简单：

编写一个函数与Oracle数据库中的concat函数的功能相同。

create or replace function myconcate(str in varchar2, str2 in varchar2) return varchar2 is

str\_temp varchar2(32767);

begin

str\_temp := str||str2;

return str\_temp;

end myconcate;

1. 编写一个函数，函数没有参数，该函数返回订单表orders中所有订单总额大于1000元的用户的个数。

答案：

create or replace function MFun1 return number is

count\_temp number;

begin

select count(\*) into count\_temp from (select user\_id from orders group by user\_id having sum(price)>1000);

return count\_temp;

end MFun1;

1. 根据上一个题，修改这个函数MFun2，这个函数含有一个入参表示钱数，根据这个参数的值来返回订单总额大于这个数的所有用户的个数。

create or replace function mfun2(price\_temp in number) return number is

count\_temp number;

begin

select count(\*) into count\_temp from (select user\_id from orders group by user\_id having sum(price)>price\_temp);

return count\_temp;

end mfun2;

1. 根据上一个题，修改这个函数MFun3，这个函数含有两个参数，一个代表的是钱数，另外一个的标识位，如果为1的话，那么不仅仅要返回订单总额大于这个值的所有用户的个数，还需要将符合条件的订单的总额进行四舍五入处理。如果第二个入参的值为0的话，那么不需要修改这个值。

create or replace function fun1(price\_temp in number,flag\_temp in char) return number is

count\_temp number;

id\_temp orders.id%type;

pri\_temp orders.price%type;

userId\_temp orders.user\_id%type;

pragma autonomous\_transaction;*--\*\*\*\*使用自治事务*

*--存储所有消费金额大于price\_temp的账单*

cursor cur is(

select \* from orders where user\_id in

(select user\_id from orders group by user\_id having sum(price)>price\_temp)

*--按照用户分组，查询消费金额大于price\_temp的用户id*

);

begin

select count(1) into count\_temp from

(select user\_id from orders group by user\_id having sum(price)>price\_temp);

if(flag\_temp='1') then

open cur;

loop

fetch cur into id\_temp,pri\_temp,userId\_temp;

exit when cur%notfound;

update orders set price=round(pri\_temp) where id=id\_temp;

end loop;

commit;

end if;

return count\_temp;

end fun1;

1. 编写一个转帐的存储过程(需要调试存储过程的系统权限------debug any procedure

debug connect session)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*方法一\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

create or replace procedure zhuanzhang(id1 in number,id2 in number,money number,out\_ret out varchar2) is

id1\_money accounts.price%type;

id2\_money accounts.price%type;

begin

select price into id1\_money from accounts where id=id1;

select price into id2\_money from accounts where id=id2;

update accounts set price=(id1\_money-money) where id=id1;

update accounts set price=(id2\_money+money) where id=id2;

commit;

out\_ret:='400';

end zhanzhang;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*方法二\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

create or replace procedure zhuanzhang

(id1 in number,id2 in number,money\_temp in number,out\_result out varchar2) is

id1\_money accounts.money%type;

id2\_money accounts.money%type;

id1\_num number;

id2\_num number;

begin

*--查询这个人是否存在*

select count(id) into id1\_num from accounts where id=id1;

select count(id) into id2\_num from accounts where id=id2;

if(id1\_num <=0 or id2\_num <= 0 ) then

out\_result:='人不存在';

else

*--余额是否足够*

select money into id1\_money from accounts where id=id1;

select money into id2\_money from accounts where id=id2;

if(id1\_money>money\_temp) then

update accounts set money=(id1\_money-money\_temp) where id=id1;

update accounts set money=(id2\_money+money\_temp) where id=id2;

commit;

out\_result:='成功';

else

out\_result:='余额不足';

end if;

end if;

*--处理异常*

exception when others then

out\_result:='系统异常，操作失败';

rollback;

end zhuanzhang;

1. 编写一个存储过程，将数据插入到mytable中，然后在将这些数据写入到mytable\_temp表。

a是

create or replace procedure myproc(a in number,c in number,b out varchar2) is

num number;

id\_temp mytable.id%type;

name\_temp mytable.name%type;

age\_temp mytable.age%type;

cursor mycur is select mm.id,mm.name,mm.age from mytable mm where id>=c;

begin

for I in 1..a loop

select max(id) into num from mytable;

insert into mytable values(num+1,'zhangsan',20);

end loop;

open mycur;

loop

fetch mycur into id\_temp,name\_temp,age\_temp;

exit when mycur%notfound;

insert into mytable\_temp(id,uname,age) values(id\_temp,name\_temp,age\_temp);

end loop;

commit;

b:='400';

EXCEPTION

WHEN OTHERS THEN

rollback;

b:='500';

end myproc;

1. 用存储过程实现将订单总额大于某个值的所有订单的价钱进行四舍五入。

create or replace procedure myfun33(price\_temp in number,

flag\_temp in char,

order\_num out number,

out\_ret out varchar2) is

count\_temp number;

id\_temp orders.id%type;

pri\_temp orders.price%type;

userId\_temp orders.user\_id%type;

cursor myc is

select \*

from orders

where user\_id in (select user\_id

from orders

group by user\_id

having sum(price) > price\_temp);

begin

select count(\*)

into count\_temp

from (select user\_id

from orders

group by user\_id

having sum(price) > price\_temp);

if (flag\_temp = '1') then

open myc;

loop

fetch myc

into id\_temp, pri\_temp, userId\_temp;

exit when myc%notfound;

update orders set price = round(pri\_temp) where id = id\_temp;

end loop;

commit;

end if;

order\_num := count\_temp;

out\_ret := '400';

EXCEPTION

WHEN OTHERS THEN

OUT\_RET := '500';

rollback;

end myfun33;