DBMS Practical 4

Dharmit Shah SYIT 330

Practical 4: Creating Procedures, Functions and Packages

1. Create and replace an empty procedure and call it

```
SQL> create or replace procedure proc as
  2  begin
  3  null;
  4  end;
  5  /

Procedure created.

SQL> execute proc;

PL/SQL procedure successfully completed.
```

2. Create a procedure and a function to display the square of a number.

Run SQL Command Line

```
SQL> create or replace procedure p1(a in int,b out int)
  2  as
  3  begin
  4  b:=a*a;
  5  end;
  6  /

Procedure created.

SQL> declare
  2  a int := &a;
  3  b int;
  4  begin
  5  p1(a,b);
  6  dbms_output.put_line(b);
  7  end;
  8  /
Enter value for a: 10
old   2: a int := &a;
new   2: a int := 10;
100

PL/SQL procedure successfully completed.
```

```
SQL> create or replace function p2(num1 in int)
  2  return int
  3  as
  4  s int;
  5  begin
  6  s := num1 * num1;
  7  return s;
  8  end;
  9  /

Function created.

SQL> select p2(6) from dual;

    P2(6)
----------
        36
```

3. Create a procedure and a function to swap two numbers.

Run SQL Command Line

```
SQL> create or replace procedure p3
  2  as
  3  a int :=10;
  4  b int :=20;
  5  c int;
  6  begin
  7  c:=a;
  8  a:=b;
  9  b:=c;
 10  dbms_output.put_line(a|| ' <--> '||b);
 11  end;
 12  /

Procedure created.

SQL> execute p3;
20 <--> 10

PL/SQL procedure successfully completed.
```

```
SQL> create or replace function f3(num1 int, num2 int)
  2  return int
  3  as
  4  temp int;
  5  num_1 int := num1;
  6  num_2 int := num2;
  7  swap int;
  8  begin
  9  temp := num_1;
 10  num_1 := num_2;
 11  num_2 := temp;
 12  swap := num_1||num_2;
 13  return swap;
 14  end;
 15  /

Function created.

SQL> select f3(5,10) from dual;

  F3(5,10)
----------
       105
```

4. Create a procedure and a function to display the greatest among two numbers.

Run SQL Command Line

```
SQL> create or replace procedure p4
  2  as
  3  a int:= &a;
  4  b int:= &b;
  5  begin
  6  if (a>b) then
  7  dbms_output.put_line(a||' is greater than '||b);
  8  else
  9  dbms_output.put_line(b||' is greater than '||a);
 10  end if;
 11  end;
 12  /
Enter value for a: 20
old   3: a int:= &a;
new   3: a int:= 20;
Enter value for b: 40
old   4: b int:= &b;
new   4: b int:= 40;

Procedure created.

SQL> execute p4;
40 is greater than 20

PL/SQL procedure successfully completed.
```

```
SQL> create or replace function f4(a in int,b in int)
  2  return int
  3  as
  4  begin
  5  if a>b then
  6  return a;
  7  else
  8  return b;
  9  end if;
 10  end;
 11  /

Function created.

SQL> select f4(3,9) from dual;

   F4(3,9)
----------
         9

SQL>
```

5. Create a procedure and a function to display the employee name whose employeeno is accepted by the user.

```
create procedure p5(en out varchar, eno in out int)
as
begin
select ename into en from emp_dharmit where empno=eno;
dbms_output.put_line(en);
end;
/
```

Results Explain Describe Saved SQL History

```
Procedure created.
```

```
declare
en varchar(20);
eno int := 7900;
begin
p5(en,eno);
end;
/
```

Results Explain Describe Saved SQL History

```
JAMES

Statement processed.
```

```
create or replace function p5(emp_n int)
return varchar
as
e_name varchar(20);
begin
select ename into e_name from emp_dharmit where empno = emp_n;
return e_name;
end;
/
```

Results Explain Describe Saved SQL History

```
Function created.
```

0.02 seconds

```
select p5(7900) from dual;
```

Results Explain Describe Saved SQL History

P5(7900)
JAMES

1 rows returned in 0.00 seconds CSV Export

6. Create a procedure and a function to display the sum of salary of the employees whose job is accepted by the user.

```
create procedure p6(j in varchar, s out int)
as
begin
select sum(sal) into s from emp_dharmit where job=j;
dbms_output.put_line(s);
end;
/
```

Results Explain Describe Saved SQL History

```
Procedure created.
```

0.09 seconds

```
declare j varchar(20) := 'CLERK';
s int;
begin
p6(j,s);
end;
/
```

Results Explain Describe Saved SQL Hist

```
4150

Statement processed.
```

```
create or replace function f6(j in varchar)
return int
as
sal int;
begin
select sum(SAL) into sal from emp_dharmit where job = j;
return sal;
end;
/
```

Results Explain Describe Saved SQL History

```
Function created.
```

```
select f6('MANAGER') from dual;
```

Results Explain Describe Saved SQL History

F6('MANAGER')
8275

7. Create a procedure to display today's date.

Run SQL Command Line

```
SQL> create procedure cur_dt(dt in out date)
  2  as
  3  begin
  4  dbms_output.put_line('The current date is: '||dt);
  5  end;
  6  /

Procedure created.

SQL> declare
  2  dt DATE;
  3  begin
  4  dt := SYSDATE ;
  5  cur_dt(dt);
  6  end;
  7  /
The current date is: 18-AUG-21

PL/SQL procedure successfully completed.
```

8. Create a procedure to find the factorial of a number.

Run SQL Command Line

```
SQL> create procedure fact(n in int,f in out int)
  2  as
  3  begin
  4  dbms_output.put_line(f);
  5  end;
  6  /

Procedure created.

SQL> declare
  2  f int := 1;
  3  n int := &n;
  4  begin
  5  while n>0 loop
  6  f := n * f;
  7  n := n - 1;
  8  end loop;
  9  fact(n,f);
 10  end;
 11  /
Enter value for n: 5
old   3: n int := &n;
new   3: n int := 5;
120

PL/SQL procedure successfully completed.
```

9. Create a procedure to display the length of a string.

Run SQL Command Line

```
SQL> create procedure leng(len_str in out int)
  2  as begin
  3  dbms_output.put_line(len_str);
  4  end;
  5  /

Procedure created.

SQL> declare
  2  len varchar(30) := '&len';
  3  len_str int;
  4  begin
  5  len_str := length(len);
  6  leng(len_str);
  7  end;
  8  /
Enter value for len: dharmit
old   2: len varchar(30) := '&len';
new   2: len varchar(30) := 'dharmit';
7

PL/SQL procedure successfully completed.
```

10. Create a function to print the reverse of a string.

```
SQL> create procedure reverse(a in varchar)
  2  as
  3  r varchar(20);
  4  begin
  5  select reverse(a) into r from dual;
  6  dbms_output.put_line(r);
  7  end;
  8  /

Procedure created.

SQL> execute reverse('dharmit');
timrahd

PL/SQL procedure successfully completed.
```

11. Create a package with a function and procedure to find the sum of first 10 natural numbers.

Run SQL Command Line

```
SQL> create or replace package pack_num
  2  as
  3  procedure
  4  sumno(n in int);
  5  function sum_no(n int) return int;
  6  end pack_num;
  7  /

Package created.

SQL> create or replace package body pack_num
  2  as
  3  procedure sumno(n in int)
  4  as
  5  begin
  6  dbms_output.put_line((n*(n+1))/2);
  7  end;
  8  function sum_no(n int) return int
  9  as
 10  begin
 11  return (n*(n+1))/2;
 12  end;
 13  end pack_num;
 14  /

Package body created.

SQL> execute pack_num.sumno(10);
55

PL/SQL procedure successfully completed.

SQL> select pack_num.sum_no(10) from dual;

PACK_NUM.SUM_NO(10)
-------------------
                 55

SQL>
```

12. Create a package with a function and procedure to print the prime numbers between 1 to 50.

```
SQL> create or replace package pack_prime
  2  as
  3  procedure primenumber(x in out number, flag in out number,no in out number,r in out number);
  4  function is_prime(p_n in number) return NUMBER;
  5  end pack_prime;
  6  /

Package created.
```

```
SQL> create or replace package body pack_prime
  2  as
  3  procedure primenumber(x in out number, flag in out number,no in out number,r in out number)
  4  as
  5  begin
  6  while x < 50 loop
  7  flag:=0;
  8  no:= x-1;
  9  while no > 1 loop
 10  r := mod(x,no);
 11  if r=0 then
 12  flag:=1;
 13  exit;
 14  end if;
 15  no := no-1;
 16  end loop;
 17  if flag = 0 then
 18  dbms_output.put_line(x);
 19  end if;
 20  x := x+1;
 21  end loop;
 22  end;
 23
 24  function is_prime(p_n in number) return
 25  NUMBER
 26  as
 27  l_stop number := ceil(sqrt(p_n));
 28  begin
 29  for i in 2..l_stop
 30  loop
 31  if ( mod(p_n,i) = 0 )
 32  then
 33  return 0;end if;
 34  end loop;
 35  return 1;
 36  end;
 37  end pack_prime;
 38  /

Package body created.
```

```
SQL> declare
  2   x number := 1;
  3   flag number := 0;
  4   no number;
  5   r number;
  6   begin
  7   pack_prime.primenumber(x,flag,no,r);
  8   end;
  9   /
1
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47

PL/SQL procedure successfully completed.

SQL>
```

```
SQL> select level from dual where pack_prime.is_prime(level) = 1 connect by rownum < 16;

     LEVEL
----------
         1
         3
         5
         7
        11
        13
        17
        19
        23
        29
        31

     LEVEL
----------
        37
        41
        43
        47

15 rows selected.
```