LAB SHEET-1

1) List empno,ename and salary?

```
SQL>   SELECT ENAME,EMPNO,SALARY FROM EMP;
```

OUTPUT:

```
ENAME        EMPNO      SAL
----------   ----------   ----------
KING          7839      5000
BLAKE         7698      2850
CLARK         7782      2450
JONES         7566      2975
MARTIN        7654      1250
```

2) List the names of all managers?

```
SQL>   SELECT ENAME FROM EMP WHERE JOB=’MANAGER’;
```

OUTPUT:

```
ENAME
----------
BLAKE
CLARK
JONES
```

3) List all clerks in deptno 30?

```
SQL>   SELECT ENAME FROM EMP WHERE JOB=’CLERK’ AND DEPTNO=30;
```

OUTPUT:

```
ENAME
----------
JAMES
```

4) List the employes to who manager is 7698?

```
SQL>   SELECT ENAME,JOB FROM EMP WHERE MANAGER=7698;
```

OUTPUT:

```
ENAME     JOB
----------   ---------
MARTIN    SALESMAN
ALLEN     SALESMAN
TURNER    SALESMAN
JAMES     CLERK
WARD      SALESMAN
```

5) List jobs in dept 20?

```
SQL>   SELECT DISTINCT(JOB) FROM EMP WHERE DEPT=20;
```

OUTPUT:

```
JOB
---------
ANALYST
CLERK
MANAGER
```

6) List employee names whose salary is between 2000 and 3000?

```
SQL>   SELECT ENAME,EMPNO,SAL FROM EMP WHERE SAL BETWEEN 2000 AND  3000;
```

OUTPUT:

```
ENAME         EMPNO       SAL
----------   ----------  ----------
BLAKE        7698      2850
CLARK        7782      2450
JONES         7566      2975
FORD         7902      3000
SCOTT        7788      3000
```

7) List employees in departments 10,20?

```
SQL>   SELECT ENAME,EMPNO,DEPTNO FROM EMP WHERE DEPTNO IN(10,20);
```

OUTPUT:

```
ENAME         EMPNO    DEPTNO
----------   ----------    ------
KING         7839          10
CLARK        7782          10
JONES         7566          20
```

8) List ename which begin with s?

```
SQL>   SELECT ENAME FROM EMP WHERE ENAME LIKE ‘S%’;
```

OUTPUT:

```
ENAME
----------
SMITH
SCOTT
```

9) List ename having 'A' in there names?

```
SQL>   SELECT ENAME FROM EMP WHERE ENAME LIKE'%A%';
```

OUTPUT:

```
ENAME
----------
BLAKE
CLARK
```

10) List employees who have joined in JAN?

```
SQL>   SELECT ENAME,TO_CHAR(HIREDATE,'MON') FROM EMP WHERE
       TO_CHAR(HIREDATE,'MON')='JAN';
```

OUTPUT:

```
ENAME      TO_
----------  ---
ADAMS      JAN
MILLER     JAN
```

11) List employees who have joined in the year 81?

```
SQL>   SELECT ENAME, TO_CHAR(HIREDATE,'YYYY') AS YEAR FROM EMP WHERE
       TO_CHAR(HIREDATE,'YYYY')=1981;
```

OUTPUT:

```
ENAME      YEAR
----------  ----
KING       1981
BLAKE      1981
```

12) List all distinct jobs?

```
SQL>   SELECT DISTINCT(JOBS) FROM EMP;
```

OUTPUT:

```
JOB
---------
ANALYST
CLERK
MANAGER
PRESIDENT
SALESMAN
```

13) List ename in alphabetical order?

```
SQL>   SELECT ENAME FROM EMP ORDER BY ENAME;
```

OUTPUT:

```
ENAME
----------
ADAMS
ALLEN
BLAKE
```

14) List ename alphabetically department wise?

```
SQL>   SELECT ENAME,DEPTNO FROM EMP ORDER BY DEPTNO;
```

OUTPUT:

```
ENAME        DEPTNO
----------  ----------
KING           10
CLARK          10
```

15) List employee names alphabetically job wise?

```
SQL>   SELECT ENAME,JOB FROM EMP ORDER BY JOB;
```

OUTPUT:

```
ENAME     JOB
----------  ---------
FORD      ANALYST
SCOTT     ANALYST
JAMES     CLERK
SMITH     CLERK
```

16) List eno,ename,sal,DA(15% of sal) and PF(10% of sal)?

```
SQL>   SELECT ENO,ENAME,SAL,(SAL*0.15) AS DA,(SAL*0.10) AS PF FROM EMP;
```

OUTPUT:

```
EMPNO   ENAME          SAL        DA        PF
----------  ----------        -------     --------    -------
  78139   KING           5000       750       500
  7698    BLAKE          2850       427.5     285
  7782    CLARK          2450       367.5     245
```

17) List enames having an experience more than 15 years?

```
SQL>   SELECT NAME,ENO,ROUND(MONTHS_BETWEEN(SYSDATE,HIREDATE)/12) AS EXP
       FROM EMP WHERE (MONTHS_BETWEEN(SYSDATE,HIREDATE)/12)>15;
```

OUTPUT:

```
ENAME        EMPNO      EXP
----------    ----------   ----------
KING         7839       30
BLAKE        7698       31
CLARK        7782       31
```

18) List ename whose commission is NULL?

```
SQL>   SELECT ENAME ,COMM. FROM EMP WHERE COMM IS NULL;
```

OUTPUT:

```
ENAME         COMM
----------    ----------
KING
BLAKE
CLARK
JONES
```

19) List employees who donot report to anybody?

```
SQL>   SELECT ENAME,MGR FROM EMP WHERE MGR IS NULL;
```

OUTPUT:

```
ENAME          MGR
----------    ----------
KING
```

20) List max sal,min sal,avg sal?

```
SQL>   SELECT MAX(SAL),MIN(SAL),AVG(SAL) FROM EMP;
```

OUTPUT:

```
  MAX(SAL)   MIN(SAL)     AVG(SAL)
  ----------  ----------     ----------------
    5000       800           2073.21429
```

21) List number of jobs?

```
SQL>   SELECT COUNT(DISTINCT(JOBS)) FROM EMP;
```

OUTPUT:

```
COUNT(DISTINCT(JOB))
--------------------
               5
```

22) List the numbers of people and average salary in deptno 30?

```
SQL>   SELECT COUNT(*),AVG(SAL) FROM EMP WHERE DEPTNO=30;
```

OUTPUT:

```
COUNT(*)   AVG(SAL)
---------   ----------
      6       1566.66667
```

23) List max sal and min sal in the designation SALESMAN and CLERK?

```
SQL>   SELECT COUNT(*),MAX(SAL),MIN(SAL) FROM EMP WHERE JOB
       IN(‘SALESMAN’,’CLERK’);
```

OUTPUT:

```
  COUNT(*)   MAX(SAL)   MIN(SAL)
----------   ----------   ----------
       8        1600         800
```

24) List the number of people and average salary of employees joined in 81,82 and 83?

```
SQL>   SELECT COUNT(*),AVG(SAL) FROM EMP WHERE TO_CHAR(HIREDATE,’YY’) IN
       (81,82,83);
```

OUTPUT:

```
  COUNT(*)   AVG(SAL)
----------   ----------
      13      2171.15385
```

25) List jobs that are unique to dept 20?

```
SQL>   SELECT DISTINCT(JOB) FROM EMP WHERE DEPTNO=20;
```

OUTPUT:

```
JOB
--------
ANALYST
CLERK
MANAGER
```

26) Display todays date and present time

```
SQL>   SELECT TO_CHAR(SYSDATE,’DD-MON-YYYY HH-MI-SS’) FROM DUAL;
```

OUTPUT:

```
TO_CHAR(SYSDATE,'DD-
--------------------
22-mar-2012 10-58-08
```

27) Display a given date as a string in different formats.

```
SQL>   SELECT TO_CHAR(SYSDATE,’DDSPTH MONTH YEAR’) FROM DUAL;
```

OUTPUT:

```
TO_CHAR(SYSDATE,'DDSPTHMONTHYEAR')
-----------------------------------------------------------------
twenty-second march     twenty twelve
```

28) List employee names and there experience in years?

```
SQL>   SELECT ENAME,ROUND(MONTHS_BETWEEN(SYSDATE,HIREDATE)/12)) AS EXP
       FROM EMP;
```

OUTPUT:

```
ENAME           EXP
----------     ----------
KING             30
BLAKE            31
CLARK            31
```

29) List empnames who joined in DEC and on MONDAY or FRIDAY?

```
SQL>   SELECT ENAME,TO_CHAR(HIREDATE,’MON’),TO_CHAR(HIREDATE,’DAY’) FROM
       EMP WHERE TO_CHAR(HIREDATE,’MON’)LIKE ‘DEC’ AND
       TO_CHAR(HIREDATE,’DAY’) IN (MONDAY,FRIDAY);
```

OUTPUT:

```
No row selected.
```

30) List ename and there joining date in the following formats
a)   SMITH 17th DEC NINETEEN EIGHTY

```
SQL>   SELECT ENAME ,TO_CHAR(HIREDATE,’DDTH MON YEAR’) FROM EMP WHERE
       ENAME LIKE ‘SMITH’;
```

OUTPUT:

```
ENAME      TO_CHAR(HIREDATE,'DDTHMONYEAR')
----------       --------------------------------------------------
SMITH      17th dec nineteen eighty
```

b)   Smith seventeenth dec nineteen eighty

```
SQL>   SELECT ENAME,TO _CHAR(HIREDATE,’DDSPTH MON YEAR’)  FROM EMP WHERE
       ENAME LIKE ‘SMITH’;
```

OUTPUT:

```
ENAME      TO_CHAR(HIREDATE,'DDSPTHMONYEAR')
---------- -      -----------------------------------------------------------
SMITH      seventeenth dec nineteen eighty
```

c) Smith weekday of joining

```
SQL>   SELECT ENAME, _CHAR(HIREDATE,’DAY’) FROM EMP WHERE ENAME LIKE
       ‘SMITH’;
```

OUTPUT:

```
ENAME      TO_CHAR(H
----------   ---------
SMITH        wednesday
```

d) Smith 17/12/1980

```
SQL>   SELECT ENAME, _CHAR(HIREDATE,’DD/MM/YYYY’) FROM EMP WHERE ENAME
       LIKE ‘SMITH’;
```

OUTPUT:

```
ENAME      TO_CHAR(HI
----------      ----------
SMITH           17/12/1980
```

LAB SHEET-2

1. **list employee names and their hire dates sorted in the order of their experience.**

```
SQL>   SELECT ENAME,HIREDATE,ROUND(MONTHS_BETWEEN(SYSDATE,HIREDATE)/12)
       EXP  FROM EMP ORDER BY ROUND(MONTHS_BETWEEN(SYSDATE,HIREDATE)/12);
```

OUTPUT:

```
 ENAME    HIREDATE     EXP
 ---------- --------- ---------- -----
 SCOTT     09-DEC-82       29
ADAMS     12-JAN-83        29
KING       17-NOV-81       30
```

2. **List managers names and their joining dates completely spelled in alphabetical order of names.**

```
SQL>   SELECT ENAME,JOB,TO_CHAR(HIREDATE,'DDSPTH MONTH YEAR') FROM EMP
       WHERE JOB='MANAGER' ORDER BY ENAME;
```

OUTPUT:

```
 ENAME     JOB          TO_CHAR(HIREDATE,'DDSPTHMONTHYEAR')
----------     ---------       -------- -------------------------------------------
BLAKE     MANAGER          first may nineteen eighty-one
CLARK     MANAGER          ninth june nineteen eighty-one
JONES      MANAGER          second april nineteen eighty-one
```

3. **List employee names and their experience in years with names arranged in descending order.**

```
SQL>   SELECT ENAME,ROUND(MONTHS_BETWEEN(SYSDATE,HIREDATE)/12) EXP FROM
       EMP  ORDER BY ENAME DESC;
```

OUTPUT:

```
 ENAME          EXP
----------         ----------
WARD            31
TURNER         31
SMITH            31
```

4. **List employee names having a minimum of 2 years experience sorted on experience.**

```
SQL>   SELECT ENAME,ROUND(MONTHS_BETWEEN(SYSDATE,HIREDATE)/12) EXP FROM
       EMP WHERE ROUND(MONTHS_BETWEEN(SYSDATE,HIREDATE)/12)>=2 ORDER BY
       ROUND(MONTHS_BETWEEN(SYSDATE,HIREDATE)/12);
```

OUTPUT:

```
ENAME          EXP
---------- -       ---------
SCOTT           29
ADAMS          29
KING             30
```

5. **List employee names with all capital letters, with all small letters and with first letter only as Capital.**

```
SQL>   SELECT ENAME,UPPER(ENAME),LOWER(ENAME),INITCAP(ENAME) FROM EMP;
```

OUTPUT:

```
ENAME     UPPER(ENAME) LOWER(ENAME) INITCAP(ENAME)
----------    --------------------  --------------------- ----------------------
KING         KING                 king                  King
BLAKE        BLAKE                 blake                 Blake
CLARK        CLARK                 clark                 Clark
```

6. **List employee names with length of the name sorted on length.**

```
SQL>   SELECT ENAME,LENGTH(ENAME) FROM EMP ORDER BY LENGTH(ENAME);
```

OUTPUT:

```
 ENAME     LENGTH(ENAME)
----------    --------------------------
KING              4
WARD              4
FORD              4
```

7. **List employee names appending Sri to the beginning and Garu to the end.**

```
SQL>   SELECT 'SRI'|| ENAME || 'GARU' FROM EMP;
```

OUTPUT:

```
 'SRI'||ENAME||'GARU’
------------------------------
sriKINGgaru
sriBLAKEgaru
sriCLARKgaru
```

8. **List employee names and month names of joining.**

```
SQL>   SELECT ENAME,TO_CHAR(HIREDATE,'MONTH') FROM EMP;
```

OUTPUT:

```
 ENAME     TO_CHAR(HIREDATE,’month’)
----------     -------------------------------------
KING         november
BLAKE       may
CLARK       june
```

9. **List employee names and year of joining in words.**

```
SQL>   SELECT ENAME,TO_CHAR(HIREDATE,'YEAR') FROM EMP;
```

OUTPUT:

```
ENAME      TO_CHAR(HIREDATE,'YEAR')
----------      --------------------------------------
KING            nineteen eighty-one
BLAKE           nineteen eighty-one
CLARK           nineteen eighty-one
```

10. **List employees names, job and salary with 5 hyphens in between.**

```
SQL>   SELECT ENAME||'------'||JOB||'------'||SAL FROM EMP;
```

OUTPUT:

```
 ENAME||'------'||JOB||'------'||SAL
--------------------------------------------------
KING------PRESIDENT------5000
BLAKE------MANAGER------2850
CLARK------MANAGER------2450
```

11. **List employee names and position of first occurrence of I in their name.**

```
SQL>   SELECT ENAME,INSTR(ENAME,’I’) FROM EMP;
```

OUTPUT:

```
  ENAME      INSTR(ENAME,'I')
----------      ----------------
KING                  2
BLAKE                 0
CLARK                 0
```

12. **List employee names and the string without first character and last character in their name.**

```
SQL>   SELECT ENAME,SUBSTR(ENAME,2,LENGTH(ENAME)-2) FROM EMP;
```

OUTPUT:

```
ENAME      SUBSTR(ENAME)
----------      ----------------------
KING            IN
BLAKE           LAK
CLARK           LAR
```

13. **List employees who joined between Apr 81 and Apr 82.**

```
SQL>   SELECT ENAME,TO_CHAR(HIREDATE,'MON YY') FROM EMP
       WHERE TO_DATE(HIREDATE) BETWEEN TO_DATE('01-APR-81') AND
        TO_DATE('30-APR-82');
```

OUTPUT:

```
 ENAME      TO_CHAR(HIREDATE,'MON YY)
----------   ----------------------------------------
KING           NOV 81
BLAKE          MAY 81
CLARK          JUN 81
```

14. **List max sal, min sal and average sal of depts. 10,30.**

```
SQL>   SELECT DEPTNO,MIN(SAL),MAX(SAL),AVG(SAL) FROM EMP WHERE
       DEPTNO IN(10,30) GROUP BY DEPTNO;
```

OUTPUT:

```
 DEPTNO   MIN(SAL)   MAX(SAL)   AVG(SAL)
-----------  --------------  ------------    -------------
 10       1300              5000      2916.66667
 30        950              2850     1566.66667
```

15. **List the designation in dept 30 but not in 20.**

```
SQL>   SELECT JOB FROM EMP WHERE DEPTNO=30 MINUS SELECT JOB FROM EMP WHERE
       DEPTNO=20;
```

OUTPUT:

```
 JOB
---------
SALESMAN
```

16. **List the number of employees ine ach department along with dept numbers.**

```
SQL>   SELECT DEPTNO,COUNT(*) FROM EMP GROUP BY DEPTNO;
```

OUTPUT:

```
 DEPTNO   COUNT(*)
 ------------   -------------
 10             3
 20             5
 30             6
```

17 . **List number of employees joined year wise.**

```
SQL>   SELECT TO_CHAR(HIREDATE,'YY'),COUNT(*) FROM EMP  GROUP BY
       TO_CHAR(HIREDATE,'YY');
```

OUTPUT:

```
   TO_CHAR(HIREDATE,’YY’)   COUNT(*)
 ---------------------------------  --------------
           80                          1
           81                          10
           82                          2
           83                          1
```

18. **List number of employees job wise.**

```
SQL>   SELECT JOB,COUNT(*) FROM EMP GROUP BY JOB;
```

OUTPUT:

```
    JOB        COUNT(*)
  ---------      --------------
 ANALYST            2
 CLERK              4
 MANAGER           3
 PRESIDENT         1
 SALESMAN          4
```

19. **List max sal, min sal, average salary dept wise.**

```
SQL>   SELECT DEPTNO,MIN(SAL),MAX(SAL),AVG(SAL) FROM EMP GROUP BY DEPTNO;
```

OUTPUT:

```
  DEPTNO   MIN(SAL)   MAX(SAL)   AVG(SAL)
 -------------    ------------   --------------  -------------
 10       1300       5000          2916.66667
 20        800       3000          2175
 30        950       2850          1566.66667
```

20. **List max sal, min sal, average salary job wise.**

```
SQL>   SELECT JOB,MIN(SAL),MAX(SAL),AVG(SAL) FROM EMP GROUP BY JOB;
```

OUTPUT:

```
   JOB        MIN(SAL)   MAX(SAL)   AVG(SAL)
   ---------      --------------   -------------  -----------
 ANALYST        3000       3000          3000
  CLERK          800        1300          1037.5
 MANAGER     2450       2975          2758.33333
 PRESIDENT    5000       5000           5000
 SALESMAN    1250       1600                1400
```

21. **List max sal, min sal for the jobs MANAGER and CLERK.**

```
SQL>   SELECT JOB,MIN(SAL),MAX(SAL) FROM EMP WHERE JOB IN('MANAGER','CLERK')
       GROUP BY JOB;
```

OUTPUT:

```
 JOB          MIN(SAL)   MAX(SAL)
 ---------       -------------  --------------
CLERK              800        1300
MANAGER        2450       2975
```

22. **List max sal, min sal AND average salary of the depts. Having a minimum 3 employees.**

```
SQL>   SELECT DEPTNO,MIN(SAL),MAX(SAL),COUNT(*) FROM EMP GROUP BY DEPTNO
       HAVING COUNT(*)>=2;
```

OUTPUT:

```
 DEPTNO   MIN(SAL)   MAX(SAL)   COUNT(*)
----------     --------------  -----------      -----------
10                   1300            5000         3
20                  800              3000         5
30                  950              2850         6
```

23. **List the number of employees in each job in each department.**

```
SQL>   SELECT DEPTNO,JOB,COUNT(*) FROM EMP GROUP BY DEPTNO,JOB;
```

OUTPUT:

```
 DEPTNO        JOB        COUNT(*)
------------      -------          -----------------
10                 CLERK                1
10                 MANAGER          1
10                 PRESIDENT         1
20                 CLERK                 2
```

24. **List MGR and the number of employees report to them in the sorted order.**

```
SQL>   SELECT MGR,COUNT(*) FROM EMP WHERE MGR IS NOT NULL
       GROUP BY MGR ORDER BY COUNT(*);
```

OUTPUT:

```
MGR   COUNT(*)
-------  -------------
7782        1
7788        1
7902        1
```

25. **List emp numbers of employees to whom a minimum of 3 people report.**

```
SQL>  SELECT MGR,COUNT(*) FROM EMP WHERE MGR IS NOT NULL GROUP BY MGR
      HAVING COUNT(*)>=3;
```

OUTPUT:

```
 MGR   COUNT(*)
---------- ----------
7698        5
7839        3
```

26. **List dept numbers having a minimum of 3 persons.**

```
SQL>  SELECT DEPTNO,COUNT(*) FROM EMP GROUP BY DEPTNO HAVING COUNT(*)>=3;
```

OUTPUT:

```
DEPTNO   COUNT(*)
----------     ----------
10                  3
20                 5
30                 6
```

27. **List names of jobs having a minimum of 3 persons in that job.**

```
SQL>  SELECT JOB,COUNT(*) FROM EMP GROUP BY JOB HAVING COUNT(*)>=3;
```

OUTPUT:

```
JOB        COUNT(*)
---------  ----------
CLERK              4
MANAGER            3
SALESMAN           4
```

28. **List names of months in which a minimum of 3 persons joined.**

```
SQL>  SELECT TO_CHAR(HIREDATE,'MONTH'),COUNT(*) FROM EMP GROUP BY
      TO_CHAR(HIREDATE,'MONTH') HAVING COUNT(*)>=3;
```

OUTPUT:

```
  TO_CHAR(HIREDATE,’MONTH’)   COUNT(*)
-------------------------------------      ----------------
DECEMBER                                  4
```

29. **List hiredates of employees having 2 or more employees having the same hiredate.**

```
SQL>   SELECT HIREDATE,COUNT(*) FROM EMP GROUP BY HIREDATE HAVING
       COUNT(*)>=2;
```

OUTPUT:

```
 HIREDATE    COUNT(*)
-------------   ----------------
03-DEC-81         2
```

30. **List departments having minimum of 3 people having a minimum of 17 years of experience.**

```
SQL>   SELECT DEPTNO,ROUND(MONTHS_BETWEEN(SYSDATE,HIREDATE)/12)
       EXP,COUNT(*) FROM EMP  WHERE
       ROUND(MONTHS_BETWEEN(SYSDATE,HIREDATE)/12)>=23 GROUP BY DEPTNO,
       ROUND(MONTHS_BETWEEN(SYSDATE,HIREDATE)/12) HAVING COUNT(*)>=3;
```

OUTPUT:

```
DEPTNO        EXP   COUNT(*)
--------------     ------ ----------
 30                31         4
```

LAB SHEET-III

1. **List employee names and dept names with which they are associated.**

```
SQL>    SELECT ENAME,DNAME  FROM  EMP,DEPT  WHERE
        EMP.DEPTNO=DEPT.DEPTNO;
```

OUTPUT:

```
ENAME      DNAME
---------- --------------
KING       ACCOUNTING
BLAKE      SALES
CLARK      ACCOUNTING
JONES      RESEARCH
```

2. **List employee names, salary and their grade.**

```
SQL>    SELECT ENAME,SAL,GRADE FROM EMP,SALGRADE WHERE SAL BETWEEN
        LOSAL AND HISAL;
```

OUTPUT:

```
ENAME          SAL      GRADE
----------     ---------- ----------
JAMES          950        1
SMITH          800        1
ADAMS          1100       1
MARTIN         1250       2
KING           5000       5
```

3. **List employee name, dept name along with grade.**

```
SQL>    SELECT  E1.ENAME,E2.ENAME  FROM EMP E1,EMP E2 WHERE
        E1.MGR=E2.EMPNO;
```

OUTPUT:

```
ENAME      DNAME             GRADE
---------- --------------    ----------
JAMES      SALES                   1
SMITH      RESEARCH                1
ADAMS      RESEARCH                1
MARTIN     SALES                   2
```

4. **List employee names and their manager names.**

```
SQL>  SELECT ENAME,DNAME,GRADE FROM EMP,DEPT,SALGRADE WHERE
      EMP.DEPTNO=DEPT.DEPTNO AND SAL BETWEEN LOSAL AND HISAL;
```

OUTPUT:

```
ENAME           SAL     GRADE
----------     ----------    ----------
JAMES           950         1
SMITH           800         1
ADAMS          1100         1
MARTIN         1250         2
```

5. **List dept name and Manager name.**

```
SQL>  SELECT E1.ENAME,E2.ENAME,DNAME FROM EMP E1,EMP E2,DEPT WHERE
      E1.DEPTNO=DEPT.DEPTNO AND E1.MGR=E2.EMPNO;
```

OUTPUT:

```
ENAME     ENAME     DNAME
----------   ----------         --------------
BLAKE     KING        SALES
CLARK     KING       ACCOUNTING
JONES      KING       RESEARCH
MARTIN    BLAKE      SALES
```

6. **List managers of various depts.. Along with grade sorted on grade.**

```
SQL>  SELECT ENAME,JOB,DEPTNO,GRADE FROM EMP,SALGRADE WHERE
      JOB='MANAGER' AND SAL BETWEEN LOSAL AND HISAL ORDER BY GRADE;
```

OUTPUT:

```
ENAME     JOB             DEPTNO     GRADE
----------    ---------            ----------       ----------
BLAKE     MANAGER        30           4
CLARK     MANAGER        10           4
JONES      MANAGER        20           4
```

7. **List employees having commission along with grade.**

```
SQL>    SELECT ENAME,COMM,GRADE FROM EMP,SALGRADE WHERE COMM IS NOT
        NULL AND SAL BETWEEN LOSAL AND HISAL;
```

OUTPUT:

```
ENAME          COMM      GRADE
----------     ----------  ----------
MARTIN         1400        2
WARD           500         2
ALLEN          300         3
TURNER           0         3
```

8. **List employees names with job manager along their manager names to whom they have to report.**

```
SQL>    SELECT E1.ENAME,E1.JOB,E2.ENAME FROM EMP E1,EMP E2 WHERE
        E1.JOB='MANAGER' AND E1.MGR=E2.EMPNO;
```

OUTPUT:

```
ENAME     JOB           ENAME
----------    ---------         ----------
BLAKE     MANAGER   KING
CLARK     MANAGER   KING
JONES     MANAGER   KING
```

9. **List names of employees who are working in the same dept of their manager.**

```
SQL>    SELECT E1.ENAME,E1.DEPTNO,E2.ENAME,E2.DEPTNO FROM EMP E1,EMP E2
        WHERE E1.MGR=E2.EMPNO AND E1.DEPTNO=E2.DEPTNO;
```

OUTPUT:

```
ENAME        DEPTNO ENAME        DEPTNO
----------        ----------   ----------        ----------
CLARK          10       KING          10
MARTIN         30       BLAKE         30
ALLEN          30       BLAKE         30
TURNER         30       BLAKE         30
```

10. List names of employees who are not working in the same dept of their managers.

```
SQL>    SELECT E1.ENAME,E1.DEPTNO,E2.ENAME,E2.DEPTNO FROM EMP E1,EMP E2
        WHERE E1.MGR=E2.EMPNO AND E1.DEPTNO<>E2.DEPTNO;
```

OUTPUT:

```
ENAME        DEPTNO ENAME        DEPTNO
----------   ----------  ---------    ----------
BLAKE           30       KING            10
JONES           20       KING            10
```

11. List names of employees having second character in their name second character in their dept name same.

```
SQL>    SELECT ENAM,DNAME FROM EMP,DEPT WHERE
        EMP.DEPTNO=DEPT.DEPTNO AND SUBSTR(ENAME,1,1)=SUBTR(DNAME,1,1);
```

OUTPUT:

```
 ENAME     DNAME
---------- --------------
MARTIN    SALES
JAMES     SALES
WARD      SALES
```

12. List employees who joined in the present month in any year and having grade and last digit in the year are same.

```
SQL>    SELECT  ENAME,GRADE,SUBSTR(TO_CHAR(HIREDATE,'YYYY'),4,1) FROM
        EMP,SALGRADE WHERE TO_CHAR('HIREDATE,'MONTH')=TO_CHAR
        (SYSDATE,'MONTH')  AND SAL BETWEEN LOSAL AND HISAL AND
        GRADE=SUBSTR(TO_CHAR(HIREDATE,'YYYY'),4,1);
```

OUTPUT:

```
No rows selected.
```

13. **List names of employees whose empno, mgr and grade given the same remainder when divided by 2.**

```
SQL>    SELECT ENAME,MOD(EMPNO,2),MOD(MGR,2),MOD(GRADE,2) FROM
        EMP,SALGRADE WHERE SAL BETWEEN LOSAL AND HISAL AND
        MOD(EMPNO,2)=MOD(MGR,2) AND MOD(MGR,2)=MOD(GRADE,2);
```

OUTPUT:

```
ENAME      MOD(EMPNO,2) MOD(MGR,2) MOD(GRADE,2)
---------- ------------ ---------- ------------
MARTIN                0          0            0
MILLER                0          0            0
FORD                  0          0            0
SCOTT                 0          0            0
```

14. **List the names of employees having grade and tens position in the deptno same.**

```
SQL>    SELECT ENAME,GRADE,SUBSTR(TO_CHAR(HIREDATE,'YYYY'),4,1),
        SUBSTR(DEPTNO,1,1) FROM EMP,SALGRADE WHERE SAL BETWEEN LOSAL
        AND HISAL AND GRADE=SUBSTR(TO_CHAR(HIREDATE,'YYYY'),4,1) AND
        SUBSTR(TO_CHAR(HIREDATE,'YYYY'),4,1)=SUBSTR(DEPTNO,1,1);
```

OUTPUT:

```
No rows selected
```

15. **List the names of employees having grade and tens position in the deptno same.**

```
SQL>    SELECT ENAME,GRADE,SUBSTR(DEPTNO,1,1) FROM EMP,SALGRADE
        WHERE SAL BETWEEN LOSAL AND HISAL AND
        GRADE=SUBSTR(DEPTNO,1,1);
```

OUTPUT:

```
ENAME      GRADE S
---------- ---------- -
ALLEN          3 3
TURNER         3 3
```

16. **List employee name,deptname and dept location of those employees having any of these three same length.**

```
SQL>    SELECT ENAME,DNAME,LOC FROM EMP,DEPT WHERE
        EMP.DEPTNO=DEPT.DEPTNO AND (LENGTH(ENAME)=LENGTH(DNAME) OR
        LENGTH(DNAME)=LENGTH(LOC) OR LENGTH(LOC)=LENGTH(ENAME));
```

OUTPUT:

```
ENAME     DNAME          LOC
---------- -------------- -------------
BLAKE     SALES          CHICAGO
ALLEN     SALES          CHICAGO
JAMES     SALES          CHICAGO
```

17. **List names of employees having month number of hiredate and grade same.**

```
SQL>    SELECT ENAME,TO_CHAR(HIREDATE,'MM'),GRADE FROM EMP,SALGRADE
        WHERE SAL BETWEEN LOSAL AND HISAL AND
        GRADE=TO_CHAR(HIREDATE,'MM');
```

OUTPUT:

```
ENAME     TO     GRADE
---------- -- ----------
ADAMS     01         1
WARD      02         2
JONES     04         4
```

18. **List names of clerks who are reporting to analyst.**

```
SQL>    SELECT E1.ENAME,E1.JOB,E2.ENAME,E2.JOB FROM EMP E1,EMP E2 WHERE
        E1.MGR=E2.EMPNO AND E1.JOB='CLERK' AND E2.JOB='ANALYST';
```

OUTPUT:

```
ENAME     JOB         ENAME      JOB
---------- ---------  ----------  ---------
SMITH     CLERK       FORD       ANALYST
ADAMS     CLERK       SCOTT      ANALYST
```

19. List emp names and their manager names having same grade.

```
SQL>    SELECT E1.ENAME,S1.GRADE,E2.ENAME,S2.GRADE FROM EMP E1,EMP E2,
        SALGRADE S1,SALGRADE S2 WHERE E1.MGR=E2.EMPNO AND E1.SAL
        BETWEEN S1.LOSAL AND S1.HISAL AND E2.SAL BETWEEN S2.LOSAL AND
        S2.HISAL AND S1.GRADE=S2.GRADE;
```

OUTPUT:

```
ENAME          GRADE ENAME          GRADE
----------      ---------- ----------      ----------
SCOTT            4       JONES            4
FORD             4       JONES            4
```

20. List emp names of employees who joined before their manager's joining date.

```
SQL>    SELECT E1.ENAME,E1.HIREDATE,E2.ENAME,E2.HIREDATE FROM EMP
        E1,EMP E2 WHERE E1.MGR=E2.EMPNO AND E1.HIREDATE<E2.HIREDATE;
```

OUTPUT:

```
ENAME      HIREDATE  ENAME      HIREDATE
----------   ---------    ----------   ---------
BLAKE      01-MAY-81 KING         17-NOV-81
CLARK      09-JUN-81  KING         17-NOV-81
JONES      02-APR-81  KING         17-NOV-81
ALLEN      20-FEB-81  BLAKE       01-MAY-81
```

LAB SHEET-4

1. **Get full details of all projects.**

```
SQL>  SELECT * FROM J;
```

OUTPUT:

```
JN JNAME      CITY
-- ---------- --------
J1 SORTER     PARIS
J2 DISPLAY   ROME
J3 OCR           ATHENS
J4 CONSOLE ATHENS
J5 RAID         LONDON
J6 EDC           OSLO
J7 TAPE          LONDON
```

2. **Get full details of all Projects in London.**

```
SQL>  SELECT * FROM J WHERE CITY='LONDON';
```

OUTPUT:

```
JN JNAME      CITY
 -- ---------- --------
 J5 RAID      LONDON
 J7 TAPE      LONDON
```

3. **Get supplier numbers for Suppliers who supply Project J1.**

```
SQL>  SELECT SNO FROM SPJ WHERE JNO='J1';
```

OUTPUT:

```
SNO
----------
S1
S2
S3
```

4. Get all shipments where the quantity is in the range 300 to 700.

```
SQL>  SELECT * FROM SPJ WHERE QTY BETWEEN 300 AND 700;
```

OUTPUT:

```
SNO        PNO        JNO       QTY
---------- ---------- ---------- ----------
S1         P1         J4              700
S2         P3         J1              400
S2         P3         J4              500
S2         P3         J5              600
S2         P3         J6              400
S3         P4         J2              500
S4         P6         J3              300
S4         P6         J7              300
S5         P5         J5              500
```

5. Get all part-color/part-city combinations.

```
SQL>  SELECT P1.PNO,P1.COLOR,P1.PNO,P2.CITY FROM P P1,P P2 WHERE
      P1.PNO=P2.PNO;
```

OUTPUT:

```
PNO   COLOR      PNO   CITY
----- ---------- ----- ----------
P1    RED        P1    LONDON
P2    GREEN      P2    PARIES
P3    BLUE       P3    ROME
P4    RED        P4    LONDON
P5    BLUE       P5    PARIES
P6    RED        P6    LONDON
```

6. Get all Supplier-number/part-number/Project-number triples such that the indicated Supplier part and Project are collocated.

```
SQL>  SELECT S.SNO,P.PNO,J.JNO FROM S,P,J,SPJ WHERE S.SNO=SPJ.SNO AND
      P.PNO=SPJ.PNO AND J.JNO=SPJ.JNO AND S.CITY=P.CITY AND P.CITY=J.CITY;
```

OUTPUT:

```
SNO   PNO   JNO
----- ----- -----
S4    P6    J7
```

7. Get all supplier-number/part-number/project-number triples such that the indicated supplier, part and project are not all collocated.

```
SQL>  SELECT S.SNO,P.PNO,J.JNO,S.CITY,P.CITY,J.CITY FROM S,P,J,SPJ WHERE
      S.SNO=SPJ.SNO AND P.PNO=SPJ.PNO AND J.JNO=SPJ.JNO AND
      S.CITY!=P.CITY AND P.CITY!=J.CITY AND J.CITY!=S.CITY;
```

```
OUTPUT:
SNO   PNO   JNO   CITY       CITY       CITY
----- ----- ----- ---------- ---------- ----------
S2    P3    J3    PARIES     ROME       ATHENS
S2    P3    J4    PARIES     ROME       ATHENS
S2    P3    J5    PARIES     ROME       LONDON
S2    P3    J6    PARIES     ROME       OSLO
S2    P3    J7    PARIES     ROME       LONDON
S3    P4    J2    PARIES     LONDON     ROME
S5    P2    J2    ATHENS     PARIES     ROME
S5    P5    J5    ATHENS     PARIES     LONDON
S5    P5    J7    ATHENS     PARIES     LONDON
S5    P6    J2    ATHENS     LONDON     ROME
```

8. Get all supplier-number/part-number/ project-number triples such that the indicated supplier, part and project are collocated.

```
SQL>  SELECT S.SNO,P.PNO,J.JNO,S.CITY,P.CITY,J.CITY FROM S,P,J,SPJ WHERE
      SPJ.SNO=S.SNO AND SPJ.PNO=P.PNO AND SPJ.JNO=J.JNO AND
      (S.CITY=P.CITY OR P.CITY=J.CITY OR J.CITY=S.CITY);
```

OUTPUT:

```
SNO   PNO   JNO   CITY       CITY       CITY
----- ----- ----- ---------- ---------- ----------
S1    P1    J1    LONDON     LONDON     PARIES
S1    P1    J4    LONDON     LONDON     ATHENS
S2    P3    J1    PARIES     ROME       PARIES
S2    P3    J2    PARIES     ROME       ROME
S2    P5    J2    PARIES     PARIES     ROME
S3    P3    J1    PARIES     ROME       PARIES
S4    P6    J3    LONDON     LONDON     ATHENS
S4    P6    J7    LONDON     LONDON     LONDON
S5    P2    J4    ATHENS     PARIES     ATHENS
S5    P1    J4    ATHENS     LONDON     ATHENS
S5    P3    J4    ATHENS     ROME       ATHENS
```

9. Get part numbers for parts supplied by a supplier in London.

```
SQL> SELECT DISTINCT P.PNO,S.CITY FROM S,P,SPJ WHERE S.SNO=SPJ.SNO AND
     P.PNO=SPJ.PNO AND S.CITY='LONDON';
```

OUTPUT:

```
  PNO   CITY
----- ----------
P1    LONDON
P6    LONDON
```

10. Get part numbers for parts supplied by a supplier in London to a project in London.

```
SQL> SELECT DISTINCT(SPJ.SNO),SPJ.PNO,J.JNO,S.CITY,J.CITY FROM SPJ,S,P,J
     WHERE J.JNO=SPJ.JNO AND S.CITY='LONDON' AND J.CITY='LONDON';
```

OUTPUT:

```
SNO        PNO        JNO   CITY       CITY
---------- ---------- ----- ---------- ----------
S2         P3         J5    LONDON     LONDON
S2         P3         J7    LONDON     LONDON
S4         P6         J7    LONDON     LONDON
S5         P5         J5    LONDON     LONDON
S5         P5         J7    LONDON     LONDON
```

11. Get all pairs of city names such that a supplier in the first city supplies a project in the second city.

```
SQL> SELECT DISTINCT S.CITY,J.CITY FROM S,J,SPJ WHERE S.CITY!=J.CITY AND
     SPJ.SNO=S.SNO AND SPJ.JNO=J.JNO;
```

OUTPUT:

```
CITY       CITY
---------- ----------
ATHENS     LONDON
ATHENS     ROME
LONDON     ATHENS
LONDON     PARIES
PARIES     ATHENS
PARIES     LONDON
PARIES     OSLO
PARIES     ROME
```

12. Get part numbers for parts supplied to any project by a supplier in the same city as that project.

```
SQL> SELECT  P.PNO,S.CITY,J.CITY FROM S,J,P,SPJ WHERE S.CITY=J.CITY AND
     SPJ.SNO=S.SNO AND SPJ.JNO=J.JNO AND SPJ.PNO=P.PNO;
```

OUTPUT:

```
PNO   CITY       CITY
----- ---------- ----------
P3    PARIES     PARIES
P3    PARIES     PARIES
P6    LONDON     LONDON
P2    ATHENS     ATHENS
P1    ATHENS     ATHENS
P3    ATHENS     ATHENS
P4    ATHENS     ATHENS
```

13. Get project numbers for projects supplied by atelast one supplier not in the same city.

```
SQL> SELECT  J.JNO,S.CITY,J.CITY FROM S,J,SPJ WHERE S.CITY!=J.CITY AND
     SPJ.SNO=S.SNO AND SPJ.JNO=J.JNO;
```

OUTPUT:

```
JNO   CITY       CITY
----- ---------- ----------
J1    LONDON     PARIES
J4    LONDON     ATHENS
J2    PARIES     ROME
J3    PARIES     ATHENS
J4    PARIES     ATHENS
J5    PARIES     LONDON
J6    PARIES     OSLO
J7    PARIES     LONDON
J2    PARIES     ROME
J2    PARIES     ROME
J3    LONDON     ATHENS
```

14. Get all pairs of part numbers such that some supplier supplies both the indicated parts.

```
SQL> SELECT DISTINCT SNO,PNO FROM SPJ GROUP BY SNO,PNO HAVING
     COUNT(*)>=2;
```

OUTPUT:

```
SNO        PNO
---------- ----------
S1         P1
S2         P3
```

15. Get the total number of projects supplied by supplier S1.

```
SQL>  SELECT COUNT(JNO) FROM SPJ WHERE SNO='S1';
```

OUTPUT:

```
COUNT(DISTINCT(JNO))
--------------------
                   2
```

16. Get the total quantity of part P1 supplied by suppliers S1.

```
SQL>  SELECT SUM(QTY) FROM SPJ WHERE SNO='S1' AND PNO='P1';
```

OUTPUT:

```
  SUM(QTY)
----------
       900
```

17. For each part being supplied to some project get the part number, the project numbers and the corresponding total quantity.

```
SQL>  SELECT P.PNO,SUM(QTY) FROM P,SPJ WHERE P.PNO=SPJ.PNO  GROUP BY
      P.PNO;
```

OUTPUT:

```
PNO     SUM(QTY)
----- ----------
P1          1000
P2           300
P3          3500
P4          1300
P5           700
P6           800
```

18. Get part numbers of parts supplied to some project in an average quantity of more than 320.

```
SQL>  SELECT P.PNO,AVG(QTY) FROM P,SPJ WHERE P.PNO=SPJ.PNO GROUP BY
      P.PNO HAVING AVG(QTY)>320;
```

OUTPUT:

```
PNO     AVG(QTY)
----- ----------
P1    333.333333
P3    388.888889
P4           650
```

19. Get project names for projects supplied by supplier S1.

```
SQL>  SELECT JNAME FROM SPJ,J WHERE J.JNO=SPJ.JNO AND SNO='S1';
```

OUTPUT:

```
JNAME
----------
SORTER
CONSOLE
```

20. Get colors of parts supplied by supplier S1.

```
SQL>  SELECT DISTINCT(P.COLOR) FROM P,SPJ WHERE P.PNO=SPJ.PNO AND
      SNO='S1';
```

OUTPUT:

```
COLOR
----------
RED
```

21. Get parts numbers for parts supplied to any project in London.

```
SQL>  SELECT DISTINCT P.PNO,J.CITY FROM P,J,SPJ WHERE P.PNO=SPJ.PNO AND
      J.JNO=SPJ.JNO AND J.CITY='LONDON';
```

OUTPUT:

```
PNO   CITY
----- ----------
P3    LONDON
P5    LONDON
P6    LONDON
```

22. Get project numbers for projects using atleast one part available from supplier S1.

```
SQL>  SELECT J.JNO FROM J,SPJ WHERE J.JNO=SPJ.JNO AND SPJ.SNO='S1';
```

OUTPUT:

```
JNO
-----
J1
J4
```

23. **Get supplier numbers for suppliers supplying atleast one part supplied by atelast one supplier who supplies atleast one red part.**

```
SQL>  SELECT DISTINCT S.SNO,P.PNO,P.COLOR FROM S,P,SPJ WHERE
      S.SNO=SPJ.SNO AND P.PNO=SPJ.PNO AND P.COLOR='RED';
```

OUTPUT:

```
SNO   PNO   COLOR
----- ----- ----------
S1    P1    RED
S3    P4    RED
S4    P6    RED
S5    P1    RED
S5    P4    RED
S5    P6    RED
```

24. **Get supplier numbers for suppliers with a status lower than that of supplier S1.**

```
SQL>  SELECT SNO FROM S WHERE STATUS<(SELECT STATUS FROM S WHERE
      SNO='S1');
```

OUTPUT:

```
SNO
-----
S2
```

25. **Get project numbers for projects whose city is first in the alphabetic list of**

```
SQL>  SELECT JNO FROM J WHERE CITY IN(SELECT MIN(CITY) FROM J);
```

OUTPUT:

```
JNO
-----
J3
J4
```

LAB SHEET-5

CREATING TABLES

....

```
SQL>    CREATE TABLE EMPLOYEE2(FNAME VARCHAR(10)NOT NULL,MINIT
        VARCHAR(10),LNAME VARCHAR(10)NOT NULL,SSN NUMBER(4)PRIMARY KEY,SEX
        VARCHAR(1),SALARY NUMBER(5),SUPERSSN NUMBER(4),DNO NUMBER(1));

SQL>   CREATE TABLE  DEPARTMENT2(DNAME  VARCHAR(15) UNIQUE,DNUMBER
        NUMBER(1)PRIMARY KEY,MGRSSN NUMBER(4)NOT NULL,FOREIGN KEY(MGRSSN)
        REFERENCES EMPLOYEE2(SSN));

SQL>   CREATE TABLE  DEPT_LOCATION(DNUMBER  NUMBER(1),DLOCATION
        VARCHAR(15),PRIMARY KEY(DNUMBER,DLOCATION),FOREIGN KEY(DNUMBER)
        REFERENCES DEPARTMENT2(DNUMBER));

SQL>    CREATE  TABLE PROJECT (PNAME VARCHAR(15)UNIQUE,PNUMBER
        NUMBER(2)PRIMARY KEY,PLOCATION VARCHAR(15),DNUM NUMBER(1)NOT
        NULL,FOREIGN KEY(DNUM) REFERENCES DEPARTMENT2(DNUMBER));

SQL>   CREATE TABLE WORKS_ON(ESSN NUMBER(4)NOT NULL,PNO NUMBER(2)NOT
        NULL,HOURS NUMBER(2,1),PRIMARY KEY(ESSN,PNO),FOREIGN KEY(ESSN)
        REFERENCES EMPLOYEE2(SSN),FOREIGN KEY(PNO) REFERENCES
        PROJECT(PNUMBER));

SQL>   CREATE TABLE  DEPENDENT(ESSN NUMBER(4)NOT NULL,D_NAME
        VARCHAR(15)NOT NULL,SEX VARCHAR(1),RELATIONSHIP VARCHAR(15),PRIMARY
        KEY(ESSN,D_NAME),FOREIGN KEY(ESSN) REFERENCES EMPLOYEE2(SSN));

SQL>   ALTER TABLE EMPLOYEE2 ADD  FOREIGN KEY (SUPERSSN) REFERENCES
        EMPLOYEE2 (SSN);

SQL>   ALTER TABLE EMPLOYEE2 ADD FOREIGN KEY(DNO) REFERENCES
        DEPARTMENT2(DNUMBER);
```

INSERTING VALUES INTO EMPLOYEE

```
SQL>    INSERT INTO E++MPLOYEE2(FNMAE,MINIT,LNAME,SSN,SEX,SALARY) VALUES
        ('JOHN','B','SMITH',2345,'M',30000);

SQL>    INSERT INTO EMPLOYEE2(FNAME,MINIT,LNAME,SSN,SEX,SALARY) VALUES
        ('FRANKLIN','T','WONG',3344,'M',40000);

SQL>    INSERT INTO EMPLOYEE2(FNAME,MINIT,LNAME,SSN,SEX,SALARY) VALUES
        ('ALICIA','J','ZELAYA',9988,'F',25000);

SQL>    INSERT INTO EMPLOYEE2(FNAME,MINIT,LNAME,SSN,SEX,SALARY) VALUES
        ('JENNIFER','S','WALLACE',8765,'F',43000);

SQL>    INSERT INTO EMPLOYEE2(FNAME,MINIT,LNAME,SSN,SEX,SALARY) VALUES
        ('RAMESH','K','NARAYANA',6688,'M',38000);

SQL>    INSERT INTO EMPLOYEE2(FNAME,MINIT,LNAME,SSN,SEX,SALARY) VALUES
        ('JOYCE','A','ENGLISH',5345,'F',25000);

SQL>    INSERT INTO EMPLOYEE2(FNAME,MINIT,LNAME,SSN,SEX,SALARY) VALUES
        ('AHMAD','V','JABBER',8798,'M',25000);

SQL>    INSERT INTO EMPLOYEE2(FNAME,MINIT,LNAME,SSN,SEX,SALARY) VALUES
        ('JAMES','E','BORG',8866,'M',55000);
```

INSERTING VALUES INTO DEPARTMENT

```
SQL>    INSERT INTO  DEPARTMENT2(DNAME,DNUMBER,MGRSSN) VALUES
        ('RESEARCH',5,3344);

SQL>    INSERT INTO  DEPARTMENT2(DNAME,DNUMBER,MGRSSN) VALUES
        ('ADMINISTRATION',4,8765);

SQL>    INSERT INTO  DEPARTMENT2(DNAME,DNUMBER,MGRSSN) VALUES
        ('HEADQUATERS',1,8866);
```

INSERTING VALUES INTO DEPT_LOCATION

```
SQL>    INSERT INTO  DEPT_LOCATION(DNUMBER,DLOCATION) VALUES (1,'HOUSTON');

SQL>   INSERT INTO  DEPT_LOCATION(DNUMBER,DLOCATION) VALUES (4,'STAFFORD');

SQL>   INSERT INTO  DEPT_LOCATION(DNUMBER,DLOCATION) VALUES (5,'BELLARIE');

SQL>   INSERT INTO  DEPT_LOCATION(DNUMBER,DLOCATION) VALUES (5,'SUGARLAND');

SQL>   INSERT INTO  DEPT_LOCATION(DNUMBER,DLOCATION) VALUES (5,'HOUSTON');
```

INSERTING VALUES INTO WORKS_ON

```
SQL>   INSERT INTO WORKS_ON(ESSN,PNO,HOURS) VALUES (2345,2,7.5);

SQL>   INSERT INTO WORKS_ON(ESSN,PNO,HOURS) VALUES (6688,3,40);

SQL>   INSERT INTO WORKS_ON(ESSN,PNO,HOURS) VALUES (5345,1,20);

SQL>   INSERT INTO WORKS_ON(ESSN,PNO,HOURS) VALUES (5345,2,20);

SQL>   INSERT INTO WORKS_ON(ESSN,PNO,HOURS) VALUES (3344,2,10);

SQL>   INSERT INTO WORKS_ON(ESSN,PNO,HOURS) VALUES (3344,3,10);

SQL>   INSERT INTO WORKS_ON(ESSN,PNO,HOURS) VALUES (3344,10,10);

SQL>   INSERT INTO WORKS_ON(ESSN,PNO,HOURS) VALUES (3344,20,20);

SQL>   INSERT INTO WORKS_ON(ESSN,PNO,HOURS) VALUES (9988,30,30);

SQL>   INSERT INTO WORKS_ON(ESSN,PNO,HOURS) VALUES (9988,10,10);

SQL>   INSERT INTO WORKS_ON(ESSN,PNO,HOURS) VALUES (8798,10,35);

SQL>   INSERT INTO WORKS_ON(ESSN,PNO,HOURS) VALUES (8798,20,5);

SQL>   INSERT INTO WORKS_ON(ESSN,PNO,HOURS) VALUES (8765,20,20);

SQL>   INSERT INTO WORKS_ON(ESSN,PNO,HOURS) VALUES (8765,30,15);
SQL>    INSERT INTO WORKS_ON(ESSN,PNO) VALUES (8866,30);

SQL>    INSERT INTO WORKS_ON(ESSN,PNO) VALUES (8866,1);
```

....

INSERTING VALUES INTO PROJECT

....

```
SQL>    INSERT INTO PROJECT(PNAME,PNUMBER,PLOCATION,DNUM)
        VALUES('PRODUCT_Y',2,'SUGARLAND',5);

SQL>    INSERT INTO PROJECT(PNAME,PNUMBER,PLOCATION,DNUM)
        VALUES('PRODUCT_Z',3,'HOUSTON',5);

SQL>    INSERT INTO PROJECT(PNAME,PNUMBER,PLOCATION,DNUM)
        VALUES('COMPUTERIZATION',10,'STAFFORD',4);

SQL>    INSERT INTO PROJECT(PNAME,PNUMBER,PLOCATION,DNUM)
        VALUES('REORGANIZATION',20,'HOUSTON',1);

SQL>    INSERT INTO PROJECT(PNAME,PNUMBER,PLOCATION,DNUM)
        VALUES('NEWBENEFITS',30,'STAFFORD',4);
```

....

INSERTING VALUES INTO DEPENDENT

....

```
SQL>    INSERT INTO DEPENDENT(ESSN,D_NAME,SEX,RELATIONSHIP) VALUES
        (3344,'ALICE','F','DAUGHTRER');

SQL>    INSERT INTO DEPENDENT(ESSN,D_NAME,SEX,RELATIONSHIP) VALUES
        (3344,'THEODORE','M','SON');

SQL>    INSERT INTO DEPENDENT(ESSN,D_NAME,SEX,RELATIONSHIP) VALUES
        (3344,'JOY','F','SPOUSE');

SQL>    INSERT INTO DEPENDENT(ESSN,D_NAME,SEX,RELATIONSHIP) VALUES
        (8765,'ABNER','M','SPOUSE');

SQL>    INSERT INTO DEPENDENT(ESSN,D_NAME,SEX,RELATIONSHIP) VALUES
        (2345,'MICHAEL','M','SON');

SQL>    INSERT INTO DEPENDENT(ESSN,D_NAME,SEX,RELATIONSHIP) VALUES
        (2345,'ALICE','F','DAUGHTER');
```

```
SQL>   INSERT INTO DEPENDENT(ESSN,D_NAME,SEX,RELATIONSHIP) VALUES
       (2345,'ELIZABETH','F','SPOUSE');
```

………………………………………………………………………

UPDATE EMPLOYEE

………………………………………………………………………

```
SQL>   UPDATE EMPLOYEE2 SET SUPERSNN=3344,DNO=5 WHERE SSN=2345;

SQL>   UPDATE EMPLOYEE2 SET SUPERSSN=8866,DNO=5 WHERE SSN=3344;

SQL>   UPDATE EMPLOYEE2 SET SUPERSSN=8765,DNO=4 WHERE SSN=9988;

SQL>   UPDATE EMPLOYEE2 SET SUPERSSN=8866,DNO=4 WHERE SSN=8765;

SQL>   UPDATE EMPLOYEE2 SET SUPERSSN=3344,DNO=5 WHERE SSN=6688;

SQL>   UPDATE EMPLOYEE2 SET SUPERSSN=3344,DNO=5 WHERE SSN=5345;

SQL>   UPDATE EMPLOYEE2 SET SUPERSSN=8765,DNO=4 WHERE SSN=8798;

SQL>   UPDATE EMPLOYEE2 SET SUPERSSN=NULL,DNO=1 WHERE SSN=8866;
```

QUERIES

1.Retrieve names of Employees who work for the Research Department.

```
SQL>   SELECT FNAME,MINIT,LNAME  FROM EMPLOYEE2 WHERE DNO IN(SELECT
       DNUMBER FROM DEPARTMENT2 WHERE DNAME='RESEARCH');
```

OUTPUT:

```
FNAME      MINIT      LNAME
---------- ---------- ----------
FRANKLIN   T         WONG
RAMESH      K         NARAYANA
JOYCE          A         ENGLISH
JOHN            B         SMITH
```

2.**For each project located in Stafford, list the project number, controlling department number and the department manager's last name**

```
SQL>   SELECT P.PNUMBER,P.DNUM,E.LNAME FROM EMPLOYEE2 E,PROJECT P WHERE
       P.PLOCATION='STAFFORD' AND P.DNUM=E.DNO;
```

OUTPUT:

```
 PNUMBER       DNUM LNAME
---------- ---------- ----------
        10          4 JABBER
        10          4 WALLACE
        10          4 ZELAYA
        30          4 JABBER
        30          4 WALLACE
        30          4 ZELAYA
```

3.**For each Employee retrieve the employee's first and last name of his/her supervisor.**

```
SQL>   SELECT (E.FNAME)EMPLOYEE_FNAME,(E.LNAME)EMPLOYEE_LNAME,(A.FNAME)
       SUPERIOR_FNAME,(A.LNAME)SUPERIOR_LNAME FROM EMPLOYEE2
       E,EMPLOYEE2 A WHERE E.SUPERSSN=A.SSN;
```

OUTPUT:

```
EMPLOYEE_F EMPLOYEE_L SUPERIOR_F SUPERIOR_L
----------        ----------        ----------        ----------
FRANKLIN      WONG            JAMES            BORG
ALICIA           ZELAYA         JENNIFER       WALLACE
JENNIFER       WALLACE      JAMES            BORG
RAMESH         NARAYANA   FRANKLIN       WONG
```

4.List all project number for projects that involve an Employee whose last name is SMITH, either as a worker or as a MANAGER of the Department that controls the project.

```
SQL>    SELECT SSN,LNAME,PNO FROM EMPLOYEE2,WORKS_ON,DEPARTMENT2 WHERE
        LNAME='SMITH'AND DNUMBER=DNO AND ESSN=SSN OR SUPERSSN=SSN;
```

OUTPUT:

```
       SSN LNAME             PNO
---------- ---------- ----------
      2345 SMITH               1
      2345 SMITH               2
```

5.Retrieve the list of Employees and the projects they are working on order by department and with in each department ordered Alphabetically by last name and first name.

```
SQL>    SELECT FNAME,LNAME,PNO,PNAME,SSN,DNO FROM
        EMPLOYEE2,WORKS_ON,PROJECT WHERE SSN=ESSN AND PNUMBER=PNO ORDER
        BY DNO,FNAME,LNAME;
```

OUTPUT:

```
FNAME        LNAME          PNO PNAME                      SSN       DNO
----------   ---------- ---------- -------------- ---------------------------- ----------
JAMES        BORG          30 NEWBENEFITS             8866        1
JAMES        BORG          1 PRODUCT_X                8866        1
AHMAD        JABBER        10 COMPUTERIZATION         8798        4
AHMAD        JABBER        20 REORGANIZATION          8798        4
ALICIA       ZELAYA        30 NEWBENEFITS             9988        4
ALICIA       ZELAYA        10 COMPUTERIZATION         9988        4
JENNIFER     WALLACE       20 REORGANIZATION          8765        4
JENNIFER     WALLACE       30 NEWBENEFITS             8765        4
FRANKLIN     WONG          2 PRODUCT_Y                3344        5
FRANKLIN     WONG          3 PRODUCT_Z                3344        5
FRANKLIN     WONG          10 COMPUTERIZATION         3344        5
```

6.Retrieve the name of each employee who has the dependent with the same first name and same sex as the employee.

```
SQL>    SELECT FNAME,D_NAME,EMPLOYEE2.SEX,DEPENDENT.SEX FROM EMPLOYEE2,
        DEPENDENT WHERE ESSN=SSN AND FNAME=D_NAME AND
        EMPLOYEE2.SEX=DEPENDENT.SEX;
```

OUTPUT:

```
no rows selected
```

7.Retrieve the name of each employee who works on all the projects controlled by department no. 5

```
SQL>   SELECT FNAME,SSN,DNUM FROM EMPLOYEE2,WORKS_ON,PROJECT WHERE PNO
       IN(SELECT PNUMBER FROM PROJECT WHERE DNUM=5)   AND PNO=PNUMBER AND
       ESSN=SSN;
```

OUTPUT:

```
FNAME           SSN      DNUM
---------- -------------------- ----------
JOHN           2345         5
JOHN           2345         5
RAMESH      6688         5
JOYCE        5345         5
JOYCE        5345         5
FRANKLIN  3344         5
FRANKLIN  3344         5
JAMES        8866         5
```

8.Retrieve the names of employees who have no dependents

```
SQL>   SELECT FNAME,SSN FROM EMPLOYEE2 WHERE SSN NOT IN(SELECT
       DISTINCT(ESSN) FROM DEPENDENT);
```

OUTPUT:

```
FNAME           SSN
---------- ----------------
ALICIA          9988
RAMESH        6688
JOYCE           5345
AHMAD         8798
JAMES           8866
```

9.List the names of Managers who have at least one dependent.

```
SQL>   SELECT SSN,FNAME FROM EMPLOYEE2 WHERE SSN IN(SELECT DISTINCT(ESSN)
       FROM EMPLOYEE2,DEPENDENT WHERE ESSN=SUPERSSN) ;
```

OUTPUT:

```
   SSN     FNAME
---------- ----------
      3344 FRANKLIN
      8765 JENNIFER
```

10.For each project on which more than two employees worth, retrieve the project number, the project name, and the no.of employees who worth on the project.

```
SQL>   SELECT P.PNUMBER,COUNT(ESSN),P.PNAME FROM PROJECT P,WORKS_ON WHERE
       P.PNUMBER=PNO AND HOURS IS NOT NULL GROUP BY P.PNUMBER,P.PNAME
       HAVING COUNT(ESSN)>2;
```

OUTPUT:

```
 PNUMBER COUNT(ESSN) PNAME
---------- ----------- ------------------------
     2         3                   PRODUCT_Y
    10         3                   COMPUTERIZATION
    20         3                   REORGANIZATION
```

11.For each project, retrieve the project number, the project name and the no.of employees from department 5 who works on the project.

```
  SQL> SELECT PNO,COUNT(ESSN),DNO FROM WORKS_ON,EMPLOYEE2 WHERE DNO=5
AND HOURS IS NOT NULL AND ESSN=SSN;
```

OUTPUT:

```
 PNO COUNT(ESSN)      DNO
---------- ----------- ---------------
     1         2            5
     2         3            5
     3         2            5
    10          1           5
    20          1           5
```

12.For each department that has more than 5 employees, retrieve the department number and the no.of its employees who are making more than 40,000.

```
SQL>   SELECT DNO,COUNT(SSN) FROM EMPLOYEE2 WHERE SALARY>40000 GROUP BY
       DNO HAVING COUNT(SSN)>5;
```

OUTPUT:

```
no rows selected
```

PL/SQL

1) W rite a program to display welcome message.

```
SQL>
         BEGIN
         DBMS_OUTPUT.PUT_LINE('HAI');
         DBMS_OUTPUT.PUT_LINE('WELCOME');
         DBMS_OUTPUT.PUT_LINE('PL/SQL PROGRAMS');
         END;
         /
```

OUTPUT:

```
                    HAI
                    WELCOME
                    PL/SQL PROGRAMS

PL/SQL procedure successfully completed.
```

2) **write a program to find sum of two integer numbers.**

```
SQL>
        DECLARE
        A NUMBER;
        B NUMBER;
         C NUMBER;
         BEGIN
        A:=&A;
        B:=&B;
        C:=A+B;
        DBMS_OUTPUT.PUT_LINE('THE SUM OF TWO INTEGERS IS:  '||C);
        END;
         /
```

OUTPUT:

```
Enter value for a: 1
old   6: A:=&A;
new   6: A:=1;
Enter value for b: 2
old   7: B:=&B;
new   7: B:=2;
THE SUM OF TWO INTEGERS IS:  3

PL/SQL procedure successfully completed.
```

3) write a program to accept empno,ename,sal & calculate bonus on the following condition 20% on ann_sal.

```
SQL>
        DECLARE
        EMPNO NUMBER;
        ENAME VARCHAR2(20);
        SAL NUMBER(7,2);
        ANU_SAL NUMBER(10,2);
        BONUS NUMBER(10,2);
        BEGIN
        EMPNO:=&EMPNO;
        ENAME:='&ENAME';
        SAL:=&SAL;
        ANU_SAL:=SAL*12;
        BONUS:=ANU_SAL*20/100;
        DBMS_OUTPUT.PUT_LINE('EMPNO: '||EMPNO);
        DBMS_OUTPUT.PUT_LINE('ENAME: '||ENAME);
        DBMS_OUTPUT.PUT_LINE('SAL:  '||SAL);
        DBMS_OUTPUT.PUT_LINE('BONUS: '||BONUS);
        END;
        /
```

OUTPUT:

```
Enter value for empno: 10
old   8: EMPNO:=&EMPNO;
new   8: EMPNO:=10;
Enter value for ename: BALU
old   9: ENAME:='&ENAME';
new   9: ENAME:='BALU';
Enter value for sal: 15000
old  10: SAL:=&SAL;
new  10: SAL:=15000;
EMPNO: 10
ENAME: BALU
SAL:  15000
BONUS: 36000

PL/SQL procedure successfully completed.
```

4) write a program to accept product no,pname,quantity,price & calculate total,discount(20% on total),net bill.

```
SQL>
        DECLARE
        PRODNO NUMBER;
        PNAME VARCHAR2(20);
        QUAN NUMBER(3);
        PRICE NUMBER(7,2);
        TOTAL NUMBER(7,2);
        DISCOUNT NUMBER(7,2);
        NET NUMBER(7,2);
        BEGIN
        PRODNO:=&PRODNO;
        PNAME:='&PNAME';
        QUAN:=&QUAN;
        PRICE:=&PRICE;
        TOTAL:=QUAN*PRICE;
        DISCOUNT:=TOTAL*20/100;
        NET:=TOTAL-DISCOUNT;
        DBMS_OUTPUT.PUT_LINE('PRODNO: '||PRODNO);
        DBMS_OUTPUT.PUT_LINE('PNAME: '||PNAME);
        DBMS_OUTPUT.PUT_LINE('QUANTITY: '||QUAN);
        DBMS_OUTPUT.PUT_LINE('PRICE: '||PRICE);
        DBMS_OUTPUT.PUT_LINE('TOTAL: '||TOTAL);
        DBMS_OUTPUT.PUT_LINE('DISCOUNT: '||DISCOUNT);
        DBMS_OUTPUT.PUT_LINE('NET BALANCE: '||NET);
        END;
       /
```

OUTPUT:

```
Enter value for prodno: 110
old  10: PRODNO:=&PRODNO;
new  10: PRODNO:=110;
Enter value for pname: PEN
old  11: PNAME:='&PNAME';
new  11: PNAME:='PEN';
Enter value for quan: 12
old  12: QUAN:=&QUAN;
new  12: QUAN:=12;
Enter value for price: 10
old  13: PRICE:=&PRICE;
new  13: PRICE:=10;
PRODNO: 110
PNAME: PEN
QUANTITY: 12
PRICE: 10
TOTAL: 120
DISCOUNT: 24
NET BALANCE: 96

PL/SQL procedure successfully completed.
```

5) write a program to accept empno,sal,calculate bonus based on the following conditions

Salary	**Bonus**
>=10000	**20% on ann_sal**
>=5000&<10000	**15% on ann_sal**
>=3000&<5000	**12% on ann_sal**
>=1500&<3000	**10% on ann_sal**
>1500	**8% on ann_sal**

```
SQL>
        DECLARE
        EMPNO NUMBER;
        SAL NUMBER(7,2);
        ANU_SAL NUMBER(7,2);
        BONUS NUMBER(7,2);
        BEGIN
        EMPNO:=&EMPNO;
        SAL:=&SAL;
        ANU_SAL:=SAL*12;
        IF SAL>=10000 THEN
        BONUS:=ANU_SAL*20/100;
        ELSIF SAL>=5000 AND SAL<10000 THEN
        BONUS:=ANU_SAL*15/100;
        ELSIF SAL>=3000 AND SAL<5000 THEN
        BONUS:=ANU_SAL*12/100;
        ELSIF SAL>=1500 AND SAL<3000 THEN
        BONUS:=ANU_SAL*10/100;
        ELSE
        BONUS:=ANU_SAL*8/100;
        END IF;
        DBMS_OUTPUT.PUT_LINE('EMPNO: '||EMPNO);
        DBMS_OUTPUT.PUT_LINE('SAL: '||SAL);
        DBMS_OUTPUT.PUT_LINE('ANU_SAL: '||ANU_SAL);
        DBMS_OUTPUT.PUT_LINE('BONUS: '||BONUS);
        END;
       /
```

OUTPUT:

```
Enter value for empno: 10
old   7: EMPNO:=&EMPNO;
new   7: EMPNO:=10;
Enter value for sal: 1000
old   8: SAL:=&SAL;
new   8: SAL:=1000;
EMPNO: 10
SAL: 1000
ANU_SAL: 12000
BONUS: 960
```

PL/SQL procedure successfully completed.

```
/
Enter value for empno: 10
old   7: EMPNO:=&EMPNO;
new   7: EMPNO:=10;
Enter value for sal: 6000
old   8: SAL:=&SAL;
new   8: SAL:=6000;
EMPNO: 10
SAL: 6000
ANU_SAL: 72000
BONUS: 10800
```

PL/SQL procedure successfully completed.

6) wap to print numbers from 10-1.

```
SQL>
     DECLARE
     I NUMBER;
     BEGIN
     DBMS_OUTPUT.PUT_LINE('THE NUMBERS ARE');
     FOR I IN REVERSE 1..10 LOOP
     DBMS_OUTPUT.PUT_LINE(I);
     END LOOP;
     END;
    /
```

OUTPUT:

```
THE NUMBERS ARE
10
9
8
7
6
5
4
3
2
1
```

PL/SQL procedure successfully completed.

7) WAP to accept a date & print next 7 days along with day.

```
SQL>
          DECLARE
          DA DATE;
          I NUMBER;
          BEGIN
          DA:='&DA';
          FOR I IN 1..7 LOOP
          DBMS_OUTPUT.PUT_LINE('THE DATE IS:'||DA);
          DA:=DA+1;
          END LOOP;
         END;
        /
```

OUTPUT:

```
Enter value for da: 10-JAN-10
old   5: DA:='&DA';
new   5: DA:='10-JAN-10';
THE DATE IS:10-JAN-10
THE DATE IS:11-JAN-10
THE DATE IS:12-JAN-10
THE DATE IS:13-JAN-10
THE DATE IS:14-JAN-10
THE DATE IS:15-JAN-10
THE DATE IS:16-JAN-10

PL/SQL procedure successfully completed.
```

PL/SQL-CURSORS

1. wap to display emp details along with ann_sal & exp

```
SQL>
        DECLARE
        CURSOR C1 IS SELECT  * FROM EMP;
        V_EC  C1%ROWTYPE;
        ANN_SAL  NUMBER(8,2);
        EXP NUMBER(8,2);
        BEGIN
        OPEN C1;
        LOOP
        FETCH C1 INTO V_EC;
        EXIT WHEN C1%NOTFOUND;
        ANN_SAL := V_EC.SAL*12;
        EXP := MONTHS_BETWEEN(SYSDATE,V_EC.HIREDATE)/12;
        DBMS_OUTPUT.PUT_LINE('EMPNO=' ||V_EC.EMPNO);
        DBMS_OUTPUT.PUT_LINE('ANN_SAL=' ||ANN_SAL);
        DBMS_OUTPUT.PUT_LINE('EXP=' ||EXP);
        END LOOP;
        CLOSE C1;
        END;
        /
```

OUTPUT:

```
                empno=7839
                ann_sal=60000
                exp=30.37
                empno=7698
                ann_sal=34200
                exp=30.91
                empno=7782
                ann_sal=29400
                exp=30.81
                empno=7566
                ann_sal=35700
                exp=30.99
                empno=7654
                ann_sal=15000
                exp=30.5
                empno=7499
                ann_sal=19200
                exp=31.11
                empno=7844
                ann_sal=18000
                exp=30.56
                empno=7900
                ann_sal=11400
                exp=30.32
                empno=7521
                ann_sal=15000
                exp=31.1
                empno=7902
                ann_sal=36000
                exp=30.32
                empno=7369
                ann_sal=9600
                exp=31.28
                empno=7788
                ann_sal=36000
                exp=29.31
                empno=7876
                ann_sal=13200
                exp=29.21
                empno=7934
                ann_sal=15600
                exp=30.18

PL/SQL procedure successfully completed.
```

2. Wap to calc bonus for all emps insert into bonus table

```
SQL>  CREATE TABLE BONUS1(EMPNO NUMBER(5) PRIMARY KEY,BONUS_AMT
      NUMBER(10,3),ADD_AMT NUMBER(10,3),ISS_DATE DATE);

SQL>
       DECLARE
       CURSOR EC IS SELECT EMPNO,SAL FROM EMP;
       V_EC EC%ROWTYPE;
       ANN_SAL NUMBER(10,2);
       B BONUS1%ROWTYPE;
       BEGIN
       OPEN EC;
       LOOP
       FETCH EC INTO V_EC;
       EXIT WHEN EC%NOTFOUND;
       ANN_SAL :=V_EC.SAL*12;
       B.BONUS_AMT := ANN_SAL*0.2;
       INSERT INTO BONUS1(EMPNO,BONUS_AMT,ADD_AMT,ISS_DATE)
      VALUES(V_EC.EMPNO,B.BONUS_AMT,1000,SYSDATE
       END LOOP;
       CLOSE EC;
       END;
       /

PL/SQL procedure successfully completed.
```

OUTPUT:

```
SQL>   SELECT *FROM BONUS1;

              EMPNO  BONUS_AMT    ADD_AMT ISS_DATE
              ------------------------------- ---------- ---------- ---------
              7839     12000          1000           29-MAR-12
              7698      6840          1000           29-MAR-12
              7782      5880          1000           29-MAR-12
              7566      7140          1000           29-MAR-12
              7654      3300          1000           29-MAR-12
              7499      4224          1000           29-MAR-12
              7844      3960          1000           29-MAR-12
              7900      2280          1000           29-MAR-12
              7521      3300          1000           29-MAR-12
              7902      7200          1000           29-MAR-12
              7369      1920          1000           29-MAR-12
```

3. wap to display empno,ename,job,sal,deptno,dname,loc,grade of all mgrs.

```
SQL>
        DECLARE
        CURSOR EC IS SELECT E.EMPNO,E.JOB,E.SAL,D.DEPTNO,D.LOC,S.GRADE FROM
        EMP E,SALGRADE S,DEPT D WHERE E.JOB='MANAGER' AND E.DEPTNO=D.DEPTNO
        AND E.SAL BETWEEN S.LOSAL AND S.HISAL;
        V_EC EC%ROWTYPE;
        BEGIN
        OPEN EC;
        LOOP
        FETCH EC INTO V_EC;
        EXIT WHEN EC%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE(V_EC.EMPNO);
        DBMS_OUTPUT.PUT_LINE(V_EC.JOB);
        DBMS_OUTPUT.PUT_LINE(V_EC.SAL);
        DBMS_OUTPUT.PUT_LINE(V_EC.DEPTNO);
        DBMS_OUTPUT.PUT_LINE(V_EC.LOC);
        DBMS_OUTPUT.PUT_LINE(V_EC.GRADE);
        END LOOP;
        CLOSE EC;
        END;
        /
```

OUTPUT:

```
                7698
                MANAGER
                2850
                30
                CHICAGO
                4
                7782
                MANAGER
                2450
                10
                NEW YORK
                4
                7566
                MANAGER
                2975

                20
                DALLAS
                4

PL/SQL procedure successfully completed.
```

For Loop:

4. Wap to display dept details

```
SQL>
        DECLARE
        CURSOR EC IS SELECT * FROM DEPT;
        BEGIN
        FOR V_EC IN EC
        LOOP
        DBMS_OUTPUT.PUT_LINE('DEPTNO='||V_EC.DEPTNO);
        DBMS_OUTPUT.PUT_LINE('DNAME='||V_EC.DNAME);
        DBMS_OUTPUT.PUT_LINE('LOC='||V_EC.LOC);
        END LOOP;
        END;
       /
```

OUTPUT:

```
                    deptno=10
                    dname=ACCOUNTING
                    loc=NEW YORK
                    deptno=20
                    dname=RESEARCH
                    loc=DALLAS
                    deptno=30
                    dname=SALES
                    loc=CHICAGO
                    deptno=40
                    dname=OPERATIONS
                    loc=BOSTON

PL/SQL procedure successfully completed.
```

5. Wap to increment all emp sal by 10% who are working in grade 2&3.

```
SQL>
        DECLARE
        CURSOR EC IS SELECT E.EMPNO,E.ENAME,E.SAL,S.GRADE FROM EMP E,SALGRADE
        S WHERE S.GRADE IN(2,3) AND E.SAL BETWEEN S.LOSAL AND S.HISAL;
        BEGIN
        FOR V_EC IN EC
        LOOP
        UPDATE EMP SET SAL=V_EC.SAL+V_EC.SAL*0.1 WHERE EMPNO=V_EC.EMPNO;
        END LOOP;
        END;
        /
```

OUTPUT:

Before :

```
EMPNO ENAME            SAL      GRADE
---------- ---------- -------------------- ----------
      7654 MARTIN         1250           2
      7521 WARD           1250           2
      7934 MILLER         1300           2
      7499 ALLEN          1600           3
      7844 TURNER         1500           3
```

After:

```
 EMPNO ENAME            SAL      GRADE
---------- ---------- ---------- ------------------
      7654 MARTIN         1375            2
      7521 WARD           1375            2
      7499 ALLEN          1760            3
      7844 TURNER         1650            3
      7934 MILLER         1430            3
```