存储过程：

create or replace procedure STUDY.proc\_1

is

cursor c is

select \* from study.t\_user for update;

begin

for v\_user in c loop

if(v\_user.\_PARENTID = 1) then

update study.T\_USER set addressid = 1 where current of c;

elsif(v\_user.\_PARENTID = 2) THEN

update study.T\_USER set addressid = 2 where current of c;

else

update study.T\_USER set addressid = 4 where current of c;

end if;

end loop;

commit;

end;

exec study.proc\_1;--调用存储过错

--带参数的存储过程

create or replace procedure study.proc\_2(v\_a in number, v\_b in number , v\_out out number, v\_all in out number) is

begin

if(v\_a > v\_b) THEN

v\_out := v\_a;

else

v\_out := v\_b;

end if;

v\_all := v\_all + 1;

end;

declare

v\_a number := 11;

v\_b number := 12;

v\_c number;

v\_d number := 123;

begin

dbms\_output.put\_line('111123');

study.proc\_2(v\_a,v\_b,v\_c,v\_d);

dbms\_output.put\_line(v\_c);

dbms\_output.put\_line(v\_d);

end;

数组类型：

declare

type table\_userid is table of study.T\_USER.id%type index by binary\_integer;--table类型声明，相当于数组

tb\_1 table\_userid;

userid study.T\_USER.id%type;--普通类型

rcd\_1 study.t\_user%rowtype;--record类型，相当于map

begin

tb\_1(0) := 100;

userid := 1;

select \* into rcd\_1 from study.T\_USER where id = tb\_1(0);

dbms\_output.put\_line(rcd\_1.username);

dbms\_output.put\_line(sql%rowcount || '条记录被修改');

end;

循环：

declare

i binary\_integer := 1;

begin

while i <= 10 loop

dbms\_output.put\_line(i);

i := i + 1;

end loop;

end;

begin

for i in 1..10 loop

dbms\_output.put\_line(i);

end loop;

end;

--类似于do while循环

declare

i binary\_integer := 1;

begin

loop

dbms\_output.put\_line(i);

i := i + 1;

exit when(i >10);

end loop;

end;

游标：

declare

cursor c is select \* from STUDY.T\_USER;

v\_user c%rowtype;

begin

open c;--打开游标

loop

fetch c into v\_user;

dbms\_output.put\_line(v\_user.id);

--dbms\_output.put\_line(c%rowcount);

exit when(c%notfound);--最近的fetch语句取不到数据

end loop;

close c;

end;

declare

cursor c is select \* from STUDY.T\_USER;

v\_user c%rowtype;

begin

open c;

fetch c into v\_user;

while(c%found) LOOP

dbms\_output.put\_line(v\_user.id);

dbms\_output.put\_line(c%rowcount);

fetch c into v\_user;

end loop;

close c;

end;

declare

cursor c is select \* from STUDY.T\_USER;

begin

for i in c loop

dbms\_output.put\_line(i.username);

end loop;

end;

--带参数的游标

declare

cursor c(uname STUDY.T\_USER.username%type, pword study.T\_USER.password%type) is

select \* from STUDY.T\_USER where USERNAME = uname and password = pword;

v\_user STUDY.T\_USER%rowtype;

begin

for v\_user in c('zhangsan','123') loop

dbms\_output.put\_line(v\_user.id);

end loop;

end;

--利用游标更新数据

declare

cursor c is

select \* from study.t\_user for update;

v\_user study.T\_USER%rowtype;

begin

for v\_user in c loop

if(v\_user.password = '123') then

update study.T\_USER set username = 'zhangsan' where current of c ;

elsif(v\_user.username is null) then

delete study.T\_USER where current of c;

else

update study.T\_USER set addressid = 1 where current of c;

end if;

end loop;

exception

when others then

rollback;

dbms\_output.put\_line('there is a exception occured!');

dbms\_output.put\_line(sqlerrm);

dbms\_output.put\_line(sqlcode);

end;

方法function：

create or replace function study.qiumi(v\_num number, v\_mi number)

return number is

declare

v\_temp number := v\_num;

begin

if(v\_mi = 0) THEN

return 1;

elsif(v\_mi = 1) THEN

return v\_num;

elsif(v\_mi > 0) then

for i in 2..v\_mi loop

v\_num := v\_num \* v\_temp;

end loop;

return v\_num;

else

for i in 2..(0 - v\_mi) loop

v\_num := v\_num \* v\_temp;

end loop;

return 1 / v\_num;

end if;

end;

select study.QIUMI(2,-2) from dual;

批量插入：

INSERT INTO t\_sc(SID,cid,g)

SELECT 2011001,001,90 FROM dual UNION ALL

SELECT 2011002,001,91 FROM dual UNION ALL

SELECT 2011003,001,92 FROM dual UNION ALL

SELECT 2011004,002,93 FROM dual UNION ALL

SELECT 2011005,002,94 FROM dual UNION ALL

SELECT 2011006,002,95 FROM dual UNION ALL

SELECT 2011007,003,96 FROM dual UNION ALL

SELECT 2011008,003,97 FROM dual UNION ALL

SELECT 2011009,004,98 FROM dual UNION ALL

SELECT 2011010,004,99 FROM dual

序列的：

insert into td\_s\_loadinfo (load\_id, load\_app\_id, load\_cache\_id, load\_state, add\_time)

select SEQ\_LOAD\_INFO\_ID.Nextval, a.\* from (

select 10 col1,10 col2,'0' col3, sysdate from dual union all

select 11 col1,11 col2,'0' col3, sysdate from dual

) a

触发器：

-- 建表

create table temp\_age(

age\_id number(5),

age number(3)

);

-- 建触发器

create or replace trigger t\_temp\_age\_check

before insert on temp\_age

for each row

begin

if :new.age < 18

then

raise\_application\_error(-20001,'age must at least 18 years old');

end if;

end;

-- 客户端程序

declare

no\_baby\_allowed exception;

pragma exception\_init(no\_baby\_allowed,-20001);

begin

insert into temp\_age(age\_id,age) values(1,20);

insert into temp\_age(age\_id,age) values(2,17);

insert into temp\_age(age\_id,age) values(3,18);

exception

when no\_baby\_allowed

then

dbms\_output.put\_line(sqlerrm);

end;

不走索引的一些情况

# like 模糊查询 前模糊或者 全模糊不走索引

explain select \* from users u where u.name like '%mysql测试'

# or 条件中只要有一个字段没有索引，改语句就不走索引

explain select \* from users u where u.name = 'mysql测试' or u.password ='JspStudy'

# 使用 union all 代替 or 这样的话有索引例的就会走索引

explain

select \* from users u where u.name = 'mysql测试'

union all

select \* from users u where u.password = 'JspStudy'

# in 走索引

explain select \* from users u where u.name in ('mysql测试','JspStudy')

# not in 不走索引

explain select \* from users u where u.name not in ('mysql测试','JspStudy')

# is null 走索引

explain select \* from users u where u.name is null

# is not null 不走索引

explain select \* from users u where u.name is not null

# !=、<> 不走索引

explain select \* from users u where u.name <> 'mysql测试'

# 隐式转换-不走索引（name 字段为 string类型，这里123为数值类型，进行了类型转换，所以不走索引,改为 '123' 则走索引）

explain select \* from users u where u.name = 123

# 函数运算-不走索引

explain select \* from users u where date\_format(upTime,'%Y-%m-%d') = '2019-07-01'

————————————————

版权声明：本文为CSDN博主「Muscleheng」的原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/Muscleheng/article/details/94393024