1) ARMSTRON OR NOT

create or replace function arm(x in number)

return number as

z number;

r number;

s number:=0;

i number;

n number;

begin

n:=x;

i:=length(n);

while i>0

loop

r:=mod(n,10);

s:=s+power(r,i);

n:=trunc(n/10);

end loop;

z:=s;

return z;

end;

/

declare

n number;

c number;

begin

n:=&n;

c:=arm(n);

if c=n

then

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

else

dbms\_output.put\_line('not arm');

end if;

end;

/

2) FACTORIAL

declare

fact number:=1;

n number;

begin

n:=&n;

for i in 1 ..n

loop

fact:=fact\*i;

end loop;

dbms\_output.put\_line('factorial of '|| n ||'is '||fact);

end;

/

3) PRIME OR NOT

declare

n number;

c number:=1;

begin

n:=&n;-

if n=1

then

dbms\_output.put\_line('not prime');

end if;

for i in 1 ..n/2

loop

if mod(n,i)=0

then

c:=c+1;

end if;

end loop;

if c=2

then

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

else

dbms\_output.put\_line("not prime");

end if ;

end;

/

4) TABLE ITEM AND VALUE RETRIEVAL

create table i\_tem(itemid int,itemname varchar(20),price int);

create or replace function getdata(id in number)

return number as

p i\_tem.itemname%type;

q i\_tem.price%type;

begin

select itemname,price into p,q from i\_tem where itemid=id;

dbms\_output.put\_line('item name item id');

dbms\_output.put\_line(p||' '||q);

return 0;

end;

declare

id number;

begin

id:=&id;

dbms\_output.put\_line(getdata(id));

end;

5)POW

create or replace function pow(a numner,b number)

return number as

r number;

begin

r:=power(a,b);

return r;

end;

declare

x number;

y number;

begin

x:=&x;

y:=&y;

dbms\_output.put\_line(pow(x,y));

end;

CURSOR

1)INTERSET FIXED DEPO

create table banky(accn int,years int,amount int,interest int);

declare

cursor ban is select \* from banky;

begin

for i in ban

loop

if i.amount<=10000

then

update banky set interest=i.amount\*0.1 where accn=i.accn;

else if i.amount>10000 and i.amount<=20000

then

update banky set interest=i.amount\*0.2 where accn=i.accn;

else

update banky set interest=i.amount\*0.3 where accn=i.accn;

end if;

end if;

end loop;

end;

2) BILL

create table billy(billnum int,name varchar(20),units int,charge float);

declare

cursor bill is select \* from billy;

begin

for i in bill

loop

if i.units<=2

then

update billy set charge=100+i.units\*0.2 where billnum=i.billnum;

else if i.units>2 and i.units<=5

then

update billy set charge=100+i.units\*0.5 where billnum=i.billnum;

else

update billy set charge=100+i.units\*0.9 where billnum=i.billnum;

end if;

end if;

end loop;

end;

3)IMPLICIT CURSOR

declare

a int;

begin

if sql%notfound

then

dbms\_output.put\_line('not updated');

else

update i\_tem set price=100;

a:=sql%rowcount;

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

end if;

end;

/

4) HIGHEST PAID EMPLOYEE

create table wors(empid int,cname varchar(25),sal int);

declare

i int :=0;

cursor wo is select \* from wors order by sal desc;

r wo%rowtype;

begin

dbms\_output.put\_line('highest paid employe are');

open wo;

loop

exit when i=3;

fetch wo into r;

dbms\_output.put\_line(r.empid||' '||r.cname||' '||r.sal);

i:=i+1;

end loop;

close wo;

end;

PROCEDURE

1)BANKING

create table banker(acn int,name varchar(30),balance int);

create or replace procedure withdrow(acc int, amount int) is

begin

update banker set balance=balance-amount where acn=acc;

end;

create or replace procedure deposit(acc int, amount int) is

begin

update banker set balance=balance+amount where acn=acc;

end;

create or replace procedure display is cursor ba is select \* from banker;

r ba%rowtype;

begin

open ba;

loop

fetch ba into r;

exit when ba%notfound;

dbms\_output.put\_line(r.acn||' '||r.name||' '||r.balance);

end loop;

close ba;

end;

declare

acc int;

amount int;

choice number;

begin

acc:=&acc;

amount:=&amount;

dbms\_output.put\_line('1.deposit');

dbms\_output.put\_line('2.withdrow');

dbms\_output.put\_line('3.display');

choice:=&choice;

if choice=1

then

deposit(acc,amount);

else if choice=2

then

withdrow(acc,amount);

else

dbms\_output.put\_line('3.display');

display();

end if;

end if;

end;

2) EXCEPTION

create table procedure(proid int,pronamne varchar(20),price int);

create or replace procedure error(pid int) is invalidpid exception;

begin

update procedure set price=price+100 where proid=pid;

if sql%notfound

then

raise invalidpid;

end if;

exception

when invalidpid

then

dbms\_output.put\_line('pid not found');

end;

create or replace procedure display is

cursor c is select \* from procedure;

r c%rowtype;

begin

open c;

loop

fetch c into r;

exit when c%notfound;

dbms\_output.put\_line(r.proid||' '||r.pronamne||' '||r.price);

end loop;

close c;

end;

/

declare

pid int;

begin

pid:=&pid;

error(pid);

display();

end;

/

TRIGGER

1) RESTRICT INSERT UPDATE DELETE

create table sample(sid int,sname varchar(20));

create or replace trigger t0 before insert or update or delete on sample for each row

begin

raise\_application\_error(-2000,'you are not permitted to do insert/delete/update');

end;

2) MINSALARY

create table profinsert(pnum int,pname varchar(20),sal int);

create or replace trigger minsal before insert on prof for each row

begin

if (:new.sal<6000)

then

raise\_application\_error(-2000,'violation of min salary constrain');

end if;

end;

3)BACKLOG

create table back(pnum int, pname varchar(20),sal int);

create or replace trigger back after delete on profo for each row

begin

insert into back values(:old.pnum,:old.pname,:old.sal);

end;

4)GOLDPRICE HISTORY

create table gold(cdate date,price int);

create table history(cdate date,price int);

create or replace trigger history after update on gold for each row

begin

insert into history values(:old.cdate,:old.price);

end;