

PL-SQL

Programming Language-structure query language

declare ---It is used to declare the variables

begin---In this we write the logic

end----This is the end of the program

example-1

declare

a number:=10;

b number:=20;

c number;

begin

c:=a+b;

dbms_output.put_line('The result is ' || c);

end;

Taking input from user

a number;

b number;

c number;

begin

a:=a;

b:=b;

c:=a+b;

dbms_output.put_line('The result is ' || c);

end;

greatest of 2 no

declare

a number;

b number;

begin

a:=a;

b:=b;

if(a>b)then

dbms_output.put_line('The greatest is ' || a);

else

dbms_output.put_line('The greatest is ' || b);

end if;

end;

The number is even or odd

declare

a number;

begin

a:=a;

if(a mod 2=0)then

dbms_output.put_line('The no is even');

else

dbms_output.put_line('The no is odd');

end if;

end;

Greatest of three no

```
declare
a number:=10;
b number:=20;
c number:=30;
begin
if(a>b and a>c)then
dbms_output.put_line('The greatest is :'| |a);
elsif(b>a and b>c)then
dbms_output.put_line('The greatest is :'| |b);
else
dbms_output.put_line('The greatest is :'| |c);
end if;
end;
```

```
declare
a number;
b number;
c number;
begin
a:=a;
b:=b;
c:=c;
if(a>b and a>c)then
dbms_output.put_line('The greatest is :'| |a);
elsif(b>a and b>c)then
dbms_output.put_line('The greatest is :'| |b);
```

```
else
dbms_output.put_line('The greatest is :'| |c);
end if;
end;
```

case-when

```
Declare
a number;
b number;
opt varchar2(20);
begin
a:=a;
b:=b;
opt:=opt;
case
when opt='add'
then dbms_output.put_line('The sum is :'| |(a+b));
when opt='sub'
then dbms_output.put_line('The sub is :'| |(a-b));
when opt='mul'
then dbms_output.put_line('The mul is :'| |(a*b));
when opt='div'
then dbms_output.put_line('The div is :'| |(a/b));
else
dbms_output.put_line(' Wrong Input: ');
end case;
end;
```

wap to enter a no and display it in words(1-5 no)

1-one

Declare

a number;

begin

a:=a;

case

when a=1

then dbms_output.put_line('One');

when a=2

then dbms_output.put_line('Two');

when a=3

then dbms_output.put_line('Three');

when a=4

then dbms_output.put_line('Four');

when a=5

then dbms_output.put_line('Five');

else

dbms_output.put_line(' Wrong Input: ');

end case;

end;

Declare

rollno number;

name varchar2(30);

address varchar2(30);

marks number;

begin

```
rollno:=:rollno;
name:=:name;
address:=:address;
marks:=:marks;
dbms_output.put_line('The rollno is :'| |rollno);
dbms_output.put_line('The name is :'| |name);
dbms_output.put_line('The address is :'| |address);
dbms_output.put_line('The marks is :'| |marks);
case
when marks >=70
then dbms_output.put_line('First Division');
when marks >=60
then dbms_output.put_line('Second Division');
when marks >=50
then dbms_output.put_line('Third Division');
else
dbms_output.put_line(' Fail: ');
end case;
end;
```

loops

```
declare
i number;
begin
for i in 1..10 loop
dbms_output.put_line(i);
end loop;
end;
```

```
-----  
  
declare  
  
i number;  
  
begin  
  
for i in reverse 1..10 loop  
dbms_output.put_line(i);  
end loop;  
end;
```

```
-----  
  
create table circle223(radius number,area number);  
  
-----
```

```
declare  
  
pi constant number:=3.14;  
  
radius number;  
  
area number;  
  
begin  
  
for radius in 5..10 loop  
area:=pi*power(radius,2);  
insert into circle223 values(radius,area);  
end loop;  
end;
```

```
-----  
  
select * from circle223;  
  
-----
```

example-1

```
-----  
  
begin  
  
update emp223 set salary=55000 where empono=101;
```

```
update emp223 set salary=54000 where empono=102;
update emp223 set salary=53000 where empono=103;
insert into emp223 values(113,'tom',32000,'dev');
insert into emp223 values(114,'jerry',22000,'dev');
end;
```

Declare

```
name varchar2(30) not null:='Kunal';
surname name%type :='Kumar';
address emp223.dept%type:='Bangalore';
begin
dbms_output.put_line(' name is :'|name);
dbms_output.put_line(' surname is :'|surname);
dbms_output.put_line(' address is :'|address);
end;
```

function :- There are two types of functions

- 1)user-defined function
- 2)pre-defined function:- String function,math function.

Example-1

```
create or replace function addno(i number,j number)return number is
begin
return i+j;
end;
```

declare


```

a number;

begin

a:=addno(10,20);

dbms_output.put_line('The sum is' || a);

end;

```

```

-----

declare

i number;

j number;

c number;

begin

i:=i;

j:=j;

c:=addno(i,j);

dbms_output.put_line('The sum is: ' || c);

end;

```

Example-2

```

-----

create or replace function addno(i number,j number)return number is

begin

return i+j;

end;

create or replace function subno(i number,j number)return number is

begin

return i-j;

end;

create or replace function mulno(i number,j number)return number is

begin

```

```

return i*j;

end;

create or replace function divno(i number,j number)return number is
begin
return i/j;

end;

```

```

declare
i number;
j number;
c number;
d number;
e number;
f number;
begin
i:=i;
j:=j;
c:=addno(i,j);
d:=subno(i,j);
e:=mulno(i,j);
f:=divno(i,j);

dbms_output.put_line('The sum is: ' || c);
dbms_output.put_line('The sub is: ' || d);
dbms_output.put_line('The mul is: ' || e);
dbms_output.put_line('The div is: ' || f);

end;

```

Example-3

create or replace function design(salary number) return varchar2 is

begin

if(salary >=50000)then

return 'CEO';

elsif(salary >=40000)then

return 'Manager';

elsif(salary >=30000)then

return 'Officer';

elsif(salary >=20000)then

return 'Worker';

else

return 'Helper';

end if;

end;

select design(45000) from dual;

-----Assignment-3-----

1)wap to enter a radius and find the area of the circle using function.

principal -1000

time-2 years

rate-10%

$si = (p * t * r) / 100;$

2)wap to enter principle amount ,time,rate and find simple interest using function.

$si = (p * t * r) / 100;$

3) wap to enter a number and find even or odd using function.

4) Create a bank table(accno,name,balance) and insert the data into the table using PL-SQL.

5) Create a employee table (empno,name,salary,designation) and delete designation-manager from the table.

Example-1

```
select * from emp223
```

```
declare
```

```
a emp223.empono%type;
```

```
b emp223.name%type;
```

```
c emp223.salary%type;
```

```
d emp223.dept%type;
```

```
begin
```

```
a:=a;
```

```
b:=b;
```

```
c:=c;
```

```
d:=d;
```

```
insert into emp223 values(a,b,c,d);
```

```
end;
```

```
declare
```

```
a emp223.empono%type;
```

```
b emp223.name%type;
```

```
c emp223.salary%type;
```

```
d emp223.dept%type;
```

```
begin
```

```
a:=a;
b:=b;
c:=c;
d:=d;
insert into emp223 values(a,b,c,d);
end;
```

Procedure

It is a pl-sql block which execute when called but donot return any value.

example-1:

```
create or replace procedure add_no(i number default 0,j number default 0,k number default 0) is
begin
dbms_output.put_line('The result is :'| |(i+j+k));
end;
```

```
begin
add_no(3,4);
end;
```

example-2

```
create or replace procedure deldev is
begin
delete from emp223 where dept='dev';
commit;
end;
```

```
-----  
begin  
deldev;  
end;
```

```
-----  
select * from emp223;  
-----
```

example-3

```
-----  
create or replace procedure insertdata(a number,b varchar2,c number,d varchar2)  
is  
begin  
insert into emp223 values(a,b,c,d);  
end;
```

```
-----  
begin  
insertdata(123,'sumeet',30000,'dev');  
insertdata(124,'rahul',31000,'dev');  
insertdata(125,'ajay',32000,'dev');  
insertdata(126,'anil',33000,'dev');  
end;
```

```
-----  
select * from emp223;  
-----
```

Trigger

```
-----  
Trigger :-The query will be fired automatically.
```

salary posting,electricity bill ,telephone bill generation.on 1st day of the month or last day of the month.

Triggers are stored programs, which are automatically executed or fired when some event occurs.

Triggers are written to be executed in response to any of the following events.

A database manipulation (DML) statement (DELETE, INSERT, or UPDATE).

A database definition (DDL) statement (CREATE, ALTER, or DROP).

Trigger Timing: BEFORE, AFTER

Trigger Event: INSERT, UPDATE, DELETE

Trigger Type: Row

Example:---

```
CREATE OR REPLACE TRIGGER t1
BEFORE INSERT ON emphasiseemp
BEGIN
IF(TO_CHAR(SYSDATE,'DY')IN('SAT','SUN')) OR
(TO_CHAR(SYSDATE,'HH24:MI') NOT BETWEEN '08' AND '15')
THEN RAISE_APPLICATION_ERROR(-20500,'You may insert into employees table only during business
hours');
END IF;
END;
```

```
insert into IBMEMP values(1009,'Vikram','ProjLead');
```

cursor

cursors:- cursors are used whenever one want to retrieve more than one record from the database.cursor is the amount of memory reserved to accomodate more than one row of the table.

cursor are divide into two types:-

1)implicit cursor

2)explicit cursor

implicit cursor:- oracle implicitly creates and open a cursor for every sql data manipulation statement that is not part of an explicit declared cursor.programmer has no control over this.

explicit cursor:-pl/sql programmer use explicit cursor only.the result of this cursor select statement can consist of more than one rows depending on query.

activeset:- it is a group of rows which are hold in the memory area of the cursor.

steps:-

1)declare (memory allocation)

2)open (activeset is created)

3)manipulation(loop/rows)

4)close(release the memory)

cursor attributes:-

1)%isopen :-when there is no activeset it will retrieve false and if is there activeset is true.return type is boolean.

2)%found:-to know whether the cursor pointer points to a valid row inside the cursor.we cannot fetch when no more rows available.return type is boolean.

3)%notfound:-(reverse)

4)%rowcount:-it will retrieve the position of the pointer within the activeset .

```
-----  
declare  
cursor c is select name from emp223 ;  
x varchar2(15);  
begin  
open c;  
loop  
fetch c into x;  
exit when c %notfound;  
dbms_output.put_line(c %rowcount||' '||x);  
end loop;  
end;
```

example-1

```
-----  
declare  
cursor c is select name from emp223;  
n varchar2(30);  
begin  
open c;  
fetch c into n;  
dbms_output.put_line(n);  
end;
```

example-2

```
-----  
declare  
cursor c is select name from emp223;
```

```
n varchar2(30);  
  
begin  
  
open c;  
  
loop  
fetch c into n;  
exit when c %notfound;  
dbms_output.put_line(n);  
end loop;  
end;
```

example-3

```
declare  
  
cursor c is select salary, name from emp223;  
  
n number;  
x varchar2(20);  
  
begin  
  
open c;  
  
loop  
fetch c into n,x;  
exit when c %notfound;  
dbms_output.put_line(x || ' get salary= ' || n);  
end loop;  
end;
```

example-1

```
create or replace function calfun(nempno emp223.empono%type)  
return number as
```

```
nsalary emp223.salary%type;
ntotal emp223.salary%type;

begin
select salary,salary*12 into nsalary, ntotal from emp223 where empono=nempno;
return ntotal;
end;
```

```
-----

declare
nresult number;

begin
nresult:=calfun(145);
dbms_output.put_line('the nresult is ' || nresult);
end;
```

```
-----

select * from emp223;
```

```
-----

select calfun(145) from dual;
```

```
-----
```