**湘潭大学**

**实 验 报 告**

**课 程： Oracle数据库**

**实验题目： 异常处理**

**学 院： 信息工程学院**

**专 业： 网络 工 程**

**班 级： 2015网络工程2**

**学 号： 2015551621**

**姓 名： 王 康**

**完成日期： 2017年11月18日**

**一、实验目的**

1. 熟悉oracle异常的概念，掌握Oracle异常处理的方法。

**二、实验内容**

有针对性地给“实验四”上机内容中的过程与函数增加异常处理部分，保留原来的过程与函数，修改后的过程与函数另外保存，如obj4\_1另存为obj6\_1，以此类推。

1. 输出名为SMITH的雇员的薪水和职位。(存储过程)

create or replace

procedure obj6\_1(name EMP.ENAME%type)

as

vsal EMP.SAL%type;

vjob EMP.JOB%type;

begin

select SAL,JOB

into vsal,vjob

from EMP

where ENAME = name;

dbms\_output.put\_line('sal:'||vsal||',job:'||vjob);

exception

when no\_data\_found then

dbms\_output.put\_line('该员工不存在！');

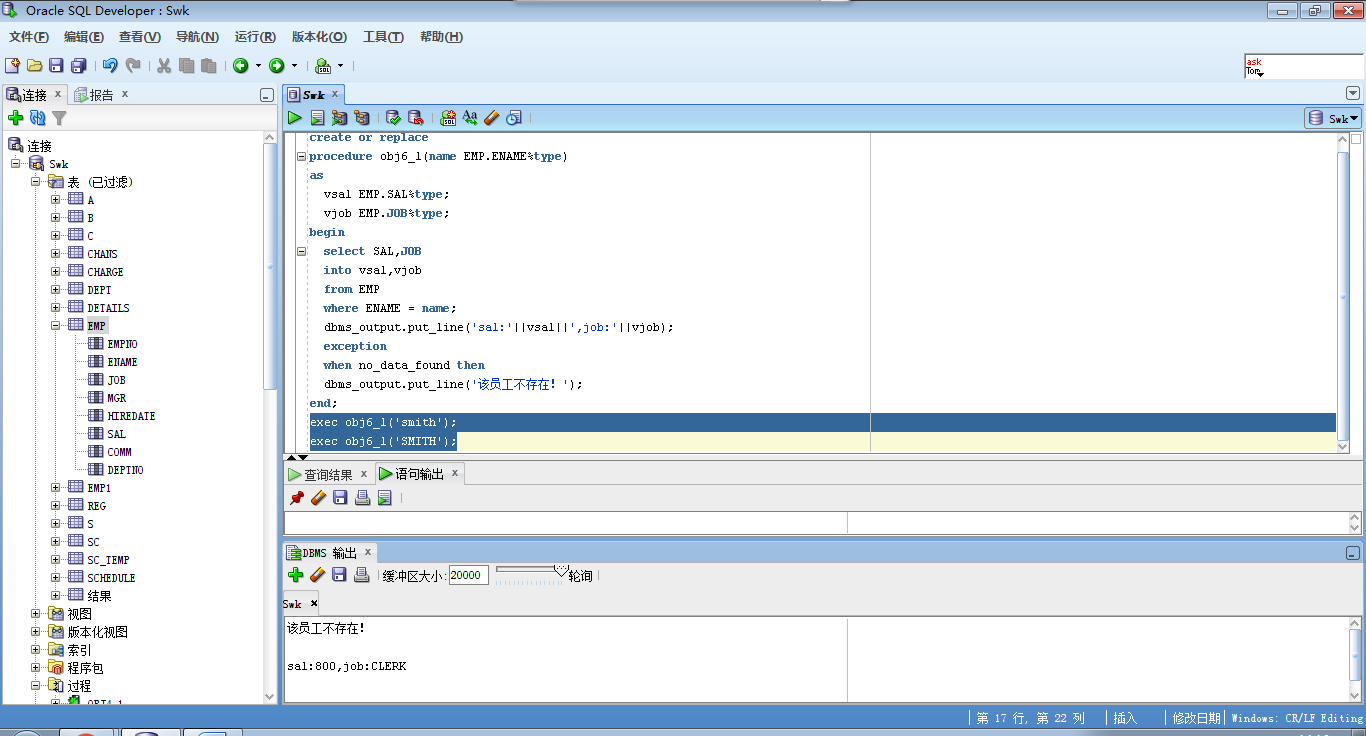
end;

调用过程：

**set serveroutput on;**

**execute obj6\_1（‘SMITH’）;**

结果如下：



1. 接收部门编号，输出部门名和地理位置。(存储过程)

create or replace

procedure obj6\_2(dno in DEPT.DEPTNO%type)

as

vname DEPT.DNAME%type;

vloc DEPT.LOC%type;

begin

select DNAME,LOC

into vname,vloc

from DEPT

where DEPTNO = dno;

dbms\_output.put\_line('name:'||vname||',loc:'||vloc);

exception

when no\_data\_found then

dbms\_output.put\_line('请输入正确的部门名。');

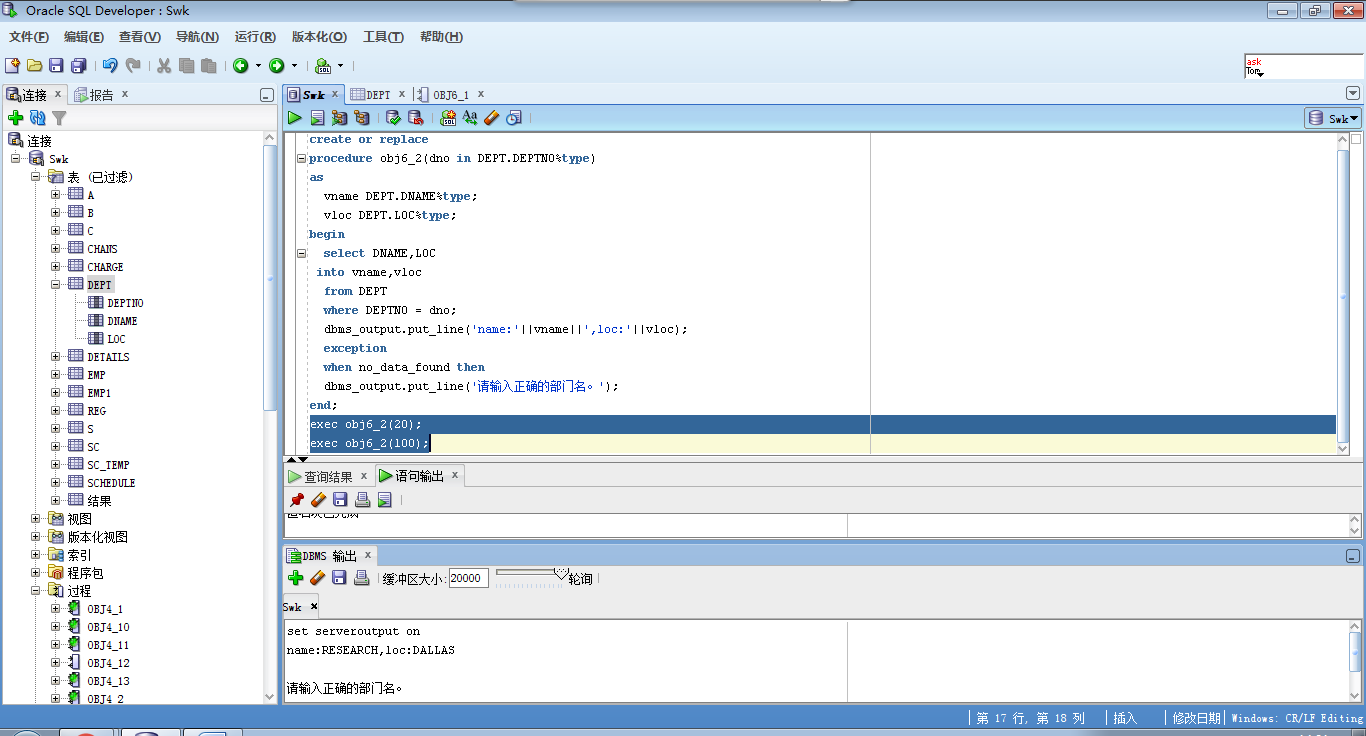
end;

调用过程：

**set serveroutput on;**

**execute obj6\_2（4799）;**

结果如下：



3、接收雇员号，输出该雇员的工资和提成，没有提成的用0替代。（用%type实现）。(存储过程)

 create or replace

procedure obj6\_3(eno in EMP.EMPNO%type)

as

vsal EMP.SAL%type;

vcomm EMP.COMM%type;

begin

select SAL,nvl(COMM,0)

into vsal,vcomm

from EMP

where EMPNO = eno;

dbms\_output.put\_line('empno:'||eno||',sal:'||vsal||',comm:'||vcomm);

exception

when no\_data\_found then

dbms\_output.put\_line('未找到该员工。');

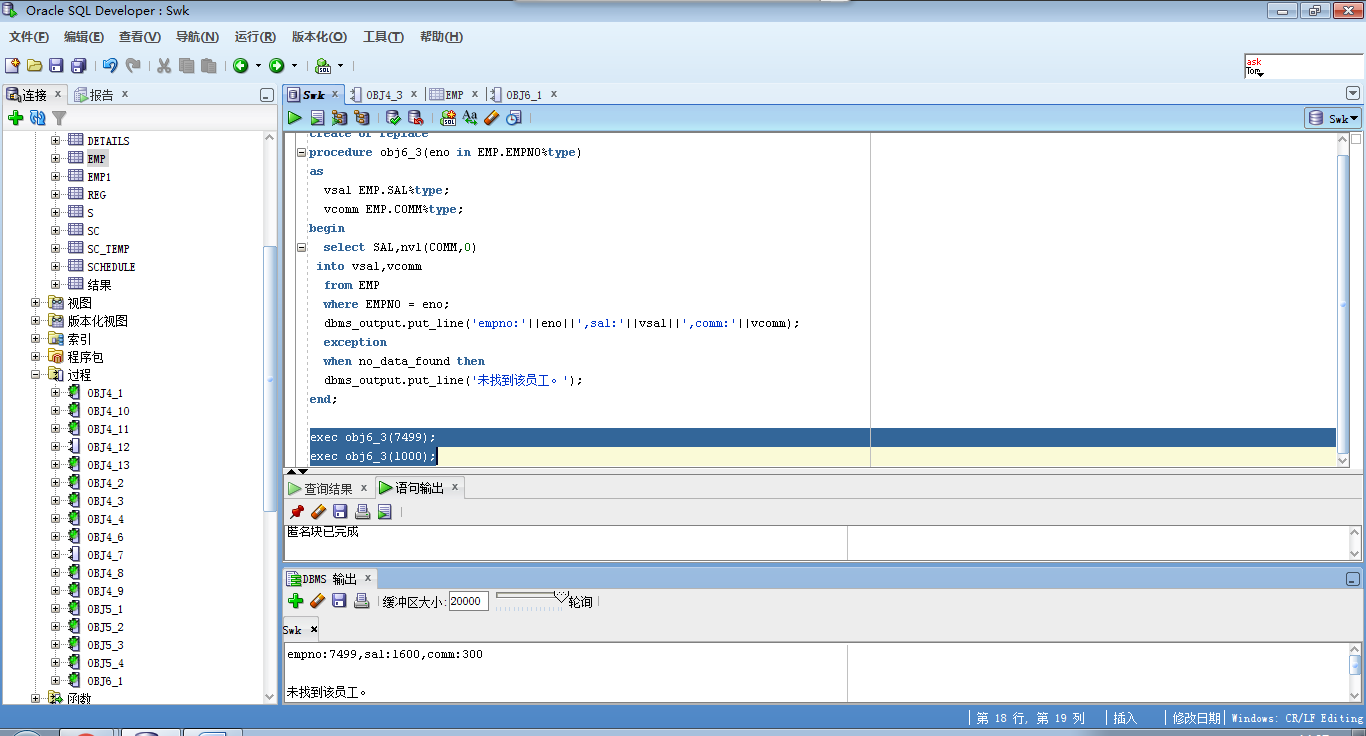
End

调用过程：

**set serveroutput on;**

**execute obj6\_3（7499）;**

结果如下：

;

4、接收雇员号，输出该雇员的所有信息，没有提成的用0替代。（用%rowtype实现）。(存储过程)

 create or replace

procedure obj6\_4(eno in EMP.EMPNO%type)

as

vemp EMP%rowtype;

begin

select EMPNO,ENAME,JOB,MGR,HIREDATE,SAL,nvl(COMM,0),DEPTNO

into vemp.empno,vemp.ename,vemp.job,vemp.mgr,vemp.hiredate,vemp.sal,vemp.comm,vemp.deptno

from EMP

where EMPNO = eno;

dbms\_output.put\_line('empno:'||eno||',ename:'||vemp.ename||',job:'||vemp.job||

',mgr:'||vemp.mgr||',hiredate:'||vemp.hiredate||',sal:'||vemp.sal||',comm:'||vemp.comm||',deptno:'||vemp.deptno);

exception

when no\_data\_found then

dbms\_output.put\_line('没有该雇员。');

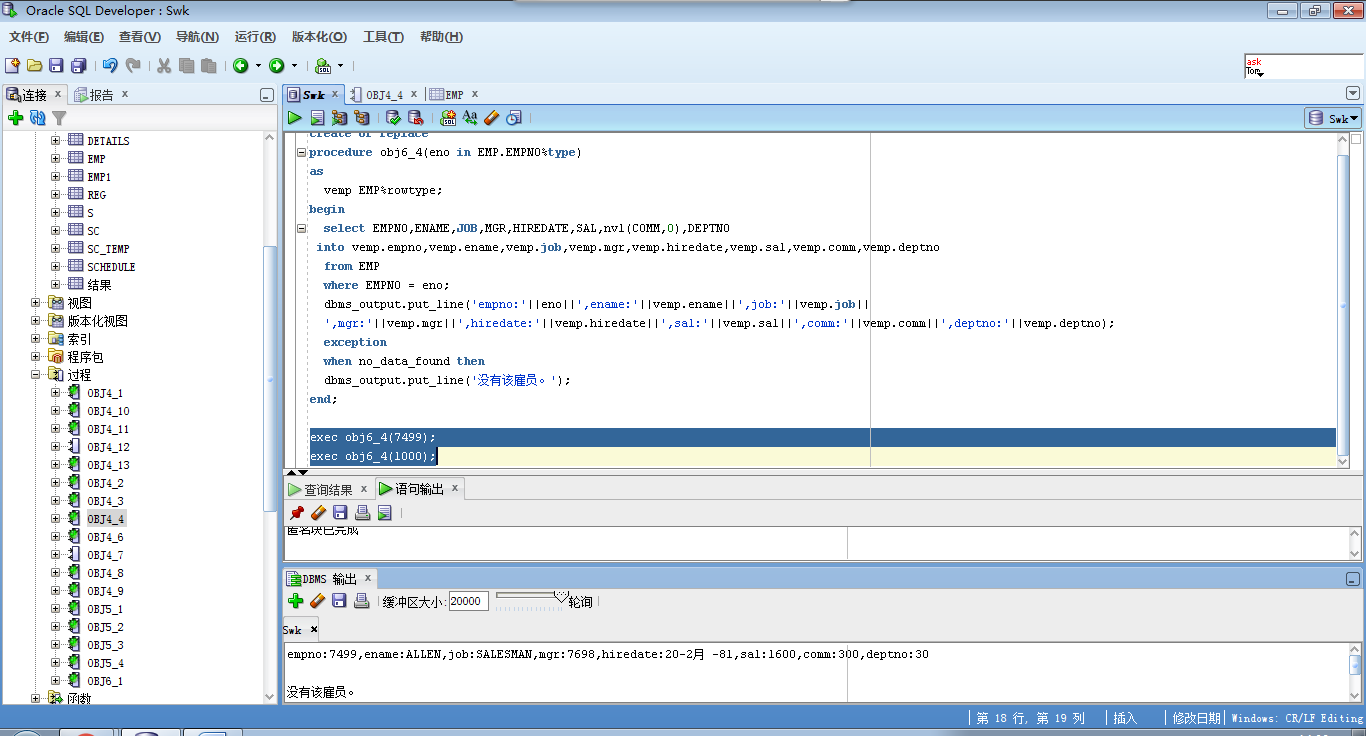
end;

调用过程：

**set serveroutput on;**

**execute obj6\_4;**

结果如下：



5、接收雇员号，输出该雇员的工资。(存储函数)

 create or replace

function obj6\_5(eno in EMP.EMPNO%type)

return number

as

vsal EMP.SAL%type;

begin

select sal into vsal

from EMP

where EMPNO = eno;

dbms\_output.put\_line('工资为：'||vsal);

exception

when no\_data\_found then

dbms\_output.put\_line('未找到该雇员。');

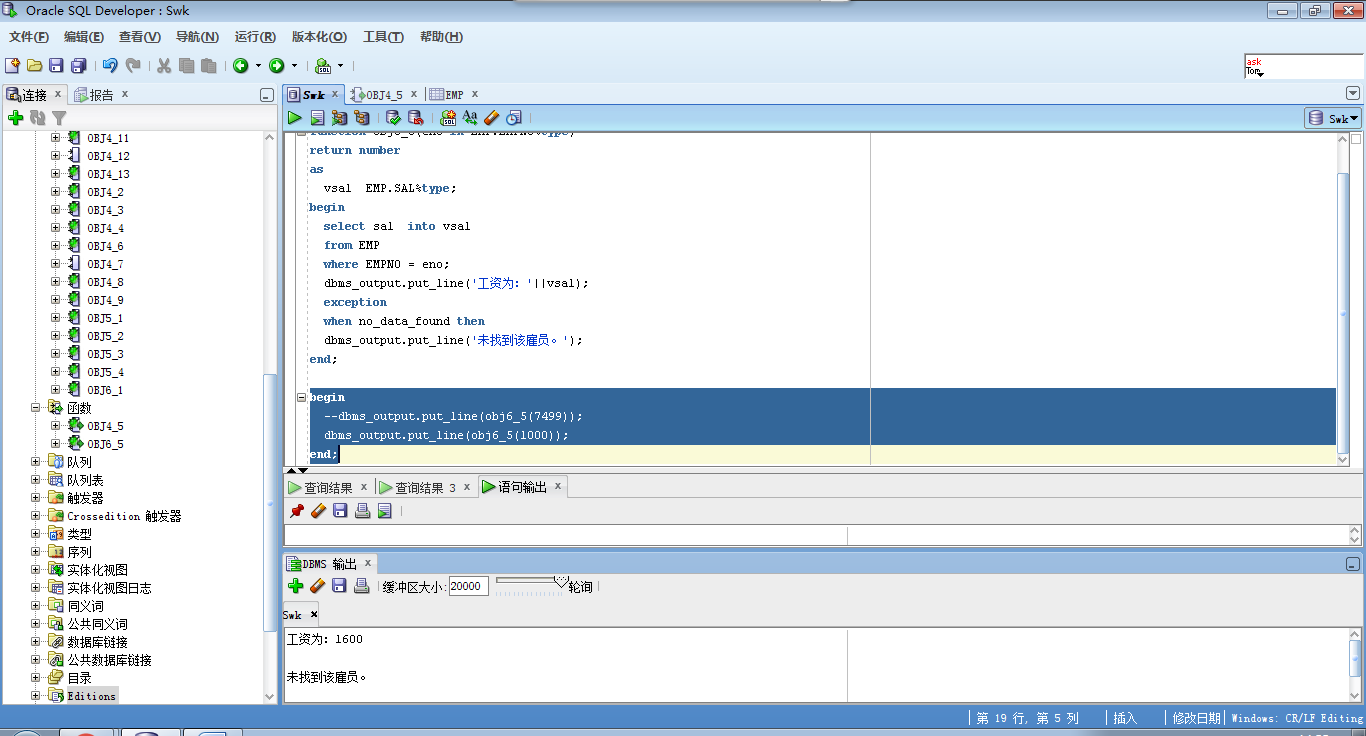
end;

调用过程：

**set serveroutput on;**

**execute obj6\_5（7499）;**

结果如下：



6、接收一个雇员名或雇员编号，判断他的job，根据job不同，为他增加相应的sal（用if-elsif实现，不要改动到基本表emp，创建一个与emp表一模一样的表emp1）。(存储过程)

    Job         raise

    clerk       +500

    salesman    +1000

    analyst     +1500

    otherwise   +3000

create or replace

procedure obj6\_6(eno in EMP1.EMPNO%type)

as

vjob emp1.job%TYPE;

vsal emp1.sal%TYPE;

vcomment varchar2(26);

begin

select job,sal into vjob,vsal from emp1 where empno=eno;

if vjob='CLERK' then

vsal:=vsal+500;

elsif vjob='SALESMAN' then

vsal:=vsal+1000;

elsif vjob='ANALYST' then

vsal:=vsal+1500;

elsif vjob='OTHERWISE' then

vsal:=vsal+2000;

end if;

vcomment:=vjob;

update emp1 set sal=vsal where empno=eno;

dbms\_output.put\_line('职位为:'||vcomment||' 薪水'||vsal||' 修改成功！！！！');

exception

when no\_data\_found then

dbms\_output.put\_line('未找到该雇员。');

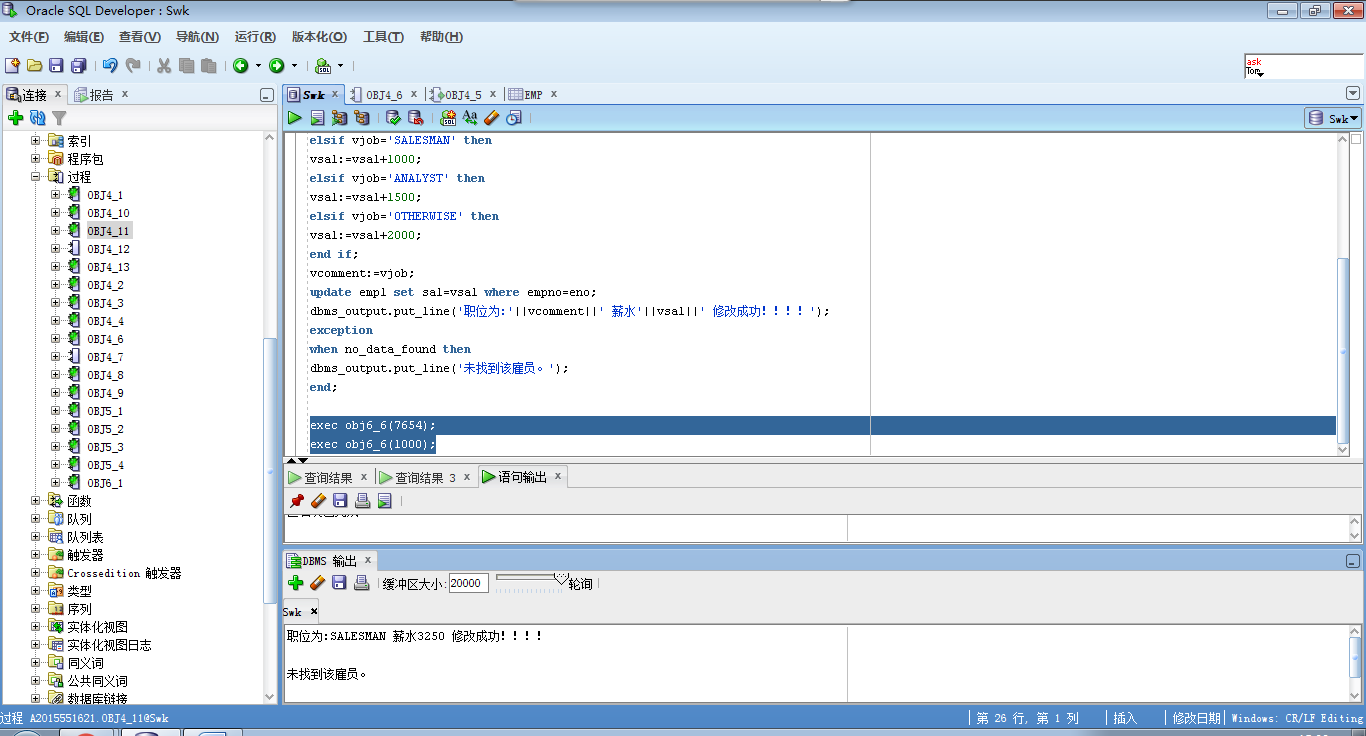
end;

调用过程：

**set serveroutput on;**

**execute obj6\_6（7499）;**

结果如下：



7、输入部门编号，按照下列加薪比例执行给该部门的雇员加薪(用CASE实现，修改emp1表的数据) (存储过程)

    deptno  raise(%)

    10      8%

    20      10%

    30      20%

    40      20%

    加薪比例以现有的sal为标准。

create or replace

procedure obj6\_7(dno in EMP1.DEPTNO%type)

as

vsal emp1.sal%TYPE;

vdeptno emp1.deptno%type;

execp EXCEPTION;

Cursor cur is select deptno,sal from emp1 where deptno=dno;

begin

for result\_row in cur LOOP

if result\_row.sal=0 then

raise execp;

end if;

case result\_row.deptno

when 10 then

vsal:=result\_row.sal\*1.08;

when 20 then

vsal:=result\_row.sal\*1.1;

when 30 then

vsal:=result\_row.sal\*1.2;

when 40 then

vsal:=result\_row.sal\*1.2;

end case;

update emp1 set sal=vsal where deptno=dno;

dbms\_output.put\_line('部门:'||dno||' 薪水:'||vsal||' 修改成功！！！！');

end LOOP;

exception

when no\_data\_found then

dbms\_output.put\_line('没有该部门。');

when execp then

dbms\_output.put\_line('该员工的工资出错。');

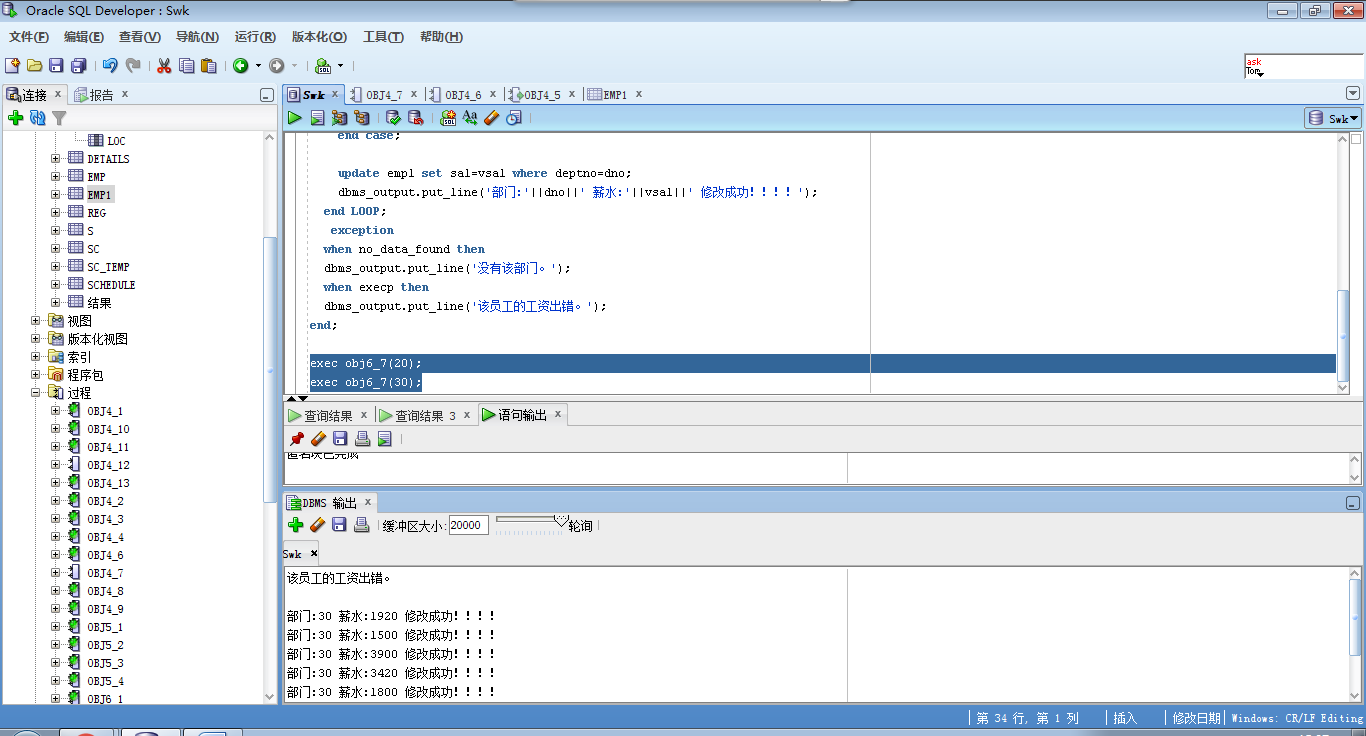
end;

调用过程：

**set serveroutput on;**

**execute obj6\_7（7499）;**

结果如下：



以下8-12题与以下表有关。先从guo模式中将这些表等复制到自己的模式中。

s： 学生基本情况表，有学号、姓名、性别、专业、生日、状态等列。

reg：注册信息表，有学号、 注册日期等列。

schedule： 学费标准表，有专业、学年、学费等列。

charge： 收费表，有学年、学号、应交学费、已交学费等列。

details： 收费明细表，有编号、学年、 学号、收费日期、学费等列。

学生每学年开学前必须注册。开始注册前要初始化学生表S，所有状态为“Registered”的学生的状态设置为"Normal"，注册后设置为"Registered"，还有“毕业生”、“开除”、“休学”等状态。注册时生成应交学费记录(charge)。学生每学年按学生所属专业收取学费，交学费时产生收费明细记录，并修改charge表中相应记录。

8、在学生表S中增加一列，用来记录学生的密码，写一个PL/SQL程序，模拟登录的过程。输入学号和密码，判断是否正确，对于登录成功和失败分别给出提示信息。(存储过程)

create or replace

procedure obj4\_8(v\_username S.SNO%type,v\_password S.PASSWORD%type)

as

v\_userpassword S.PASSWORD%type;

begin

select password into v\_userpassword from S where SNO = v\_username; --查询密码，如果密码正确即可以登录(也对用户名进行检测是否存在);

if v\_password=v\_userpassword then --用户名存在密码正确，可成功登录

dbms\_output.put\_line('登录成功！！！');

else

dbms\_output.put\_line('登录失败！！！');

end if;

exception --查询不到用户，抛出异常！！！

when no\_data\_found then

dbms\_output.put\_line('没有该用户！');

end;

调用过程：

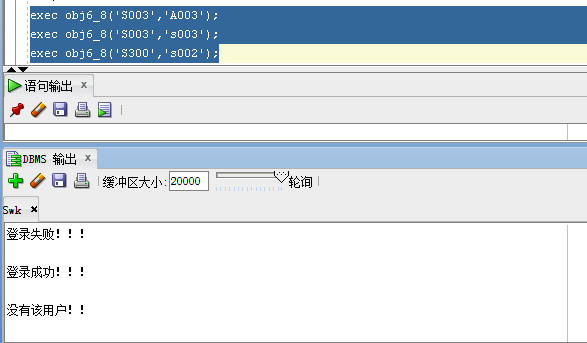
**set serveroutput on;**

exec obj6\_8('S003','A003');

exec obj6\_8('S003','s003');

exec obj6\_8('S300','s002');

结果如下：



9、编写一个向学费标准表添加记录的过程。

create or replace

procedure obj6\_9(v\_SPECIALITY SCHEDULE.SPECIALITY%type,v\_SHYEAR SCHEDULE.SHYEAR %type,v\_TUITION SCHEDULE.TUITION%type)

as

excep EXCEPTION;

num integer;

begin

select count(\*) n into num from SCHEDULE where SPECIALITY = v\_SPECIALITY and SHYEAR = v\_SHYEAR;

if num>=1 then

raise excep;

else

insert into SCHEDULE(SPECIALITY,SHYEAR,TUITION) values(v\_speciality,v\_shyear,v\_tuition);

dbms\_output.put\_line('插入成功。');

end if;

exception

when excep then

dbms\_output.put\_line('已存在该专业在该年的收费记录');

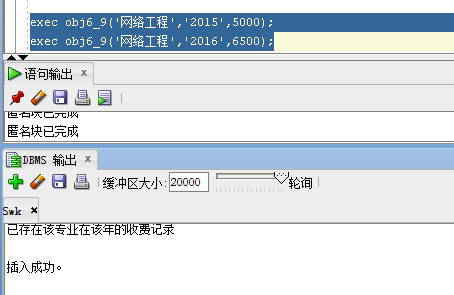
end;

调用过程：

exec obj6\_9('网络工程','2015',5000);

exec obj6\_9('网络工程','2016',6500);

结果如下：



10、编写一个学生注册的过程，注册日期默认为当天，以学号为参数。

 create or replace

procedure obj6\_10(v\_sno REG.SNO%type)

as

v\_regdate REG.REGDATE%type;

excep EXCEPTION;

num integer;

begin

select count(\*) n into num from REG where sno = v\_sno;

if num>=1 then

raise excep;

else

select to\_timestamp(sysdate,'DD-MON-RR') into v\_regdate from dual;

insert into REG(SNO,REGDATE) values(v\_sno,v\_regdate);

dbms\_output.put\_line('注册成功。');

end if;

exception

when excep then

dbms\_output.put\_line('该用户名已存在');

end;

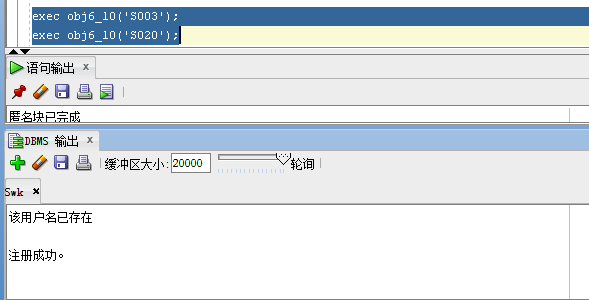
调用过程：

**set serveroutput on;**

exec obj6\_10('S003');

exec obj6\_10('S020');

结果如下：



11、编写一个收学费的过程，收费日期默认为当天，以学年、 学号、学费为参数。

 create or replace

procedure obj6\_11(v\_shyear in CHARGE.SHYEAR%type,v\_sno in CHARGE.SNO%type,v\_balance in CHARGE.BALANCE%type)

as

v\_regdate REG.REGDATE%type;

begin

select to\_timestamp(sysdate,'dd-mon-rr') into v\_regdate from dual;

insert into REG(SNO,REGDATE) values(v\_sno,v\_regdate);

dbms\_output.put\_line('插入成功');

exception

when value\_error then

dbms\_output.put\_line('数据类型错误');

when too\_many\_rows then

dbms\_output.put\_line('该学生已收过费了');

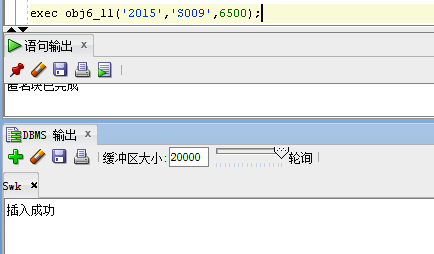
end;

调用过程：

**set serveroutput on;**

**execute obj6\_11（‘2015’,’S009’,6500）;**

结果如下：



12、编写一个过程，输出指定学年的欠费情况（含欠费人数、欠费总金额）。

 create or replace

procedure obj6\_12(year charge.shyear%type)

as

sno\_sum number:=0;

money\_sum CHARGE.TUITION%type:=0;

Cursor cur is select TUITION,BALANCE from CHARGE where year=shyear;

begin

for result\_row in cur LOOP

if result\_row.TUITION>result\_row.BALANCE

then sno\_sum:=sno\_sum+1;money\_sum:=money\_sum+result\_row.TUITION-result\_row.BALANCE;

end if;

end LOOP;

dbms\_output.put\_line('sno\_sum:'||sno\_sum||',money\_sum:'||money\_sum);

exception

when value\_error then

dbms\_output.put\_line('数据类型错误');

when too\_many\_rows then

dbms\_output.put\_line('返回多行数据');

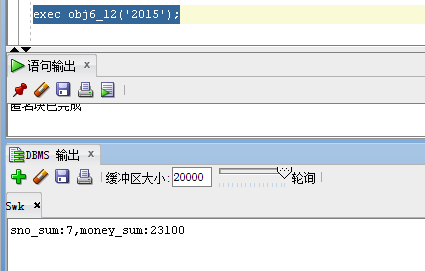
end;

调用过程：

**set serveroutput on;**

**execute obj6\_12(‘2015’);**

结果如下：



13、输出如下九九乘法表。(存储过程)

create or replace

procedure obj6\_13

as

i int;

j int;

begin

for i in 1..9 loop

for j in 1..i loop

dbms\_output.put(i||'\*'||j||'='||(i\*j)||' ');

end loop;

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

end loop;

exception

when value\_error then

dbms\_output.put\_line('数据类型错误！');

when too\_many\_rows then

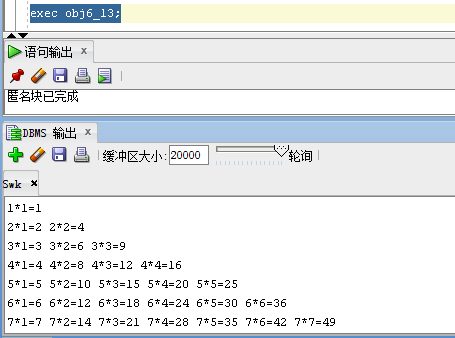
dbms\_output.put\_line('返回多行匹配的数据！');

end;

调用过程：

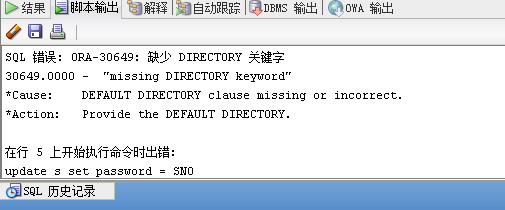
exec obj6\_13;

结果如下：



**三、实验问题分析**

**实验问题1：关于调用过程失败的问题的出现及解决方案**

****

解决办法：忘记添加关键字，添加关键字

2.在第8项编译后出现警告。

说明: C:\Users\xuesheng\Downloads\cuowu6_8.PNG

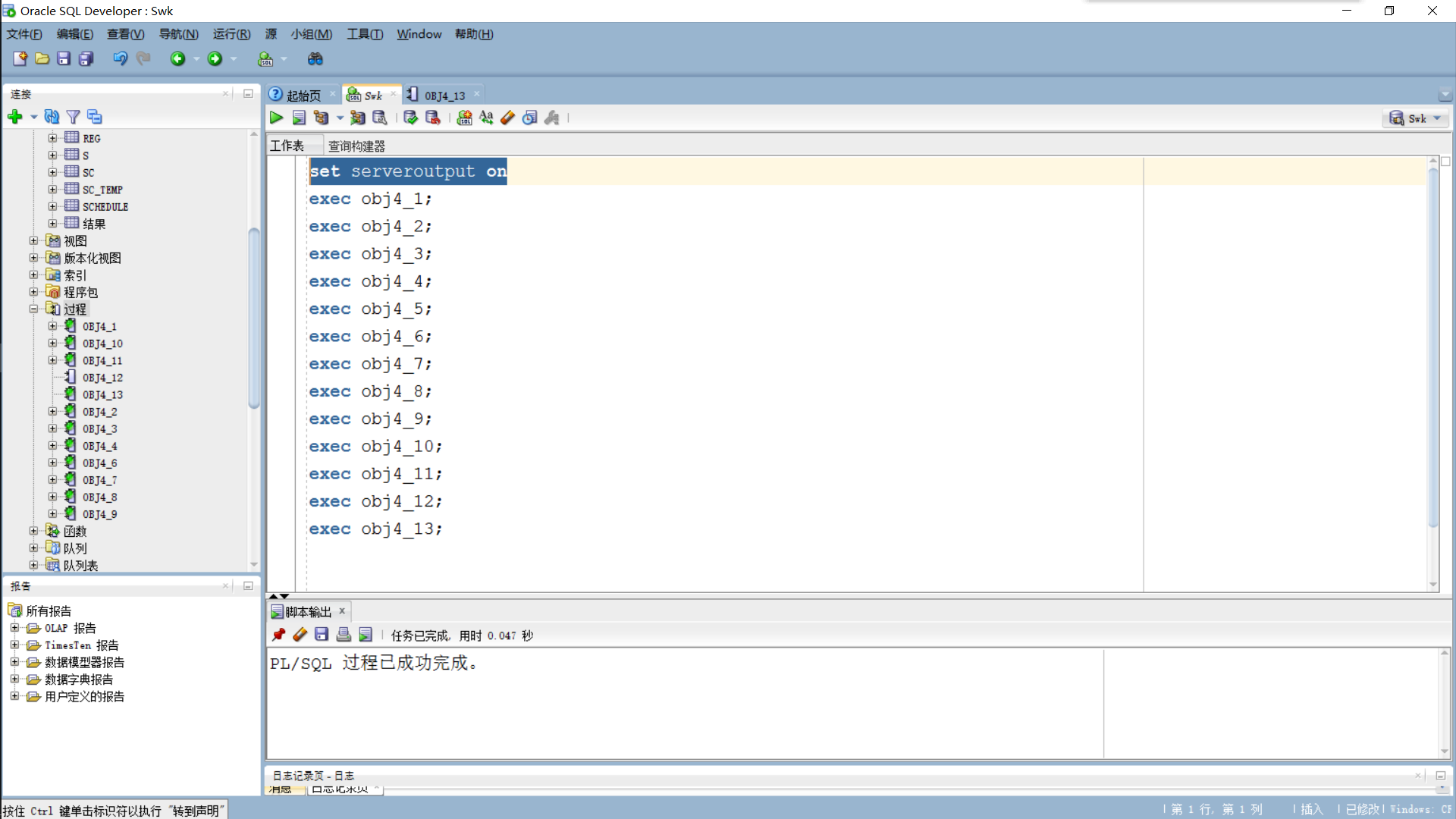
解决方法：代码中的:=写成了=，修改后编译成功。

**实验问题2：关于PL\SQL语句不输出问题的出现及解决方案**

未开启输出服务，执行开启输出服务的语句

Set serveroutput on；

开启后正常输出。



**四、小结**

本次实验通过对于oracle异常的使用，对于前面实验中的内容，增加异常处理，巩固oracle相关知识。

在实验中也遇到了很多的问题，一开始对于异常的使用还不是很熟练，在多次练习之后，才明白了异常的使用方法。

通过使用异常，可以使我们更加清楚的知道问题出在什么地方，有利于我们查找和解决问题。