

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

```
SQL> set serveroutput on
SQL> create procedure aniket as
  2  begin
  3  null;
  4  end;
  5  /

Procedure created.

SQL> execute aniket;

PL/SQL procedure successfully completed.

SQL> create or replace procedure aniket as
  2  begin
  3  dbms_output.put_line('Hello Aniket');
  4  end;
  5  /

Procedure created.

SQL> execute aniket;
Hello Aniket
```

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

```
Oracle SQL*Plus
File Edit Search Options Help

SQL> create procedure square(n in int, s out int)
  2  as
  3  begin
  4  s:=n*n;
  5  end square;
  6  /

Procedure created.

SQL> declare
  2  n int :=&n;
  3  s int;
  4  begin
  5  square(n,s);
  6  dbms_output.put_line('square of the number is:'||s);
  7  end square;
  8  /
Enter value for n: 5
old  2: n int :=&n;
new  2: n int :=5;
square of the number is:25

PL/SQL procedure successfully completed.
```

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

```
Oracle SQL*Plus
File Edit Search Options Help

SQL> create procedure swap(a in out int, b in out int)is
  2  c number;
  3  begin
  4  c :=a;
  5  a :=b;
  6  b :=c;
  7  end;
  8  /

Procedure created.

SQL> declare
  2  a int :=4;
  3  b int :=15;
  4  c int;
  5  begin
  6  swap(a,b);
  7  dbms_output.put_line(a||' '||b);
  8  end;
  9  /
15 4

PL/SQL procedure successfully completed.
```

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

```
Oracle SQL*Plus
File Edit Search Options Help

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

Procedure created.

SQL> declare
  2  a int:=&a;
  3  b int:=&b;
  4  begin
  5  greater_num(a,b);
  6  end;
  7  /
Enter value for a: 15
old  2: a int:=&a;
new  2: a int:=15;
Enter value for b: 47
old  3: b int:=&b;
new  3: b int:=47;
greater number is:47

PL/SQL procedure successfully completed.

Oracle SQL*Plus
File Edit Search Options Help

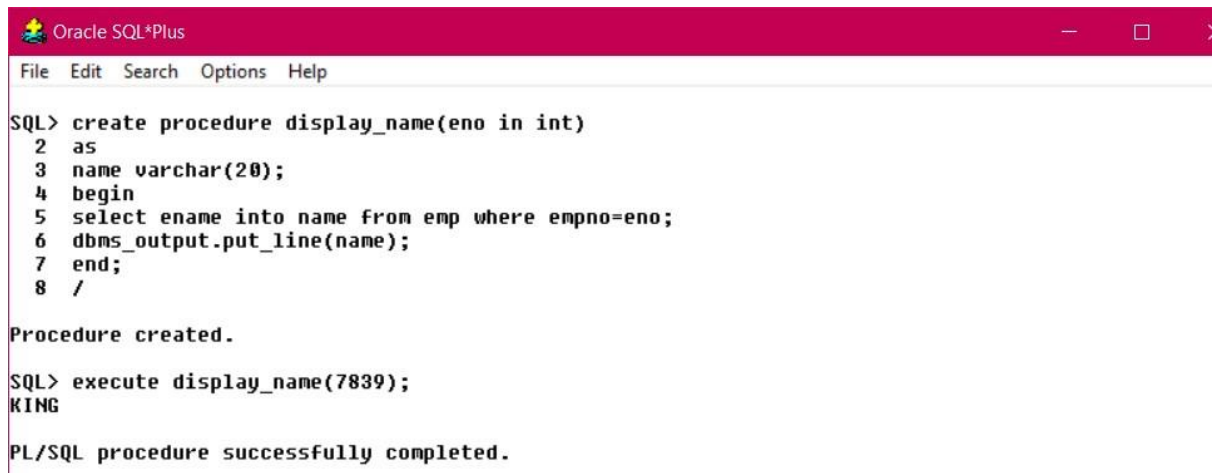
SQL> create or replace function greater(a in int, b in int)
  2  return int
  3  as
  4  begin
  5  if a>b then
  6  return a;
  7  else
  8  return b;
  9  end if;
 10 end;
 11 /

Function created.

SQL> select greater(50,47) from dual;

GREATER(50,47)
-----
              50
```

5. Create a procedure and a function to display the employee name whose employee no is accepted by user.



```
Oracle SQL*Plus
File Edit Search Options Help

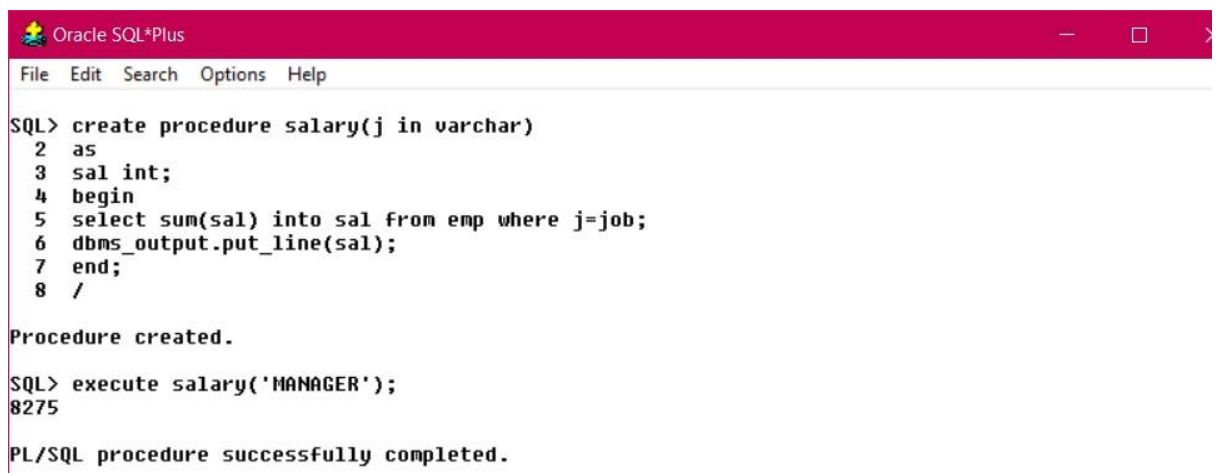
SQL> create procedure display_name(eno in int)
2  as
3  name varchar(20);
4  begin
5  select ename into name from emp where empno=eno;
6  dbms_output.put_line(name);
7  end;
8  /

Procedure created.

SQL> execute display_name(7839);
KING

PL/SQL procedure successfully completed.
```

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



```
Oracle SQL*Plus
File Edit Search Options Help

SQL> create procedure salary(j in varchar)
2  as
3  sal int;
4  begin
5  select sum(sal) into sal from emp where j=job;
6  dbms_output.put_line(sal);
7  end;
8  /

Procedure created.

SQL> execute salary('MANAGER');
8275

PL/SQL procedure successfully completed.
```

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



```
Oracle SQL*Plus
File Edit Search Options Help

SQL> create procedure today_date
2  as
3  m date;
4  begin
5  select sysdate into m from dual;
6  dbms_output.put_line(m);
7  end;
8  /

Procedure created.

SQL> execute today_date;
13-AUG-21

PL/SQL procedure successfully completed.
```

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

```

Oracle SQL*Plus
File Edit Search Options Help

SQL> create procedure factorial(num in int)
  2 as
  3 fact int :=1;
  4 begin
  5 for i in 1..num loop
  6 fact := fact * i;
  7 end loop;
  8 dbms_output.put_line(fact);
  9 end;
 10 /

Procedure created.

SQL> execute factorial(6);
720

PL/SQL procedure successfully completed.

```

9. Create a procedure to display length of a number.

```

SQL> create procedure stringlength(str in varchar)
  2 as
  3 n date;
  4 begin
  5 dbms_output.put_line(length(str));
  6 end;
  7 /

Procedure created.

SQL> execute stringlength('HEY ANIKET');
10

PL/SQL procedure successfully completed.

```

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

```

Oracle SQL*Plus
File Edit Search Options Help

SQL> create function reverse
  2 return number
  3 as
  4 rev number :=0;
  5 n number:=&n;
  6 begin
  7 while (n>0) loop
  8 rev := rev*10+mod(n,10);
  9 n :=floor(n/10);
 10 end loop;
 11 return rev;
 12 end;
 13 /

Enter value for n: 598
old 5: n number:=&n;
new 5: n number:=598;

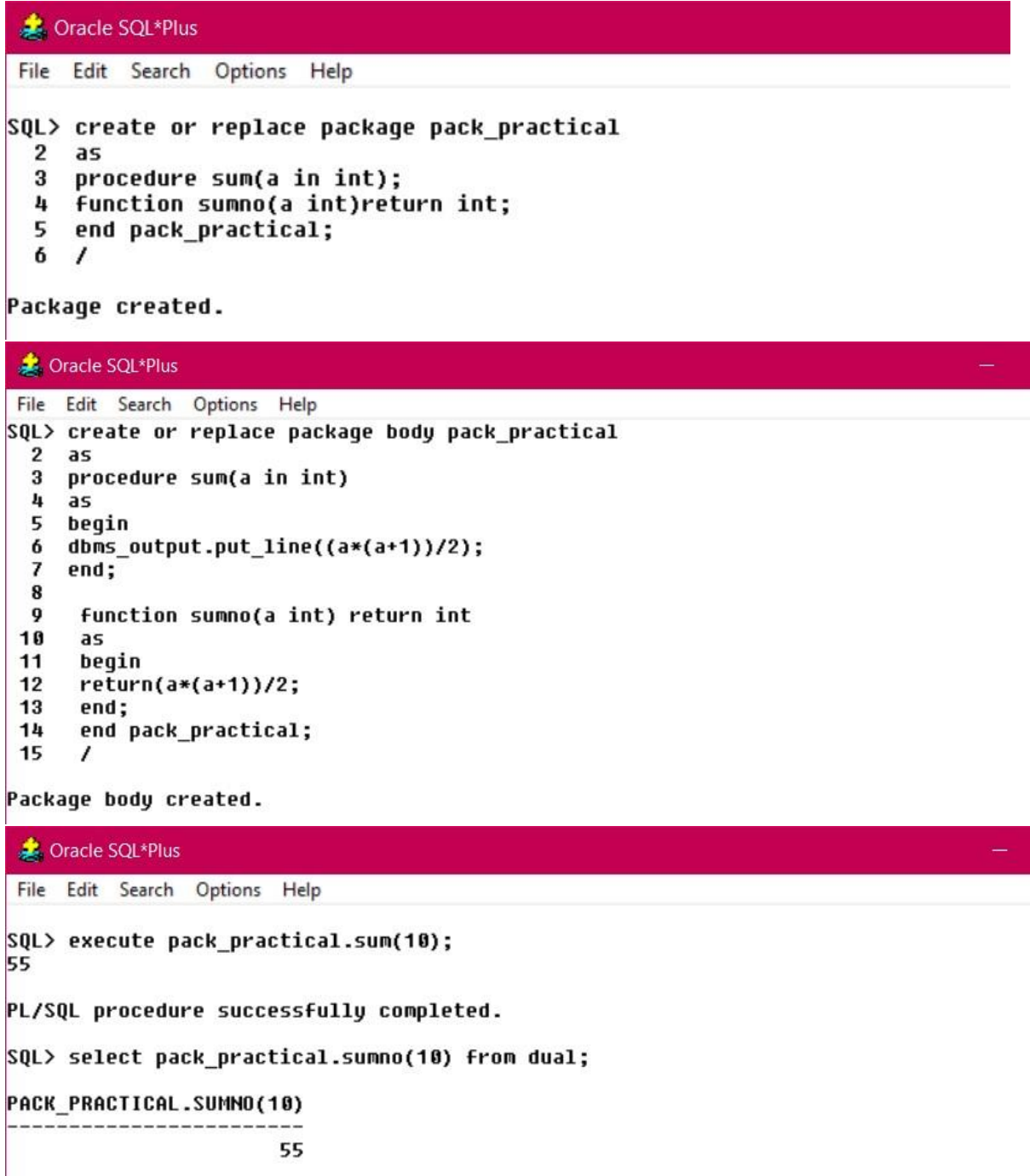
Function created.

SQL> select reverse from dual;

 REVERSE
-----
      895

```

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



The image shows three screenshots of the Oracle SQL\*Plus command-line interface. The first screenshot shows the creation of a package named 'pack\_practical' with two subprograms: a procedure 'sum' and a function 'sumno'. The second screenshot shows the creation of the package body for 'pack\_practical', which implements the logic for the 'sum' procedure and the 'sumno' function. The third screenshot shows the execution of the 'sum' procedure with an argument of 10, resulting in the output 55, and a query that selects the result of the 'sumno' function from a dual table, also returning 55.

```
Oracle SQL*Plus
File Edit Search Options Help

SQL> create or replace package pack_practical
2  as
3  procedure sum(a in int);
4  function sumno(a int) return int;
5  end pack_practical;
6  /

Package created.

Oracle SQL*Plus
File Edit Search Options Help

SQL> create or replace package body pack_practical
2  as
3  procedure sum(a in int)
4  as
5  begin
6  dbms_output.put_line((a*(a+1))/2);
7  end;
8
9  function sumno(a int) return int
10 as
11 begin
12 return(a*(a+1))/2;
13 end;
14 end pack_practical;
15 /

Package body created.

Oracle SQL*Plus
File Edit Search Options Help

SQL> execute pack_practical.sum(10);
55

PL/SQL procedure successfully completed.

SQL> select pack_practical.sumno(10) from dual;

PACK_PRACTICAL.SUMNO(10)
-----
                        55
```

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

```
Oracle SQL*Plus
File Edit Search Options Help
SQL> create or replace package pack_prime
2 as
3 procedure proc(a in int);
4 function func(a int) return varchar;
5 end pack_prime;
6 /

Package created.

SQL> create or replace package body pack_prime
2 as
3
4 procedure proc(a in int)
5 as
6 b number :=0;
7 c number :=2;
8 begin
9 while c<a loop
10 if remainder(a,c)=0 then
11 b:=1;
12 exit;
13 end if;
14 c:=c+1;
15 end loop;
16 if b=0 then
17 dbms_output.put_line(a);
18 end if;
19 end;
20
21 function func(a int)return varchar
22 as
23 b number:=0;
24 c number:=2;
25 prime varchar(130):='';
26 begin
27 while c<a loop
28 if remainder(a,c)=0 then
29 b:=1;
30 exit;
31 end if;
32 c := c+1;
33 end loop;
34 if b=0 then
35 prime :=prime||a;
36 end if;
37 return prime;
38 end;
39 end pack_prime;
40 /

Package body created.
```

```
Oracle SQL*Plus
File Edit Search Options Help
SQL> begin
  2 for i in 2..50 loop
  3   pack_prime.proc(i);
  4 end loop;
  5 end;
  6 /
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47

PL/SQL procedure successfully completed.

SQL> declare
  2 prime varchar(130);
  3 begin
  4   for i in 2..50 loop
  5     prime := pack_prime.func(i);
  6     dbms_output.put_line(prime);
  7   end loop;
  8 end;
  9 /
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47

PL/SQL procedure successfully completed.

SQL>
```