Name: Gaurav Kedia
Roll No: E-39

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

subprograms using Oracle 11g.

```
SQL> SET SERVEROUTPUT ON
SQL> -- PRINT HELLO WORLD



HELLO WORLD !
```

PL/SQL procedure successfully completed.

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

```
DECLARE

num1 NUMBER;
num2 NUMBER;
num3 NUMBER;

BEGIN
num1 := 23;
num2 := 3;
num3 := 56;
        IF num1>num2 AND num1>num3 THEN
                DBMS_OUTPUT.PUT_LINE('maximum number  is : '||num1);
        ELSIF num2>num1 AND num2>num3 THEN
                DBMS_OUTPUT.PUT_LINE('maximum number is : ' || num2);
        ELSE
                DBMS_OUTPUT.PUT_LINE('MAXIMUM NUMEBR IS : ' || num3);
        END IF;
END;
/



SQL> @D:\dbms\notes\max_num.sql
num2 := 3;
*
ERROR at line 9:
ORA-06550: line 9, column 1:
PLS-00371: at most one declaration for 'NUM2' is permitted
```

```
ORA-06550: line 9, column 1:
PL/SQL: Statement ignored
ORA-06550: line 10, column 1:
PLS-00201: identifier 'NUM3' must be declared
ORA-06550: line 10, column 1:
PL/SQL: Statement ignored
ORA-06550: line 11, column 10:
PLS-00371: at most one declaration for 'NUM2' is permitted
ORA-06550: line 11, column 2:
PL/SQL: Statement ignored


SQL> @D:\dbms\notes\max_num.sql
MAXIMUM NUMEBR IS : 56

PL/SQL procedure successfully completed.
```

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)

```
DECLARE

n NUMBER;
x NUMBER;
y NUMBER;
BEGIN
n:=&n;
x:=REMAINDER(n,100);
y:=REMAINDER(n,4);
IF (x=0 and y=0) or x!=0 THEN
        DBMS_OUTPUT.PUT_LINE('NOT A LEAP YEAR:');
ELSE
        DBMS_OUTPUT.PUT_LINE('IS A LEAP YEAR:!!');
END IF;

END;
/


SQL> @D:\dbms\notes\leap_year.sql
Enter value for n: 2000
old   7: n:=&n;
new   7: n:=2000;
NOT A LEAP YEAR:
```

PL/SQL procedure successfully completed.

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

```
DECLARE
num number:=&amp;num;
i integer:=1;

BEGIN
WHILE i&lt;=10 LOOP
dbms_output.put_line(num||' x '||i|| ' = '||(num*i));
i:=i+1;
END LOOP;
END;
/
```

```
SQL> @D:\dbms\notes\MUL-TABLE.SQL"
Enter value for sv_num: 5
old   2: V_NUM NUMBER := &SV_NUM;
new   2: V_NUM NUMBER := 5;
5 X 1 = 5
5 X 2 = 10
5 X 3 = 15
5 X 4 = 20
5 X 5 = 25
5 X 6 = 30
5 X 7 = 35
5 X 8 = 40
5 X 9 = 45
5 X 10 = 50
```

PL/SQL procedure successfully completed.

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

```
DECLARE
i integer:=1;

BEGIN
LOOP
EXIT WHEN i&gt;10;
```

```
IF(mod(i,2)!=0) THEN
dbms_output.put_line(I ||'IS ODD' );
END IF;

i:=i+1;
END LOOP;
END;
 /
SQL> @D:\dbms\notes\ ODDNUMS.SQL"
1 IS ODD
3 IS ODD
5 IS ODD
7 IS ODD
9 IS ODD

PL/SQL procedure successfully completed.
```

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

```
BEGIN

For i IN REVERSE 1..10 LOOP

dbms_output.put_line(i);

END LOOP;

END;

/

SQL> @D:\dbms\notes\ REV.SQL"
10
9
8
7
6
5
4
3
2
1

PL/SQL procedure successfully completed.
```

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

swapping.

```
DECLARE
   num1 number:=&c;
   num2 number:=&num2;
   temp number;

 BEGIN
   dbms_output.put_line(' BEFORE SWAP:'); dbms_output.put_line('NUM1:'||num1);
dbms_output.put_line('NUM2:'||num2);
num1='||num1||' num2='||num2);
   temp:=num1;
   num1:=num2;
  num2:=temp;
dbms_output.put_line(' AFTER SWAP:'); dbms_output.put_line('NUM1:'||num1);
dbms_output.put_line('NUM2:'||num2);


 END;
 /

SQL> @D:\dbms\notes\ SWAP.SQL"
Enter value for num1: 2
old   2: num1 NUMBER := & num1;
new   2: num2 NUMBER := 2;
Enter value for num1: 3
old   3: num1 NUMBER := & num2;
new   3: num2 NUMBER := 3;
BEFORE SWAP:
NUM1: 2
NUM2: 3
AFTER SWAP:
NUM1: 3
NUM2: 2

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.

```
DECLARE
   v_empno emp.empno%TYPE;
   v_ename emp.ename%TYPE;
   v_deptno emp.deptno%TYPE;
   v_sal emp.sal%TYPE;
```

```
BEGIN
  select empno, ename,deptno,sal
  into v_empno,v_ename,v_deptno,v_sal
 from emp
 where empno=&EMPNO;
 dbms_output.put_line('EMPNO :'||ename);
 dbms_output.put_line('ENAME :'||ename);
 dbms_output.put_line('DEPTNO :'||v_deptno);
 dbms_output.put_line('SALARY :'||sal);




  EXCEPTION
  when no_data_found then
  dbms_output.put_line('Sorry,no such employee exist.');

END;
/
```

```
SQL> @D:\dbms\notes\ EMP.SQL
Enter value for empno: 7782
old   2: V_ENO NUMBER := & EMPNO;
new   2: V_ENO NUMBER := 7782;
EMPNO : 7782
ENAME : CLARK
DEPTNO : 10
SALARY : 2450

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.

```
Declare
v_empno emp.empno%TYPE;
v_sal emp.sal%TYPE;
  grade varchar2(1);
```

```
Begin
 select sal
 into v_sal
 from emp
where empno=&empno;

if(v_sal>3000)then
  grade:= 'A';
elsif(v_sal>2000)then
  grade:= 'B';
elsif(v_sal>1000)then
 grade:= 'C';
else
  grade:= 'D';
end if;

dbms_output.put_line('EMPLOYEE GRADE :'||grade);




exception
  when no_data_found then
  dbms_output.put_line(NO DATA FOUND FOR EMPNO:                    ');

end;
/
```

```
SQL> @D:\dbms\notes\ GRADE.SQL
Enter value for empno: 7782
old   2: V_ENO NUMBER := &SV_ENO;
new   2: V_ENO NUMBER := 7782;
EMPLOYEE GRADE : B

PL/SQL procedure successfully completed.

SQL> @D:\dbms\notes\ GRADE.SQL
Enter value for empno: 102
old   2: V_ENO NUMBER := &SV_ENO;
new   2: V_ENO NUMBER := 102;
NO DATA FOUND FOR EMPNO:

PL/SQL procedure successfully completed.
```

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

```
Declare

v_ename emp.ename%TYPE;
v_sal emp.sal%TYPE;

Begin
select ename,sal
into v_ename,v_sal
from emp e1
where 4-1=(select count(distinct sal)
from emp e2
where e2.sal&gt;e1.sal);
dbms_output.put_line(' EMPLOYEE WITH 4TH HIGHEST SALARY: ');
dbms_output.put_line(' EMPLOYEE NAME :'||v_ename);
dbms_output.put_line(' SALARY : '||v_sal);

exception
when no_data_found then
dbms_output.put_line('Sorry,no such employee exist.');

end;
/

SELECT ENAME, SAL
  2  FROM EMP E1
  3  WHERE 5-1 = (SELECT COUNT(DISTINCT SAL) FROM EMP E2 WHERE E2.SAL > E1.SAL);

ENAME           SAL
---------- ----------
CLARK          2450

SQL> SELECT DISTINCT SAL
  2  FROM EMP;

      SAL
----------
     5000
     2450
     1300
     2850
     1250
     2975
     1100
     3000
      800
     1600
     1500
```

```
      SAL
----------
      950

12 rows selected.

SQL> @D:\dbms\notes\HIGH-SAL-4.SQL"
EMPLOYEE WITH 4TH HIGHEST SALARY:
EMPLOYEE NAME : BLAKE
EMPLOYEE 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.

```
Declare
d date;

Begin
 select to_date('&Rental_Date/&Rental_Month/&Rental_Year', 'DD/MM/YYYY')+3
 into d
 from dual;
dbms_output.put_line(d);
 dbms_output.put_line('Return_Date:'||extract(day from d));
dbms_output.put_line('Return_Month:'||extract(month from d));
dbms_output.put_line('Return_Year:'||extract(year from d));

 end;
 /


SQL> @D:\dbms\notes\ DATE.SQL
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.

```
DECLARE
v_empno emp.empno%TYPE;
v_ename emp.ename%TYPE;
v_sal emp.sal%TYPE;
v_comm emp.comm%TYPE;
result emp.sal%TYPE;

8 BEGIN
select empno, ename,sal,nvl(comm,0)
into v_empno,v_ename,v_sal,v_comm
from emp
where empno=&amp;EMPNO;
result:=v_sal+v_comm;
dbms_output.put_line('EMPLOYEE NAME'||v_ename);
dbms_output.put_line('EMPLOYEE SALARY :'|| v_sal);
dbms_output.put_line('EMPLOYEE COMMISION :'|| v_comm);

dbms_output.put_line('EMPLOYEE TOTAL COMPENSATION ||result*12);

EXCEPTION
when no_data_found then
dbms_output.put_line(&#39;Sorry,no such employee exist.&#39;);

END;
/
SQL> @D:\dbms\notes\ COMP.SQL
Enter value for sv_empno: 7654
old   2: V_EMPNO EMP.EMPNO%TYPE := &SV_EMPNO;
new   2: V_EMPNO EMP.EMPNO%TYPE := 7654;
EMPLOYEE NAME : MARTIN
EMPLOYEE SALARY : 1250
EMPLOYEE COMMISION : 1400
EMPLOYEE TOTAL COMPENSATION : 2650

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

```
Declare

v_ename emp.ename%TYPE;
v_sal emp.sal%TYPE;

Begin
select ename,sal
into v_ename,v_sal
from emp e1
where 1-1=(select count(distinct sal)
from emp e2
where e2.sal&gt;e1.sal);
dbms_output.put_line('EMPLOYEE WITH HIGHEST SALARY');
dbms_output.put_line('EMPLOYEE NAME'||v_ename);
dbms_output.put_line('EMPLOYEE NAME'||v_sal);


exception
when no_data_found then
dbms_output.put_line(&#39;Sorry,no such employee exist.&#39;);

end;
/

SQL> @D:\dbms\notes\ HIGH-SAL.SQL"
EMPLOYEE WITH HIGHEST SALARY:
EMPLOYEE NAME : KING
EMPLOYEE SALARY : 5000

PL/SQL procedure successfully completed.
```