26)

A

create or replace package mypack1 is

procedure product\_table(a in number);

function product(a in number, b in number) return number;

end;

B

create or replace package body mypack1 as

procedure product\_table(a in number) as

i number;

begin

i:=1;

while(i<=10)

loop

dbms\_output.put\_line(a||'x'||i||'='||a\*i);

i:=i+1;

end loop;

end product\_table;

function product(a in number, b in number) return number as

c number;

begin

c:= a\*b;

return c;

end product;

end;

C

declare

x number :=2;

y number :=3;

z number;

begin

mypack1.product\_table(x);

z:= mypack1.product(x,y);

dbms\_output.put\_line('product of ' ||x || ' and '||y||' is '||z);

end;

27

A

create or replace package pack1 is

procedure p1(p in programmer.pname%type);

function f1(t in software.title%type) return software.pname%type;

end;

B

create or replace package body pack1 as

procedure p1( p in programmer.pname%type) as

s programmer.salary%type;

begin

select salary into s from programmer where pname = p;

dbms\_output.put\_line('Salary is '|| s);

end p1;

function f1(t in software.title%type) return software.pname%type as

x software.pname%type;

begin

select pname into x from software where title=t;

return x;

end f1;

end;

c

declare

a programmer.pname%type := 'Anand';

b software.title%type := 'Read Me';

r software.pname%type;

begin

pack1.p1(a);

r:=pack1.f1(b);

dbms\_output.put\_line('Name is'||r);

end;

28

A

create or replace package pack2 is

procedure p2(x in software.DEV\_D%type);

function f2(y in studies.PNAME%type)return studies.SPLACE%type;

end;

b

create or replace package body pack2 is

procedure p2(x in software.DEV\_D%type) as

cursor s is select\*from software where DEV\_D=x;

t s%rowtype;

begin

open s;

loop

fetch s into t;

exit when s%notfound;

dbms\_output.put\_line(t.TITLE);

end loop;

close s;

end p2;

function f2(y in studies.PNAME%type)return studies.SPLACE%type as

r studies.SPLACE%type;

begin

select SPLACE into r from studies where PNAME=Y;

return r;

end f2;

end;

**c**

declare

a software.dev\_d%type:='C';

b studies.pname%type:='Anand';

r studies.splace%type;

begin

pack2.p2(a);

r:=pack2.f2(b);

dbms\_output.put\_line('name of the institution :'||r);

end;

29)

declare

a programmer.pname%type := 'Anand';

datbirth programmer.doj%type;

invalid\_age\_exception exception;

begin

select doj into datbirth from programmer where pname=a;

if(sysdate-datbirth)/365 > 18 then

dbms\_output.put\_line(a||' is eligible to vote ');

else

RAISE invalid\_age\_exception;

end if;

EXCEPTION

when invalid\_age\_exception then

dbms\_output.put\_line('sorry '||a||' is not eligible to vote ');

end;

30

A

create table student1(rollno number(3) primary key,

sname varchar2(15),

marks1 number(3),

marks2 number(3),

total number(4));

b

create or replace trigger auto\_cal before insert on student1 for each row

declare

begin

:new.total := :new.marks1 + :new.marks2;

end;

c

select \* from student1;

31

A

create table prog as select pname, salary from programmer;

b

create table update\_prog

(pname varchar2(20),

old\_salary number(7,2),

new\_salary number(7,2),

dt date,

time varchar2(10));

c

create or replace trigger update\_status after update on prog for each row

begin

insert into update\_prog values(:old.pname, :old.salary, :new.salary, sysdate, substr(current\_timestamp,11,8));

end;

d

select \* from update\_prog;

32

A

create table student

(roll number(3) primary key,

sname varchar2(15),

age number(3));

b

create or replace trigger age before insert on student

declare

begin

update student set age=age+1;

end;

c

select \* from student;