CLAUSES: (07-11-2024)
========

- Clause is a statement which is used to add to sql query for providing some additional facilities are filtering rows,sorting values,grouping similar data,finding sub total and grand total based on columns automatically.
- Oracle supports the following clauses are:
    - WHERE
    - ORDER BY
    - GROUP BY
    - HAVING
    - ROLLUP
    - CUBE

syntax:
=======

<SQL QUERY>+<Clause Statement>;

WHERE:
=======

- filtering rows before grouping data in the table.(i.e work on individual rows)
- it can be used in "SELECT,UPDATE,DELETE" commands only.

syntax:
=======

where <filtering condition>;

EX:
SQL> SELECT * FROM EMP WHERE EMPNO=7788;
SQL> UPDATE EMP SET SAL=34000 WHERE JOB='MANAGER';
SQL> DELETE FROM EMP WHERE SAL=5000;

ORDER BY:
=========

- to arrange a specific column values either in ascending order (or) in descending order.
- by default oracle clause will arrange the values in ascending if we want to arrange the values in descending order then we must use "DESC" keyword.
- it can be used in "SELECT" command only.

syntax:
======

select * / <list of columns> from <table name> order by <column name1> <asc/desc>,<column name2> <asc/desc>,.....;

EX:
SQL> SELECT * FROM EMP ORDER BY SAL;
SQL> SELECT * FROM EMP ORDER BY SAL DESC;

SQL> SELECT * FROM EMP ORDER BY ENAME;
SQL> SELECT * FROM EMP ORDER BY ENAME DESC;

SQL> SELECT * FROM EMP ORDER BY HIREDATE;
SQL> SELECT * FROM EMP ORDER BY HIREDATE DESC;

NOTE:
======
- order by clause can apply on not only column names eventhough we can apply on the position of column in SELECT query;

EX:
SQL> SELECT EMPNO,ENAME,SAL FROM EMP ORDER BY 1;
SQL> SELECT EMPNO,ENAME,SAL FROM EMP ORDER BY 2 DESC;
SQL> SELECT EMPNO,ENAME,SAL FROM EMP ORDER BY 3 ;

EX:
waq to display employees who are working under deptno is 20 and arrange those employees salaries in descending?
SQL> SELECT * FROM EMP WHERE DEPTNO=20 ORDER BY SAL DESC;

EX:
waq to arrange deptno's are ascending and also arrange thoes employees salaries in descending order from each deptno wise?
SQL> SELECT * FROM EMP ORDER BY DEPTNO,SAL DESC;

ORDER BY with NULL clauses:
==========
i) NULLS FIRST
ii) NULLS LAST

i) NULLS FIRST:
==========
- by default order by clause on NULLS in ascending order,
  First : Values
  Later : Nulls

Ex:
SQL> SELECT * FROM EMP ORDER BY COMM;
- To overcome the above problem we must use "NULLS FIRST" clause.

Solution:
========
SQL> SELECT * FROM EMP ORDER BY COMM NULLS FIRST;

ii) NULLS LAST:
==========

- by default order by clause on NULLS in descending order,
  - First : Nulls
  - Later : Values

Ex:

```
SQL> SELECT * FROM EMP ORDER BY COMM DESC;
```

- To overcome the above problem we must use "NULLS LAST" clause.

Solution:
========

```
SQL> SELECT * FROM EMP ORDER BY COMM DESC NULLS LAST;
```

GROUP BY:
=========

- it is used to divide groups based on columns.
- when we use "group by" clause we must use "grouping / aggregative" functions to get the result.
- it can be used in "SELECT" query only.

syntax:
======

```
select <column name1>,<column name2>,.....,<aggregative function name1>,.....
from <table name> group by <column name1>,<column name2>,.....;
```

EX:
waq to find out total no.of employees are working in a company?

```
SQL> SELECT COUNT(*) AS TOTAL_NO_OF_EMPLOYEES FROM EMP;

TOTAL_NO_OF_EMPLOYEES
---------------------------------------------
                   14
```

EX:
waq to find out no.of employees are working in "SALESMAN" job?

```
SQL> SELECT COUNT(*) AS NO_OF_EMPLOYEES FROM EMP WHERE JOB='SALESMAN';

NO_OF_EMPLOYEES
-----------------------------------
             4
```

EX:
waq to find out no.of employees are working in each job wise?