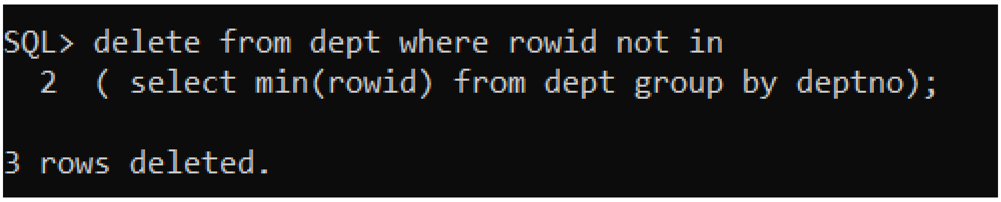
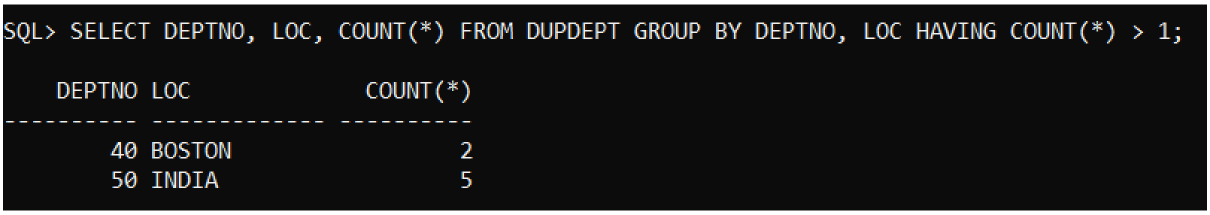
Interview Questions and Answers

1. Remove duplicates from a table
   SELECT * FROM DUPDEPT WHERE ROWID NOT IN (SELECT MIN(ROWID)FROM DUPDEPT GROUP BY DEPTNO);



```
SQL> delete from dept where rowid not in
  2  ( select min(rowid) from dept group by deptno);

3 rows deleted.
```

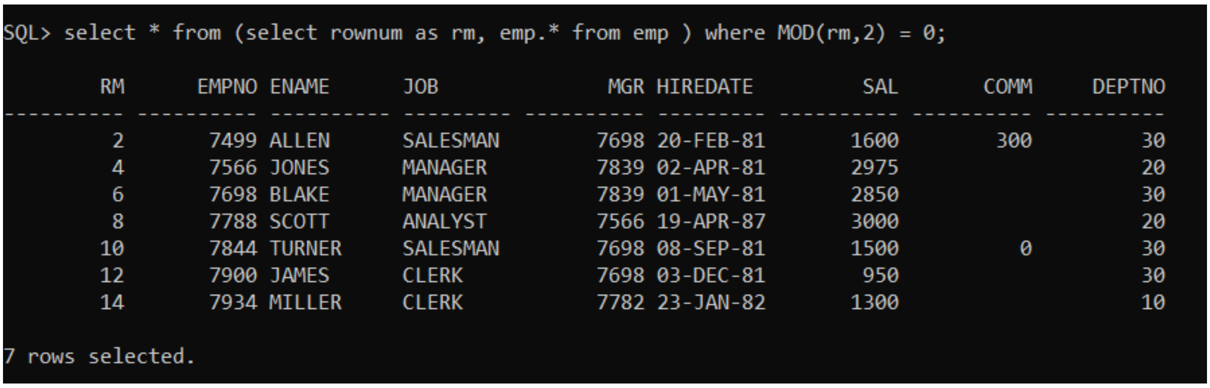
2. 4 Ways to find and display duplicate record in SQL
   a. Using Rowid function (refer question 1)
   b. Using Count function
      SELECT DEPTNO, LOC , COUNT(*) FROM DUPDEPT GROUP BY DEPTNO, LOC HAVING COUNT(*)>1;



```
SQL> SELECT DEPTNO, LOC, COUNT(*) FROM DUPDEPT GROUP BY DEPTNO, LOC HAVING COUNT(*) > 1;

    DEPTNO LOC             COUNT(*)
---------- ------------- ----------
        40 BOSTON                 2
        50 INDIA                  5
```

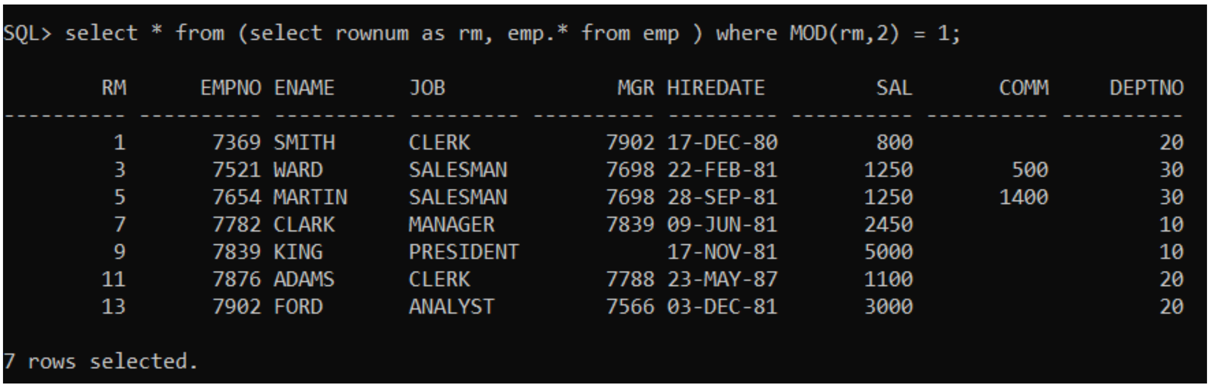
   c. Using Lag function and Case statement
   d. Using Row_num and Partition Function

3. Display every alternate record in a table (even rows)
   SELECT * FROM (SELECT ROWNUM AS RM, EMP.* FROM EMP) WHERE MOD(RM,2) =0;



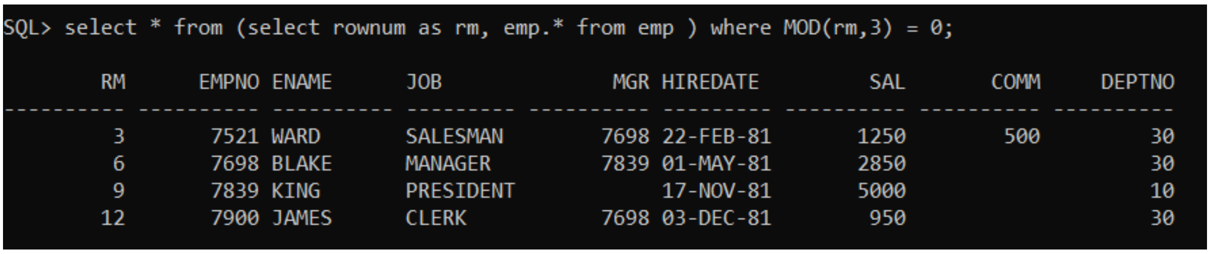
```
SQL> select * from (select rownum as rm, emp.* from emp ) where MOD(rm,2) = 0;

        RM      EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- ---------- --------- ---------- --------- ---------- ---------- ----------
         2       7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
         4       7566 JONES      MANAGER         7839 02-APR-81       2975                    20
         6       7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
         8       7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
        10       7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
        12       7900 JAMES      CLERK           7698 03-DEC-81        950                    30
        14       7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

7 rows selected.
```

4. Display every alternate record in a table (odd rows)
   SELECT * FROM (SELECT ROWNUM AS RM, EMP.* FROM EMP) WHERE MOD(RM,2) =1;



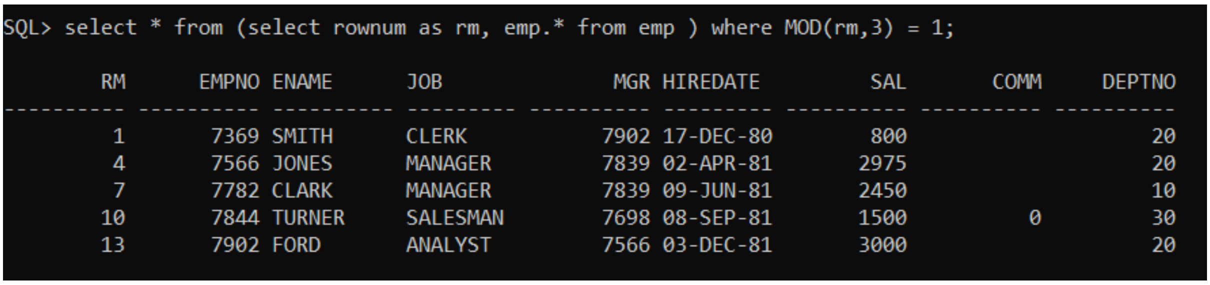
```
SQL> select * from (select rownum as rm, emp.* from emp ) where MOD(rm,2) = 1;

        RM      EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- ---------- --------- ---------- --------- ---------- ---------- ----------
         1       7369 SMITH      CLERK           7902 17-DEC-80        800                    20
         3       7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
         5       7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
         7       7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
         9       7839 KING       PRESIDENT            17-NOV-81       5000                    10
        11       7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
        13       7902 FORD       ANALYST         7566 03-DEC-81       3000                    20

7 rows selected.
```

5. Display every 3rd record in a table (even rows)
   SELECT * FROM (SELECT ROWNUM AS RM, EMP.* FROM EMP) WHERE MOD(RM,3) =0;



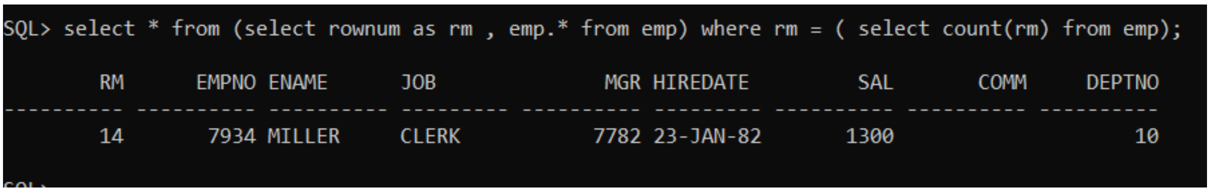
```
SQL> select * from (select rownum as rm, emp.* from emp ) where MOD(rm,3) = 0;

        RM      EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- ---------- --------- ---------- --------- ---------- ---------- ----------
         3       7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
         6       7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
         9       7839 KING       PRESIDENT            17-NOV-81       5000                    10
        12       7900 JAMES      CLERK           7698 03-DEC-81        950                    30
```

6. Display every 3rd record in a table (odd rows)
   SELECT * FROM (SELECT ROWNUM AS RM, EMP.* FROM EMP) WHERE MOD(RM,3) =1;



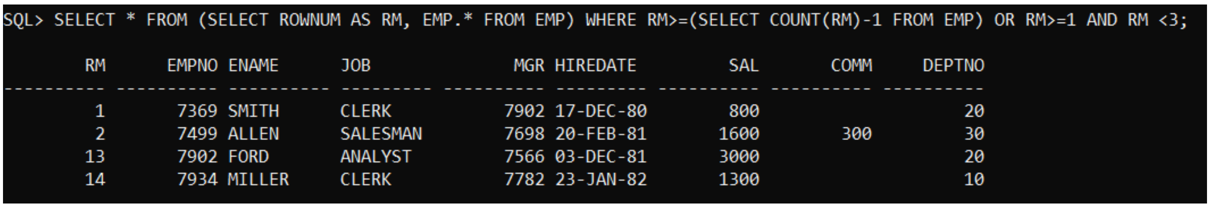
```
SQL> select * from (select rownum as rm, emp.* from emp ) where MOD(rm,3) = 1;

        RM      EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- ---------- --------- ---------- --------- ---------- ---------- ----------
         1       7369 SMITH      CLERK           7902 17-DEC-80        800                    20
         4       7566 JONES      MANAGER         7839 02-APR-81       2975                    20
         7       7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
        10       7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
        13       7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
```

7. Display the last record in a table
   SELECT * FROM (SELECT ROWNUM AS RM, EMP.* FROM EMP) WHERE RM= (SELECT COUNT (*) FROM EMP);



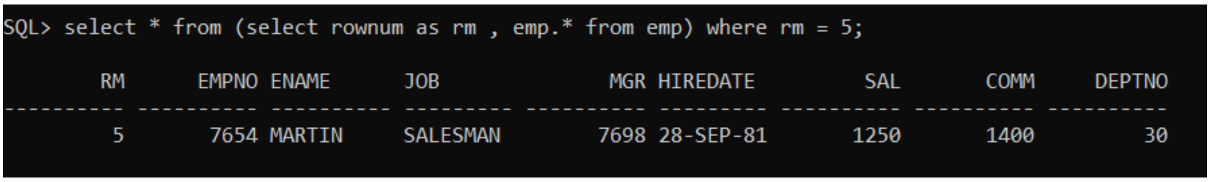
```
SQL> select * from (select rownum as rm , emp.* from emp) where rm = ( select count(rm) from emp);

        RM      EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- ---------- --------- ---------- --------- ---------- ---------- ----------
        14       7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

SQL>
```

8. Display the First 2 row and last 2 rows in a table



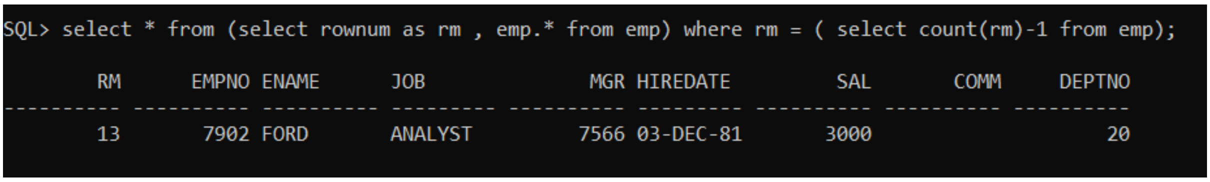
```
SQL> SELECT * FROM (SELECT ROWNUM AS RM, EMP.* FROM EMP) WHERE RM>=(SELECT COUNT(RM)-1 FROM EMP) OR RM>=1 AND RM <3;

        RM      EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- ---------- --------- ---------- --------- ---------- ---------- ----------
         1       7369 SMITH      CLERK           7902 17-DEC-80        800                    20
         2       7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
        13       7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
        14       7934 MILLER     CLERK           7782 23-JAN-82       1300                    10
```

9. Display the 5th record in a table



```
SQL> select * from (select rownum as rm , emp.* from emp) where rm = 5;

        RM      EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- ---------- --------- ---------- --------- ---------- ---------- ----------
         5       7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
```

10. Display the last but one record in a table

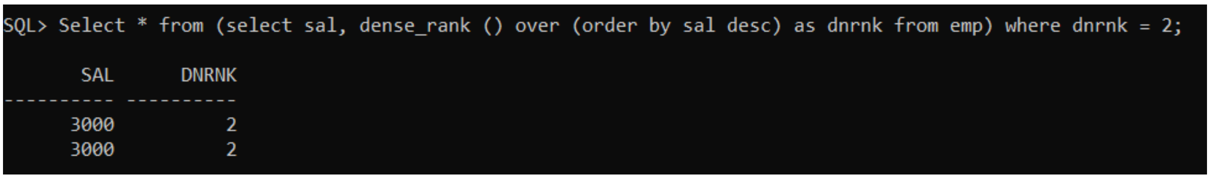


```
SQL> select * from (select rownum as rm , emp.* from emp) where rm = ( select count(rm)-1 from emp);

        RM      EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- ---------- --------- ---------- --------- ---------- ---------- ----------
        13       7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
```

Rank, Dense_Rank and Rownum are all Analytical function

11. Display the employee's details who are having second highest salary.

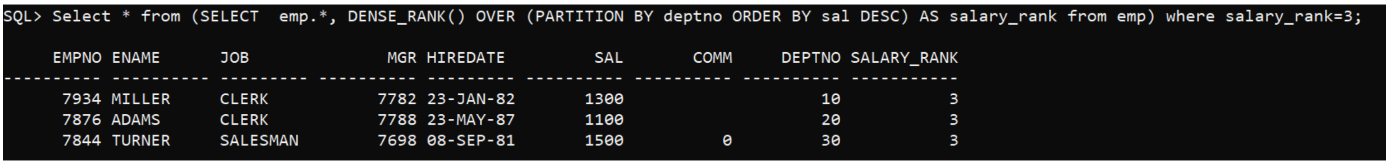
Ans: SELECT * FROM (SELECT SAL, DENSE_RANK () OVER (ORDER BY SAL DESC) AS DRNK FROM EMP) WHERE DRNK = 2;



```
SQL> Select * from (select sal, dense_rank () over (order by sal desc) as dnrnk from emp) where dnrnk = 2;

       SAL      DNRNK
---------- ----------
      3000          2
      3000          2
```

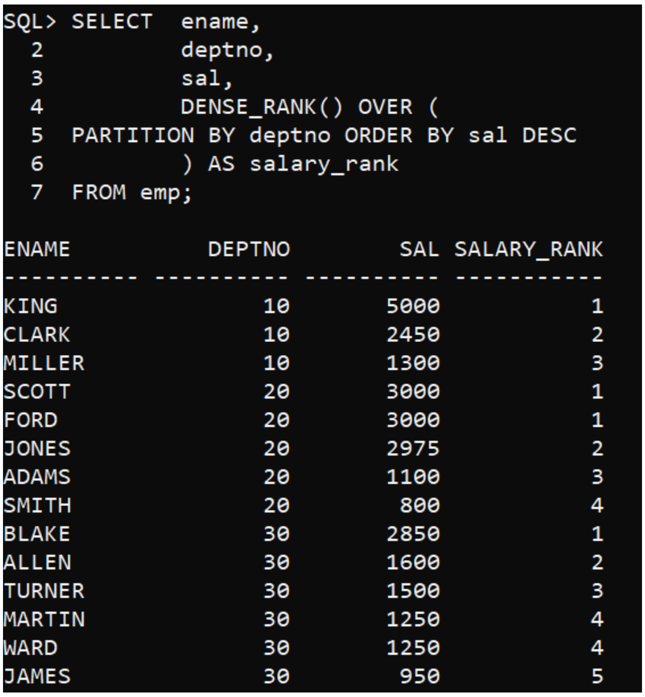
12. Find the 3rd Highest Sal from each dept

Ans: SELECT * FROM (SELECT EMP.*, DENSE_RANK () OVER (PARTITION BY DEPTNO ORDER BY SAL DESC) AS salary_rank FROM EMP) WHERE salary_rank = 3;



```
SQL> Select * from (SELECT  emp.*, DENSE_RANK() OVER (PARTITION BY deptno ORDER BY sal DESC) AS salary_rank from emp) where salary_rank=3;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO SALARY_RANK
---------- ---------- --------- ---------- --------- ---------- ---------- ---------- -----------
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10           3
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20           3
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30           3
```

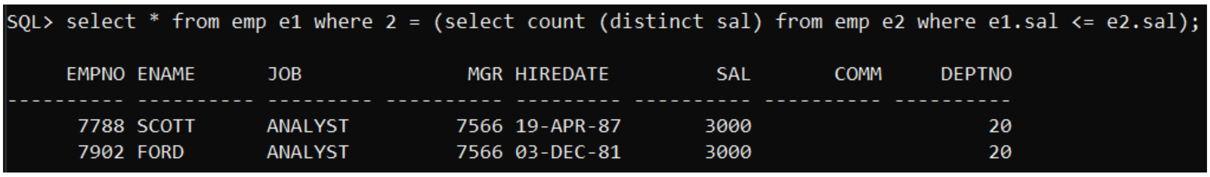
13. SELECT ENAME, DEPTNO, SAL, DENSE_RANK () OVER (PARTITION BY DEPTNO ORDER BY SAL DESC) AS salary_rank FROM EMP;



```
SQL> SELECT  ename,
             deptno,
             sal,
             DENSE_RANK() OVER (
     PARTITION BY deptno ORDER BY sal DESC
             ) AS salary_rank
     FROM emp;

ENAME          DEPTNO        SAL SALARY_RANK
---------- ---------- ---------- -----------
KING               10       5000           1
CLARK              10       2450           2
MILLER             10       1300           3
SCOTT              20       3000           1
FORD               20       3000           1
JONES              20       2975           2
ADAMS              20       1100           3
SMITH              20        800           4
BLAKE              30       2850           1
ALLEN              30       1600           2
TURNER             30       1500           3
MARTIN             30       1250           4
WARD               30       1250           4
JAMES              30        950           5
```

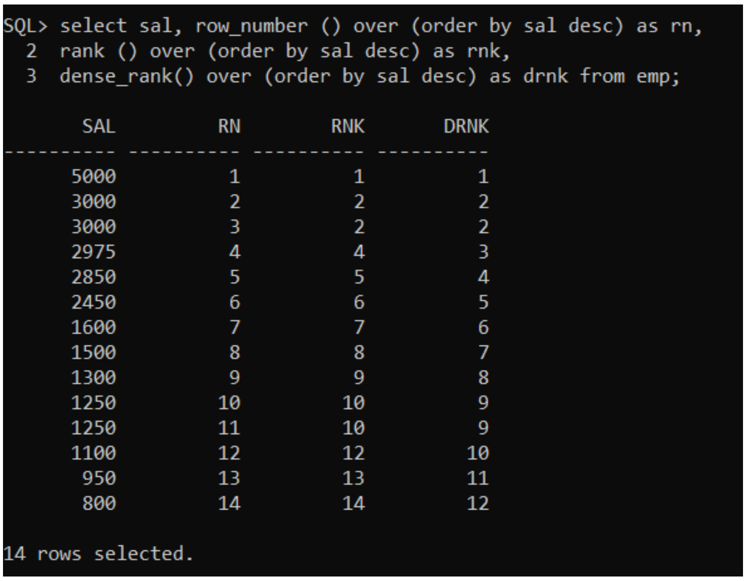
14. Find the 2nd Highest Sal using Joins in Oracle

Ans: SELECT * FROM EMP E1 WHERE 2 = (SELECT COUNT (DISTINCT SAL) EMP E2 WHERE E1.SAL <= E2.SAL);



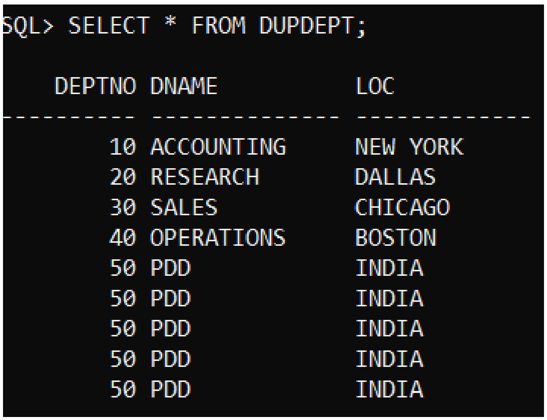
```
SQL> select * from emp e1 where 2 = (select count (distinct sal) from emp e2 where e1.sal <= e2.sal);

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
```

15. Implement Ranking & Dense Ranking functions, why do we need Analytical functions?

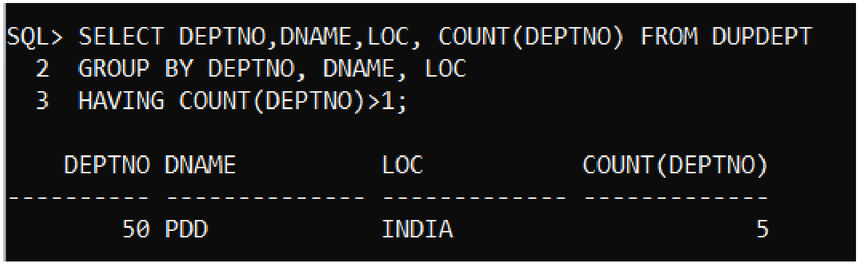


```
SQL> select sal, row_number () over (order by sal desc) as rn,
     rank () over (order by sal desc) as rnk,
     dense_rank() over (order by sal desc) as drnk from emp;

       SAL         RN        RNK       DRNK
---------- ---------- ---------- ----------
      5000          1          1          1
      3000          2          2          2
      3000          3          2          2
      2975          4          4          3
      2850          5          5          4
      2450          6          6          5
      1600          7          7          6
      1500          8          8          7
      1300          9          9          8
      1250         10         10          9
      1250         11         10          9
      1100         12         12         10
       950         13         13         11
       800         14         14         12

14 rows selected.
```

16. Display the duplicate record in a table.

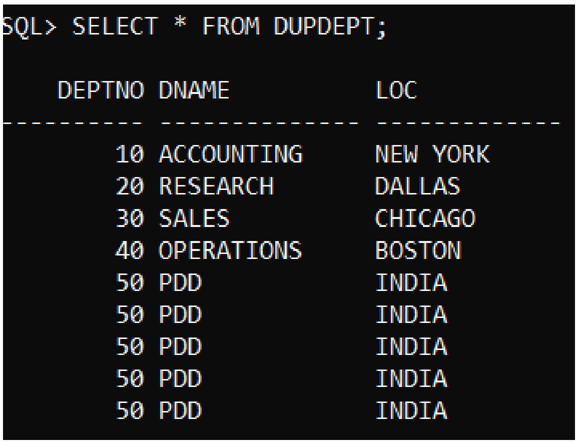


```
SQL> SELECT * FROM DUPDEPT;

    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       DALLAS
        30 SALES          CHICAGO
        40 OPERATIONS     BOSTON
        50 PDD            INDIA
        50 PDD            INDIA
        50 PDD            INDIA
        50 PDD            INDIA
        50 PDD            INDIA
```

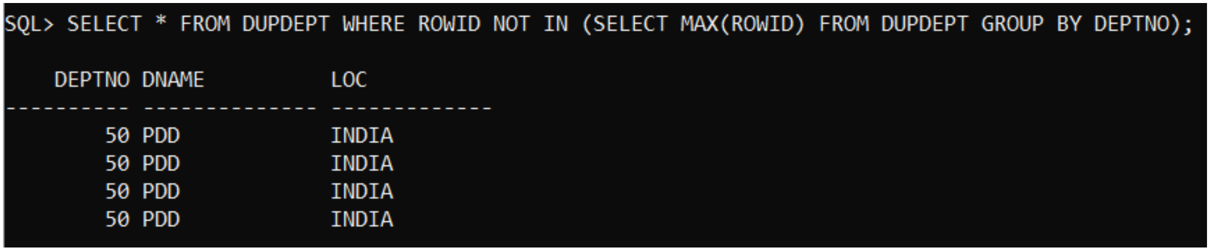


```
SQL> SELECT DEPTNO,DNAME,LOC, COUNT(DEPTNO) FROM DUPDEPT
  2  GROUP BY DEPTNO, DNAME, LOC
  3  HAVING COUNT(DEPTNO)>1;

    DEPTNO DNAME          LOC           COUNT(DEPTNO)
---------- -------------- ------------- -------------
        50 PDD            INDIA                     5
```

17. Display the duplicate records in a table

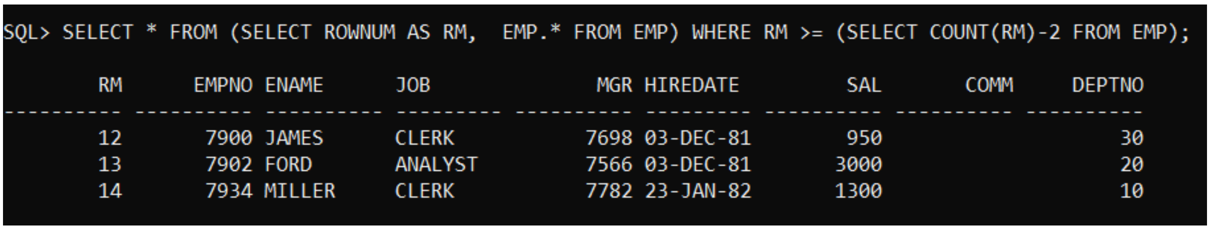


```
SQL> SELECT * FROM DUPDEPT;

    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       DALLAS
        30 SALES          CHICAGO
        40 OPERATIONS     BOSTON
        50 PDD            INDIA
        50 PDD            INDIA
        50 PDD            INDIA
        50 PDD            INDIA
        50 PDD            INDIA
```



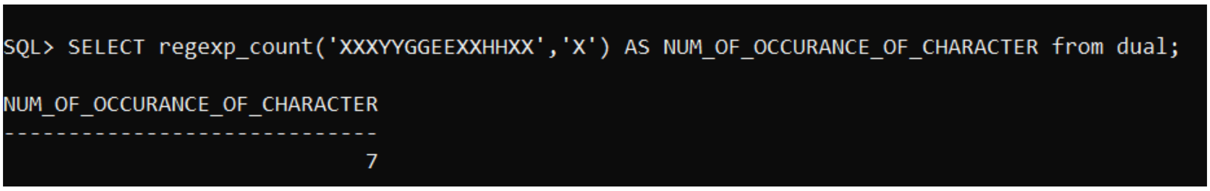
```
SQL> SELECT * FROM DUPDEPT WHERE ROWID NOT IN (SELECT MAX(ROWID) FROM DUPDEPT GROUP BY DEPTNO);

    DEPTNO DNAME          LOC
---------- -------------- -------------
        50 PDD            INDIA
        50 PDD            INDIA
        50 PDD            INDIA
        50 PDD            INDIA
```

18. Display the last 3 records from the EMP table.



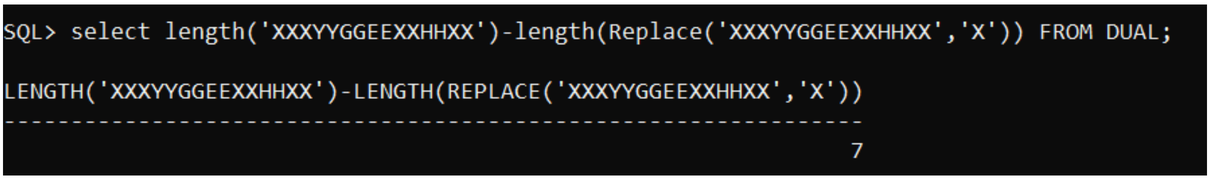
```
SQL> SELECT * FROM (SELECT ROWNUM AS RM,  EMP.* FROM EMP) WHERE RM >= (SELECT COUNT(RM)-2 FROM EMP);

        RM      EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- ---------- --------- ---------- --------- ---------- ---------- ----------
        12       7900 JAMES      CLERK           7698 03-DEC-81        950                    30
        13       7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
        14       7934 MILLER     CLERK           7782 23-JAN-82       1300                    10
```

19. Count the number of occurrences of any character in a string
X in the below example
Method 1



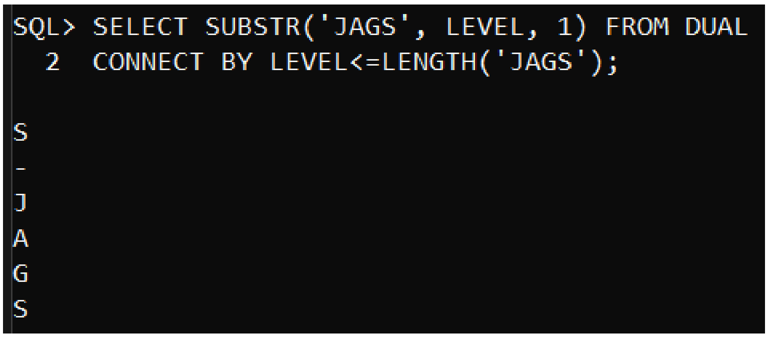
```
SQL> SELECT regexp_count('XXXYYGGEEXXHHXX','X') AS NUM_OF_OCCURANCE_OF_CHARACTER from dual;

NUM_OF_OCCURANCE_OF_CHARACTER
-----------------------------
                            7
```

Method 2



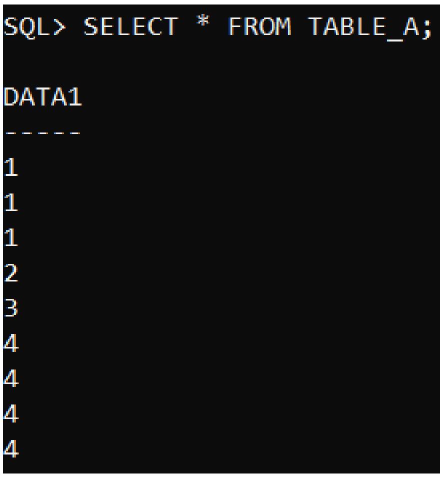
```
SQL> select length('XXXYYGGEEXXHHXX')-length(Replace('XXXYYGGEEXXHHXX','X')) FROM DUAL;

LENGTH('XXXYYGGEEXXHHXX')-LENGTH(REPLACE('XXXYYGGEEXXHHXX','X'))
----------------------------------------------------------------
                                                               7
```

20. Print the input string JAGS as below output
J
A
G
S
Ans:

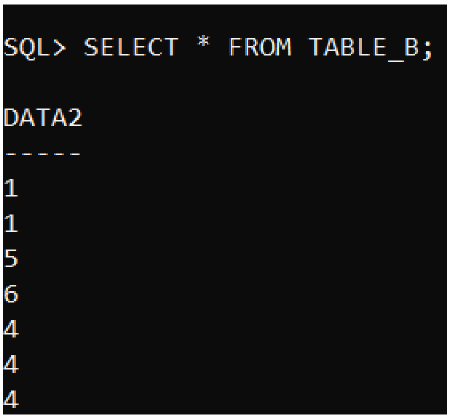


```
SQL> SELECT SUBSTR('JAGS', LEVEL, 1) FROM DUAL
  2  CONNECT BY LEVEL<=LENGTH('JAGS');

S
-
J
A
G
S
```

21. There are 2 tables as below



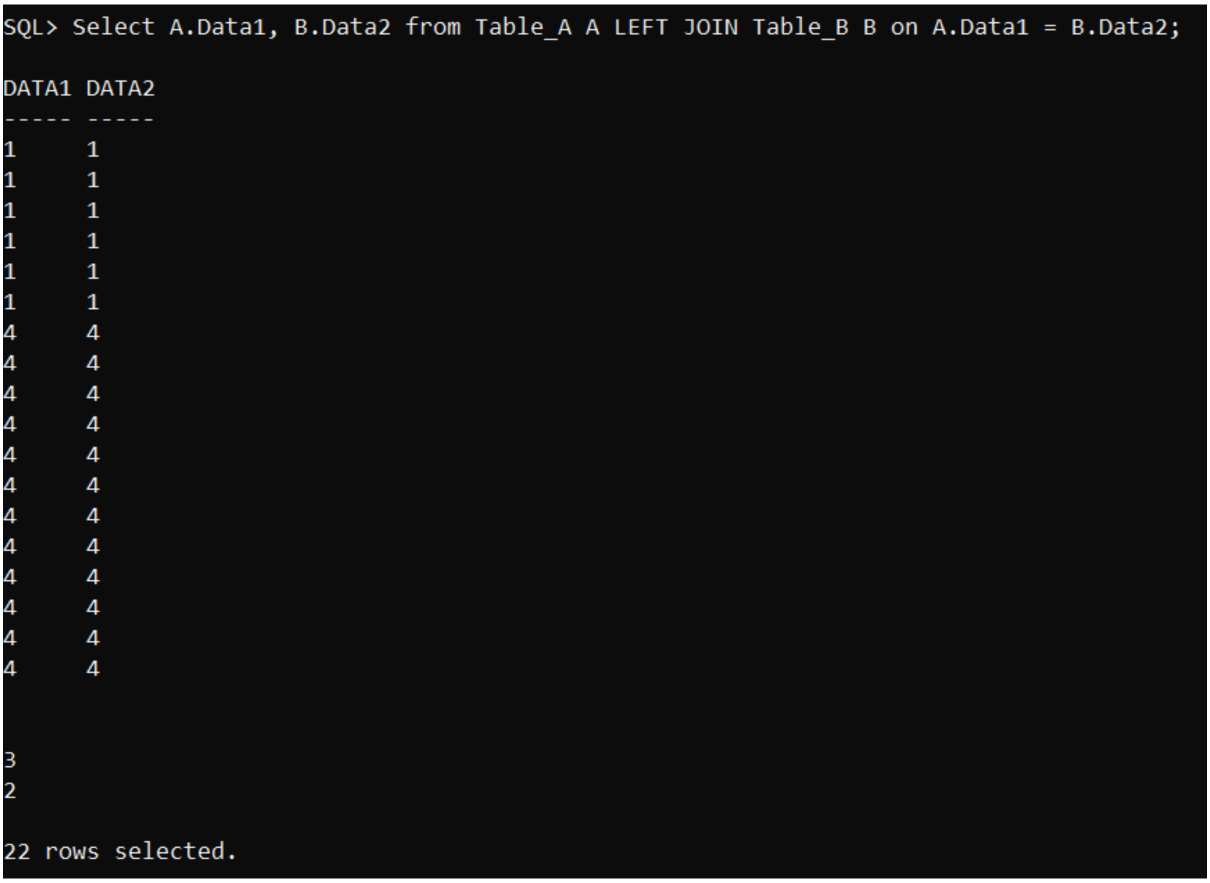
```
SQL> SELECT * FROM TABLE_A;

DATA1
-----
1
1
1
2
3
4
4
4
4
```



```
SQL> SELECT * FROM TABLE_B;

DATA2
-----
1
1
5
6
4
4
4
```

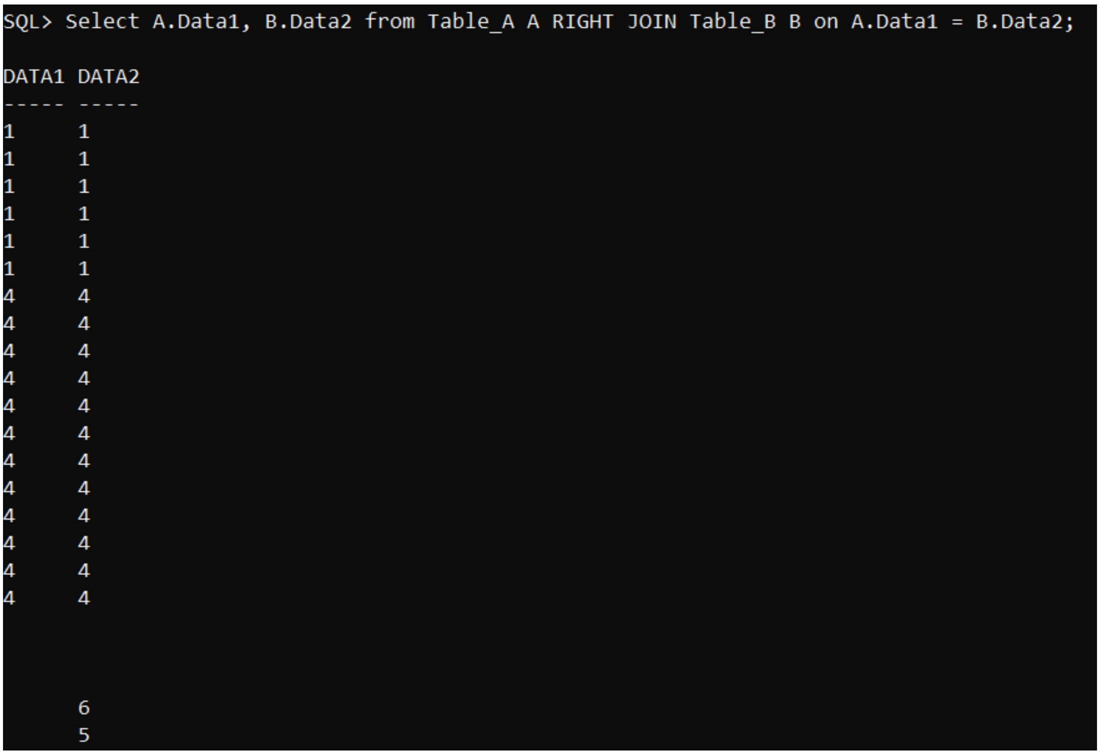
What is the Output of Left Join, Right Join, Inner Join and Full Outer Join?
Ans:

Left Join will give 22 records as below (Here blank row is also count as one Rec)



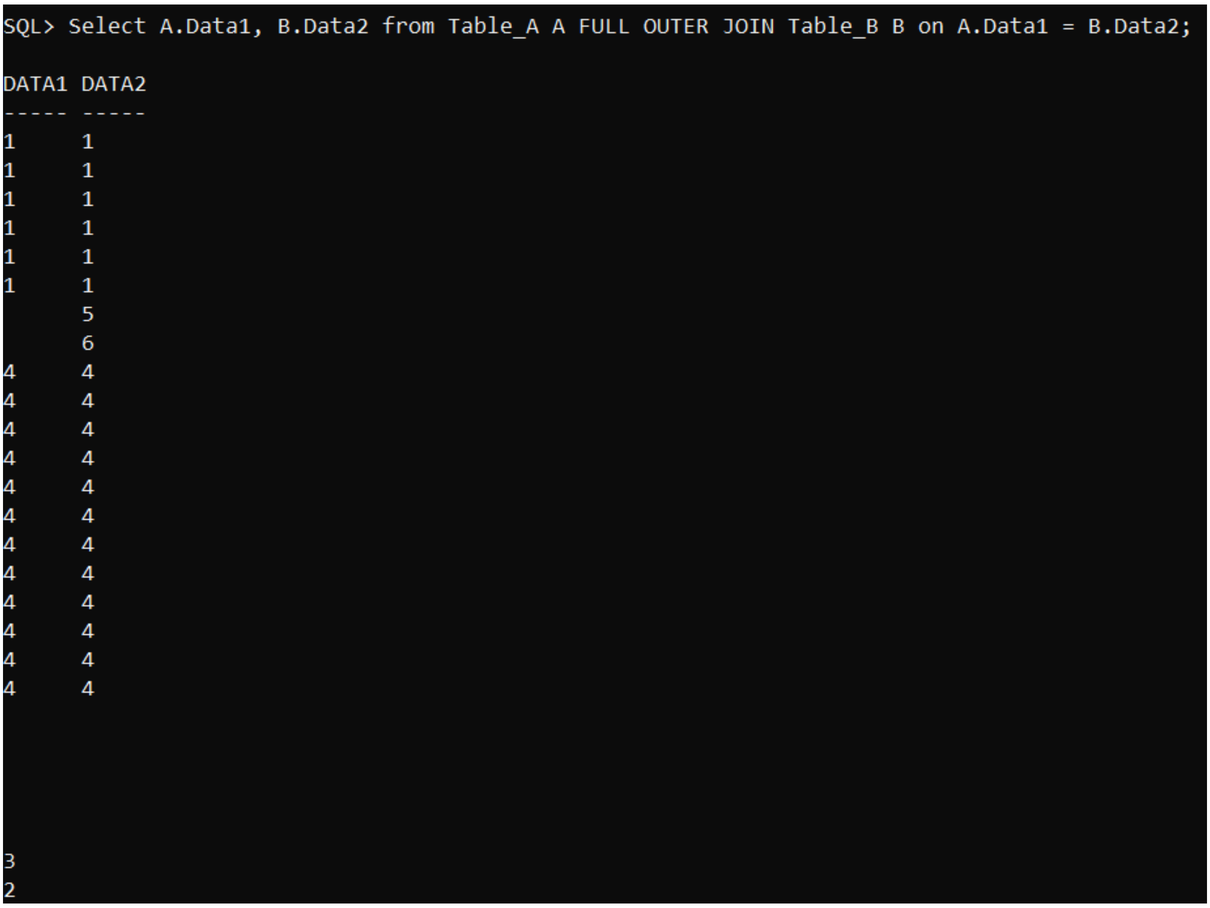
```
SQL> Select A.Data1, B.Data2 from Table_A A LEFT JOIN Table_B B on A.Data1 = B.Data2;

DATA1 DATA2
----- -----
1     1
1     1
1     1
1     1
1     1
1     1
4     4
4     4
4     4
4     4
4     4
4     4
4     4
4     4
4     4
4     4
4     4
4     4


3
2

22 rows selected.
```

Right Join will give 23 records as below (Here blank row is also count as one Rec)



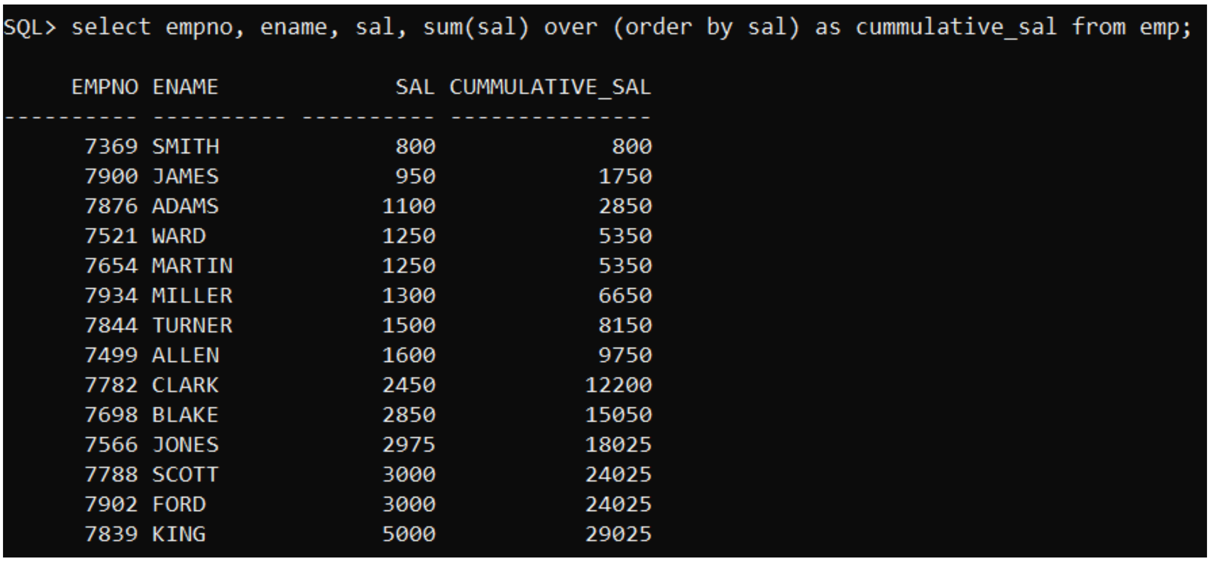
```
SQL> Select A.Data1, B.Data2 from Table_A A RIGHT JOIN Table_B B on A.Data1 = B.Data2;

DATA1 DATA2
----- -----
1     1
1     1
1     1
1     1
1     1
1     1
4     4
4     4
4     4
4     4
4     4
4     4
4     4
4     4
4     4
4     4
4     4
4     4



      6
      5
```

Full Outer Join will give 27 records as below (Here blank row is also count as one Rec)



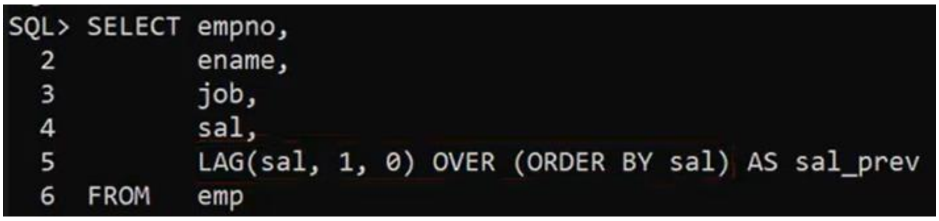
```
SQL> Select A.Data1, B.Data2 from Table_A A FULL OUTER JOIN Table_B B on A.Data1 = B.Data2;

DATA1 DATA2
----- -----
1     1
1     1
1     1
1     1
1     1
1     1
      5
      6
4     4
4     4
4     4
4     4
4     4
4     4
4     4
4     4
4     4
4     4
4     4
4     4






3
2
```

22. How to find the Cumulative salary of an Employee?



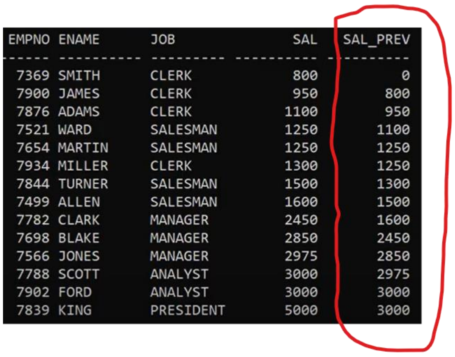
```
SQL> select empno, ename, sal, sum(sal) over (order by sal) as cummulative_sal from emp;

     EMPNO ENAME             SAL CUMMULATIVE_SAL
---------- ---------- ---------- ---------------
      7369 SMITH             800             800
      7900 JAMES             950            1750
      7876 ADAMS            1100            2850
      7521 WARD             1250            5350
      7654 MARTIN           1250            5350
      7934 MILLER           1300            6650
      7844 TURNER           1500            8150
      7499 ALLEN            1600            9750
      7782 CLARK            2450           12200
      7698 BLAKE            2850           15050
      7566 JONES            2975           18025
      7788 SCOTT            3000           24025
      7902 FORD             3000           24025
      7839 KING             5000           29025
```

23. Lag and Lead Analytical functions



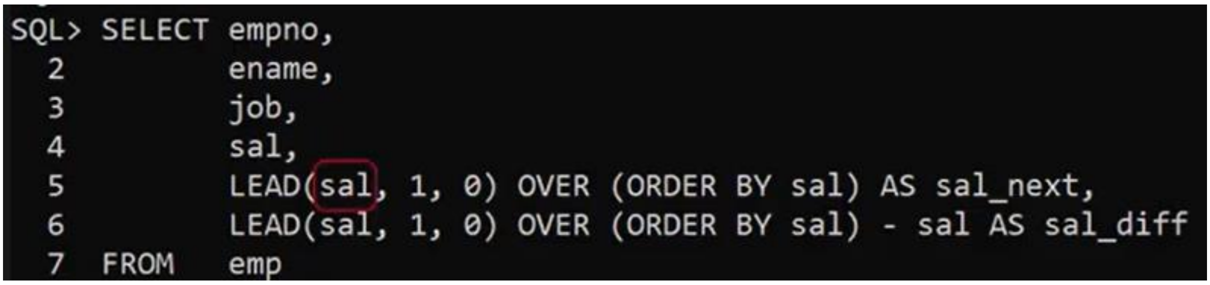
```
SQL> SELECT empno,
            ename,
            job,
            sal,
            LAG(sal, 1, 0) OVER (ORDER BY sal) AS sal_prev
     FROM   emp
```

Output:



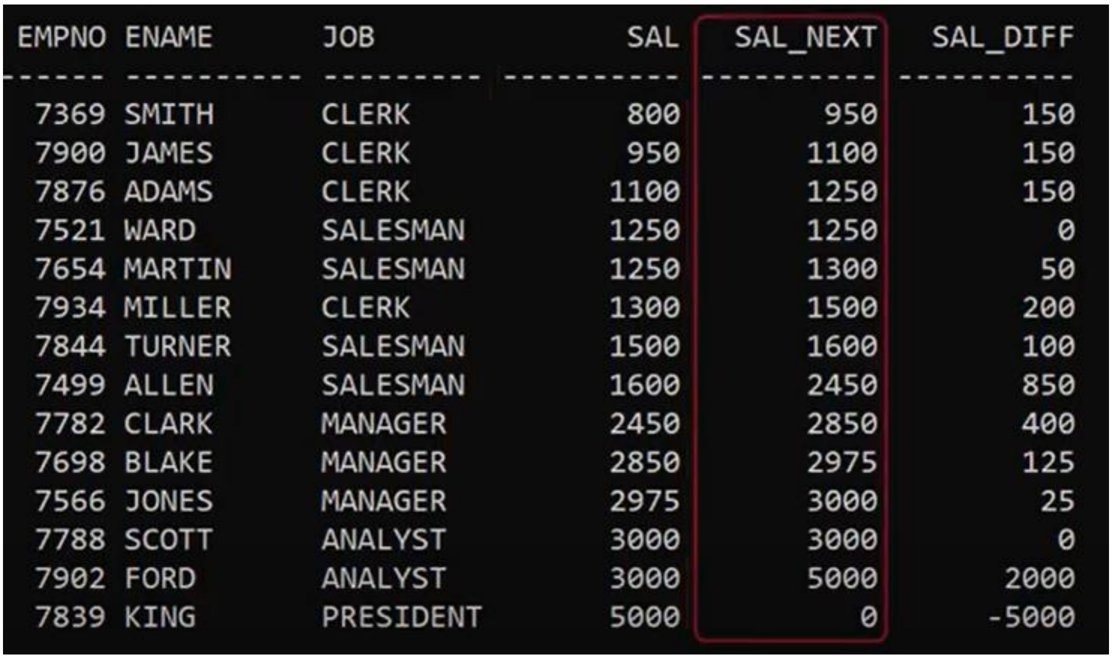
```
EMPNO ENAME      JOB             SAL   SAL_PREV
----- ---------- --------- ---------- ----------
 7369 SMITH      CLERK           800          0
 7900 JAMES      CLERK           950        800
 7876 ADAMS      CLERK          1100        950
 7521 WARD       SALESMAN       1250       1100
 7654 MARTIN     SALESMAN       1250       1250
 7934 MILLER     CLERK          1300       1250
 7844 TURNER     SALESMAN       1500       1300
 7499 ALLEN      SALESMAN       1600       1500
 7782 CLARK      MANAGER        2450       1600
 7698 BLAKE      MANAGER        2850       2450
 7566 JONES      MANAGER        2975       2850
 7788 SCOTT      ANALYST        3000       2975
 7902 FORD       ANALYST        3000       3000
 7839 KING       PRESIDENT      5000       3000
```

Lead Analytical Function



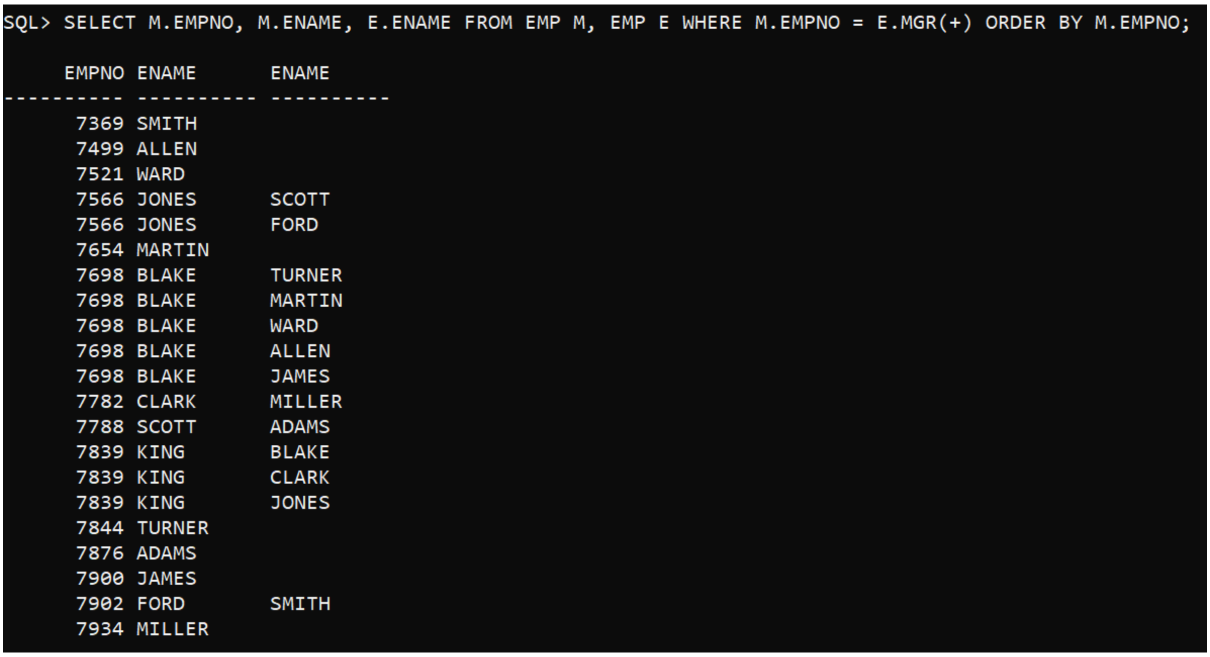
```
SQL> SELECT empno,
  2         ename,
  3         job,
  4         sal,
  5         LEAD(sal, 1, 0) OVER (ORDER BY sal) AS sal_next,
  6         LEAD(sal, 1, 0) OVER (ORDER BY sal) - sal AS sal_diff
  7  FROM   emp
```

Output:



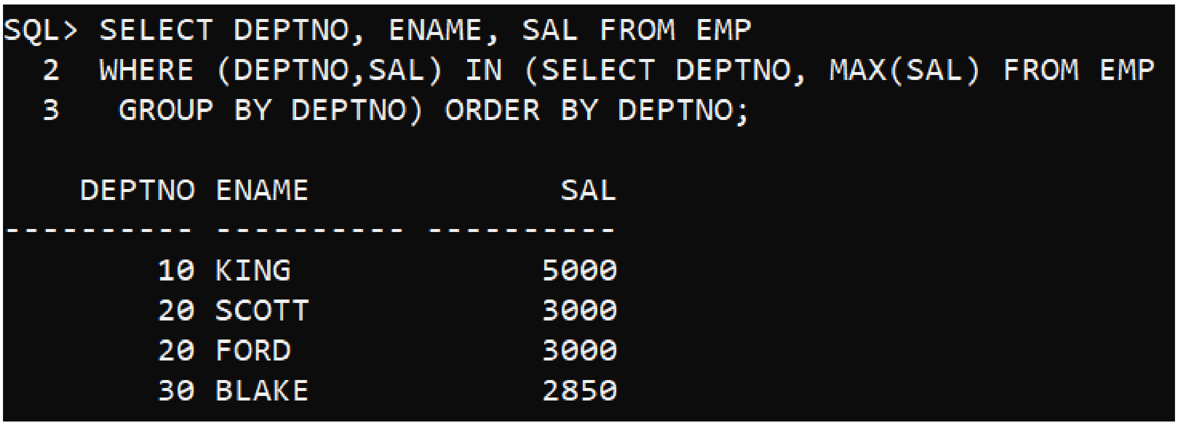
```
EMPNO ENAME      JOB              SAL   SAL_NEXT   SAL_DIFF
----- ---------- --------- ---------- ---------- ----------
 7369 SMITH      CLERK            800        950        150
 7900 JAMES      CLERK            950       1100        150
 7876 ADAMS      CLERK           1100       1250        150
 7521 WARD       SALESMAN        1250       1250          0
 7654 MARTIN     SALESMAN        1250       1300         50
 7934 MILLER     CLERK           1300       1500        200
 7844 TURNER     SALESMAN        1500       1600        100
 7499 ALLEN      SALESMAN        1600       2450        850
 7782 CLARK      MANAGER         2450       2850        400
 7698 BLAKE      MANAGER         2850       2975        125
 7566 JONES      MANAGER         2975       3000         25
 7788 SCOTT      ANALYST         3000       3000          0
 7902 FORD       ANALYST         3000       5000       2000
 7839 KING       PRESIDENT       5000          0      -5000
```

24. Display the employees reporting to each Manager
    SELECT M.EMPNO, M.ENAME, E.ENAME FROM EMP M, EMP E WHERE M.EMPNO = E.MGR(+)
    ORDER BY M.EMPNO;



```
SQL> SELECT M.EMPNO, M.ENAME, E.ENAME FROM EMP M, EMP E WHERE M.EMPNO = E.MGR(+) ORDER BY M.EMPNO;

     EMPNO ENAME      ENAME
---------- ---------- ----------
      7369 SMITH
      7499 ALLEN
      7521 WARD
      7566 JONES      SCOTT
      7566 JONES      FORD
      7654 MARTIN
      7698 BLAKE      TURNER
      7698 BLAKE      MARTIN
      7698 BLAKE      WARD
      7698 BLAKE      ALLEN
      7698 BLAKE      JAMES
      7782 CLARK      MILLER
      7788 SCOTT      ADAMS
      7839 KING       BLAKE
      7839 KING       CLARK
      7839 KING       JONES
      7844 TURNER
      7876 ADAMS
      7900 JAMES
      7902 FORD       SMITH
      7934 MILLER
```

25. Highest Salary in each department with name
    Ans: SELECT DEPTNO, ENAME, SAL FROM EMP WHERE (DEPTNO,SAL) IN (SELECT DEPTNO, MAX(SAL) FROM EMP GROUP BY DEPTNO) ORDER BY DEPTNO;



```
SQL> SELECT DEPTNO, ENAME, SAL FROM EMP
  2  WHERE (DEPTNO,SAL) IN (SELECT DEPTNO, MAX(SAL) FROM EMP
  3   GROUP BY DEPTNO) ORDER BY DEPTNO;

    DEPTNO ENAME             SAL
---------- ---------- ----------
        10 KING             5000
        20 SCOTT            3000
        20 FORD             3000
        30 BLAKE            2850
```

26. Copy the structure of a table from another table
    CREATE TABLE EMP2 AS SELECT * FROM EMP WHERE 1=0;

27. Different RDBMS you are aware of?
    MySQL, PostgreSQL, SQL Server, Oracle, etc.

28. Explain the Need, Pro's, Cons of Views?
    Views can help us have our own copy of data/table, Data integrity, saves system run time

Advantages of views

Security

Each user can be given permission to access the database only through a small set of views that contain the specific data the user is authorized to see, thus restricting the user's access to stored data

Query Simplicity

A view can draw data from several different tables and present it as a single table, turning multi-table queries into single-table queries against the view.

Structural simplicity

Views can give a user a "Personalized" view of the database structure, presenting the database as a set of virtual tables that make sense for that user.

Consistency

A view can present a consistent, unchanged image of the structure of the database, even if the underlying source tables are split, restructured, or renamed.

Data Integrity

If data is accessed and entered through a view, the DBMS can automatically check the data to ensure that it meets the specified integrity constraints.

Logical data independence.

View can make the application and database tables to a certain extent independent. If there is no view, the application must be based on a table. With the view, the program can be established in view of above, to view the program with a database table to be separated.

Disadvantages of views

Performance

Views create the appearance of a table, but the DBMS must still translate queries against the view into queries against the underlying source tables. If the view is defined by a complex, multi-table query then simple queries on the views may take considerable time.

Update restrictions

When a user tries to update rows of a view, the DBMS must translate the request into an update on rows of the underlying source tables. This is possible for simple views, but more complex views are often restricted to read-only.

29. Difference between VIEWs and MATERIALIZED Views?

VIEW	MATERIALIZED View
View is just a named query. It doesn't store anything. When there is a query on the	Stores data physically and gets updated periodically. While querying MATERIALIZED

view, it runs the query of the view definition. Actual data comes from table	View, it gives data from MATERIALIZED View.
Online data from underlying base tables.	Offline/Delayed data from underlying base tables.
No need to refresh the data. Since the data is directly fetched from base tables.	MATERIALIZED View can be set to refresh manually, on a set schedule, or based on the database detecting a change in data from one or the underlying tables.

30. Difference between ROWID and ROWNUM?

Rowid

For each row in the database, the ROWID pseudo column returns the address of the row.

Rownum

For each row returned by a query, the ROWNUM pseudo column returns a number indicating the order in which Oracle selects the row from a table or set of joined rows. The first row selected has a ROWNUM of 1, the second has 2, and so on.

Difference

The actual difference between rowid and rownum is, that rowid is a permanent unique identifier for that row. However, the rownum is temporary. If you change your query, the rownum number will refer to another row, the rowid won't.

So, the ROWNUM is a consecutive number which applicable for a specific SQL statement only. In contrary the ROWID, which is a unique ID for a row.

31. Difference between DROP, DELETE and TRUNCATE?

DROP	DELETE	TRUNCATE
DDL Statement	DML Statement	DDL Statement
Remove Meta data & detail data	Remove only detail data & not the space	Remove only detail data but Meta data remains
Cannot Rollback	Can be Rollback	Cannot Rollback

<u>SDC2</u>

SOURCE

SOURCE QUALIFIER

LOOKUP Transformation

EXPRESSION Transformation

ROUTER Transformation

SEQUECNE Generator

UPDATE Strategy

TARGET

IF(IsNULL OF custid,True,false)

If(custid=lkpcustid and col1=lkpcol1… or coln=lkpcol1)

Use MD5 for N number of columns

DIFFERENCE BETWEEN INSERT AND UPDATE

DD UPDATE AND DD INSERT