Employee Table

```
SQL> set linesize 200;
SQL> select * from emp;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL
COMM      DEPTNO
---------- ---------- --------- ---------- --------- ----------
      7839 KING       PRESIDENT            17-NOV-81       5000
10
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850
30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450
10
      7566 JONES      MANAGER         7839 02-APR-81       2975
20
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250
1400          30
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600
300          30
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500
0          30
      7900 JAMES      CLERK           7698 03-DEC-81        950
30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250
500          30
      7902 FORD       ANALYST         7566 03-DEC-81       3000
20
      7369 SMITH      CLERK           7902 17-DEC-80        800
20

     EMPNO ENAME      JOB              MGR HIREDATE         SAL
COMM      DEPTNO
---------- ---------- --------- ---------- --------- ---------- ----------
      7788 SCOTT      ANALYST         7566 09-DEC-82       3000
20
      7876 ADAMS      CLERK           7788 12-JAN-83       1100
20
      7934 MILLER     CLERK           7782 23-JAN-82       1300
10

14 rows selected
```

Worksheet-1

1.list all the empno,ename and salary from emp.

```
SQL> select empno,ename,sal from emp;

     EMPNO ENAME                 SAL
---------- ---------- ----------
      7839 KING                 5000
      7698 BLAKE                2850
      7782 CLARK                2450
      7566 JONES                2975
      7654 MARTIN               1250
      7499 ALLEN                1600
      7844 TURNER               1500
      7900 JAMES                 950
      7521 WARD                 1250
      7902 FORD                 3000
      7369 SMITH                 800

     EMPNO ENAME                 SAL
---------- ---------- ----------
      7788 SCOTT                3000
      7876 ADAMS                1100
      7934 MILLER               1300

14 rows selected.
```

2. list the names of all managers

```
SQL> select ename from emp where job='MANAGER';

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

3. list all clerks in deptno 30

```
SQL> select ename from emp where deptno=30 and job='CLERK';

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

4. list the employee to whome the manager is 7698

```
SQL> select ename from emp where mgr=7698 and job!='MANAGER';

ENAME
```

```
---------------
MARTIN
ALLEN
TURNER
JAMES
WARD
```

5. list the jobs in deptno 20

```
SQL> select job from emp where deptno=20;

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

6. list the employees whose salary is between 2000 and 3000

```
SQL> select ename from emp where sal>=2000 and sal<=3000;

ENAME
---------------
BLAKE
CLARK
JONES
FORD
SCOTT
```

7. list the employees in the department 10,20

```
SQL> select ename from emp where deptno=10 or deptno=20 ;

ENAME
---------------
KING
CLARK
JONES
FORD
SMITH
SCOTT
ADAMS
MILLER

8 rows selected.
```

8. list the employees whose names begin with 'S'

```
SQL>  select ename from emp where ename like 'S%';
```

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

9. list the employees having 'A' in their names

```
SQL> select ename from emp where ename like '%A%' ;

ENAME
---------------
BLAKE
CLARK
MARTIN
ALLEN
JAMES
WARD
ADAMS

7 rows selected.
```

10. list the employees who had joined in jan

```
SQL> select ename from emp where hiredate like '%JAN%' ;

ENAME
---------------
ADAMS
MILLER
```

11. list the employees who had joined in the year 81

```
SQL> select ename from emp where hiredate like '%81' ;

ENAME
---------------
KING
BLAKE
CLARK
JONES
MARTIN
ALLEN
TURNER
JAMES
WARD
FORD

10 rows selected.
```

12. list all the distint jobs

```
SQL> select distinct(job) from emp ;

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

13. list the employee names in alphabetical order

```
SQL> select ename from emp order by ename ;

ENAME
----------
ADAMS
ALLEN
BLAKE
CLARK
FORD
JAMES
JONES
KING
MARTIN
MILLER
SCOTT

ENAME
----------
SMITH
TURNER
WARD

14 rows selected.
```

14. list the employee names alphabetically departmentwise

```
SQL> select ename,deptno from emp order by deptno ;

ENAME          DEPTNO
---------------------------------------------------------------------
KING               10
CLARK              10
MILLER             10
JONES              20
SCOTT              20
ADAMS              20
SMITH              20
FORD               20
BLAKE              30
MARTIN             30
```

```
ALLEN                  30

ENAME              DEPTNO
-------------- ---------------
TURNER                 30
JAMES                  30
WARD                   30

14 rows selected.
```

15. list the employee names alphabetically jobwise

```
SQL> select ename,job from emp order by job ;

ENAME       JOB
-------------- -------------
FORD        ANALYST
SCOTT       ANALYST
JAMES       CLERK
SMITH       CLERK
MILLER      CLERK
ADAMS       CLERK
BLAKE       MANAGER
CLARK       MANAGER
JONES       MANAGER
KING        PRESIDENT
MARTIN      SALESMAN

ENAME       JOB
-------------- -------------
ALLEN       SALESMAN
WARD        SALESMAN
TURNER      SALESMAN

14 rows selected.
```

16. list empno,ename and sal with DA(15% sal)and PF(10% of sal)

```
SQL> select empno,ename,sal,(0.15*sal)da,(0.10*sal)pf from emp
;

     EMPNO ENAME              SAL         DA         PF
---------- ---------- ---------- ---------- ----------
      7839 KING              5000        750        500
      7698 BLAKE             2850      427.5        285
      7782 CLARK             2450      367.5        245
      7566 JONES             2975     446.25      297.5
      7654 MARTIN            1250      187.5        125
      7499 ALLEN             1600        240        160
      7844 TURNER            1500        225        150
      7900 JAMES                 950      142.5         95
      7521 WARD                 1250      187.5        125
      7902 FORD                 3000        450        300
      7369 SMITH                 800        120         80

     EMPNO ENAME                 SAL         DA         PF
---------- ---------- ---------- ---------- ----------
      7788 SCOTT                3000        450        300
      7876 ADAMS                1100        165        110
      7934 MILLER               1300        195        130

14 rows selected.
```

17. list employee names whose comission is null

```
SQL> select ename from emp where comm is null ;

ENAME
----------
KING
BLAKE
CLARK
JONES
JAMES
FORD
SMITH
SCOTT
ADAMS
MILLER

10 rows selected.
```

18. list maximum salary,minimum salary,average salary from emp

```
SQL> select max(sal),min(sal),avg(sal) from emp ;

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

19. list the number of jobs

```
SQL> select count(distinct(job)) from emp ;

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

20. list the number of people and avg salary in deptno 30

```
SQL> select count(empno),avg(sal) from emp where deptno=30 ;

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

21. list the maximum and minimum salary in the designation 'SALESMEN' and 'CLERK'

```
SQL> select count(*),max(sal),min(sal),avg(sal) from emp where
job in ('SALESMAN','CLERK') ;

  COUNT(*)   MAX(SAL)   MIN(SAL)   AVG(SAL)
---------- ---------- ---------- ----------
         8       1600        800    1218.75
```

22. 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) ;

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

23. display todays date and present time

```
SQL> select to_char(sysdate,'dd-mm-yyyy hh-mi-ss') from dual ;

TO_CHAR(SYSDATE,'DD
-------------------
16-03-2023 10-57-44
```

24. list the employee names an dtheir joining dates in the following formats
A. SMITH   17 dec nineteen eighty

```
SQL> select ename,to_char(hiredate,'dd mon year')from emp
where ename like 'SMITH' ;

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

B. SMITH   seventeenth dec nineteen eighty

```
SQL> select ename,to_char(hiredate,'ddspth mon year') from emp
where ename like 'SMITH' ;

ENAME      TO_CHAR(HIREDATE,'DDSPTHMONYEAR')
---------- ----------------------------------------------------
```

```
SMITH      seventeenth dec nineteen eighty
```

C. SMITH weekday of joining

```
SQL> select ename,to_char(hiredate,'day') from emp where ename
like 'SMITH' ;


ENAME      TO_CHAR(H
---------- ---------
SMITH      Wednesday
```

D.SMITH 17/12/80

```
SQL> select ename,to_char(hiredate,'dd/mm/yy') from emp where
ename like 'SMITH' ;

ENAME      TO_CHAR(
---------- --------
SMITH      17/12/80
```

25. list the employee names and their experience in the years

```
SQL> select
ename,round(months_between(sysdate,hiredate)/12)exp from emp ;

ENAME             EXP
---------------------
KING               41
BLAKE              42
CLARK              42
JONES              42
MARTIN             41
ALLEN              42
TURNER             42
JAMES              41
WARD               42
FORD               41
SMITH              42

ENAME             EXP
---------------------
SCOTT              40
ADAMS              40
MILLER             41

14 rows selected.
```

26. list the employee names who joined in DEC and on wednesday or Friday

```
SQL> select ename,to_char(hiredate,'day mon') from emp where
to_char(hiredate,'day mon') in('wednesd
ay dec' , 'friday dec');

ENAME          TO_CHAR(HIRED
-------------- ------------------
SMITH          wednesday dec
```

27. display a given day as astring in different formats

```
SQL> select to_char(sysdate,'ddspth month year ') from dual;

TO_CHAR(SYSDATE,'DDSPTHMONTHYEAR')
------------------------------------------------------------------------------
sixteenth march     twenty twenty-three
```

28. list the employees who dont report to anybody

```
SQL> select ename,mgr from emp where mgr is null ;

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

Worksheet-2

1)list employee names and their hiredates sorted in the order of their experience.

```
SQL> select ename,round(months_between(sysdate,hiredate)/12)
as experience from emp order by experie
nce;

ENAME          EXPERIENCE
-------------- --------------
SCOTT                  40
ADAMS                  40
KING                   41
JAMES                  41
MILLER                 41
FORD                   41
BLAKE                  42
CLARK                  42
JONES                  42
SMITH                  42
WARD                   42

ENAME          EXPERIENCE
-------------- --------------
MARTIN                 42
ALLEN                  42
TURNER                 42

14 rows selected.
```

2)list the managers names and their joining dates completely spelled in alphabetical order of names.

```
SQL>select ename,hiredate from emp where job='MANAGER' order
by ename;

ENAME          HIREDATE
-------------- -------------
BLAKE          01-MAY-81
CLARK          09-JUN-81
JONES          02-APR-81
```

3)list employee names and their experience in years with names arranged in descending order.

```
SQL> select ename,round((months_between(sysdate,hiredate))/12)
as experience from emp order by ename
 desc;

ENAME          EXPERIENCE
-------------- --------------
```

```
WARD                   42
TURNER                 42
SMITH                  42
SCOTT                  40
MILLER                 41
MARTIN                 42
KING                   41
JONES                  42
JAMES                  41
FORD                   41
CLARK                  42

ENAME          EXPERIENCE
-------------- --------------
BLAKE                  42
ALLEN                  42
ADAMS                  40

14 rows selected.
```

4)list the employee names havving a minimum 2 years experience sorted on experience.

```
SQL> select
ename,round((months_between(sysdate,hiredate))/12)as
experience from emp where round((mo


ENAME          EXPERIENCE
-------------- --------------
SCOTT                  40
ADAMS                  40
KING                   41
JAMES                  41
MILLER                 41
FORD                   41
BLAKE                  42
CLARK                  42
JONES                  42
SMITH                  42
WARD                   42

ENAME          EXPERIENCE
-------------- --------------
MARTIN                 42
ALLEN                  42
TURNER                 42

14 rows selected.
```

5)list employee names with all capital letters with all small letters and with first letter only capital.

```
SQL>select upper(ename),lower(ename),initcap(ename) from emp;

UPPER(ENAM LOWER(ENAM INITCAP(EN
---------- ---------- ----------
KING       king       King
BLAKE      blake      Blake
CLARK      clark      Clark
JONES      jones      Jones
MARTIN     martin     Martin
ALLEN      allen      Allen
TURNER     turner     Turner
JAMES      james      James
WARD       ward       Ward
FORD       ford       Ford
SMITH      smith      Smith

UPPER(ENAM LOWER(ENAM INITCAP(EN
---------- ---------- ----------
SCOTT      scott      Scott
ADAMS      adams      Adams
MILLER     miller     Miller

14 rows selected.
```

6)list the employee names with length of the name sorted on length.

```
SQL> select ename,length(ename) from emp order by
length(ename);

ENAME      LENGTH(ENAME)
------------------------
KING                   4
WARD                   4
FORD                   4
BLAKE                  5
CLARK                  5
ALLEN                  5
SCOTT                  5
ADAMS                  5
SMITH                  5
JAMES                  5
JONES                  5

ENAME      LENGTH(ENAME)
------------------------
MARTIN                 6
TURNER                 6
MILLER                 6

14 rows selected.
```

7)list the employee names appending sri to the begining and garu to the ending.

```
SQL>select 'SRI '||'   '||ename||' '||'GARU' from emp;

'SRI'||''||ENAME||''||'
--------------------------------
SRI    KING    GARU
SRI    BLAKE  GARU
SRI    CLARK  GARU
SRI    JONES  GARU
SRI    MARTIN  GARU
SRI    ALLEN  GARU
SRI    TURNER  GARU
SRI    JAMES  GARU
SRI    WARD  GARU
SRI    FORD  GARU
SRI    SMITH  GARU

'SRI'||''||ENAME||''||'
--------------------------------
SRI    SCOTT  GARU
SRI    ADAMS  GARU
SRI    MILLER  GARU

14 rows selected.
```

8)list the employee names and month names of joining.

```
SQL>select ename,to_char(hiredate,'month') from emp;

ENAME       TO_CHAR(H
---------- ---------
KING        november
BLAKE       may
CLARK       june
JONES       april
MARTIN      september
ALLEN       february
TURNER      september
JAMES       december
WARD        february
FORD        december
SMITH       december

ENAME       TO_CHAR(H
---------- ---------
SCOTT       december
ADAMS       january
MILLER      january
```

```
14 rows selected.
```

9) list employee names and year of joining in wards.

```
SQL>select ename,to_char(hiredate,'year') from emp;

ENAME          TO_CHAR(HIREDATE,'YEAR')
-------------- ------------------------------------------------------------
KING           nineteen eighty-one
BLAKE          nineteen eighty-one
CLARK          nineteen eighty-one
JONES          nineteen eighty-one
MARTIN         nineteen eighty-one
ALLEN          nineteen eighty-one
TURNER         nineteen eighty-one
JAMES          nineteen eighty-one
WARD           nineteen eighty-one
FORD           nineteen eighty-one
SMITH          nineteen eighty

ENAME          TO_CHAR(HIREDATE,'YEAR')
-------------- ------------------------------------------------------------
SCOTT          nineteen eighty-two
ADAMS          nineteen eighty-three
MILLER         nineteen eighty-two

14 rows selected.
```

10) list employee names,job and salary with five hypens inn between.

```
SQL> select ename ||'-----'||job||'-----'||sal from emp;

ENAME||'-----'||JOB||' --------'||SAL
-------------------------------------------------------------------------------------
-----------------
KING-----PRESIDENT---------5000
BLAKE-----MANAGER --------2850
CLARK-----MANAGER --------2450
JONES-----MANAGER --------2975
MARTIN-----SALESMAN-------- 1250
ALLEN-----SALESMAN---------1600
TURNER-----SALESMAN-------- 1500
JAMES-----CLERK -------- 950
WARD-----SALESMAN --------1250
FORD-----ANALYST -------- 3000
SMITH-----CLERK -------- 800

ENAME||'-----'||JOB||' --------'||SAL
-------------------------------------------------------------------------------------
-
```

```
SCOTT-----ANALYST --------3000
ADAMS-----CLERK -------- 1100
MILLER-----CLERK --------1300
```

11)list employee names and position of first occurence of I in thier name.

```
 SQL>select ename,instr(ename,'I') from emp;

ENAME       INSTR(ENAME,'I')
------------------------------------------------------------------
KING                          2
BLAKE                         0
CLARK                         0
JONES                          0
MARTIN                        5
ALLEN                         0
TURNER                        0
JAMES                         0
WARD                          0
FORD                          0
SMITH                         3

ENAME       INSTR(ENAME,'I')
------------------------------------------------------------------
SCOTT                         0
ADAMS                         0
MILLER                        2

14 rows selected.
```

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;

ENAME       SUBSTR(EN
---------- ---------
KING        IN
BLAKE       LAK
CLARK       LAR
JONES       ONE
MARTIN      ARTI
ALLEN       LLE
TURNER      URNE
JAMES       AME
WARD        AR
FORD        OR
SMITH       MIT

ENAME       SUBSTR(EN
-------------- -------------
SCOTT          COT
ADAMS          DAM
MILLER         ILLE

14 rows selected.
```

13) list employee who joined between apr 81 and apr 82.

```
SQL>select ename,hiredate from emp where hiredate between '01-
APR-81' and '30-APR-82';

ENAME          HIREDATE
-------------- -------------
KING           17-NOV-81
BLAKE          01-MAY-81
CLARK          09-JUN-81
JONES          02-APR-81
MARTIN         28-SEP-81
TURNER         08-SEP-81
JAMES          03-DEC-81
FORD           03-DEC-81
MILLER         23-JAN-82

9 rows selected.
```

14) list max(SAL),min(SAL),avg(SAL) of dept 10,30

```
SQL> select min(sal),max(sal),avg(sal) from emp where deptno
in (10,30) group by deptno;

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

15) list the designation in deptno 30 but not in 20

```
SQL> select job from emp where deptno=30 minus select job from
emp where deptno=20;

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

16) list the numbers of employee in each department along with dept number.

```
SQL> select deptno,count(*) from emp group by deptno;

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

17) list the number of employee joined year wise.

```
SQL> select count(*),to_char(hiredate,'yy')as year from emp
group by to_char(hiredate,'yy');

  COUNT(*) YE
---------- --
         1 80
        10 81
         2 82
         1 83
```

18) list number of employee jobwise.

```
SQL> select job,count(*) from emp group by job;

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

19) list the max(sal),min(sal),avg(sal) deptwise.

```
SQL> select max(Sal),min(Sal),avg(Sal),deptno from emp group
by deptno;

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

20) list max(sal),min(sal),avg(sal) jobwise.

```
SQL>select max(Sal),min(Sal),avg(Sal),job from emp group by
job;

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

21) list max(sal),min(sal) for the job manager and clerk.

```
SQL>select max(Sal),min(Sal),job from emp where job
in('MANAGER','CLERK') group by job;

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

22) list the max(sal),min(sal),avg(sal) of the dept having a minimum of 3 employees

```
SQL>select max(sal),min(sal),avg(sal) from emp group by deptno
having count(*)>=3;


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

23) list the number of employee in each job in each department.

```
SQL> select count(*),job,deptno from emp group by deptno,job;

  COUNT(*) JOB           DEPTNO
---------- --------- ----------
         1 CLERK             10
         1 MANAGER           10
         1 PRESIDENT         10
         2 CLERK             20
         2 ANALYST           20
         1 MANAGER           20
         1 CLERK             30
         1 MANAGER           30
         4 SALESMAN          30

9 rows selected.
```

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(*);

       MGR   COUNT(*)
---------- ----------
```

```
      7782          1
      7788          1
      7902          1
      7566          2
      7839          3
      7698          5

6 rows selected.
```

25) list emp numbers of employee to whom a minimum of 3 people report.

```
SQL> select mgr,count(*) from emp group by mgr having
count(*)>=3;

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

26) list the dept number having a minimum of 3 persons.

```
SQL> select deptno from emp group by deptno having
count(*)>=3;

    DEPTNO
----------
        10
        20
        30
```

27) list names of jjobs having a minimum of 3 persons in that job .

```
SQL> ed
Wrote file afiedt.buf

  1* select job from emp group by job having count(*)>=3
  2  /

JOB
---------
CLERK
MANAGER
SALESMAN
```

28) list names of months in which a minimum 3persons joined.

```
SQL> select to_char(hiredate,'month') from emp group by
to_char(hiredate,'month') having count(*)>=3
;

TO_CHAR(H
---------
```

```
december
```

29) list hiredates of employees having 2 or more employee having the same hiredate.

```
SQL> select hiredate,count(*) from emp group by hiredate
having count(*)>=2;

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

30) list departments having minimum of 3 people having a minimum of 17 years of experience.

```
  1 select
deptno,count(*),round(months_between(sysdate,hiredate)/12)
from emp where round(months_b
  2* /12) having count(*)>=3;

    DEPTNO   COUNT(*)
ROUND(MONTHS_BETWEEN(SYSDATE,HIREDATE)/12)
---------- ---------- -------------------------------------------
--
        30          5
42


emp 2.txt
Displaying emp 2.txt.
```

Worksheet-3

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

```
SQL>select ename,dname from emp,dept where
emp.deptno=dept.deptno;

ENAME          DNAME
-------------- --------------------
KING           ACCOUNTING
BLAKE          SALES
CLARK          ACCOUNTING
JONES          RESEARCH
MARTIN         SALES
ALLEN          SALES
TURNER         SALES
JAMES          SALES
WARD           SALES
FORD           RESEARCH
SMITH          RESEARCH

ENAME          DNAME
-------------- --------------------
SCOTT          RESEARCH
ADAMS          RESEARCH
MILLER         ACCOUNTING

14 rows selected.
```

2. List employee names, salary and their grade.

```
SQL>select ename,sal,grade from emp,salgrade where sal between
losal and hisal;

ENAME                     SAL          GRADE
-------------- -------------- --------------
JAMES                     950              1
SMITH                     800              1
ADAMS                    1100              1
MARTIN                   1250              2
WARD                     1250              2
MILLER                   1300              2
ALLEN                    1600              3
TURNER                   1500              3
BLAKE                    2850              4
CLARK                    2450              4
JONES                    2975              4

ENAME                     SAL          GRADE
-------------- -------------- --------------
FORD                     3000              4
```

```
SCOTT             3000          4
KING              5000          5

14 rows selected.
```

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

```
SQL> select ename,dname,grade from emp,dept,salgrade where
emp.deptno=dept.deptno and sal between lo
sal and hisal;

ENAME      DNAME                   GRADE
-------------------------------- ----------
JAMES      SALES                       1
SMITH      RESEARCH                    1
ADAMS      RESEARCH                    1
MARTIN     SALES                       2
WARD       SALES                       2
MILLER     ACCOUNTING                  2
ALLEN      SALES                       3
TURNER     SALES                       3
BLAKE      SALES                       4
CLARK      ACCOUNTING                  4
JONES      RESEARCH                    4

ENAME      DNAME                   GRADE
-------------------------------- ----------
FORD       RESEARCH                    4
SCOTT      RESEARCH                    4
KING       ACCOUNTING                  5

14 rows selected.
```

4. List employee names and their manager names.

```
SQL>select e1.ename,e2.ename from emp e1,emp e2 where
e1.mgr=e2.empno;

ENAME      ENAME
---------- ----------
BLAKE      KING
CLARK      KING
JONES      KING
MARTIN     BLAKE
ALLEN      BLAKE
TURNER     BLAKE
JAMES      BLAKE
WARD       BLAKE
FORD       JONES
SMITH      FORD
SCOTT      JONES
```

```
ENAME           ENAME
--------------- ---------------
ADAMS           SCOTT
MILLER          CLARK

13 rows selected.
```

5. List dept name and Manager name.

```
SQL>select dname,ename from emp,dept where
emp.deptno=dept.deptno and job='MANAGER';

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

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

```
SQL>select dname,ename,grade from emp,dept,salgrade where
emp.deptno=dept.deptno and job='MANAGER' and
 sal between losaland hisal order by grade;

DNAME               ENAME                GRADE
------------------- -------------- --------------
SALES               BLAKE                    4
ACCOUNTING          CLARK                    4
RESEARCH            JONES                    4
```

7. List employees having commission along with grade.

```
SQL> select comm,ename,grade from emp,salgrade where sal
between losal and hisal and comm is not nul
l;

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

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

```
SQL> select e1.ename,e2.ename,e1.job,e2.job from emp e1,emp e2
where e1.mgr=e2.empno and e1.job='MAN
AGER';

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

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

```
SQL> select e1.ename,e2.ename,e1.deptno,e2.deptno from emp
e1,emp e2 where e1.mgr=e2.empno and e1.de
ptno=e2.deptno;

ENAME      ENAME          DEPTNO     DEPTNO
---------- ---------- ---------- ----------
CLARK      KING               10         10
MARTIN     BLAKE              30         30
ALLEN      BLAKE              30         30
TURNER     BLAKE              30         30
JAMES      BLAKE              30         30
WARD       BLAKE              30         30
FORD       JONES              20         20
SMITH      FORD               20         20
SCOTT      JONES              20         20
ADAMS      SCOTT              20         20
MILLER     CLARK              10         10

11 rows selected.
```

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

```
SQL> select e1.ename,e2.ename,e1.deptno,e2.deptno from emp
e1,emp e2 where e1.mgr=e2.empno and e1.de
ptno!=e2.deptno;


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

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

```
SQL>select ename,dname from emp,dept where
emp.deptno=dept.deptno and substr(ename,1,1)=substr
```

```
(dname,1,1);

no rows selected
```

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,to_char(hiredate,'month'),grade from
emp,salgrade
where sal between losal and hisal and
to_char(hiredate,'month')=to_char(sysdate,'month') and to
_char(hiredate,'Y')=grade;

no rows selected
```

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

```
SQL>select ename,empno,mgr,grade from emp,salgrade where sal
between losal and hisal and mod(
mgr,2)=mod(empno,2) and mod(mgr,2)=mod(grade,2);

ENAME                EMPNO        MGR      GRADE
------------------------ ------------------------ ------------------------ ------------------------
MARTIN                7654       7698          2
MILLER                7934       7782          2
FORD                  7902       7566          4
SCOTT                 7788       7566          4
```

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

```
SQL> select ename,grade,deptno from emp,salgrade where sal
between losal and hisal and grade=substr(
deptno,1,1);

ENAME                GRADE     DEPTNO
------------------------ ------------------------ ------------------------
ALLEN                    3         30
TURNER                   3         30
```

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

```
SQL> select ename,grade,deptno from emp,salgrade where sal
between losal and hisal and grade=substr(
deptno,1,1);

ENAME                GRADE     DEPTNO
------------------------ ------------------------ ------------------------
ALLEN                    3         30
```

```
TURNER                3         30
```

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)
and
length(dname)=length(loc);

no rows selected
```

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 a
nd to_char(hiredate,'MM')=grade;

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='CLE
RK' and e2.job='ANALYST';

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

19. List emp names and thrie 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 s
1.grade=s2.grade;

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,e2.ename,e1.hiredate,e2.hiredate from emp
e1,emp e2 where e1.mgr=e2.empno and e
1.hiredate<=e2.hiredate;

ENAME       ENAME       HIREDATE   HIREDATE
----------- ----------- ---------- ----------
BLAKE       KING        01-MAY-81  17-NOV-81
CLARK       KING        09-JUN-81  17-NOV-81
JONES       KING        02-APR-81  17-NOV-81
ALLEN       BLAKE       20-FEB-81  01-MAY-81
WARD        BLAKE       22-FEB-81  01-MAY-81
SMITH  FORD  17-DEC-80 03-DEC-81

6 rows selected.
```

Worksheet-4

**1. EMPLOYEE(FNAME,MINIT,LNAME,SSN,SEX,SALARY,SUPERSSNIDNO)
CONSTRAINTS: FNAME,LNAME,SSN,DNO NOT NULL
PRIMARY KEY(SSN)
FOREIGN KEY (SUPERSSN) REFERENCES EMPLOYEE(SSN)
FOREIGN KEY(DNO) REFERENCES DEPARTMENT(DNUMBER)**

```
SQL> create table employee2 (FNAME character(10) not
null,MINIT character(5) not null,LNAMEcharacter(10) not
null,SSN number(4)
 not null primary key,SEX character(3) not null,SALARY
number(5)not null,SUPERSSN number(4),DNO number(1) not null);
Table created.

SQL> insert into
employeevalues('&fname','&minit','&lname',&ssn,'&sex',&salary,
&superssn,&dno);
Enter value for fname: JOHN
Enter value for minit: B
Enter value for lname: SMITH
Enter value for ssn: 2345
Enter value for sex: M
Enter value for salary: 30000
Enter value for superssn: 3344
Enter value for dno: 5
old 1: insert into employee
values('&fname','&minit','&lname',&ssn,'&sex',&salary,&superss
n,&dno
)
new 1: insert into employee
values('JOHN','B','SMITH',2345,'M',30000,3344,5)
1 row created.

SQL> /
Enter value for fname: FRANKIN
Enter value for minit: T
Enter value for lname: WONG
Enter value for ssn: 3344
Enter value for sex: M
Enter value for salary: 40000
Enter value for superssn: 8866
Enter value for dno: 5
old 1: insert into employee
values('&fname','&minit','&lname',&ssn,'&sex',&salary,&superss
n,&dno
)
new 1: insert into employee
values('FRANKIN','T','WONG',3344,'M',40000,8866,5)
1 row created.
```

```
SQL> /
Enter value for fname: JENNIFER
Enter value for minit: S
Enter value for lname: WALLACE
Enter value for ssn: 8765
Enter value for sex: F
Enter value for salary: 43000
Enter value for superssn: 8866
Enter value for dno: 4
old 1: insert into employee
values('&fname','&minit','&lname',&ssn,'&sex',&salary,&superss
n,&dno
)
new 1: insert into employee
values('JENNIFER','S','WALLACE',8765,'F',43000,8866,4)
1 row created.

SQL> /
Enter value for fname: ALICIA
Enter value for minit: J
Enter value for lname: ZELAYA
Enter value for ssn: 9988
Enter value for sex: F
Enter value for salary: 25000
Enter value for superssn: 8765
Enter value for dno: 4
old 1: insert into employee
values('&fname','&minit','&lname',&ssn,'&sex',&salary,&superss
n,&dno
)
new 1: insert into employee
values('ALICIA','J','ZELAYA',9988,'F',25000,8765,4)
1 row created.

SQL> /
Enter value for fname: RAMESH
Enter value for minit: K
Enter value for lname: NARAYANA
Enter value for ssn: 6688
Enter value for sex: M
Enter value for salary: 38000
Enter value for superssn: 3344
Enter value for dno: 5
old 1: insert into employee
values('&fname','&minit','&lname',&ssn,'&sex',&salary,&superss
n,&dno
)
new 1: insert into employee
values('RAMESH','K','NARAYANA',6688,'M',38000,3344,5)
1 row created.
```

```
SQL> insert into employee
values('&fname','&minit','&lname',&ssn,'&sex',&salary,&superss
n,&dno
);
Enter value for fname: JAMES
Enter value for minit: E
Enter value for lname: BORG
Enter value for ssn: 8866
Enter value for sex: M
Enter value for salary: 55000
Enter value for superssn: NULL
Enter value for dno: 1
old 1: insert into employee
values('&fname','&minit','&lname',&ssn,'&sex',&salary,&superss
n,&dno
)
new 1: insert into employee
values('JAMES','E','BORG',8866,'M',55000,NULL,1)
1 row created.

alter table employee add constraint employee_SUPERSSN_FK
foreign
key(SUPERSSN) references emplo
yee(SSN);
Table altered.

SQL> select * from employee;

FNAME                M LNAME       SSN  S      SALARY   SUPERSSN
DNO
-------------------- - ---------- ----- - ---------- ---------- ------
---------------------

JOHN                 B SMITH      2345 M      30000       3344
5
FRANKLIN             T WONG       3344 M      40000       8866
5
ALICIA               J ZELAYA     9988 F      25000       8765
4
JENNIFER             S WALLACE    8765 F      43000       8866
4
RAMESH               K NARAYANA   6688 M      38000       3344
5
JOYCE                A ENGLISH    5345 F      25000       3344
5
AHMAD                V JABBER     8798 M      25000       8765
4
JAMES                E BORG       8866 M      55000
1

8 rows selected.
```

```
SQL> alter table employee add constraint employee_DNO_fk
foreign key(DNO) references department(DNUMBER);
Table altered.
```

2.DEPARTMENT(DNAME,DNUMBER,MGRSSN)
CONSTRAINTS: DNAME,DNUMBER,MGRSSN NOTNULL
PRIMARY KEY (DNUMBER) UNIQUE (DNAME),
FOREIGN KEY(MGRSSN) REFERENCES EMPLOYEE(SSN)

```
SQL> create table department(DNAME character(15) not null
unique,DNUMBER number(1) not
null,MGRSSN number(4) not null);
Table created.

SQL> INSERT INTO DEPARTMENT VALUES('&DNAME',&DNUMBER,&MGRSSN);
Enter value for dname: RESEARCH
Enter value for dnumber: 5
Enter value for mgrssn: 3344
old 1: INSERT INTO DEPARTMENT
VALUES('&DNAME',&DNUMBER,&MGRSSN)
new 1: INSERT INTO DEPARTMENT VALUES('RESEARCH',5,3344)
1 row created.

SQL> /
Enter value for dname: ADMINISTRATION
Enter value for dnumber: 4
Enter value for mgrssn: 8765
old 1: INSERT INTO DEPARTMENT
VALUES('&DNAME',&DNUMBER,&MGRSSN)
new 1: INSERT INTO DEPARTMENT VALUES('ADMINISTRATION',4,8765)
1 row created.

SQL> /
Enter value for dname: HEADQUATERS
Enter value for dnumber: 1
Enter value for mgrssn: 8866
old 1: INSERT INTO DEPARTMENT
VALUES('&DNAME',&DNUMBER,&MGRSSN)
new 1: INSERT INTO DEPARTMENT VALUES('HEADQUATERS',1,8866)
1 row created.


SQL> select * from department;

DNAME                  DNUMBER     MGRSSN
-------------------  ------------  ------------
RESEARCH                     5       3344
ADMINISTRATION               4       8765
HEADQUATERS                  1       8866
```

```
SQL> alter table department add constraint
department_DNUMBER_pk primary key(DNUMBER);
Table altered.

SQL> alter table department add constraint
department_MGRSSN_fk foreign key(MGRSSN) references emplo
yee(SSN);
Table altered.
```

3. DEPT_LOCATIONS(DNUMBER,DLOCATION)
CONSTRAINTS: DNUMBER.DLOCATION NOTNULL
PRIMARY KEY(DNUMBER,DLOCATION)
FOREIGN KEY(DNUMBER) REFERENCES DEPARTMENT(DNUMBER)

```
SQL> create table dept_locations (DNUMBER number(1) not
null,DLOCATION character(10) not null);
Table created.

SQL> insert into dept_locations values(&DNUMBER,'&DLOCATION');
Enter value for dnumber: 1
Enter value for dlocation: HOUSTON
old 1: insert into dept_locations
values(&DNUMBER,'&DLOCATION')
new 1: insert into dept_locations values(1,'HOUSTON')
1 row created.

SQL> /
Enter value for dnumber: 4
Enter value for dlocation: STAFFORD
old 1: insert into dept_locations
values(&DNUMBER,'&DLOCATION')
new 1: insert into dept_locations values(4,'STAFFORD')
1 row created.

SQL> /
Enter value for dnumber: 5
Enter value for dlocation: BELLARIE
old 1: insert into dept_locations
values(&DNUMBER,'&DLOCATION')
new 1: insert into dept_locations values(5,'BELLARIE')
1 row created.

SQL> /
Enter value for dnumber: 5
Enter value for dlocation: SUGARLAND
old 1: insert into dept_locations
values(&DNUMBER,'&DLOCATION')
new 1: insert into dept_locations values(5,'SUGARLAND')
1 row created.
```

```
SQL> /
Enter value for dnumber: 5
Enter value for dlocation: HOUSTON
old 1: insert into dept_locations
values(&DNUMBER,'&DLOCATION')
new 1: insert into dept_locations values(5,'HOUSTON')
1 row created.


SQL> select * from dept_location;

   DNUMBER DLOCATION
---------- ----------
         1 houston
         4 stafford
         5 bellarie
         5 sugarland
         5 houston


SQL> alter table dept_locations add constraint dept_DNUMBER_fk
foreign key(DNUMBER) references depar
tment(DNUMBER);
Table altered.
```

4. PROJECT(PNAME,PNUMBER,PLOCATIOIM,DNUM)
CONSTRAINTS: PNAME.PNUMBER.DNUM NOTNULL
PRIMARY KEY(PNUMBER) UNIQUE(PNAME)
FOREIGN KEY(DNUM) REFERENCES DEPARTMENT(DNUMBER)

```
SQL> create tablr project(PNAME character(15) not null
unique,PNUMBERnumber(2) not null primary key,
PLOCATION character(10) not null,DNUM number(1) not null);
Table created.

SQL> insert into project
values('&PNAME',&PNUMBER,'&PLOCATION',&DNUM);
Enter value for pname: PRODUCT_X
Enter value for pnumber: 1
Enter value for plocation: BELLARIE
Enter value for dnum: 5
old 1: insert into project
values('&PNAME',&PNUMBER,'&PLOCATION',&DNUM)
new 1: insert into project values('PRODUCT_X',1,'BELLARIE',5)
1 row created.

SQL> /
Enter value for pname: PRODUCT_Y
Enter value for pnumber: 2
Enter value for plocation: SUGARLAND
Enter value for dnum: 5
old 1: insert into project
values('&PNAME',&PNUMBER,'&PLOCATION',&DNUM)
new 1: insert into project values('PRODUCT_Y',2,'SUGARLAND',5)
1 row created.

SQL> /
Enter value for pname: PRODUCT_Z
Enter value for pnumber: 3
Enter value for plocation: HOUSTON
Enter value for dnum: 5
old 1: insert into project
values('&PNAME',&PNUMBER,'&PLOCATION',&DNUM)
new 1: insert into project values('PRODUCT_Z',3,'HOUSTON',5)
1 row created.

SQL> /
Enter value for pname: COMPUTERIZATION
Enter value for pnumber: 10
Enter value for plocation: STAFFORD
Enter value for dnum: 4
old 1: insert into project
values('&PNAME',&PNUMBER,'&PLOCATION',&DNUM)
new 1: insert into project
values('COMPUTERIZATION',10,'STAFFORD',4)
1 row created.

SQL> /
Enter value for pname: REORGANIZATION
Enter value for pnumber: 20
Enter value for plocation: HOUSTON
Enter value for dnum: 1
old 1: insert into project
values('&PNAME',&PNUMBER,'&PLOCATION',&DNUM)
new 1: insert into project
values('REORGANIZATION',20,'HOUSTON',1)
1 row created.

SQL> /
Enter value for pname: NEWBENEFITS
Enter value for pnumber: 30
Enter value for plocation: STAFFORD
Enter value for dnum: 4
old 1: insert into project
values('&PNAME',&PNUMBER,'&PLOCATION',&DNUM)
new 1: insert into project
values('NEWBENEFITS',30,'STAFFORD',4)
1 row created.

SQL> select * from dependent;
```

```
PNAME                   PNUMBER PLOCATION         DNUM
----------------------------------------------------- ---------------- ----------------
PRODUCT_X                     1 BELLARIE             5
PRODUCT_Y                     2 SUGARLAND            5
PRODUCT_Z                     3 HOUSTON              5
COMPUTERIZATION              10 STAFFORD             4
REORGANIZATION               20 HOUSTON              1
NEWBENEFITS                  30 STAFFORD             4

6 rows selected.

SQL> alter table project add constraint project_DNUM_fk
foreign
key(DNUM) references department(DNUM
BER);
Table altered.
```

5. WORKS_ON(ESSN,PNO,HOURS)
CONSTRAINTS: ESSN,PNO NOTNULL
PRIMARY KEY(ESSN,PNO)
FOREIGN KEY(ESSN) REFERENCES EMPLOYEE(SSN)
FOREIGN KEY(PNO) REFERENCES PROJECT(PNUMBER)

```
SQL> create table works_on(ESSN number(4) not null,PNO
number(2) not null,HOURS number(5));
Table created.

SQL> insert into works_on values(&ESSN,&PNO,&HOURS);
Enter value for essn: 2345
Enter value for pno: 1
Enter value for hours: 32.5
old 1: insert into works_on values(&ESSN,&PNO,&HOURS)
new 1: insert into works_on values(2345,1,32.5)
1 row created.

SQL> /
Enter value for essn: 2345
Enter value for pno: 2
Enter value for hours: 7.5
old 1: insert into works_on values(&ESSN,&PNO,&HOURS)
new 1: insert into works_on values(2345,2,7.5)
1 row created.

SQL> /
Enter value for essn: 6688
Enter value for pno: 3
Enter value for hours: 40
old 1: insert into works_on values(&ESSN,&PNO,&HOURS)
new 1: insert into works_on values(6688,3,40)
```

```
1 row created.

SQL> /
Enter value for essn: 5345
Enter value for pno: 1
Enter value for hours: 20
old 1: insert into works_on values(&ESSN,&PNO,&HOURS)
new 1: insert into works_on values(5345,1,20)
1 row created.

SQL> /
Enter value for essn: 5345
Enter value for pno: 2
Enter value for hours: 20
old 1: insert into works_on values(&ESSN,&PNO,&HOURS)
new 1: insert into works_on values(5345,2,20)
1 row created.

SQL> /
Enter value for essn: 3344
Enter value for pno: 2
Enter value for hours: 10
old 1: insert into works_on values(&ESSN,&PNO,&HOURS)
new 1: insert into works_on values(3344,2,10)
1 row created.

SQL> /
Enter value for essn: 3344
Enter value for pno: 3
Enter value for hours: 10
old 1: insert into works_on values(&ESSN,&PNO,&HOURS)
new 1: insert into works_on values(3344,3,10)
1 row created.

SQL> /
Enter value for essn: 3344
Enter value for pno: 10
Enter value for hours: 10
old 1: insert into works_on values(&ESSN,&PNO,&HOURS)
new 1: insert into works_on values(3344,10,10)
1 row created.

SQL> /
Enter value for essn: 3344
Enter value for pno: 20
Enter value for hours: 10
old 1: insert into works_on values(&ESSN,&PNO,&HOURS)
new 1: insert into works_on values(3344,20,10)
1 row created.

SQL> /
```

```
Enter value for essn: 9988
Enter value for pno: 30
Enter value for hours: 30
old 1: insert into works_on values(&ESSN,&PNO,&HOURS)
new 1: insert into works_on values(9988,30,30)
1 row created.

SQL> /
Enter value for essn: 9988
Enter value for pno: 10
Enter value for hours: 10
old 1: insert into works_on values(&ESSN,&PNO,&HOURS)
new 1: insert into works_on values(9988,10,10)
1 row created.

SQL> /
Enter value for essn: 8798
Enter value for pno: 10
Enter value for hours: 35
old 1: insert into works_on values(&ESSN,&PNO,&HOURS)
new 1: insert into works_on values(8798,10,35)
1 row created.

SQL> /
Enter value for essn: 8798
Enter value for pno: 20
Enter value for hours: 5
old 1: insert into works_on values(&ESSN,&PNO,&HOURS)
new 1: insert into works_on values(8798,20,5)
1 row created.

SQL> /
Enter value for essn: 8765
Enter value for pno: 20
Enter value for hours: 20
old 1: insert into works_on values(&ESSN,&PNO,&HOURS)
new 1: insert into works_on values(8765,20,20)
1 row created.

SQL> /
Enter value for essn: 8765
Enter value for pno: 30
Enter value for hours: 15
old 1: insert into works_on values(&ESSN,&PNO,&HOURS)
new 1: insert into works_on values(8765,30,15)
1 row created.

SQL> insert into works_on values(&ESSN,&PNO,&HOURS);
Enter value for essn: 8866
Enter value for pno: 30
Enter value for hours: null
```

```
old 1: insert into works_on values(&ESSN,&PNO,&HOURS)
new 1: insert into works_on values(8866,30,null)
1 row created.


SQL> /
Enter value for essn: 8866
Enter value for pno: 1
Enter value for hours: null
old 1: insert into works_on values(&ESSN,&PNO,&HOURS)
new 1: insert into works_on values(8866,1,null)
1 row created.


SQL> select * from works_on;

      ESSN         PNO      HOURS
---------- ---------- ----------
      2345           1         33
      2345           2          8
      6688           3         40
      5345           1         20
      5345           2         20
      3344           2         10
      3344           3         10
      3344          10         10
      3344          20         10
      9988          30         30
      9988          10         10

      ESSN         PNO      HOURS
---------- ---------- ----------
      8798          10         35
      8798          20          5
      8765          20         20
      8765          30         15
      8866          30
      8866           1

17 rows selected.


SQL> alter table works_on add constraint work_ESSN_fk foreign
key(ESSN)
references employee(SSN);
Table altered.

SQL> alter table works_on add constraint work_PNO_fk foreign
key(PNO)
references project(PNUMBER);
Table altered.
```

6. DEPENDENT(ESSN,D_NAME,SEX,RELATIONSHIP)
CONSTRAINTS': ESSN,D_NAME NOTNULL
PRIMARY KEY(ESSN,D_NAME)
FOREIGN KEY(ESSN) REFERENCES EMPLOYEE(SSN)

```
SQL> create table dependent
(ESSN number(4) not null,D_NAME character(15) not null,SEX
character(3),RELATIONSHIP character(15));
Table created.

SQL> insert into dependent
values(&ESSN,'&D_NAME','&SEX','&RELATIONSHIP');
Enter value for essn: 3344
Enter value for d_name: ALICE
Enter value for sex: F
Enter value for relationship: DAUGHTER
old 1: insert into dependent
values(&ESSN,'&D_NAME','&SEX','&RELATIONSHIP'
new 1: insert into dependent
values(3344,'ALICE','F','DAUGHTER')
1 row created.

SQL> /
Enter value for essn: 3344
Enter value for d_name: THEODORE
Enter value for sex: M
Enter value for relationship: SON
old 1: insert into dependent
values(&ESSN,'&D_NAME','&SEX','&RELATIONSHIP'
new 1: insert into dependent values(3344,'THEODORE','M','SON')
1 row created.

SQL> /
Enter value for essn: 3344
Enter value for d_name: JOY
Enter value for sex: F
Enter value for relationship: SPOUSE
old 1: insert into dependent
values(&ESSN,'&D_NAME','&SEX','&RELATIONSHIP'
new 1: insert into dependent values(3344,'JOY','F','SPOUSE')
1 row created.

SQL> /
Enter value for essn: 8765
Enter value for d_name: ABNER
Enter value for sex: M
Enter value for relationship: SPOUSE
old 1: insert into dependent
values(&ESSN,'&D_NAME','&SEX','&RELATIONSHIP'
new 1: insert into dependent values(8765,'ABNER','M','SPOUSE')
1 row created.

SQL> /
Enter value for essn: 2345
Enter value for d_name: MICHAEL
Enter value for sex: M
Enter value for relationship: SON
old 1: insert into dependent
values(&ESSN,'&D_NAME','&SEX','&RELATIONSHIP'
new 1: insert into dependent values(2345,'MICHAEL','M','SON')
1 row created.

SQL> /
Enter value for essn: 2345
Enter value for d_name: ALICE
Enter value for sex: F
Enter value for relationship: DAUGHTER
old 1: insert into dependent
values(&ESSN,'&D_NAME','&SEX','&RELATIONSHIP'
new 1: insert into dependent
values(2345,'ALICE','F','DAUGHTER')
1 row created.

SQL> /
Enter value for essn: 2345
Enter value for d_name: ELIZABETH
Enter value for sex: F
Enter value for relationship: SPOUSE
old 1: insert into dependent
values(&ESSN,'&D_NAME','&SEX','&RELATIONSHIP'
new 1: insert into dependent
values(2345,'ELIZABETH','F','SPOUSE')
1 row created.

SQL> select * from dependent;

      ESSN D_NAME               SEX RELATIONSHIP
---------- -------------------- --- --------------------
      3344 ALICE                f   DAUGHTER
      3344 THEODORE             M   SON
      3344 JOY                  F   SPOUSE
      8765 ABNER                M   SPOUSE
      2345 MICHAEL              M   SON
      2345 ALICE                F   DAUGHTER
      2345 ELIZABETH            F   SPOUSE

7 rows selected.

SQL> alter table dependent add constraint depen_ESSN_fk
foreign
```

```
key(ESSN) references employee(SSN);
Table altered.
```

STUDENT TABLE CREATION

1.create a table to store student information.

```
SQL> create table std(stdno number(4),
sname varchar2(25),
sphno number(10),
sgmail varchar(50),
per number(7,2));
Table created.
```

2.list the discription of the std table.

```
SQL> desc std ;
 Name                                      Null?    Type
 ----------------------------------------- -------- ------------------
----------
 STDNO                                              NUMBER(4)
 SNAME                                              VARCHAR2(25)
 SPHNO                                              NUMBER(10)
 SGMAIL                                             VARCHAR2(50)
 PER                                                NUMBER(7,2)
```

3.add an attribute (stdaddress) to the student table.

```
SQL> alter table std add (stdaddress varchar2(50));

Table altered.
```

4.list the discription of the altered table.

```
SQL> desc std;
 Name                                      Null?    Type
 ----------------------------------------- -------- ------------------
----------
 STDNO                                              NUMBER(4)
 SNAME                                              VARCHAR2(25)
 SPHNO                                              NUMBER(10)
 SGMAIL                                             VARCHAR2(50)
 PER                                                NUMBER(7,2)
 STDADDRESS                                         VARCHAR2(50)
```

5.insert student data to the std table.

```
SQL>  insert into std
values(71,'t.sasank',9848657390,'sasanktalakayala@gmail.com',84.75,'on
gole');

1 row created.


SQL> insert into std
values(72,'k.sandeep',8765478676,'sandeepkakarla@gmail.com',58.65,'tal
luru');
```

```
1 row created.

SQL>  insert into std
values(73,'k.joshi',9948945690,'joshikotapuri@gmail.com',65.24,'medera
metla');


1 row created.

SQL>  insert into std values(74,'m.manthru
naik',7656894890,'manthrunaik@gmail.com',75.72,'markapuram');

1 row created.

SQL>  insert into std
values(75,'g.jagadeesh',9848948099,'jagadeesh@gmail.com',52.18,'ongole
');

1 row created.

SQL>   insert into std values(76,'m.sai
kiran',970151209,'saikiranmaraka@gmail.com',76.98,'palasa');


1 row created.

SQL>   insert into std
values(77,'d.sairam',8898765676,'sairamderangula@gmail.com',83.32,'raj
empeta');

1 row created.


SQL>  insert into std
values(78,'y.ashwini',8090789644,'yerraashwini@gmail.com',79.66,'vijay
awada');

1 row created.

SQL> insert into std
values(79,'p.shabna',7698754845,'patanshabna@gmail.com',78.18,'sarpava
ram') ;

1 row created.

SQL>  insert into std
values(80,'n.sujitha',9989515648,'sujithanimmana@gmail.com',87.76,'nel
lore');

1 row created.
```

6.delete the record where stdno=71.

```
SQL>  delete std where stdno=71;

1 row deleted.
```

7.list the student names from std.

```
SQL> select stdno,sname from std;

     STDNO SNAME
---------- -------------------------
        72 k.sandeep
        73 k.joshi
        74 m.manthru naik
        75 g.jagadeesh
        76 m.sai kiran
        77 d.sairam
        78 y.ashwini
        79 p.shabna
        80 n.sujitha

9 rows selected.
```

8.insert new record for the no 71.

```
SQL>  insert into std
values(71,'t.rajasekhar',9848657390,'rajasekhartalakayala@gmail.com',8
4.75,'on
gole');

1 row created.
```

9.list the students whose persentsge is greater than 80.

```
SQL> select sname,stdno,per from std where per>=80 order by stdno;

SNAME                          STDNO        PER
------------------------- ---------- ----------
t.rajasekhar                      71      84.75
d.sairam                          77      83.32
n.sujitha                         80      87.76
```

10. insert the values to the table from keyboard .

```
SQL> insert into std
values(&stdno,'&sname',&sphno,'&sgmail',&per,'&stdaddress');
Enter value for stdno: 81
Enter value for sname: s.sameera
Enter value for sphno:  6545789524
Enter value for sgmail: sameerasaaghar@gmail.com
Enter value for per: 76.45
Enter value for stdaddress: guntur
old   1: insert into std
values(&stdno,'&sname',&sphno,'&sgmail',&per,'&stdaddress')
new   1: insert into std values(81,'s.sameera',
6545789524,'sameerasaaghar@gmail.com',76.45,'guntur'

1 row created.

SQL> /
Enter value for stdno: 82
Enter value for sname: j.nithin
Enter value for sphno: 7548956298
Enter value for sgmail: jaggamnithin@gmail.com
Enter value for per: 66.23
Enter value for stdaddress: chirala
old   1: insert into std
values(&stdno,'&sname',&sphno,'&sgmail',&per,'&stdaddress')
new   1: insert into std
values(82,'j.nithin',7548956298,'jaggamnithin@gmail.com',66.23,'chiral
a')

1 row created.

SQL> /
Enter value for stdno: 83
Enter value for sname: r.riteshkumar
Enter value for sphno: 9701223485
Enter value for sgmail: ramireddyritesh@gmail.com
Enter value for per: 74.85
Enter value for stdaddress: bapatla
old   1: insert into std
values(&stdno,'&sname',&sphno,'&sgmail',&per,'&stdaddress')
new   1: insert into std
values(83,'r.riteshkumar',9701223485,'ramireddyritesh@gmail.com',74.85
,'bap

1 row created.

SQL> /
Enter value for stdno: 84
Enter value for sname: danni martin
Enter value for sphno: 4854697645
Enter value for sgmail: dannimartin@gmail.com
Enter value for per: 84.45
Enter value for stdaddress: california
old   1: insert into std
values(&stdno,'&sname',&sphno,'&sgmail',&per,'&stdaddress')
new   1: insert into std values(84,'danni
martin',4854697645,'dannimartin@gmail.com',84.45,'californ

1 row created.

SQL> /
Enter value for stdno: 85
Enter value for sname: remiel morningstar
Enter value for sphno: 3568954275
Enter value for sgmail: remiel@gmail.com
Enter value for per: 82.68
Enter value for stdaddress: texax
old   1: insert into std
values(&stdno,'&sname',&sphno,'&sgmail',&per,'&stdaddress')
new   1: insert into std values(85,'remiel
morningstar',3568954275,'remiel@gmail.com',82.68,'texax')

1 row created.

SQL> commit;

Commit complete.
```

11.list the student names whose percentage is less than 60.

```
SQL> select sname,stdno,per from std where per<60 ;

SNAME                          STDNO        PER
------------------------- ---------- ----------
k.sandeep                         72      58.65
g.jagadeesh                       75      52.18
```

12.list the student names who has the letter'y' in their names.

```
SQL>  select sname from std where sname like '%y%';

SNAME
-------------------------
y.ashwini
```

13.count the total no records in the table.

```
SQL> select count(sname) from std;

COUNT(SNAME)
------------
          15
```

14.list the student names along with their percentage by the order of their names.

```
SQL> select stdno,sname from std order by sname;

     STDNO SNAME
---------- -------------------------
        77 d.sairam
        84 danni martin
        75 g.jagadeesh
        82 j.nithin
        73 k.joshi
        72 k.sandeep
        74 m.manthru naik
        76 m.sai kiran
        80 n.sujitha
        79 p.shabna
        83 r.riteshkumar

     STDNO SNAME
---------- -------------------------
        85 remiel morningstar
        81 s.sameera
        71 t.rajasekhar
        78 y.ashwini

15 rows selected.
```

15.list the students from ongole.

```
SQL> select sname from std where stdaddress='ongole';

SNAME
-------------------------
g.jagadeesh
```

16.Print the student table.

```
SQL> select * from std;

     STDNO SNAME                          SPHNO SGMAIL
PER STDADDRESS
---------- ------------------------- ---------- ----------------------
---------------------------- -
        72 k.sandeep                 8765478676
sandeepkakarla@gmail.com                                58.65 talluru
        73 k.joshi                   9948945690
joshikotapuri@gmail.com                                 65.24
mederametla
        74 m.manthru naik            7656894890 manthrunaik@gmail.com
75.72 markapuram
        75 g.jagadeesh               9848948099 jagadeesh@gmail.com
52.18 ongole
        76 m.sai kiran                970151209
saikiranmaraka@gmail.com                                76.98 palasa
        77 d.sairam                  8898765676
sairamderangula@gmail.com                               83.32
rajempeta
        78 y.ashwini                 8090789644 yerraashwini@gmail.com
79.66 vijayawada
        79 p.shabna                  7698754845 patanshabna@gmail.com
78.18 sarpavaram
        80 n.sujitha                 9989515648
sujithanimmana@gmail.com                                87.76 nellore
        71 t.rajasekhar              9848657390
rajasekhartalakayala@gmail.com                          84.75 on

gole

     STDNO SNAME                          SPHNO SGMAIL
PER STDADDRESS
---------- ------------------------- ---------- ----------------------
---------------------------- -


        81 s.sameera                 6545789524
sameerasaaghar@gmail.com                                76.45 guntur
        82 j.nithin                  7548956298 jaggamnithin@gmail.com
66.23 chirala
        83 r.riteshkumar             9701223485
ramireddyritesh@gmail.com                               74.85 bapatla
        84 danni martin              4854697645 dannimartin@gmail.com
84.45 california
        85 remiel morningstar        3568954275 remiel@gmail.com
82.68 texax
15 rows selected.
```

PL/SQL Introduction

PL/SQL is a combination of SQL along with the procedural features of programming languages. It was developed by Oracle Corporation in the early 90's to enhance the capabilities of SQL. PL/SQL is one of three key programming languages embedded in the Oracle Database, along with SQL itself and Java.

SQL stands for Structured Query Language i.e. used to perform operations on the records stored in database such as inserting records, updating records, deleting records, creating, modifying and dropping tables, views etc.

Basic PL/SQL Programs

1)Write a PL/SQL program to print Hi BEC.

```
SQL> declare

  2   var varchar2(40):='Hi BEC';

  3   begin

  4   dbms_output.put_line(var);

  5   end;

  6 /

Hi BEC
```

2)Write a PL/SQL program to print addition of 2 numbers.

```
PL/SQL procedure successfully completed.

SQL> declare

  2   a number(2):=10;

  3   b number(2):=20;

  4   c number(2);

  5   begin

  6   c:=a+b;

  7   dbms_output.put_line('sum is '||c);

  8   end;

  9   /
```

```
sum is 30
```

3) Write a PL/SQL program to print the area of circle,circumcenter of circle ,diameter of circle.

```
PL/SQL procedure successfully completed.

SQL>

  1  declare
  2  pi number(5,2):=3.14;
  3  d number(5,2);
  4  a number(5,2);
  5  r number(5,2):=5.2;
  6  c number(5,2);
  7  begin
  8  d:=r*2;
  9  c:=2*pi*r;
 10  a:=pi*r*r;
 11  dbms_output.put_line('r'||r);
 12  dbms_output.put_line('d'||d);
 13  dbms_output.put_line('c'||c);
 14  dbms_output.put_line('a'||a);
 15* end;
SQL> /

r5.2
d10.4
c32.66
a84.91
```

4) Write a python to print the sum and read two keyboard.

```
PL/SQL procedure successfully completed.

SQL>

1  declare
  2  a number(5):=&a;
  3  b number(5):=&b;
  4  c number(5);
  5  begin
  6  c:=a+b;
  7 dbms_output.put_line('sum is '||c);
  8* end;
SQL> /
Enter value for a: 10
old  2:  a  number(5):=&a;
new  2: a number(5):=10;
Enter value for b: 23
old  3:  b  number(5):=&b;
new  3:  b  number(5):=23;
sum is 33

PL/SQL procedure successfully completed.
```

PL/SQL Programs

1.Write a PL/SQL program to print employee number of an employee as well as the corresponding MGR NO.

```
  SQL> declare
  2
  3   e1 emp.empno%type;
  4   e2 emp.mgr%type;
  5   begin
  6   e1:=&empno;
  7   select mgr into e2 from emp where empno=e1;
  8   dbms_output.put_line('empno:'||e1 ||'mgr:'||e2);
  9   exception
 10   when no_data_found then
 11   dbms_output.put_line('wrong input');
 12   end;
 13   /
Enter value for empno: 7566
old   6: e1:=&empno;
new   6: e1:=7566;
empno:7566mgr:7839


PL/SQL procedure successfully completed.
```

2. Write a PL/SQL program using FOR/WHILE LOOPS to list out month names and month numbers.

i)using for loop

```
 SQL> declare
  2   d date;
  3   i number;
  4   begin
  5  for i in 0..11
  6  loop
  7  select add_months(to_date('01-jan-05'),i)into d from
dual;
  8
dbms_output.put_line(to_char(d,'month')||''||to_char(d,'mm'));
  9  end loop;
 10  end;
 11  /
january  01
february 02
march    03
april    04
may      05
june     06
july     07
august   08
september09
october  10
november 11
december 12
PL/SQL procedure successfully completed.
```

ii) using while loop

```
SQL> Declare
  2  d date;
  3
  4  i number;
  5
  6  begin
```

```
i:=0;
while i<=11
loop
select add_months(to_date('01-jan-05'),i)into d from dual;
dbms_output.put_line(to_char(d,'month')||' '||to_char(d,'mm'));
i:=i+1;
end loop;
end;
/
january   01
february  02
march     03
april     04
may       05
june      06
july      07
august    08
september 09
october   10
november  11
december  12

PL/SQL procedure successfully completed.
```

3. Write a PL/SQL program to update commission of an employee (employee number as input) As per the following norms.

i) If commission is NULL, make it as 10% of salary

ii) If comm. < 200 make comm. = 200

iii) If comm. <300 make comm. = 300

```
 SQL> declare
   2
   3  c1 emp.comm%type;
   4  s1 emp.sal%type;
   5  e1 emp.empno%type;
   6
   7  begin
   8
   9  e1:=&e1;
  10  select comm,sal into c1,s1 from emp where empno=e1;
  11  if c1 is null then
  12  c1:=s1*(0.1);
  13
  14  elsif c1>200 and c1<300 then
  15  c1:=200;
  16  else
  17  c1:=c1+c1*(0.1);
  18  end if;
  19  dbms_output.put_line('empno is:'||e1);
  20  dbms_output.put_line('comm is:'||c1);
  21  end;
  22  /
Enter value for e1: 7876
old   9: e1:=&e1;
new   9: e1:=7876;
empno is:7876
comm is:110
PL/SQL procedure successfully completed.
```

4.write a PL/SQL program to get no.of employees whose salary is in between given range.

```
Program::

   SQL> declare
  2  losal emp.sal%type;
  3  hisal emp.sal%type;
  4  count1 number;
  5  begin
  6  losal:=&losal;
  7  hisal:=&hisal;
  8  select count(empno) into count1 from emp
  9  where sal between losal and hisal;
 10  dbms_output.put_line('no. of employees:'|| count1);
 11  end;
 12  /
Enter value for losal: 2850
old   6: losal:=&losal;
new   6:  losal:=2850;
Enter value for hisal: 5000
old   7: hisal:=&hisal;
new   7: hisal:=5000;
no. of employees:5


PL/SQL procedure successfully completed.
```

5.Write a PL/SQL program to list out , DEPT NO,DNAME, NO OF EMPLOYEES,MAX(SAL),MIN(SAL), AVG(SAL)In each dept. If a dept has no employees then display“employees are not there in this dept”.

```
 SQL> declare
  2  d1 emp.deptno%type;
  3  dn dept.dname%type;
  4  cn number;
  5  mi number;
  6  mx number;
  7  ag number(10,4);
  8  cnt exception;
  9  begin
 10  d1:=&d1;
 11  select dname into dn from dept where deptno=d1;
 12  select count(empno),min(sal),max(sal),avg(sal) into
 13  cn,mi,mx,ag from emp where deptno=d1;
 14  if cn=0 then
 15  raise cnt;
 16  else
 17  dbms_output.put_line('deptno:'||d1);
 18  dbms_output.put_line('dname:'||dn);
 19  dbms_output.put_line('no of employees:'||cn);
 20  dbms_output.put_line('min sal:'||mi);
 21  dbms_output.put_line('max sal:'||mx);
 22  dbms_output.put_line('avg sal:'||ag);
 23  end if;
 24  exception
 25  when cnt then
 26  dbms_output.put_line('there is no employees in that
dept');
 27  end;
 28  /
Enter value for d1: 10
old  10: d1:=&d1;
new  10: d1:=10;
```

```
deptno:10

dname:ACCOUNTING

no of employees:3

min sal:1300

max sal:5000

avg sal:2916.6667

PL/SQL procedure successfully completed.
```

Functions and Procedure

Function is one of the two available Named Blocks in Pl/SQL, the other being Procedure. In PL/SQL, a function takes one or more parameter and returns one value.

The main difference between a PL/SQL function and a PL/SQL procedure is that a function returns the value while a procedure does not.

A procedure is a group of PL/SQL statements that you can call by its name. These sub-programs can not return a value directly and are mainly used to perform a particular task. Stored procedures offer useful in the areas of memory allocation, development, performance, security and integrity.

1.Write a program to check whether the given number is prime or not.

```
SQL> declare
  2  n number;
  3  i number;
  4  flag number;
  5  begin
  6  i:=2;
  7  flag:=1;
  8  n:=&n;
  9  for i in 2..n/2
 10  loop
 11  if mod(n,i)=0
 12  then
 13  flag:=0;
 14  exit;
 15  end if;
 16  end loop;
 17  if flag=1
 18  then
 19  dbms_output.put_line('prime');
 20  else
 21  dbms_output.put_line('not prime');
 22   end if;
 23   end;
 24   /
Enter value for n: 7
old   9: n:=&n;
new   9: n:=7;
prime

PL/SQL procedure successfully completed.
```

2. Write a program to reverse the given number.

```
SQL> declare
  2   str1 varchar2(50):='&str';s
  3   str2 varchar2(50);
  4   len number;
  5   i number;
  6   begin
  7   len:=length(str1);
  8   for i in reverse 1..len
  9    loop
 10   str2:=str2 || substr(str1,i,1);
 11   end loop;
 12   dbms_output.put_line('Reverse of String is:'||str2);
 13   end;
 14   /
Enter value for str: bapatla
old   2: str1 varchar2(50):='&str';
new   2: str1 varchar2(50):='bapatla';
Reverse of String is:altapab
```

```
PL/SQL procedure successfully completed.
```

3. Write a program to find the factorial of given number.

```
SQL> declare
  2  n number;
  3  fac number:=1;
  4  i number;
  5  begin
  6  n:=&n;
  7  for i in 1..n
  8   loop
  9   fac:=fac*i;
 10  end loop;
 11  dbms_output.put_line('factorial='||fac);
 12  end;
 13  /
Enter value for n: 5
old   7: n:=&n;
new   7: n:=5;
factorial=120

PL/SQL procedure successfully completed.
```

Cursors

A **cursor** is a pointer to this context area. PL/SQL controls the context area through a cursor. A cursor holds the rows (one or more) returned by a SQL statement. The set of rows the cursor holds is referred to as the **active set**.

You can name a cursor so that it could be referred to in a program to fetch and process the rows returned by the SQL statement, one at a time. There are two types of cursors −

- Implicit cursors
- explicit cursors

Implicit Cursors

Implicit cursors are automatically created by Oracle whenever an SQL statement is executed, when there is no explicit cursor for the statement. Programmers cannot control the implicit cursors and the information in it.

Whenever a DML statement (INSERT, UPDATE and DELETE) is issued, an implicit cursor is associated with this statement. For INSERT operations, the cursor holds the data that needs to be inserted. For UPDATE and DELETE operations, the cursor identifies the rows that would be affected.

Attributes in implicit cursors:

%FOUND

Returns TRUE if an INSERT, UPDATE, or DELETE statement affected one or more rows or a SELECT INTO statement returned one or more rows. Otherwise, it returns FALSE.

%NOTFOUND

The logical opposite of %FOUND. It returns TRUE if an INSERT, UPDATE, or DELETE statement affected no rows, or a SELECT INTO statement returned no rows. Otherwise, it returns FALSE.

%ISOPEN

Always returns FALSE for implicit cursors, because Oracle closes the SQL cursor automatically after executing its associated SQL statement.

%ROWCOUNT

Returns the number of rows affected by an INSERT, UPDATE, or DELETE statement, or returned by a SLECT INTO statement.

1. Write a PL/SQL program to list employee names whose salar is more than their Manager (to whom the/report) salary.

```
SQL> set serveroutput ons

SQL> declare
```

```
  2  cursor c1 is
  3  select e1.ename,e1.sal,e2.ename,e2.sal
  4  from emp e1,emp e2 where
  5   e1.mgr=e2.empno
  6   and e1.sal>e2.sal;
  7  e1 emp.ename%type;
  8  e2 emp.sal%type;
  9  e3 emp.ename%type;
 10  e4 emp.sal%type;
 11  begin
 12   open c1;
 13   loop
 14  fetch c1 into e1,e2,e3,e4;exit when c1%notfound;
 15 dbms_output.put_line(e1||'---'||e2||'---'||e3||'---
'||e4);
 16  end loop;
 17  close c1;
 18  end;
 19  /
FORD---3000---JONES---2975
SCOTT---3000---JONES---2975
PL/SQL procedure successfully completed.
```

2. Write a PL/SQL program to list names of Employees in Alphabetical order along with the position where position is the position of employee in the list sorted by salary in decreasing order.

```
SQL> declare
  2  sal1 emp.sal%type;
  3  c number;
  4   cursor c1 is
  5  select ename from emp order by ename;
```

```
cursor c2 is
select ename,sal from emp order by sal desc;
begin
for i in c1 loop
for j in c2 loop
c:=c2%rowcount;
if(i.ename=j.ename) then
 dbms_output.put_line(i.ename||''||c);
end if;
end loop;end loop;
end;
/
ADAMS12
ALLEN7
BLAKE5
CLARK6
FORD2
JAMES13
JONES4
KING1
MARTIN10
MILLER9
SCOTT3
SMITH14
TURNER8
WARD11
PL/SQL procedure successfully completed.
```

Explicit Cursors

Explicit cursors are programmer-defined cursors for gaining more control over the **context area**. An explicit cursor should be defined in the declaration section of the PL/SQL Block. It is created on a SELECT Statement which returns more than one row.

Working with an explicit cursor includes the following steps −

- Declaring the cursor for initializing the memory
- Opening the cursor for allocating the memory
- Fetching the cursor for retrieving the data
- Closing the cursor to release the allocated memory

1. Write a PL/SQL program to display TOP 3 earners of the company (list names& salary).

```
SQL> declare
ena emp.ename%type;
sa emp.sal%type;
 cursor c1 is
select ename,sal from emp order by sal desc;
begin
open c1;
loop
fetch c1 into ena,sa;
exit when c1%rowcount>3;
dbms_output.put_line(ena||'........... '||sa);
end loop;
close c1;
end;
/
KING............5000
FORD............3000
SCOTT...........3000
PL/SQL procedure successfully completed.
```

2. List empname, manager name chain for each employee as follows

SMITH -------FORD ------ JONES----------- KIN

```
SQL> declare
  2  cursor c1 is
  3  select empno,ename,mgr from emp;
  4  eno emp.empno%type;
  5  mno emp.mgr%type;
  6  ena emp.ename%type;
  7   begin
  8  dbms_output.put_line('emp names and manager chain');
  9  dbms_output.put_line('king');
 10  for m in c1 loop
 11  eno:=m.empno;
 12  mno:=m.mgr;
 13  while mno is not null
 14  loop
 15  select ename into ena from emp where empno=eno;
 16  dbms_output.put(ena);
 17  select mgr into mno from emp where empno=eno;
 18  eno:=mno;
 19  end loop;
 20  dbms_output.put_line(' ');
 21  end loop;
 2j2  end;
 23  /
emp names and manager chain
king
BLAKEKING
CLARKKING
```

```
JONESKING

MARTINBLAKEKING

ALLENBLAKEKING

TURNERBLAKEKING

JAMESBLAKEKING

WARDBLAKEKING

FORDJONESKING

SMITHFORDJONESKING

SCOTTJONESKING

ADAMSSCOTTJONESKING

MILLERCLARKKING

PL/SQL procedure successfully completed.
```

3. List the empno,enmae,salary for each employee.

```
SQL> DECLARE
     c_empno emp.empno%type;
     c_ename emp.ename%type;
     c_sal emp.sal%type;
     CURSOR c_emp is
     SELECT empno, ename,sal FROM emp;
     BEGIN
     OPEN c_emp;
     LOOP
     FETCH c_emp into c_empno, c_ename, c_sal;
     EXIT WHEN c_emp%notfound;
     dbms_output.put_line(c_empno || ' ' || c_ename || ' ' ||
c_sal);
     END LOOP;
     CLOSE c_emp;
     END;
```

```
 16  /



7839 KING   5500

7698 BLAKE  3350

7782 CLARK  2950

7566 JONES  3475

7654 MARTIN 1750

7499 ALLEN  2100

7844 TURNER 2000

7900 JAMES  1450

7521 WARD   1750

7902 FORD   3500

7369 SMITH  1300

7788 SCOTT  3500

7876 ADAMS  1600

7934 MILLER 1800

PL/SQL procedure successfully completed.
```

Packages

Packages are schema objects that groups logically related PL/SQL types, variables, and subprograms.

A package will have two mandatory parts −

1.Package Specification

- The specification is the interface to the package. It just **DECLARES** the types, variables, constants, exceptions, cursors, and subprograms that can be referenced from outside the package. In other words, it contains all information about the content of the package, but excludes the code for the subprograms.
- All objects placed in the specification are called **public** objects. Any subprogram not in the package specification but coded in the package body is called a **private** object.

2.Package Body

- The package body has the codes for various methods declared in the package specification and other private declarations, which are hidden from the code outside the package.
- The **CREATE PACKAGE BODY** Statement is used for creating the package body.

```
SQL> create table customers(ID number(2),Name varchar2(20),Age
number(3),Address varchar2(50),Salary number(7,2));

Table created.

SQL> desc customers;

 Name                                          Null?    Type

 ----------------------------------------- -------- ----------
--------------------------

 ID                                                     NUMBER(2)

 NAME
VARCHAR2(20)

 AGE                                                    NUMBER(3)

 ADDRESS
VARCHAR2(50)

 SALARY
NUMBER(7,2)
```

```
SQL> insert into customers
values(1,'ramesh',32,'Ahmedabad',3000.00);

1 row created.

SQL> /

  insert into customer1 values(2,'khilan',25,'delhi',3000.00);

1 row created.

SQL> /

   insert into customer1 values(3,'kaushik',23,'kota',3000.00)

1 row created.

SQL> /

  insert into customer1
values(4,'chaitali',25,'mumbai',7500.00);

1 row created.

SQL> /

  insert into customer1
values(5,'hardik',27,'bhopal',9500.00);

1 row created.

SQL> /

  insert into customer1 values(6,'komal',22,'MP',5500.00)

1 row created.

SQL> set linesize 200;

SQL> select * from customers;


        ID NAME                            AGE ADDRESS
SALARY

---------- -------------------- ---------- -------------------
------------------------------ ------

         1 ramesh                           32 Ahmedabad
3000

         2 khilan                           25 delhi
3000
```

```
         3 kaushik                       23 kota
3000

         4 chaitali                      25 mumbai
7500

         5 hardik                        27
bhopal                                                    9500

         6 komal                         22 MP
5500


6 rows selected.

SQL> commit;

Commit complete.
```

Package Specification

```
SQL> CREATE PACKAGE cust_sal AS
  2     PROCEDURE find_sal(c_id customers.id%type);
  3  END cust_sal;
  4  /
Output:
Package created.
```

Package Body

```
SQL> CREATE OR REPLACE PACKAGE BODY cust_sal AS
   2  PROCEDURE find_sal(c_id customers.id%TYPE) IS
   3  c_sal customers.salary%TYPE;
   4  BEGIN
   5  SELECT salary INTO c_sal
   6  FROM customers
   7  WHERE id = c_id;
   8  dbms_output.put_line('Salary: '|| c_sal);
   9  END find_sal;
  10  END cust_sal;
  11  /
```

Output:

```
Package body created.
```

Using the Package Elements

```
SQL>DECLARE
   2  code customers.id%type := &cc_id;
   3  BEGIN
   4  cust_sal.find_sal(code);
   5  END;
   6    /
```

Output:

```
Enter value for cc_id: 1
Salary: 3000
PL/SQL procedure successfully completed.
```

Package Specification

```
 SQL>CREATE OR REPLACE PACKAGE c_package AS
   2  -- Adds a customer
   3   PROCEDURE addCustomer(c_id   customers.id%type,
   4  c_name customers.Name%type,
   5   c_age  customers.age%type,
   6   c_addr customers.address%type,
   7   c_sal  customers.salary%type);
   8   -- Removes a customer
   9    PROCEDURE delCustomer(c_id  customers.id%TYPE);
  10  --Lists all customers
  11   PROCEDURE listCustomer;
  12   END c_package;
  13   /
```

Output:

```
Package created.
```

Creating the Package Body

```
SQL>CREATE OR REPLACE PACKAGE BODY c_package AS
   2  PROCEDURE addCustomer(c_id  customers.id%type,
   3  c_name customers.Name%type,
   4  c_age  customers.age%type,
   5  c_addr  customers.address%type,
   6  c_sal   customers.salary%type)
   7  IS
   8  BEGIN
   9  INSERT INTO customers (id,name,age,address,salary)
   10   VALUES(c_id, c_name, c_age, c_addr, c_sal);
   11 END addCustomer;
   12  PROCEDURE delCustomer(c_id   customers.id%type) IS
   13   BEGIN
   14    DELETE FROM customers
   15   WHERE id = c_id;
   16   END delCustomer;
   17  PROCEDURE listCustomer IS
   18  CURSOR c_customers is
   19 SELECT  name FROM customers;
   20  TYPE c_list is TABLE OF customers.Name%type;
   21  name_list c_list := c_list();
   22  counter integer :=0;
   23  BEGIN
   24  FOR n IN c_customers LOOP
   25  counter := counter +1;
   26   name_list.extend;
   27  name_list(counter) := n.name;
   28   dbms_output.put_line('Customer(' ||counter||
'        ')'||name_list(counter));
```

```
 END LOOP;
END listCustomer;
END c_package;
/
```

Output:

```
Package body created.
```

Triggers

Triggers are stored programs, which are automatically executed or fired when some events occur. Triggers are, in fact, written to be executed in response to any of the following events −

- A **database manipulation (DML)** statement (DELETE, INSERT, or UPDATE)
- A **database definition (DDL)** statement (CREATE, ALTER, or DROP).
- A **database operation** (SERVERERROR, LOGON, LOGOFF, STARTUP, or SHUTDOWN).

Triggers can be defined on the table, view, schema, or database with which the event is associated.

Benefits of Triggers

Triggers can be written for the following purposes −

- Generating some derived column values automatically
- Enforcing referential integrity
- Event logging and storing information on table access
- Auditing
- Synchronous replication of tables
- Imposing security authorizations
- Preventing invalid transactions

```
SQL >CREATE OR REPLACE TRIGGER display_salary_changes
2   BEFORE DELETE OR INSERT OR UPDATE ON customers
3   FOR EACH ROW
4   WHEN (NEW.ID > 0)
5   DECLARE
6   sal_diff number;
7   BEGIN
8   sal_diff := :NEW.salary  - :OLD.salary;
9    dbms_output.put_line('Old salary: ' || :OLD.salary);
10   dbms_output.put_line('New salary: ' || :NEW.salary);
11   dbms_output.put_line('Salary difference: ' || sal_diff);
12   END;
13   /
Trigger created.
```