exp. no. 1
1. seq stmts
2. area of circle
3. name& roll

exp. no. 2
1. creat seq
2. create table
3. use seq in table;
4. display table
 exp. no.3 all till here

```
SQL> set serveroutput on;
--------------------------------------------------
```

1. Hello world

```
SQL> begin
  2  dbms_output.put_line('Hello SYCS');
  3  end;
  4  /
Hello SYCS

-----------------------------------------------------------------
```

2.Addition of two number

```
SQL> declare
  2  a int:=10;
  3  b int:=11;
  4  c int;
  5  begin
  6  c  :=a+b;
  7  dbms_output.put_line('Addition is '|| c);
  8  end;
  9  /
Addition is 21

---------------------------------------------------------------------------
```

3.Loop in PL/SQL

```
SQL> declare
  2  v_num number :=1;
  3  begin
  4  loop
  5     dbms_output.put_line('Current value of v_number : '||v_num);
  6     v_num:= v_num+1;
  7     exit when v_num>10;
  8  end loop;
  9  end;
 10  /
Current value of v_number : 1
Current value of v_number : 2
Current value of v_number : 3
Current value of v_number : 4
Current value of v_number : 5
Current value of v_number : 6
Current value of v_number : 7
Current value of v_number : 8
Current value of v_number : 9
Current value of v_number : 10

PL/SQL procedure successfully completed.


-----------------------------------------------------------------------------
```

```
##area of circle
----------
declare
r int:=&r;
a float;
begin
a:=3.14*r*r;
dbms_output.put_line('Area is :'||a);
end;
/
o/p

Enter the value of r : 10
Area is : 314

-------------------------------------------------------------------------------
take name and roll no and print

SQL> DECLARE
  2    v_name VARCHAR2(20);
  3    v_rollno NUMBER;
  4  BEGIN
  5    v_name := '&name';
  6    v_rollno := &rollno;
  7    DBMS_OUTPUT.PUT_LINE('Name: ' || v_name);
  8    DBMS_OUTPUT.PUT_LINE('Roll No: ' || v_rollno);
  9  END;
 10  /
Enter value for name: shubham
Enter value for rollno: 41
Name: shubham
Roll No: 41

--------------------------------------------------------------------------------
create table and set sequence and insert value

SQL> create sequence seq1
  2  start with 1
  3  increment by 1
  4  minvalue 0
  5  maxvalue 100
  6  nocycle;

Sequence created.

SQL> create table student(id number(30),name varchar(20));

Table created.

SQL> insert into student values(seq1.nextval,'shubham');

1 row created.

SQL> insert into student values(seq1.nextval,'Gaurav');

1 row created.

SQL> insert into student values(seq1.nextval,'Nilesh');

1 row created.

SQL> insert into student values(seq1.nextval,'Komal');

1 row created.
```

```
SQL> insert into student values(seq1.nextval,'Himanshu');

1 row created.

SQL> select * from student;

        ID NAME
---------- --------------------
         1 shubham
         2 Gaurav
         3 Nilesh
         4 Komal
         5 Himanshu

---------------------------------------------------------------------------------
-------------------

         create table for empyloe insert 5 values

         SQL> create sequence seq2
  2  start with 1
  3  increment by 1
  4  minvalue 0
  5  maxvalue 100
  6  nocycle;

Sequence created.

SQL> create table emp_41(id number(30),name varchar(20), salary number(30), Job
varchar(50), dept_id number);

Table created.


SQL> insert into emp_41 values(seq2.nextval,'shubham', 23000,'manager',102);

1 row created.

SQL> insert into emp_41 values(seq2.nextval,'komal', 20000,'clerk',101);

1 row created.

SQL> insert into emp_41 values(seq2.nextval,'Nilesh', 25000,'finance',103);

1 row created.

SQL> insert into emp_41 values(seq2.nextval,'Gaurav', 2500000,'hr',104);

1 row created.

SQL> insert into emp_41 values(seq2.nextval,'Kundan', 15000,'clerk',105);

1 row created.

SQL> select * from emp_41;

        ID NAME                     SALARY
---------- -------------------- ----------
JOB                                                   DEPT_ID
-------------------------------------------------- ----------
         1 shubham                   23000
manager                                                   102
```

```
         2 komal                     20000
clerk                                                     101

         3 Nilesh                    20000
finance                                                   103


        ID NAME                     SALARY
---------- -------------------- ----------
JOB                                                   DEPT_ID
-------------------------------------------------- ----------
         4 Gaurav                  2500000
hr                                                        104

         5 Kundan                    15000
clerk
```


Q2

```
SQL> DECLARE
  2    emp_sal emp_41.salary%type;
  3    emp_name emp_41.name%type;
  4    emp_job emp_41.job%type;
  5    emp_no emp_41.id%type;
  6  BEGIN
  7    emp_no:=&emp_no;
  8    SELECT  name, job , salary
  9    INTO emp_name, emp_job, emp_sal
 10    FROM emp_41
 11    where id=emp_no;
 12    DBMS_OUTPUT.PUT_LINE('Details: ' || emp_name || ' ' || emp_job || ' ' ||
emp_sal);
 13  END;
 14  /
Enter value for emp_no: 2
Details: komal clerk 20000
```


Day-2 28-07-2024

Q3

```
SQL> declare
  2  num1 int:=&num1;
  3  num2 int:=&num2;
  4  begin
  5  if num1>num2 then
  6  dbms_output.put_line(num1 ||'is greater');
  7  else
  8  dbms_output.put_line(num2 ||'is greater');
  9  end if;
 10  end;
 11  /
Enter value for num1: 23
Enter value for num2: 34
34is greater
```


###Example

```
##compare two string are same or not

SQL> declare
  2  s1 varchar(12):='&s1';
  3  s2 varchar(12):='&s2';
  4  begin
  5  if s1 like s2 then
  6  dbms_output.put_line(s1||' is same as'||s2);
  7  else
  8  dbms_output.put_line(s1||' is not same as'||s2);
  9  end if;
 10   end;
 11   /
Enter value for s1: sycs
Enter value for s2: sy-cs
sycs is not same assy-cs
--------------------------------------------------------------------------------
----------
##else if lader program
#calculate the percentage and show grade

SQL> declare
  2  sub1 int:=&sub1;
  3  sub2 int:=&sub2;
  4  percent float;
  5  begin
  6  percent:=((sub1+sub2)/200)*100;
  7  if percent<35 then
  8  dbms_output.put_line('grade fail');
  9  elsif percent>35 and percent<=59 then
 10  dbms_output.put_line('grade second class');
 11  elsif percent>=60 and percent<=75 then
 12  dbms_output.put_line('grade first class');
 13  elsif percent>75 and percent<=100 then
 14  dbms_output.put_line('grade istinction');
 15  else
 16  dbms_output.put_line('invalid grade');
 17  end if;
 18  end;
 19  /
Enter value for sub1: 67
Enter value for sub2: 78
grade first class


--------------------------------------------------------------------------------
------------------
greet good Morning

SQL> declare
  2  hour varchar(10):=to_char(sysdate,'hh24');
  3  begin
  4  if hour>=16 then
  5  dbms_output.put_line('Good Evening');
  6  elsif hour>=12 then
  7  dbms_output.put_line('Good Afternoon');
  8  else
  9  dbms_output.put_line('Good Morning');
 10  end if;
 11  end;
 12  /
Good Afternoon

--------------------------------------------------------------------------------
```

----

check salery from table

```
SQL> declare
  2      salary int;
  3      begin
  4      select salary into salary from employe where
  5      empid=101;
  6      if salary>=5000 and salary<=10000 then
  7      dbms_output.put_line( 'salary lies in range ');
  8      else
  9      dbms_output.put_line( 'salary does not lies in range');
 10     end if;
 11    end;
 12    /
salary does not lies in range
```

----

write a pl/sql block to accept job from employe table , give the following raise in the salary by sys 9% system 8% hr 7%

```
SQL> declare
  2  job varchar(20):= '&job';
  3  begin
  4  if job ='sys' then
  5  update employe set salary=salary+(9/100*salary);
  6  elsif job ='system' then
  7  update employe set salary=salary+(8/100*salary);
  8  elsif job='hr' then
  9  update employe  set salary=salary+(7/100*salary);
 10  else
 11  dbms_output.put_line( 'Enter correct job');
 12  end if;
 13  end;
 14  /
Enter value for job: hr

PL/SQL procedure successfully completed.

SQL> select * from employe;

     EMPID ENAME                              SALARY OWNER
---------- ------------------------------ ---------- --------------------
      DEPT
----------
       101 ramesh                              22898 lead excutive
        20

       102 suresh                              22898 system
        20

       103 nikhil                              22898 sys
        10


     EMPID ENAME                              SALARY OWNER
---------- ------------------------------ ---------- --------------------
      DEPT
----------
       104 raj                            229231.878 hr
        10

       105 mohali                            4465.11 manager
```

```
        12
```

----

case statement

```
SQL> declare
  2  grade char(1):='&grade';
  3  begin
  4  case grade
  5  when 'O' then dbms_output.put_line('Outstanding');
  6  when 'E' then dbms_output.put_line('Excellent');
  7  when 'G' then dbms_output.put_line('Good');
  8  when 'S' then dbms_output.put_line('Satisfactory');
  9  when 'F' then dbms_output.put_line('Fail');
 10  else dbms_output.put_line('Invaild Grade');
 11  end case;
 12  end;
 13  /
Enter value for grade: E
Excellent

PL/SQL procedure successfully completed.

SQL> /
Enter value for grade: f
Invaild Grade

PL/SQL procedure successfully completed.
```

Expriment-4

----

1. Simple loop

```
SQL> declare
  2  a int:=1;
  3  begin
  4  loop
  5  if a>10 then
  6  exit;
  7  end if;
  8  dbms_output.put_line('num:'||a);
  9  a:=a+1;
 10  end loop;
 11  end;
 12  /
num:1
num:2
num:3
num:4
num:5
num:6
num:7
num:8
num:9
num:10

PL/SQL procedure successfully completed.
----------------------------------------
SQL> declare
  2  a int:=10;
  3  begin
  4  loop
  5  if a<0 then
  6  exit;
  7  end if;
  8  dbms_output.put_line('num:'||a);
  9  a:=a-1;
 10  end loop;
 11  end;
 12  /
num:10
num:9
num:8
num:7
num:6
num:5
num:4
num:3
num:2
num:1
num:0

PL/SQL procedure successfully completed.
---------------------------------------------------------------------------------
----------

FOR LOOP
--------
##reversed
----
SQL> declare
  2  n int:=1;
  3  begin
  4  for n in reverse 1..10
  5  loop
  6  dbms_output.put_line('num: '||n);
  7  end loop;
  8  end;
  9  /
num: 10
num: 9
num: 8
num: 7
num: 6
num: 5
num: 4
num: 3
num: 2
num: 1

###unreversed
--------------

SQL> declare
  2  n int:=1;
  3  begin
  4  for n in 1..10
  5  loop
  6  dbms_output.put_line('num: '||n);
  7  end loop;
  8  end;
  9  /
num: 1
num: 2
num: 3
```

```
num: 4
num: 5
num: 6
num: 7
num: 8
num: 9
num: 10

##total of 1 to 100
-------------------
SQL> declare
  2  i int:=0;
  3  total int:=0;
  4  begin
  5  for i in 1..10 loop
  6  total:=total+i;
  7  end loop;
  8  dbms_output.put_line('total : '||total);
  9  end;
 10  /
total : 55

--------------------------------------------------------------------------------
---------
## While loop

SQL> declare
  2  i int:=1;
  3  total int:=0;
  4  begin
  5  while i<10 loop
  6  total:=total+i;
  7  i:=i+2;
  8  end loop;
  9  dbms_output.put_line('total : '||total);
 10  end;
 11  /
total : 25

PL/SQL procedure successfully completed.

SQL> declare
  2  i int:=1;
  3  total int:=0;
  4  begin
  5  while i<20 loop
  6  total:=total+i;
  7  i:=i+2;
  8  end loop;
  9  dbms_output.put_line('total : '||total);
 10  end;
 11  /
total : 100

--------------------------------------------------------------------------------
-------------------------
## create table AOC and apply operations

SQL> create table aoc1(sno int, radius int, area float);

Table created.

SQL> declare
  2  radius int:=1;
  3  area float;
  4  sno int:=1;
  5  begin
  6  while radius<10 loop
  7  area:=3.14*radius*radius;
  8  insert into aoc1 values(sno, radius , area);
  9  sno:=sno+1;
 10  radius:=radius+1;
 11  end loop;
 12  end;
 13  /

PL/SQL procedure successfully completed.

SQL> select * from aoc1;

       SNO     RADIUS       AREA
---------- ---------- ----------
         1          1       3.14
         2          2      12.56
         3          3      28.26
         4          4      50.24
         5          5       78.5
         6          6     113.04
         7          7     153.86
         8          8     200.96
         9          9     254.34

9 rows selected.

--------------------------------------------------------------------------------
---------------------
```

## using while make factorials program and insert into table

```
SQL> create table fact1(num int, fact int);

Table created.

SQL> declare
  2  fact int:=1;
  3  n int:=&n;
  4  num int:=n;
  5  begin
  6  while n>0 loop
  7  fact:=fact*n;
  8  n:=n-1;
  9  end loop;
 10  insert into fact1 values(num,fact);
 11  end;
 12  /
Enter value for n: 12

PL/SQL procedure successfully completed.

SQL> select * from fact1;

       NUM       FACT
---------- ----------
        12  479001600

--------------------------------------------------------------------------------
---------------------------
```

### looping statements

#Continue statement

```
SQL> begin
  2  for num in 1..10 loop
  3  continue when mod(num,2)!=0;
  4  dbms_output.put_line(num);
  5  end loop;
  6  end;
  7  /
2
4
6
8
10
```


#Goto Statement

Expriment-5

____

```
SQL> begin
  2  goto s3;
  3  <<s1>>
  4  dbms_output.put_line('section 1');
  5  goto s4;
  6  <<s2>>
  7  dbms_output.put_line('section 2');
  8  goto s1;
  9  <<s3>>
 10  dbms_output.put_line('section 3');
 11  goto s2;
 12  <<s4>>
 13  dbms_output.put_line('section 4');
 14  end;
 15  /
```

o/p

____

```
section 3
section 2
section 1
section 4
```

example-2

____

```
SQL> declare
  2  i number;
  3  begin
  4  for i in 1..10 loop
  5  if i=5 then
  6  goto loop_end;
  7  end if;
  8  dbms_output.put_line('Current value of i :'|| i);
  9  end loop;
 10  <<loop_end>>
 11  dbms_output.put_line('Loop ended');
 12  end;
 13  /
```

O/p

____

```
Current value of i :1
Current value of i :2
Current value of i :3
Current value of i :4
Loop ended
```

___

### Expriment-6

```
SQL> CREATE OR REPLACE PROCEDURE proc_emp_41
    (p_id IN NUMBER,
     p_name IN VARCHAR2,
     p_sal IN NUMBER,
     p_job IN VARCHAR2,
     p_deptid IN NUMBER)
 AS
 BEGIN
    INSERT INTO emp_41(id, name, salary, job, dept_id)
     VALUES (p_id, p_name, p_sal, p_job, p_deptid);

     COMMIT;

     DBMS_OUTPUT.PUT_LINE('Record inserted successfully!');
 END;
 /

Procedure created.
SQL> exec proc_emp_41 (6,'yadav',3456,'hr',105);

PL/SQL procedure successfully completed.
SQL> select * from emp_41;

      ID NAME          SALARY       JOB           DEPT_ID
---------- ----------     ----------       ----------          ----------
       1 shubham       23000          manager         102
       2 komal        20000          clerk           101
       3 Nilesh       25000           finance          103
       4 Gaurav      2500000            hr              104
       5 Kundan        15000            clerk           105
       6 yadav        3456              hr              105
```

Expriment-7

###Function Example-1

```
SQL> CREATE OR REPLACE FUNCTION cal_emp_b(emp_id IN NUMBER)
  RETURN NUMBER
 AS
  bonus NUMBER;
 BEGIN
  SELECT salary * 0.1 INTO bonus FROM emp_41
  WHERE id = emp_id;
  RETURN bonus;
 END;
 /

Function created.
SQL> DECLARE
  bonus NUMBER;
  emp_id NUMBER := 2;
 BEGIN
  bonus := cal_emp_b(emp_id);
  DBMS_OUTPUT.PUT_LINE('Updated bonus: ' || TO_CHAR(bonus));
 END;
 /

PL/SQL procedure successfully completed.
```

###Function Example-2

```
SQL> create or replace function calculate_fact(num in number)
  2  return number
  3  as
  4  begin
  5  if num=0 then
  6  return 1;
  7  else
  8  return num*calculate_fact(num-1);
  9  end if;
 10  end;
 11  /

Function created.


SQL> declare
  2  factorial number;
  3  begin
  4  factorial:=calculate_fact(7);
  5  dbms_output.put_line('factorial of 7 :'||factorial);
  6  end;
  7  /
factorial of 7 :5040
PL/SQL procedure successfully completed.
```

##function example-3

```
SQL> create or replace function count_emp(cnt in number)
  2  return number
  3  as
  4  emp_count number;
  5  begin
  6  select count(*) into emp_count from emp_41;
  7  return emp_count;
  8  end;
  9  /

Function created.

SQL> declare
  2  emp_count number;
  3  cnt number;
  4  begin
  5  emp_count:=count_emp(cnt);
  6  dbms_output.put_line('number of employees :'||emp_count);
  7  end;
  8  /
number of employees :5

PL/SQL procedure successfully completed.
```