insert into emp(empno,ename) values(2000,'a');

记录变量new 存储SCOTT更新后的整行信息 new.sal

create or replace trigger t2

after insert

on emp

for each row

begin -- 执行体里用new old 要在new old 前面加:

insert into emp\_bak(empno,ename) values(:new.empno,:new.ename);

end;

delete from emp where ename='a';

记录变量old 存储SCOTT更新前的整行信息 old.sal

create or replace trigger t3

after delete

on emp

for each row

begin -- 执行体里用new old 要在new old 前面加:

delete from emp\_bak where ename= :old.ename;

end;

利用分支结构 将同一个表下的多个事件放在一起，做一个触发器

create or replace trigger t1

after update or insert or delete

on emp

for each row

begin

if updating then

update emp\_bak set sal= :new.sal, job=:new.job where ename=:old.ename;

elsif inserting then

insert into emp\_bak(empno,ename) values(:new.empno,:new.ename);

else

delete from emp\_bak where ename= :old.ename;

end if;

end;

触发约束用法

update emp set sal=5000 where deptno=20;

当更新EMP表里部门编号为20时，我才同步更新EMP\_BAK表里内容。

记录变量new 存储SCOTT更新后的整行信息 new.sal

记录变量old 存储SCOTT更新前的整行信息 old.sal

when 只能在行级触发器来使用。

方法一：

create or replace trigger t1

after update

on emp

for each row

when (old.deptno=20) --when里用new old 不要在new old 前面加:

begin -- 执行体里用new old 要在new old 前面加:

update emp\_bak set sal= :new.sal, job=:new.job where ename=:old.ename;

end;

方法二：

create or replace trigger t1

after update

on emp

for each row

begin -- 执行体里用new old 要在new old 前面加:

if :old.deptno=20 then

update emp\_bak set sal= :new.sal, job=:new.job where ename=:old.ename;

end if;

end;

**应用场景二：数据确认**

**例如：只能涨工资，不能降工资。**

**update emp set sal=2000 where ename='SCOTT';**

create or replace trigger t1

before update

on emp

for each row

begin

if :old.sal>:new.sal then

raise\_application\_error(-20000,' you can not reduce my sal.');

end if;

end;

**应用场景三：级联修改触发器**

**例如：当部门表的某个部门编号修改时，职员表对门的部门编号也得做出相应的修改。**

update dept1 set deptno=21 where deptno=20;

create table emp1 as select \* from emp;

create table dept1 as select \* from dept;

create or replace trigger t1

after update

on dept1

for each row

begin

update emp1 set deptno=:new.deptno where deptno=:old.deptno;

end;

**应用场景四：复杂的安全性检查**

**例如：禁止非工作时间插入表emp**

**create or replace trigger t1**

**before insert**

**on emp**

**begin**

if trim(to\_char(sysdate,'day')) in ('friday','sunday') or to\_char(sysdate,'hh24') not between 9 and 24 then

raise\_application\_error(-20000,'inhibitting input.');

end if;

end;

delete from emp1;

create or replace trigger t1

after delete

on emp1

for each row

begin

dbms\_output.put\_line('errors');

end;