Practical-5

Aim: To write and execute PL/SQL blocks (with exception handling).

1. **Write a PL-SQL block to find greatest among three given numbers.**

```
SQL> DECLARE
  2  num1 number:=&num1;
  3  num2 number:=&num2;
  4  num3 number:=&num3;
  5  result number;
  6
  7  BEGIN
  8  if(num1>num2) then
  9      if(num1>num3) then
 10  result:=num1;
 11      else
 12  result:=num3;
 13      END IF;
 14  else
 15      if(num2>num3) then
 16  result:=num2;
 17      else
 18  result:=num3;
 19      END IF;
 20  END IF;
 21  dbms_output.put_line('Greatest among three given numbers: '||result);
 22  END;
 23  /


 Enter value for num1: -2
 old   2: num1 number:=&num1;
 new   2: num1 number:=-2;
 Enter value for num2: 90
 old   3: num2 number:=&num2;
 new   3: num2 number:=90;
 Enter value for num3: 79
 old   4: num3 number:=&num3;
 new   4: num3 number:=79;

 Greatest among three given numbers: 90
 PL/SQL procedure successfully completed.
```

2. **Write a PL-SQL block to find out if a year is a leap year.(A leap year is divisible by 4 but not by 100,or it is divisible by 400)**

```
SQL> DECLARE
  2  year integer:=&year;
  3
  4  BEGIN
  5  if(mod(year,400)=0) then
  6    dbms_output.put_line(year||' is a Leap Year');
  7  elsif(mod(year,100)=0) then
  8    dbms_output.put_line(year||' is not a Leap Year');
  9  elsif(mod(year,4)=0) then
 10    dbms_output.put_line(year||' is a Leap Year');
 11  else
 12    dbms_output.put_line(year||' is not a Leap Year');
 13  END IF;
 14  END;
```

```
 15  /


Enter value for year: 2000
old   2: year number:=&year;
new   2: year number:=2000;
2000 is a Leap Year
PL/SQL procedure successfully completed.

SQL> /
Enter value for year: 1500
old   2: year number:=&year;
new   2: year number:=1500;
1500 is not a Leap Year
PL/SQL procedure successfully completed.

SQL> /
Enter value for year: 2004
old   2: year number:=&year;
new   2: year number:=2004;
2004 is a Leap Year
PL/SQL procedure successfully completed.
```

3. Input a number with a substitution variable, and then print its multiplication table using a While loop.

```
SQL> DECLARE
  2  num number:=&num;
  3  i integer:=1;
  4
  5  BEGIN
  6     WHILE i<=10 LOOP
  7        dbms_output.put_line(num||'*'||i||'='||(num*i));
  8        i:=i+1;
  9     END LOOP;
 10  END;
 11  /


Enter value for num: 17
old   2: num number:=&num;
new   2: num number:=17;
17*1=17
17*2=34
17*3=51
17*4=68
17*5=85
17*6=102
17*7=119
17*8=136
17*9=153
17*10=170
PL/SQL procedure successfully completed.
```

4. Write a PL-SQL block to print all odd numbers between 1 and 10 using a basic loop.

```
SQL> DECLARE
  2  i integer:=1;
  3
  4  BEGIN
  5     LOOP
  6     EXIT WHEN i>10;
  7       IF(mod(i,2)!=0) THEN
  8          dbms_output.put_line(i);
  9       END IF;
```

```
 10     i:=i+1;
 11     END LOOP;
 12  END;
 13  /
1
3
5
7
9

PL/SQL procedure successfully completed.
```

5. Using a for loop, print the value 10 to 1 in reverse order.

```
SQL> BEGIN
  2    For i IN REVERSE 1..10 LOOP
  3      dbms_output.put_line(i);
  4    END LOOP;
  5  END;
  6  /


10
9
8
7
6
5
4
3
2
1

PL/SQL procedure successfully completed.
```

6. Write a PL-SQL program to swap the values of two variables. Print the variables before and after swapping.

```
SQL> DECLARE
  2  num1 number:=&num1;
  3  num2 number:=&num2;
  4  temp number;
  5
  6  BEGIN
  7  dbms_output.put_line('Before swapping: num1='||num1||' num2='||num2);
  8  temp:=num1;
  9  num1:=num2;
 10  num2:=temp;
 11  dbms_output.put_line('After swapping: num1='||num1||' num2='||num2);
 12  END;
 13  /


Enter value for num1: 12321
old   2: num1 number:=&num1;
new   2: num1 number:=12321;
Enter value for num2: 153
old   3: num2 number:=&num2;
new   3: num2 number:=153;
Before swapping: num1=12321 num2=153
After swapping: num1=153 num2=12321

PL/SQL procedure successfully completed.
```

Use scott/tiger schema for Q.1,2,3,5,6.

1) An employee no. is entered from keyboard, Write a PL-SQL program to find empno, ename, deptno, sal from emp table. Raise suitable exception, if employee no does not exist.

```
SQL> DECLARE
  2  v_empno emp.empno%TYPE;
  3  v_ename emp.ename%TYPE;
  4  v_deptno emp.deptno%TYPE;
  5  v_sal emp.sal%TYPE;
  6
  7  BEGIN
  8  select empno, ename,deptno,sal
  9  into v_empno,v_ename,v_deptno,v_sal
 10  from emp
 11  where empno=&EMPNO;
 12  dbms_output.put_line('Employee_No.:'||v_empno||',
Employee_Name:'||v_ename||', Department_No.:'||v_deptno||', Salary:'||v_sal);
 13
 14  EXCEPTION
 15  when no_data_found then
 16  dbms_output.put_line('Sorry,no such employee exist.');
 17
 18  END;
 19  /


Enter value for empno: 7369
old  11: where empno=&EMPNO;
new  11: where empno=7369;
Employee_No.:7369, Employee_Name:SMITH, Department_No.:20, Salary:800
PL/SQL procedure successfully completed.

SQL> /
Enter value for empno: 7839
old  11: where empno=&EMPNO;
new  11: where empno=7839;
Employee_No.:7839, Employee_Name:KING, Department_No.:10, Salary:5000
PL/SQL procedure successfully completed.

SQL> /
Enter value for empno: 1001
old  11: where empno=&EMPNO;
new  11: where empno=1001;
Sorry,no such employee exist.
PL/SQL procedure successfully completed.
```

2) An employee no. is entered from keyboard; Write a PL-SQL program to find grade of an employee in emp relation based on employee salary.
If sal>3000$ then grade is A
If sal>2000$ then grade is B
If sal >1000$ then grade is C
Otherwise grade is D
Raise suitable exception, if employee name does not exist.

```
SQL> Declare
  2    v_empno emp.empno%TYPE;
  3    v_sal emp.sal%TYPE;
  4    grade varchar2(1);
  5
  6  Begin
  7  select sal
  8  into v_sal
  9  from emp
 10  where empno=&empno;
 11
 12  if(v_sal>3000)then
 13    grade:= 'A';
 14  elsif(v_sal>2000)then
 15    grade:= 'B';
 16  elsif(v_sal>1000)then
 17    grade:= 'C';
 18  else
 19    grade:= 'D';
 20  end if;
 21
 22  dbms_output.put_line('Grade: '||grade||' , Salary:'||v_sal);
 23
 24  exception
 25    when no_data_found then
 26    dbms_output.put_line('Sorry,no such employee exist.');
 27
 28  end;
 29  /

Enter value for empno: 7839
old  10: where empno=&empno;
new  10: where empno=7839;
Grade: A , Salary:5000
PL/SQL procedure successfully completed.

SQL> /
Enter value for empno: 7566
old  10: where empno=&empno;
new  10: where empno=7566;
Grade: B , Salary:2975
PL/SQL procedure successfully completed.

SQL> /
Enter value for empno: 7499
old  10: where empno=&empno;
new  10: where empno=7499;
Grade: C , Salary:1600
PL/SQL procedure successfully completed.

SQL> /
Enter value for empno: 7369
old  10: where empno=&empno;
```

```
new  10: where empno=7369;
Grade: D , Salary:800
PL/SQL procedure successfully completed.

SQL> /
Enter value for empno: 1001
old  10: where empno=&empno;
new  10: where empno=1001;
Sorry,no such employee exist.
PL/SQL procedure successfully completed.
```

3) Write a PL_SQL program to compute employee name with fourth largest salary.

```
SQL> Declare
  2
  3  v_ename emp.ename%TYPE;
  4  v_sal emp.sal%TYPE;
  5
  6  Begin
  7  select ename,sal
  8  into v_ename,v_sal
  9  from emp e1
 10  where 4-1=(select count(distinct sal)
 11  from emp e2
 12  where e2.sal>e1.sal);
 13  dbms_output.put_line(v_ename||' has 4th largest salary ='||v_sal);
 14
 15  exception
 16  when no_data_found then
 17  dbms_output.put_line('Sorry,no such employee exist.');
 18
 19  end;
 20  /


BLAKE has 4th largest salary =2850

PL/SQL procedure successfully completed.
```

4) You went to a video store and rented a DVD that is due in 3 days from the rental date. Input the rental date, rental month, and rental year. Calculate and print the return date, return month, and return year.

```
SQL> Declare
  2  d date;
  3
  4  Begin
  5  select to_date('&Rental_Date/&Rental_Month/&Rental_Year', 'DD/MM/YYYY')+3
  6  into d
  7  from dual;
  8  dbms_output.put_line(d);
  9  dbms_output.put_line('Return_Date:'||extract(day from d));
 10  dbms_output.put_line('Return_Month:'||extract(month from d));
 11  dbms_output.put_line('Return_Year:'||extract(year from d));
 12
 13  end;
 14  /
```

```
Enter value for rental_date: 31
Enter value for rental_month: 12
Enter value for rental_year: 2012
old   5: select to_date('&Rental_Date/&Rental_Month/&Rental_Year',
'DD/MM/YYYY')+3
new   5: select to_date('31/12/2012', 'DD/MM/YYYY')+3
03-JAN-13
Return_Date:3
Return_Month:1
Return_Year:2013

PL/SQL procedure successfully completed.
```

5) Write a PL-SQL block to ask a user to input a employee Id.Retrieve the employee's name, Sal and commission. Print the name and sum of salary and commission. Also write exception, if employee Id is invalid.

```
SQL> DECLARE
  2  v_empno emp.empno%TYPE;
  3  v_ename emp.ename%TYPE;
  4  v_sal emp.sal%TYPE;
  5  v_comm emp.comm%TYPE;
  6  result emp.sal%TYPE;
  7
  8  BEGIN
  9  select empno, ename,sal,nvl(comm,0)
 10  into v_empno,v_ename,v_sal,v_comm
 11  from emp
 12  where empno=&EMPNO;
 13  result:=v_sal+v_comm;
 14  dbms_output.put_line('Employee_No.:'||v_empno||',
Employee_Name:'||v_ename||', '||v_sal||'+'||v_comm||'='||result);
 15  dbms_output.put_line('Package:'||result*12);
 16
 17  EXCEPTION
 18  when no_data_found then
 19  dbms_output.put_line('Sorry,no such employee exist.');
 20
 21  END;
 22  /


Enter value for empno: 7499
old  12: where empno=&EMPNO;
new  12: where empno=7499;
Employee_No.:7499, Employee_Name:ALLEN, 1600+300=1900
Package:22800
PL/SQL procedure successfully completed.

SQL> /
Enter value for empno: 7369
old  12: where empno=&EMPNO;
new  12: where empno=7369;
Employee_No.:7369, Employee_Name:SMITH, 800+0=800
Package:9600
PL/SQL procedure successfully completed.

SQL> /
Enter value for empno: 1001
old  12: where empno=&EMPNO;
new  12: where empno=1001;
Sorry,no such employee exist.
PL/SQL procedure successfully completed.
```

6)Write PL-SQL program to compute the highest salary in the EMP table, also print the name of Employee earning highest salary

```
SQL> Declare
  2
  3  v_ename emp.ename%TYPE;
  4  v_sal emp.sal%TYPE;
  5
  6  Begin
  7  select ename,sal
  8  into v_ename,v_sal
  9  from emp e1
 10  where 1-1=(select count(distinct sal)
 11  from emp e2
 12  where e2.sal>e1.sal);
 13  dbms_output.put_line(v_ename||' has highest salary ='||v_sal);
 14
 15  exception
 16  when no_data_found then
 17  dbms_output.put_line('Sorry,no such employee exist.');
 18
 19  end;
 20  /


KING has highest salary =5000

PL/SQL procedure successfully completed.
```

OR

```
SQL> Declare
  2
  3  v_ename emp.ename%TYPE;
  4  v_sal emp.sal%TYPE;
  5
  6  Begin
  7  select ename,sal
  8  into v_ename,v_sal
  9  from emp
 10  where sal=(select max(sal)
 11  from emp);
 12  dbms_output.put_line(v_ename||' has highest salary ='||v_sal);
 13
 14  exception
 15  when no_data_found then
 16  dbms_output.put_line('Sorry,no such employee exist.');
 17
 18  end;
 19  /


KING has highest salary =5000

PL/SQL procedure successfully completed.
```