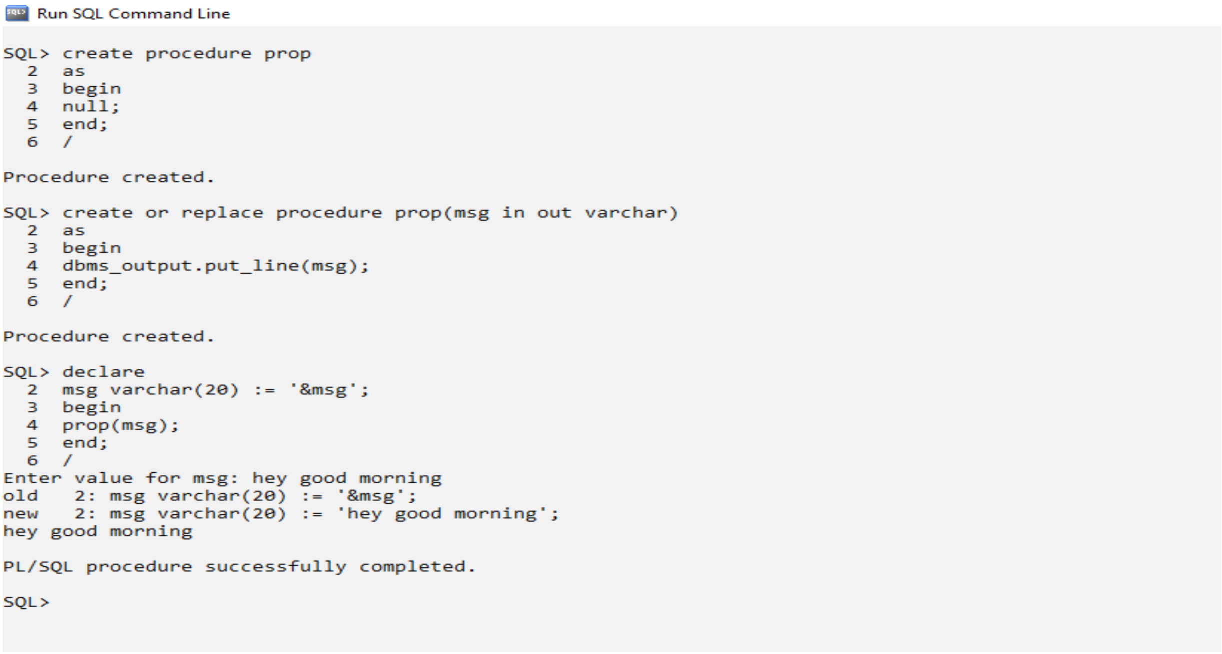
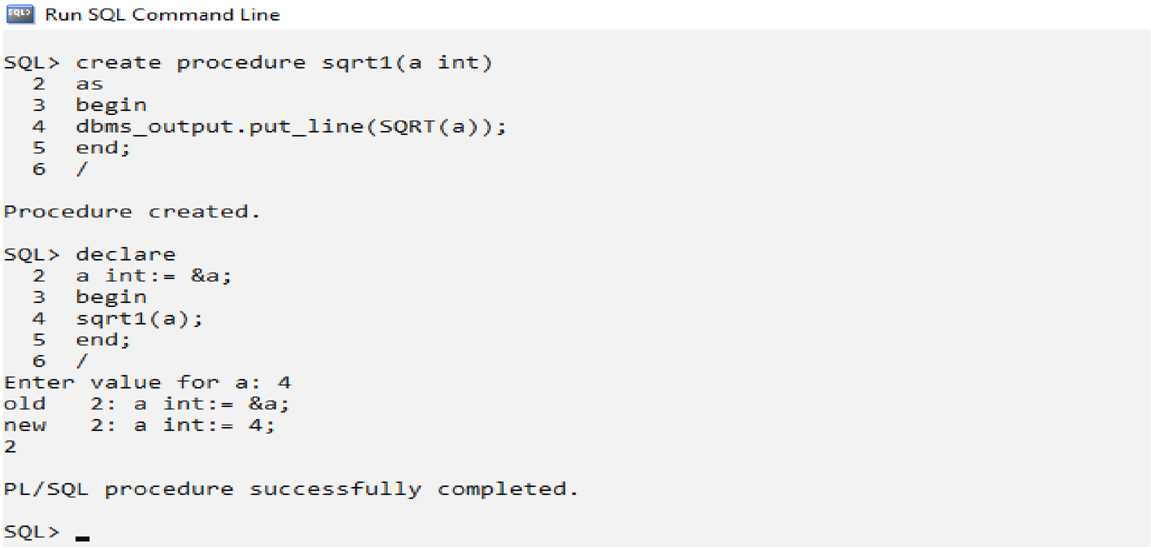
Practical 4: Creating Procedures, Functions and Packages

1. Create and replace an empty procedure and call it.



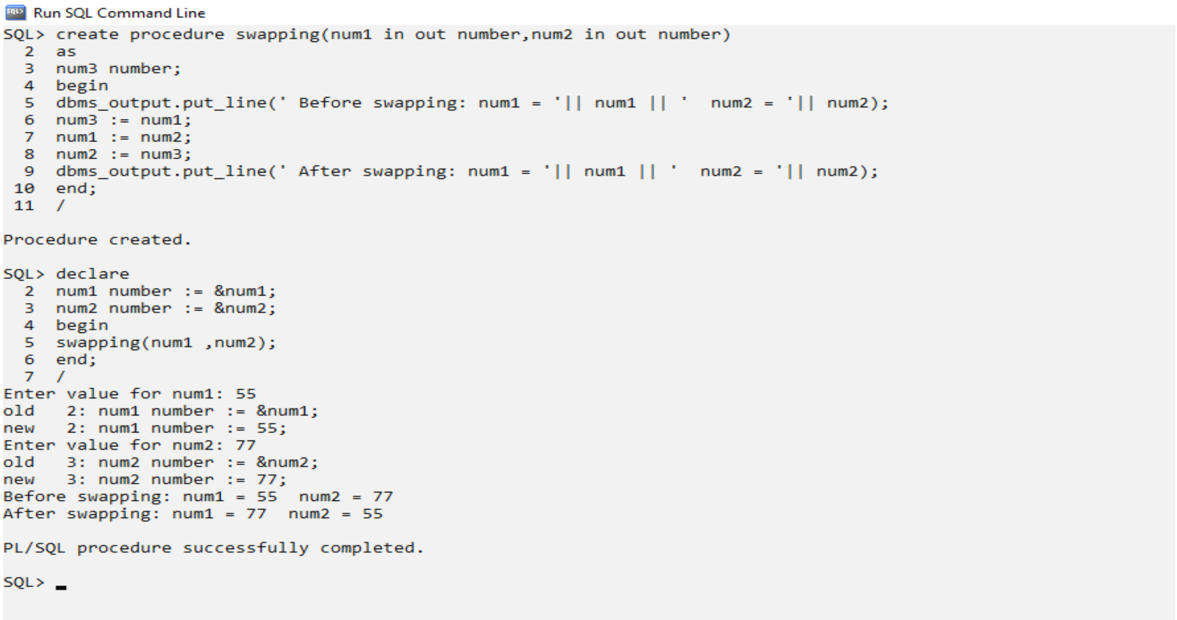
```
SQL> create procedure prop
  2  as
  3  begin
  4  null;
  5  end;
  6  /

Procedure created.

SQL> create or replace procedure prop(msg in out varchar)
  2  as
  3  begin
  4  dbms_output.put_line(msg);
  5  end;
  6  /

Procedure created.

SQL> declare
  2  msg varchar(20) := '&msg';
  3  begin
  4  prop(msg);
  5  end;
  6  /
Enter value for msg: hey good morning
old   2: msg varchar(20) := '&msg';
new   2: msg varchar(20) := 'hey good morning';
hey good morning

PL/SQL procedure successfully completed.

SQL>
```

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



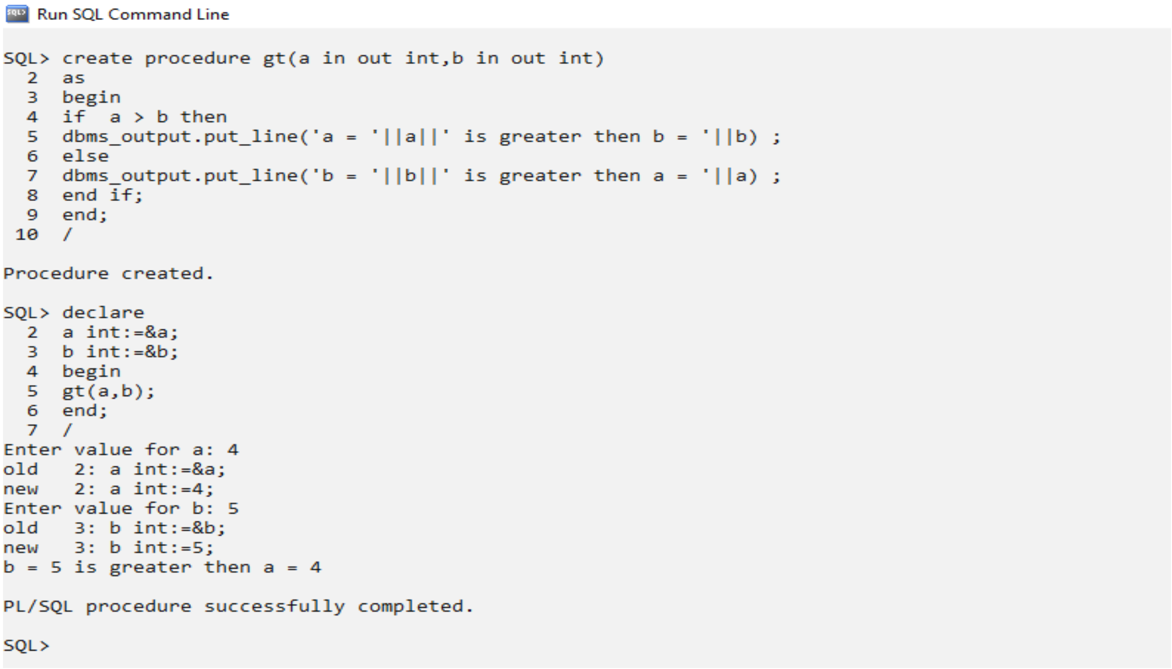
```
SQL> create procedure sqrt1(a int)
  2  as
  3  begin
  4  dbms_output.put_line(SQRT(a));
  5  end;
  6  /

Procedure created.

SQL> declare
  2  a int:= &a;
  3  begin
  4  sqrt1(a);
  5  end;
  6  /
Enter value for a: 4
old   2: a int:= &a;
new   2: a int:= 4;
2

PL/SQL procedure successfully completed.

SQL> _
```

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



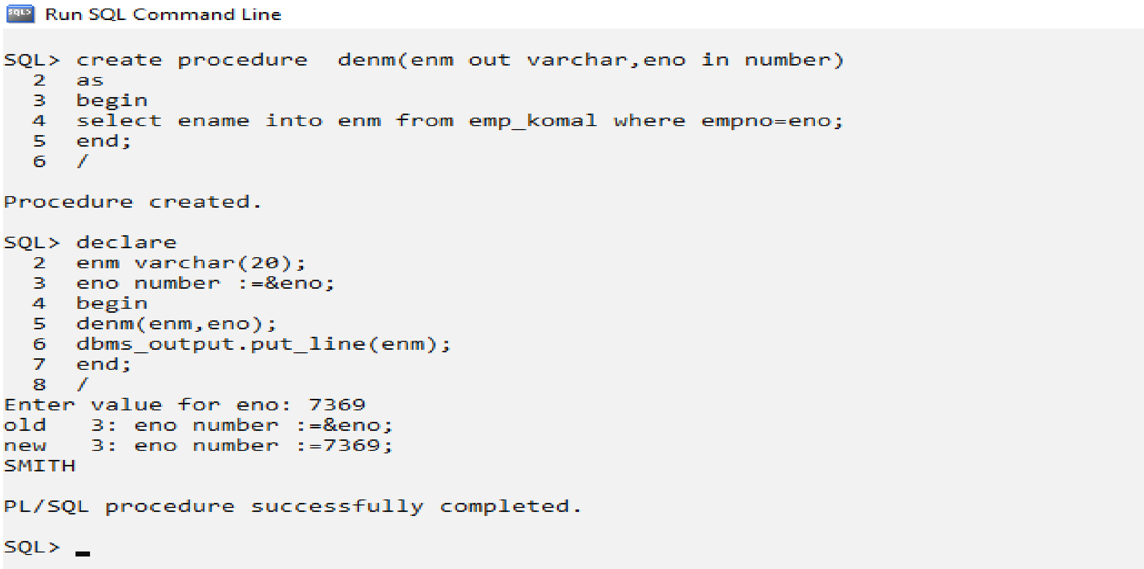
```
SQL> create procedure swapping(num1 in out number,num2 in out number)
  2  as
  3  num3 number;
  4  begin
  5  dbms_output.put_line(' Before swapping: num1 = '|| num1 || '  num2 = '|| num2);
  6  num3 := num1;
  7  num1 := num2;
  8  num2 := num3;
  9  dbms_output.put_line(' After swapping: num1 = '|| num1 || '  num2 = '|| num2);
 10  end;
 11  /

Procedure created.

SQL> declare
  2  num1 number := &num1;
  3  num2 number := &num2;
  4  begin
  5  swapping(num1 ,num2);
  6  end;
  7  /
Enter value for num1: 55
old   2: num1 number := &num1;
new   2: num1 number := 55;
Enter value for num2: 77
old   3: num2 number := &num2;
new   3: num2 number := 77;
Before swapping: num1 = 55  num2 = 77
After swapping: num1 = 77  num2 = 55

PL/SQL procedure successfully completed.

SQL> _
```

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



```
SQL> create procedure gt(a in out int,b in out int)
  2  as
  3  begin
  4  if  a > b then
  5  dbms_output.put_line('a = '||a||' is greater then b = '||b) ;
  6  else
  7  dbms_output.put_line('b = '||b||' is greater then a = '||a) ;
  8  end if;
  9  end;
 10  /

Procedure created.

SQL> declare
  2  a int:=&a;
  3  b int:=&b;
  4  begin
  5  gt(a,b);
  6  end;
  7  /
Enter value for a: 4
old   2: a int:=&a;
new   2: a int:=4;
Enter value for b: 5
old   3: b int:=&b;
new   3: b int:=5;
b = 5 is greater then a = 4

PL/SQL procedure successfully completed.

SQL>
```

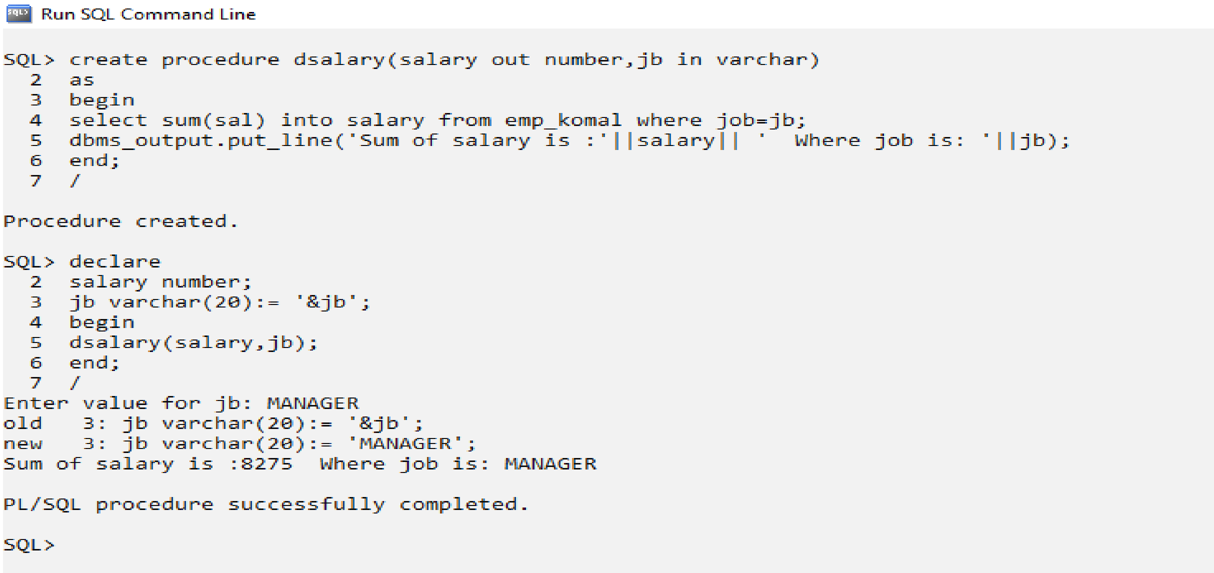
5. Create a procedure and a function to display the employee name whose employeeno is accepted by the user.



```
SQL> create procedure  denm(enm out varchar,eno in number)
  2  as
  3  begin
  4  select ename into enm from emp_komal where empno=eno;
  5  end;
  6  /

Procedure created.

SQL> declare
  2  enm varchar(20);
  3  eno number :=&eno;
  4  begin
  5  denm(enm,eno);
  6  dbms_output.put_line(enm);
  7  end;
  8  /
Enter value for eno: 7369
old   3: eno number :=&eno;
new   3: eno number :=7369;
SMITH

PL/SQL procedure successfully completed.

SQL> _
```

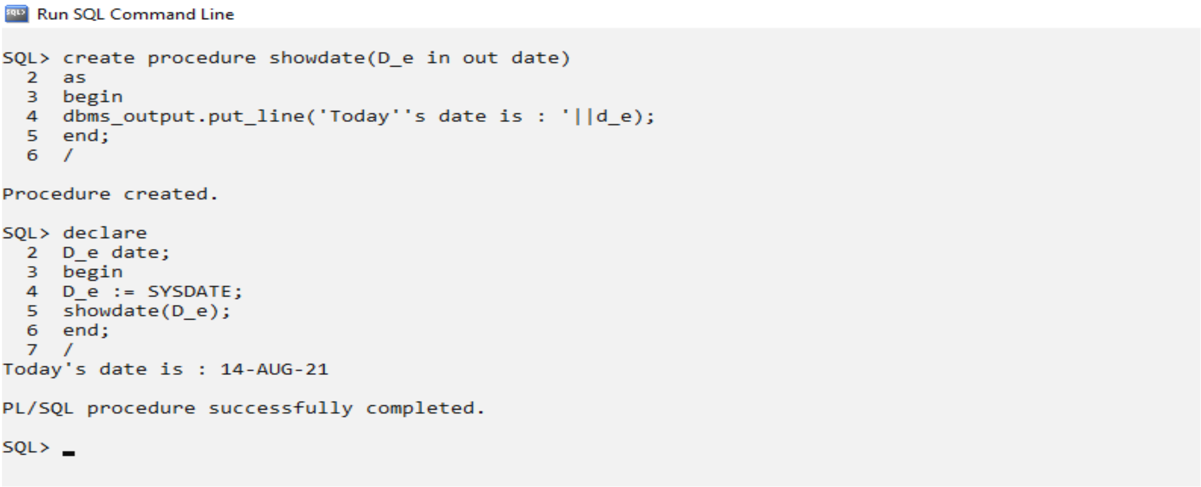
6. Create a procedure and a function to display the sum of salary of the employees

whose job is accepted by the user.



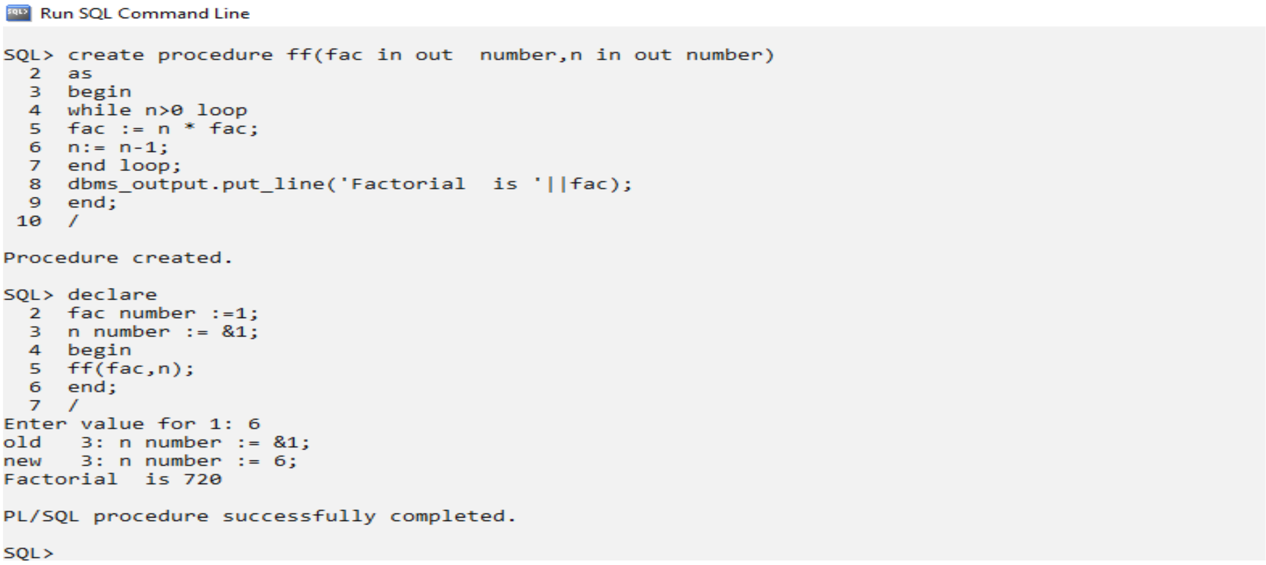
```
SQL> create procedure dsalary(salary out number,jb in varchar)
  2  as
  3  begin
  4  select sum(sal) into salary from emp_komal where job=jb;
  5  dbms_output.put_line('Sum of salary is :'||salary||  '  Where job is: '||jb);
  6  end;
  7  /

Procedure created.

SQL> declare
  2  salary number;
  3  jb varchar(20):= '&jb';
  4  begin
  5  dsalary(salary,jb);
  6  end;
  7  /
Enter value for jb: MANAGER
old   3: jb varchar(20):= '&jb';
new   3: jb varchar(20):= 'MANAGER';
Sum of salary is :8275  Where job is: MANAGER

PL/SQL procedure successfully completed.

SQL>
```

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



```
SQL> create procedure showdate(D_e in out date)
  2  as
  3  begin
  4  dbms_output.put_line('Today''s date is : '||d_e);
  5  end;
  6  /

Procedure created.

SQL> declare
  2  D_e date;
  3  begin
  4  D_e := SYSDATE;
  5  showdate(D_e);
  6  end;
  7  /
Today's date is : 14-AUG-21

PL/SQL procedure successfully completed.

SQL> _
```

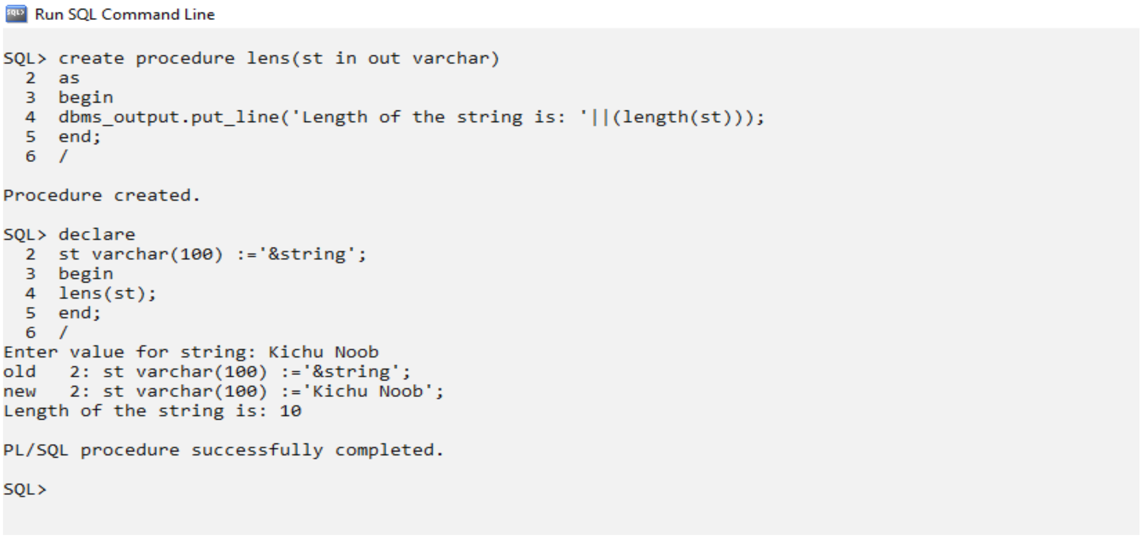
8. Create a procedure to find the factorial of a number.



Run SQL Command Line

```
SQL> create procedure ff(fac in out  number,n in out number)
  2  as
  3  begin
  4  while n>0 loop
  5  fac := n * fac;
  6  n:= n-1;
  7  end loop;
  8  dbms_output.put_line('Factorial  is '||fac);
  9  end;
 10  /

Procedure created.

SQL> declare
  2  fac number :=1;
  3  n number := &1;
  4  begin
  5  ff(fac,n);
  6  end;
  7  /
Enter value for 1: 6
old   3: n number := &1;
new   3: n number := 6;
Factorial  is 720

PL/SQL procedure successfully completed.

SQL>
```

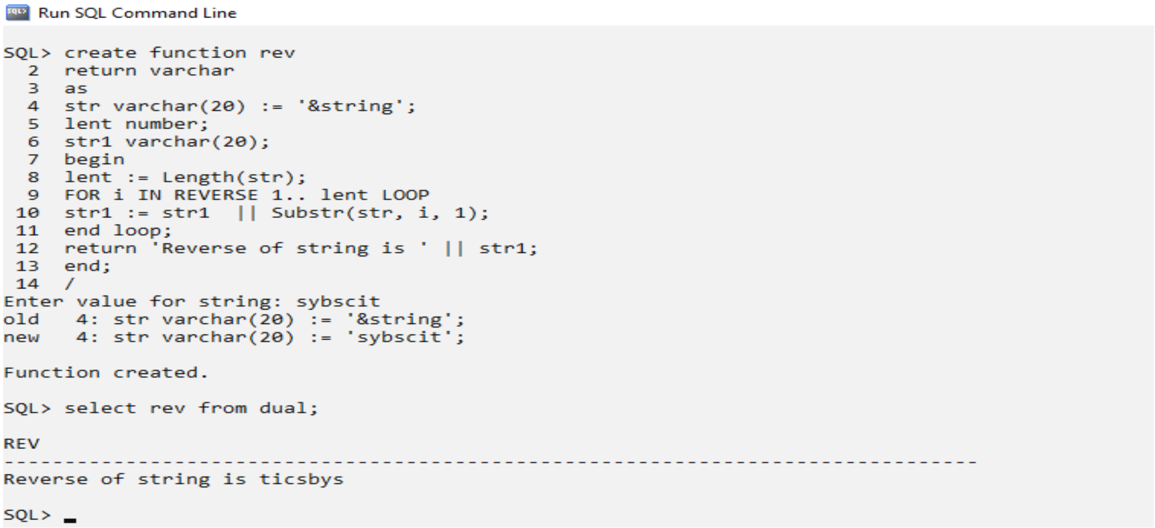
9. Create a procedure to display the length of a string.



Run SQL Command Line

```
SQL> create procedure lens(st in out varchar)
  2  as
  3  begin
  4  dbms_output.put_line('Length of the string is: '||(length(st)));
  5  end;
  6  /

Procedure created.

SQL> declare
  2  st varchar(100) :='&string';
  3  begin
  4  lens(st);
  5  end;
  6  /
Enter value for string: Kichu Noob
old   2: st varchar(100) :='&string';
new   2: st varchar(100) :='Kichu Noob';
Length of the string is: 10

PL/SQL procedure successfully completed.

SQL>
```

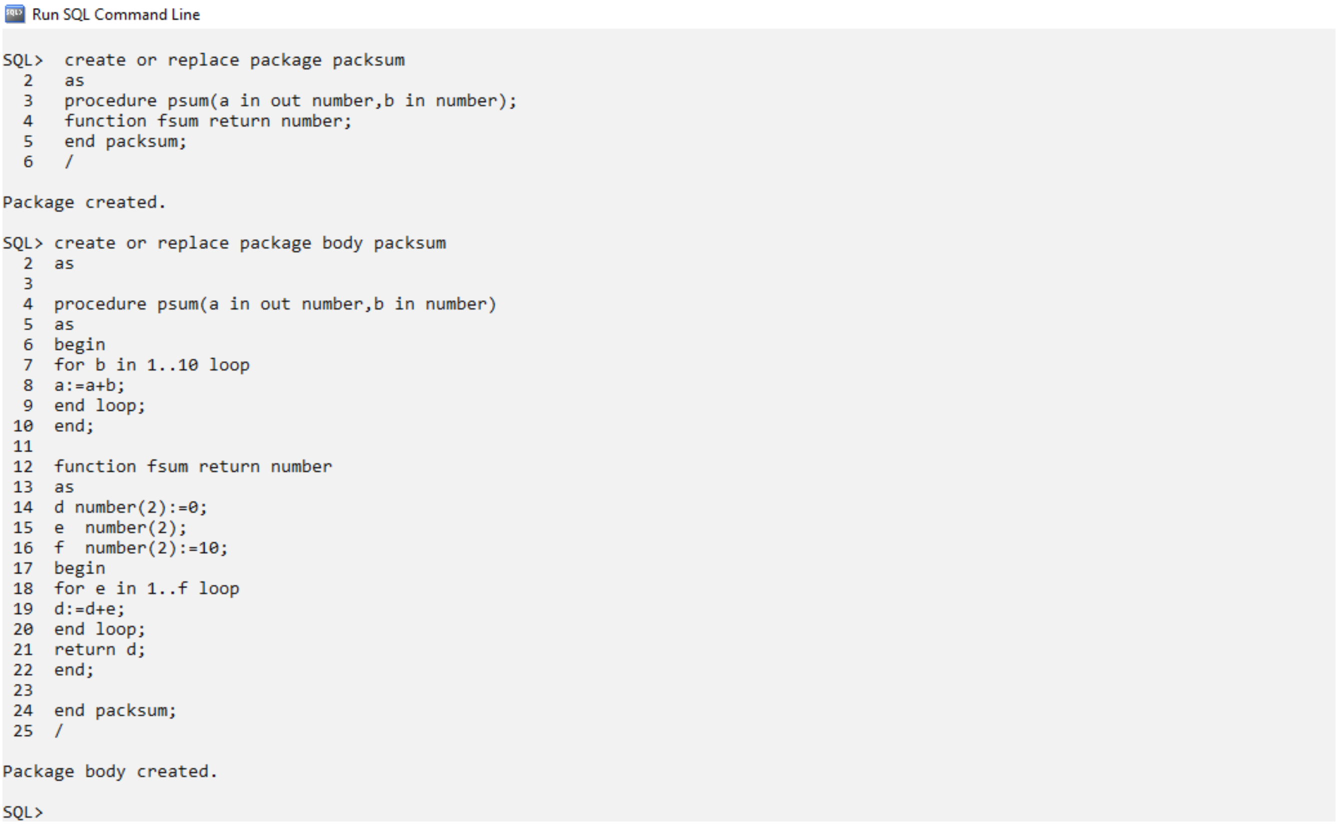
10. Create a function to print the reverse of a string.



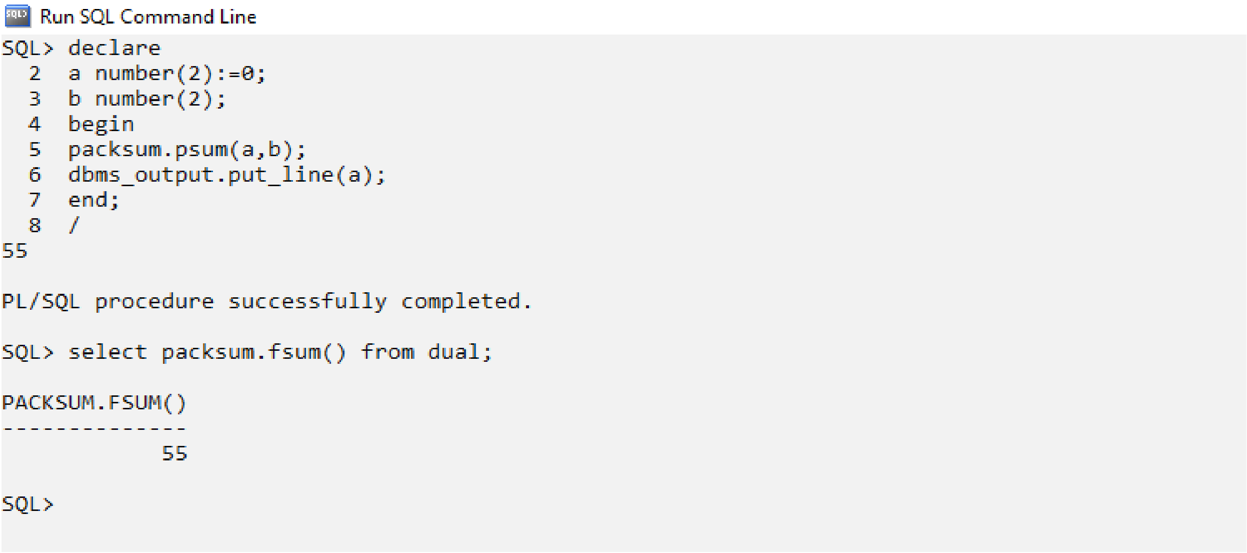
Run SQL Command Line

```
SQL> create function rev
  2  return varchar
  3  as
  4  str varchar(20) := '&string';
  5  lent number;
  6  str1 varchar(20);
  7  begin
  8  lent := Length(str);
  9  FOR i IN REVERSE 1.. lent LOOP
 10  str1 := str1  || Substr(str, i, 1);
 11  end loop;
 12  return 'Reverse of string is ' || str1;
 13  end;
 14  /
Enter value for string: sybscit
old   4: str varchar(20) := '&string';
new   4: str varchar(20) := 'sybscit';

Function created.

SQL> select rev from dual;

REV
--------------------------------------------------------------------------------
Reverse of string is ticsbys

SQL> _
```

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

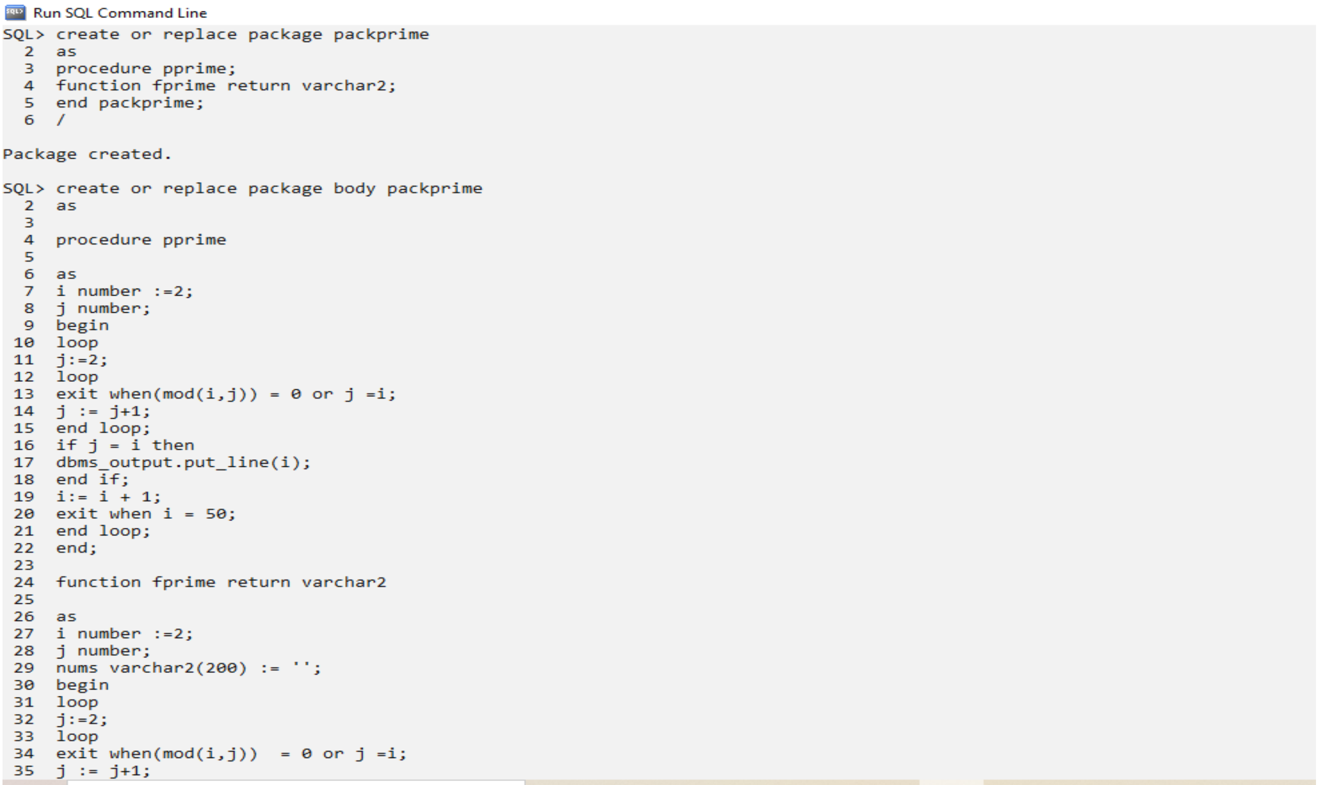


```
SQL>  create or replace package packsum
  as
  procedure psum(a in out number,b in number);
  function fsum return number;
  end packsum;
  /

Package created.

SQL> create or replace package body packsum
 as

 procedure psum(a in out number,b in number)
 as
 begin
 for b in 1..10 loop
 a:=a+b;
 end loop;
 end;

 function fsum return number
 as
 d number(2):=0;
 e  number(2);
 f  number(2):=10;
 begin
 for e in 1..f loop
 d:=d+e;
 end loop;
 return d;
 end;

 end packsum;
 /

Package body created.

SQL>
```

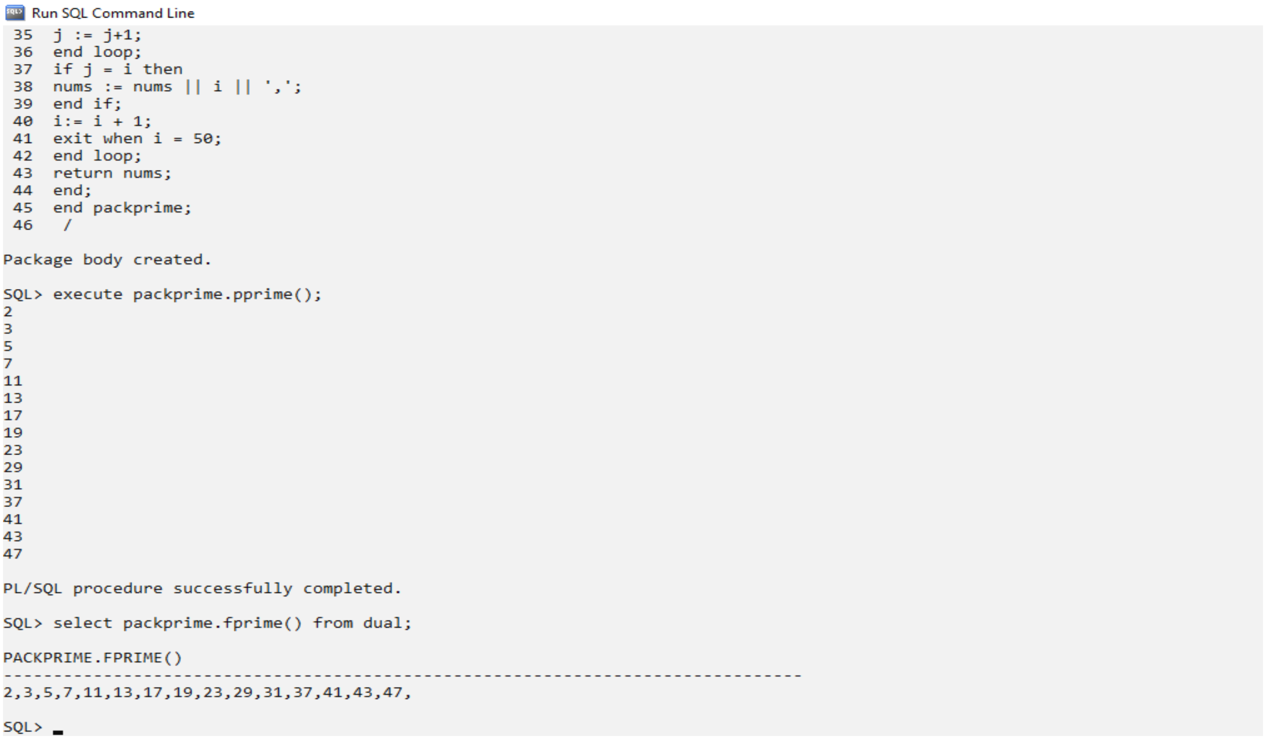


```
SQL> declare
 a number(2):=0;
 b number(2);
 begin
 packsum.psum(a,b);
 dbms_output.put_line(a);
 end;
 /
55

PL/SQL procedure successfully completed.

SQL> select packsum.fsum() from dual;

PACKSUM.FSUM()
--------------
            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 packprime
as
procedure pprime;
function fprime return varchar2;
end packprime;
/

Package created.

SQL> create or replace package body packprime
as

procedure pprime

as
i number :=2;
j number;
begin
loop
j:=2;
loop
exit when(mod(i,j)) = 0 or j =i;
j := j+1;
end loop;
if j = i then
dbms_output.put_line(i);
end if;
i:= i + 1;
exit when i = 50;
end loop;
end;

function fprime return varchar2

as
i number :=2;
j number;
nums varchar2(200) := '';
begin
loop
j:=2;
loop
exit when(mod(i,j))  = 0 or j =i;
j := j+1;
```



```
j := j+1;
end loop;
if j = i then
nums := nums || i || ',';
end if;
i:= i + 1;
exit when i = 50;
end loop;
return nums;
end;
end packprime;
 /

Package body created.

SQL> execute packprime.pprime();
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47

PL/SQL procedure successfully completed.

SQL> select packprime.fprime() from dual;

PACKPRIME.FPRIME()
--------------------------------------------------------------------------------
2,3,5,7,11,13,17,19,23,29,31,37,41,43,47,

SQL> _
```