Assignment 9

1. Using cursor display the details of all those employees from emp table whose sum of sal and comm. Is more than 25000;

Sol.

```
SET SERVEROUTPUT ON;
DECLARE
  Vempno emp.empno%TYPE;
  Vename emp.ename%TYPE;
  Vsal emp.sal%TYPE;
  Vdeptno emp.deptno%TYPE;
  CURSOR C1 IS
    SELECT empno, ename, sal, deptno
    FROM emp
    WHERE sal + NVL(comm, 0) > 25000;
BEGIN
  OPEN C1;
  LOOP
    FETCH C1 INTO Vempno, Vename, Vsal, Vdeptno;
    EXIT WHEN C1%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE(Vempno || ' ' || Vename || ' ' || Vsal || ' ' || Vdeptno);
  END LOOP;
  CLOSE C1;
END;
/
```

2. Write a block which uses a Cursor For Loop to select the five hightest earners from emp table and write their details to the message table which has the following three attributes: EmpNumber, Name and Salary.

Sol.
Create message table

```
CREATE TABLE message (
  EmpNumber NUMBER,
  Name VARCHAR2(100),
  Salary NUMBER
);

SET SERVEROUTPUT ON
DECLARE
  CURSOR C2 IS
    SELECT empno, sal, ename
    FROM emp
    ORDER BY sal DESC;
BEGIN
  FOR record1 IN C2 LOOP
    INSERT INTO message (EmpNumber, Name, Salary)
    VALUES (record1.empno, record1.ename, record1.sal);
```

```
    EXIT WHEN C2%ROWCOUNT = 5;
  END LOOP;
  COMMIT;
END;
/
```

3. **Create a table Emp_sal_inc that have three column (Emp_id,Cur_sal,Inc_date). Now write a PL/SQL block of code will allow 2% salary increment of all employee of Computer Science department. After that a record is to be inserted into the above table.**

```
CREATE TABLE emp_sal (
  emp_id VARCHAR2(8) PRIMARY KEY,
  cur_sal NUMBER(7,2) NOT NULL,
  inc_date DATE NOT NULL
);

SET SERVEROUTPUT ON
DECLARE
  inc NUMBER;
  emp_sal_rec emp%ROWTYPE;
  CURSOR cur IS
    SELECT *
    FROM emp
    WHERE deptno = (SELECT deptno FROM dept WHERE dname LIKE 'RESEARCH');
BEGIN
  FOR emp_sal_rec IN cur LOOP
    inc := emp_sal_rec.sal * 1.02;
    INSERT INTO emp_sal (emp_id, cur_sal, inc_date)
    VALUES (emp_sal_rec.empno, inc, SYSDATE);
  END LOOP;
  COMMIT;
END;
/
```

4. Write a PL/SQL block of code to update the location of specific department number that will be taken from user. Display an appropriate message using SQL%FOUND based on existence of the record in the Department table and display an appropriate message using SQL%FOUND based on the non-existence of the record in Department Table.

```
DECLARE
  dno NUMBER := &deptno;
BEGIN
  UPDATE dept
  SET loc = 'KOLKATA'
  WHERE deptno = dno;

  IF SQL%FOUND THEN
    DBMS_OUTPUT.PUT_LINE('Updated');
```

```
 ELSE
   DBMS_OUTPUT.PUT_LINE('Data NOT Found');
 END IF;
END;
/
```

5. Write a PL/SQL block of code using your own exception handling that will show an error message whenever you want to insert a null value in a not null column.

```
SET SERVEROUTPUT ON
DECLARE
user_error EXCEPTION;
PRAGMA EXECPTION_INIT(user_error, -0150);
BEGIN
insert into emp values( NULL, 'Subir', 'RESEARCH', 7900, '12-MAY-2002', 90000,1200,200);
EXCEPTION
WHEN user_error THEN
DBMS_OUTPUT.PUT_LINE ('CANNOT INSERT NULL VALUES');

END;
/
```

6. Write a PL/SQL block of code which raised 'NO DATA FOUND' exception when there is no employee joined in 1997 at the time of displaying 1997 joined employees.

```
SET SERVEROUTPUT ON
DECLARE
Vename emp.ename%type;
Vjob emp.job%type;
BEGIN
SELECT ename, job INTO Vename, Vjob FROM emp
WHERE do BETWEEN '01-JAN-97' AND '31-DEC-97';
dbms_output.put_line(Vename ||' ' || Vjob);
EXCEPTION
WHEN NO_DATA_FOUND THEN
dbms_output.put_line(' No Employee hired in 97');

END;
/
```

7. Write a PL/SQL Block of code to find the QOH amount from item_master table. If QOH is less than 1 raise an user define exception "INSUFFICIENT STOCK".

```
SET SERVEROUTPUT ON
DECLARE
OUT_OF_STOCK EXCEPTION;
QTYOH item_master.QOH%type;
BEGIN
SELECT QOH into QTYOH from item_master;
IF QTYOH <1 THEN
RAISE OUT_OF_STOCK;
END IF;
```

```
EXCEPTION
WHEN OUT_OF_STOCK THEN
dbms_output.put_line('INSUFFICIENT STOCK');

END;
/
```