Experiment 7

```
CREATE TABLE emp (
    empno NUMBER PRIMARY KEY,
    ename VARCHAR2(50),
    job VARCHAR2(50),
    mgr NUMBER,
    hiredate DATE,
    sal NUMBER,
    comm NUMBER,
    deptno NUMBER
);

CREATE TABLE dept (
    deptno NUMBER PRIMARY KEY,
    dname VARCHAR2(50),
    loc VARCHAR2(50)
);

INSERT INTO dept VALUES (10, 'ACCOUNTING', 'NEW YORK');
INSERT INTO dept VALUES (20, 'RESEARCH', 'DALLAS');
INSERT INTO dept VALUES (30, 'SALES', 'CHICAGO');

INSERT INTO emp VALUES (101, 'JOHN', 'MANAGER', NULL, SYSDATE, 6000, NULL, 10);
INSERT INTO emp VALUES (102, 'SMITH', 'CLERK', 101, SYSDATE, 4500, NULL, 20);
INSERT INTO emp VALUES (103, 'ALLEN', 'SALESMAN', 101, SYSDATE, 7000, NULL, 30);
INSERT INTO emp VALUES (104, 'CLARK', 'ANALYST', 101, SYSDATE, 8000, NULL, 20);
INSERT INTO emp VALUES (105, 'WARD', 'SALESMAN', 103, SYSDATE, 5500, NULL, 10);

COMMIT;
```

1. Write a procedure to add an amount of Rs.1000 for the employees whose salaries is greater than 5000 and who belongs to the deptno passed as an argument .

```
CREATE OR REPLACE PROCEDURE ADD_SALARY (p_deptno NUMBER) AS
BEGIN
    UPDATE emp
    SET sal = sal + 1000
    WHERE sal > 5000 AND deptno = p_deptno;

    DBMS_OUTPUT.PUT_LINE('Salary updated for department ' || p_deptno);

    COMMIT;
END;
/


EXEC ADD_SALARY(10);
```

2. Write a PL/SQL block to update the salary of the employee with a 10% increase whose empno is to be passed as an argument for the procedure.

```
CREATE OR REPLACE PROCEDURE INCREASE_SALARY (p_empno NUMBER) AS
BEGIN
    UPDATE emp
    SET sal = sal * 1.10
    WHERE empno = p_empno;

    DBMS_OUTPUT.PUT_LINE('Salary updated for employee ' || p_empno);

    COMMIT;
END;
/


EXEC INCREASE_SALARY(103);
```

3. Write a function to find the salary of the employee who is working in the deptno 20(to bepassed as an argument).

```
CREATE OR REPLACE FUNCTION GET_SALARY (p_deptno NUMBER) RETURN NUMBER IS
    v_salary NUMBER;
BEGIN
    SELECT sal INTO v_salary FROM emp WHERE deptno = p_deptno AND ROWNUM = 1;
    RETURN v_salary;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RETURN NULL;
END;
/

VAR salary NUMBER;
EXEC :salary := GET_SALARY(20);
PRINT salary;
```

4. Write a function to find the nature of job of the employee whose deptno is 20(to be passed asan argument)

```
CREATE OR REPLACE FUNCTION GET_JOB (p_deptno NUMBER) RETURN VARCHAR2 IS
    v_job VARCHAR2(50);
BEGIN
    SELECT job INTO v_job FROM emp WHERE deptno = p_deptno AND ROWNUM = 1;
    RETURN v_job;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RETURN 'No Employee Found';
END;
/

VAR job VARCHAR2(50);
EXEC :job := GET_JOB(20);
PRINT job;
```

5. Write a PL/SQL block to obtain the department name of the employee who works for deptno30.

```
DECLARE
    v_dname VARCHAR2(50);
BEGIN
    SELECT dname INTO v_dname FROM dept WHERE deptno = 30;


    DBMS_OUTPUT.PUT_LINE('Department Name: ' || v_dname);
END;
    /




    SET SERVEROUTPUT ON;
```