Assignment – 6

Name – Abhishek Pratap Singh Roll No – 197103

Sec – A

Emp Table:

```
Name      Null?    Type
--------- -------- ------------
EMPNO     NOT NULL NUMBER(38)
ENAME              VARCHAR2(20)
EJOB               VARCHAR2(20)
MGR                NUMBER(38)
HIREDATE           DATE
SAL                NUMBER(38)
COMM               NUMBER(38)
DEPTNO             NUMBER(38)
```

	EMPNO	ENAME	EJOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7369	SMITH	CLERK	7902	17-12-80	800	(null)	20
2	7499	ALLEN	SALESMAN	7698	20-02-81	1600	300	30
3	7521	WARD	SALESMAN	7698	22-02-81	1250	500	30
4	7566	JONES	MANAGER	7839	02-04-81	2975	(null)	20
5	7654	MARTIN	SALESMAN	7698	28-09-81	1250	1400	30
6	7698	BLAKE	MANAGER	7839	01-05-81	2850	(null)	30
7	7782	CLARK	MANAGER	7839	09-06-81	2450	(null)	10
8	7788	SCOTT	ANALYST	7566	19-04-87	3000	(null)	20
9	7839	KING	PRESIDENT	(null)	17-11-81	5000	(null)	10
10	7844	TURNER	SALESMAN	7698	08-09-81	1500	0	30
11	7876	ADAMS	CLERK	7788	23-05-87	1100	(null)	20
12	7900	JAMES	CLERK	7698	03-12-81	950	(null)	30
13	7902	FORD	ANALYST	7566	03-12-81	3000	(null)	20
14	7934	MILLER	CLERK	7782	23-01-82	1300	(null)	10

Queries:

1	Write the PL/SQL script to display the employee_name, job, salary and department_number from the employee table

```
DECLARE
 CURSOR CUR IS
 SELECT ENAME, EJOB, SAL, DEPTNO FROM EMP;
BEGIN
 FOR ITEM IN CUR
 LOOP
```

```
 DBMS_OUTPUT.PUT_LINE (
 'ENAME = ' || ITEM.ENAME || CHR(9) || 'JOB = ' || ITEM.EJOB || ' ' || CHR(9
)
|| 'SALARY = ' || ITEM.SAL || CHR(9) || ' DEPARTMENT_NO = ' || ITEM.DEPTNO
 );
 END LOOP;

END;
```

```
ENAME = SMITH   JOB = CLERK     SALARY = 800     DEPARTMENT_NO = 20
ENAME = ALLEN   JOB = SALESMAN  SALARY = 1600    DEPARTMENT_NO = 30
ENAME = WARD    JOB = SALESMAN  SALARY = 1250    DEPARTMENT_NO = 30
ENAME = JONES   JOB = MANAGER   SALARY = 2975    DEPARTMENT_NO = 20
ENAME = MARTIN  JOB = SALESMAN  SALARY = 1250    DEPARTMENT_NO = 30
ENAME = BLAKE   JOB = MANAGER   SALARY = 2850    DEPARTMENT_NO = 30
ENAME = CLARK   JOB = MANAGER   SALARY = 2450    DEPARTMENT_NO = 10
ENAME = SCOTT   JOB = ANALYST   SALARY = 3000    DEPARTMENT_NO = 20
ENAME = KING    JOB = PRESIDENT     SALARY = 5000    DEPARTMENT_NO = 10
ENAME = TURNER  JOB = SALESMAN  SALARY = 1500    DEPARTMENT_NO = 30
ENAME = ADAMS   JOB = CLERK     SALARY = 1100    DEPARTMENT_NO = 20
ENAME = JAMES   JOB = CLERK     SALARY = 950     DEPARTMENT_NO = 30
ENAME = FORD    JOB = ANALYST   SALARY = 3000    DEPARTMENT_NO = 20
ENAME = MILLER  JOB = CLERK     SALARY = 1300    DEPARTMENT_NO = 10
```

2 Write a PL/SQL script to increase the salary as per following criteria:

SALARY AMT	INCREMENTED BY
<1200	8%
<2500	12%
<4500	15%
OTHERWISE	20%

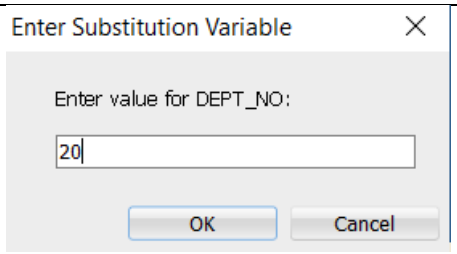
```
 DECLARE
 CURSOR CUR IS
 SELECT EMPNO, SAL FROM EMP;
 EMP_SAL EMP.SAL%TYPE;
 EMP_ID EMP.EMPNO%TYPE;
BEGIN
 FOR ITEM IN CUR
 LOOP
 EMP_SAL := ITEM.SAL;
 EMP_ID := ITEM.EMPNO;
 IF EMP_SAL<1200 THEN
 EMP_SAL := 1.08 * EMP_SAL;
 ELSIF EMP_SAL < 2500 THEN
```

```
 EMP_SAL := 1.12 * EMP_SAL;
 ELSIF EMP_SAL < 4500 THEN
 EMP_SAL := 1.15 * EMP_SAL;
 ELSE
 EMP_SAL := 1.2 * EMP_SAL;
 END IF;
UPDATE EMP SET SAL = EMP_SAL
WHERE EMPNO = EMP_ID;
 END LOOP;
END LOOP;
/
SELECT * FROM EMP;
```

	EMPNO	ENAME	EJOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7369	SMITH	CLERK	7902	17-12-80	864	(null)	20
2	7499	ALLEN	SALESMAN	7698	20-02-81	1792	300	30
3	7521	WARD	SALESMAN	7698	22-02-81	1400	500	30
4	7566	JONES	MANAGER	7839	02-04-81	3421	(null)	20
5	7654	MARTIN	SALESMAN	7698	28-09-81	1400	1400	30
6	7698	BLAKE	MANAGER	7839	01-05-81	3278	(null)	30
7	7782	CLARK	MANAGER	7839	09-06-81	2744	(null)	10
8	7788	SCOTT	ANALYST	7566	19-04-87	3450	(null)	20
9	7839	KING	PRESIDENT	(null)	17-11-81	6000	(null)	10
10	7844	TURNER	SALESMAN	7698	08-09-81	1680	0	30
11	7876	ADAMS	CLERK	7788	23-05-87	1188	(null)	20
12	7900	JAMES	CLERK	7698	03-12-81	1026	(null)	30
13	7902	FORD	ANALYST	7566	03-12-81	3450	(null)	20
14	7934	MILLER	CLERK	7782	23-01-82	1456	(null)	10

3	Write the PL/SQL script to display the employee_name, job, salary of particular department that is input by user using parameter.

```
 DECLARE
 DEPT_NO EMP.DEPTNO%TYPE := &DEPT_NO;
 CURSOR CUR IS
SELECT ENAME, EJOB , SAL FROM EMP
WHERE DEPTNO = DEPT_NO;
BEGIN
 FOR ITEM IN CUR
 LOOP
 DBMS_OUTPUT.PUT_LINE(
 'DEPTNO = ' || DEPT_NO || CHR(9) ||
'NAME = ' || ITEM.ENAME || CHR(9) ||
'JOB = ' || ITEM.EJOB || CHR(9) ||
'SALARY = ' || ITEM.SAL
 );
 END LOOP;
END;
```




```
DEPTNO = 20 NAME = SMITH    JOB = CLERK SALARY = 864
DEPTNO = 20 NAME = JONES    JOB = MANAGER   SALARY = 3421
DEPTNO = 20 NAME = SCOTT    JOB = ANALYST   SALARY = 3450
DEPTNO = 20 NAME = ADAMS    JOB = CLERK SALARY = 1188
DEPTNO = 20 NAME = FORD JOB = ANALYST   SALARY = 3450
```

4	Write a PL/SQL script to display the name, salary and bonus (salary * .12) for each employee using cursor for loop.

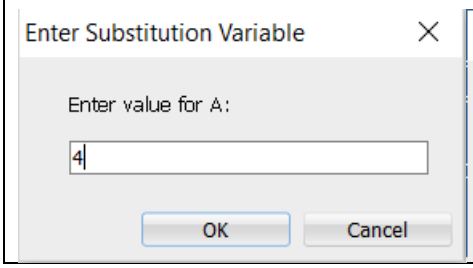
```
 DECLARE
 CURSOR CUR IS
SELECT ENAME, SAL FROM EMP;
BEGIN
 FOR ITEM IN CUR
 LOOP
 DBMS_OUTPUT.PUT_LINE(
 'NAME = ' || ITEM.ENAME || CHR(9) ||
'SALARY = ' || ITEM.SAL || CHR(9) ||
'BONUS = ' || ITEM.SAL*0.12
 );
 END LOOP;
END;
```

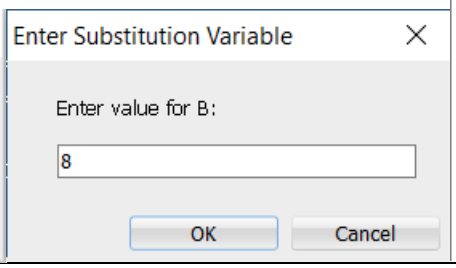
```
NAME = MARTIN   SALARY = 1400   BONUS = 168
NAME = BLAKE    SALARY = 3278   BONUS = 393.36
NAME = CLARK    SALARY = 2744   BONUS = 329.28
NAME = SCOTT    SALARY = 3450   BONUS = 414
NAME = KING SALARY = 6000   BONUS = 720
NAME = TURNER   SALARY = 1680   BONUS = 201.6
NAME = ADAMS    SALARY = 1188   BONUS = 142.56
NAME = JAMES    SALARY = 1026   BONUS = 123.12
NAME = FORD SALARY = 3450   BONUS = 414
NAME = MILLER   SALARY = 1456   BONUS = 174.72
```

5	Write a PL/SQL procedure called Multi_table that take two numbers as parameter and display the product of first number till second number;

```
 CREATE OR REPLACE PROCEDURE MULTI_TABLE(A INT, B INT)
AS
C INT;
```

```
BEGIN
 FOR I IN 1..B LOOP
 C := A*I;
 DBMS_OUTPUT.PUT_LINE(A || ' * ' || I || ' = ' || C);
 END LOOP;
END;
/
DECLARE
A INT := &A;
B INT := &B;
BEGIN
 MULTI_TABLE(A,B);
END;
```





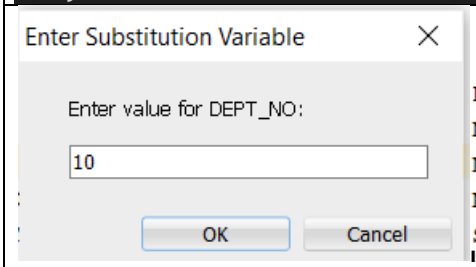

```
4 * 1 = 4
4 * 2 = 8
4 * 3 = 12
4 * 4 = 16
4 * 5 = 20
4 * 6 = 24
4 * 7 = 28
4 * 8 = 32
```

6	Write a PL/SQL procedure that take the department_number as parameter and display the name and salary of employees working in that department and return the sum of salary of such employees using out parameter.

```
CREATE OR REPLACE PROCEDURE DEPT_DETAILS(DEPT_NO IN EMP.DEPTNO%TYPE,
SUM_SAL OUT EMP.SAL%TYPE) IS CURSOR CUR IS
SELECT ENAME, SAL FROM EMP
WHERE DEPTNO = DEPT_NO;
BEGIN
DBMS_OUTPUT.PUT_LINE('DEPARTMENT NO = ' || DEPT_NO);
SUM_SAL := 0;
FOR ITEM IN CUR
LOOP
SUM_SAL := SUM_SAL + ITEM.SAL;
DBMS_OUTPUT.PUT_LINE(
'NAME = ' || ITEM.ENAME || CHR(9) ||
'SALARY = ' || ITEM.SAL
);
END LOOP;
END;
/

DECLARE
DEPT_NO EMP.DEPTNO%TYPE := &DEPT_NO;
SUM_SAL EMP.SAL%TYPE;
```

```
BEGIN
 DEPT_DETAILS(DEPT_NO,SUM_SAL);
 DBMS_OUTPUT.PUT_LINE('SUM OF SALARY = ' || SUM_SAL);
END;
```

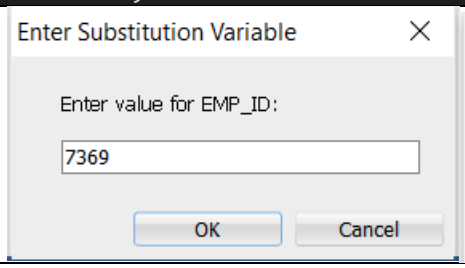


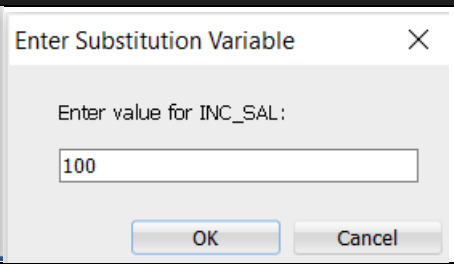

```
DEPARTMENT NO = 10
NAME = CLARK    SALARY = 2744
NAME = KING SALARY = 6000
NAME = MILLER   SALARY = 1456
SUM OF SALARY = 10200
```

7	Write a procedure raise_sal, which increases the salary of an employee. It accepts employee's number and salary increment amount.

```
 CREATE OR REPLACE PROCEDURE RAISE_SAL(EMP_ID IN EMP.EMPNO%TYPE, INC_SAL IN
EMP.SAL%TYPE)
IS
BEGIN
 UPDATE EMP
 SET SAL = SAL + INC_SAL
 WHERE EMPNO = EMP_ID;
END;
/

DECLARE
 EMP_ID EMP.DEPTNO%TYPE := &EMP_ID;
 INC_SAL EMP.SAL%TYPE := &INC_SAL;
BEGIN
 RAISE_SAL(EMP_ID,INC_SAL);
 DBMS_OUTPUT.PUT_LINE('SALARY OF ' || emp_id || ' IS INCREASED BY ' || INC_S
AL);
END;
/
SELECT * FROM EMP;
```



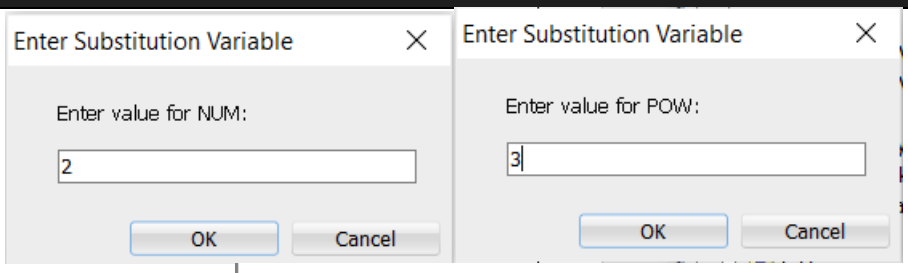


1	7369	SMITH	CLERK	7902	17-12-80	964	(null)	20
2	7499	ALLEN	SALESMAN	7698	20-02-81	1792	300	30
3	7521	WARD	SALESMAN	7698	22-02-81	1400	500	30
4	7566	JONES	MANAGER	7839	02-04-81	3421	(null)	20
5	7654	MARTIN	SALESMAN	7698	28-09-81	1400	1400	30
6	7698	BLAKE	MANAGER	7839	01-05-81	3278	(null)	30
7	7782	CLARK	MANAGER	7839	09-06-81	2744	(null)	10
8	7788	SCOTT	ANALYST	7566	19-04-87	3450	(null)	20
9	7839	KING	PRESIDENT	(null)	17-11-81	6000	(null)	10
10	7844	TURNER	SALESMAN	7698	08-09-81	1680	0	30
11	7876	ADAMS	CLERK	7788	23-05-87	1188	(null)	20
12	7900	JAMES	CLERK	7698	03-12-81	1026	(null)	30
13	7902	FORD	ANALYST	7566	03-12-81	3450	(null)	20
14	7934	MILLER	CLERK	7782	23-01-82	1456	(null)	10

8	Write a procedure raise_sal, which increases the salary of an employee. It accepts employee's number and salary increment amount.

```
 CREATE OR REPLACE FUNCTION NUM_POWER(NUM NUMBER, POW NUMBER)
RETURN NUMBER
IS
P NUMBER := 1;
BEGIN
 FOR I IN 1..POW LOOP
 P := P*NUM;
 END LOOP;
RETURN P;
END;
/

DECLARE
 NUM NUMBER := &NUM;
 POW NUMBER := &POW;
BEGIN
 DBMS_OUTPUT.PUT_LINE(NUM || ' RAISED TO POWER ' || POW || ' IS ' || NUM_POW
ER(NUM,POW)
);
END;
/
```




```
2 RAISED TO POWER 3 IS 8
```

9	Write a set of triggers to maintain the employee_name and department_name fields redundantly in the employee-department relation , so that you donot have to join the employee and department tables just to get a simple department listing.

```
 CREATE TABLE emp_dept_rel
(
emp_name VARCHAR(20),
dept_name VARCHAR(20)
);

CREATE OR REPLACE TRIGGER add_to_dep_emp
AFTER INSERT ON emp
FOR EACH ROW

DECLARE
dept_name VARCHAR(20);
BEGIN
 SELECT dname into dept_name FROM DEPT WHERE deptno=:NEW.deptno;
 INSERT INTO emp_dept_rel VALUES(:NEW.ename, dept_name);
 dbms_output.put_line(:NEW.ename || ' AND ' || dept_name || ' added in table
 ');
END;

INSERT INTO EMP
VALUES (7936,'ABHISHEK','CLERK',7782,'23-01-1982',1300,null,10);

select * from emp_dept_rel;
```

```
Trigger ADD_TO_DEP_EMP compiled
```

```
ABHISHEK AND ACCOUNTING added in table


1 row inserted.
```

	EMP_NAME	DEPT_NAME
1	KILLER	ACCOUNTING
2	ABHISHEK	ACCOUNTING

10	Write a trigger that verifies the joining date when a new row is inserted in the Employee table. Joining date should be greater or equal to current date.

```
CREATE OR REPLACE TRIGGER DATE_VALIDATION
BEFORE INSERT ON emp
FOR EACH ROW
```

```
BEGIN
IF(:NEW.hiredate < sysdate)THEN
  RAISE_APPLICATION_ERROR(-
20125, 'HIREDATE SHOULD BE GREATER OR EQUAL TO CURRENT DATE');
END IF;
END;
/

INSERT INTO EMP
VALUES (6970,'ABHISHEK','ANALYST',7902,'14-04-21',800,null,20);

INSERT INTO EMP
VALUES (6970,'ABHISHEK','ANALYST',7902,'15-04-21',800,null,20);
```

```
Trigger DATE_VALIDATION compiled

Error starting at line : 1 in command -
INSERT INTO EMP
VALUES (6970,'ABHISHEK','ANALYST',7902,'13-04-21',800,null,20)
Error report -
ORA-20125: HIREDATE SHOULD BE GREATER OR EQUAL TO CURRENT DATE
ORA-06512: at "AVI.DATE_VALIDATION", line 3
ORA-04088: error during execution of trigger 'AVI.DATE_VALIDATION'


ABHISHEK AND RESEARCH added in table


1 row inserted.
```

11	Write a trigger that is fired before any row is inserted in the Employee table.

```
 CREATE or replace TRIGGER insertion
BEFORE INSERT ON EMP
FOR EACH ROW
BEGIN
dbms_output.put_line(:NEW.ENAME || ' ADDED');
END;
/

INSERT INTO EMP
VALUES (6971,'ABHISHEK','MANAGER',7902,'15-04-21',800,null,20);
```

```
Trigger INSERTION compiled
```

	ABHISHEK ADDED ABHISHEK AND RESEARCH added in table 1 row inserted.