1. Create a procedure to read a character and print whether it is a vowel or not.

```
SQL> create procedure vowel(ch in out char)

2 as

3 begin

4 case ch

5 when 'a' then dbms_output.put_line('vowel');

6 when 'A' then dbms_output.put_line('vowel');

7 when 'e' then dbms_output.put_line('vowel');

8 when 'E' then dbms_output.put_line('vowel');

9 when 'i' then dbms_output.put_line('vowel');

10 when 'I' then dbms_output.put_line('vowel');

11 when 'o' then dbms_output.put_line('vowel');

12 when '0' then dbms_output.put_line('vowel');

13 when 'u' then dbms_output.put_line('vowel');

14 when 'U' then dbms_output.put_line('vowel');

15 else dbms_output.put_line('Not a vowel');

16 end case;

17 end;

18 /

Procedure created.

SQL> declare

2 ch char := '&ch';

3 begin

4 vowel(ch);

5 end;

6 /

Enter value for ch: D

old 2: ch char := '&ch';

new 2: ch char := 'D';

Not a vowel

PL/SQL procedure successfully completed.
```

2. Create a block to print the numbers from 1 to 10 using FOR Loop.

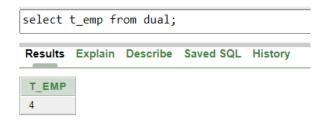
```
SQL> declare
2 i int;
3 begin
4 for i in 1..10 loop
5 dbms_output.put_line(i);
6 end loop;
7 end;
8 /
1
2
3
4
5
6
7
8
9
10
PL/SQL procedure successfully completed.
```

3. Create a function to print the total number of employees working as 'CLERK'.

```
create function t_emp
return int
as
c int;
begin
select count(*) into c from emp_dharmit where job='CLERK';
return c;
end;
/

Results Explain Describe Saved SQL History
```

Function created.



4. Create a block to print even numbers from 2 to 20 and terminate the loop using EXIT statement.

```
SQL> declare
 2 i int := 0;
 3 begin
 4 loop
    dbms_output.put_line(i);
    exit when i=20;
  8
    end loop;
    end;
 10
10
12
14
16
18
20
PL/SQL procedure successfully completed.
```

5. Explain the significance of replace keyword with example.

Ans: It is used to change the definition of existing object.

```
SQL> create or replace function func_d
 2 return int
 4 a int := &a;
 5 b int := &b;
 6 c int;
 7 begin
 8 c := a + b;
 9 return c;
10 end;
11
Enter value for a: 16
old
    4: a int := &a;
new
    4: a int := 16;
Enter value for b: 14
    5: b int := &b;
old
     5: b int := 14;
Function created.
SQL> select func_d from dual;
   FUNC_D
        30
```

```
SQL> create or replace function func_d
 2 return int
 4 a int := &a;
5 b int := &b;
 6 c int;
  7 begin
  8 c := a * b;
 9 return c;
 10
     end;
Enter value for a: 5
old 4: a int := &a;
new 4: a int := 5;
Enter value for b: 5
old 5: b int := &b;
     5: b int := 5;
new
Function created.
SQL> select func_d from dual;
    FUNC_D
         25
```

If we don't use replace keyword then error will occur