The GROUP BY Clause ====>
==========

* The GROUP BY clause is used in a SELECT statement to grouprows into a set of summary rows by values of columns or expressions.

* The GROUP BY clause then returns one row per group.

* The GROUP BY clause is often used with aggregate functions such as AVG(), COUNT(), MAX(), MIN() and SUM().

* Syntax:
- SELECT column_list FROM <table_name> GROUP BY <col_name>;

WAQ to display number of employees in each department.
- select deptno, count(*) from emp group by deptno;

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

WAQ to display average salary of each department.
- select deptno, avg(sal) from emp group by deptno;

```
    DEPTNO   AVG(SAL)
---------- ----------
        30 1566.66667
        20       2175
        10 2916.66667
```

WAQ to display number of employees hired every year.
- select to_char(hiredate, 'YYYY') as Year, count(*) from emp group by to_char(hiredate, 'YYYY');

```
YEAR   COUNT(*)
---- ----------
1987          2
1980          1
1982          1
1981         10
```

WAQ to display max salary of each job.
- select job, max(sal) from emp group by job;

```
JOB         MAX(SAL)
--------- ----------
CLERK           1300
SALESMAN        1600
PRESIDENT       5000
MANAGER         2975
```

```
ANALYST         3000
```

GROUP BY And NULL =====>
==========

When we GROUP BY a column that contains NULL values for some rows, all the rows with NULL values are placed into a single group and presented as one summary row in the output.

- select comm, count(*) from emp where comm is not null group by comm;

```
      COMM   COUNT(*)
---------- ----------
      1400          1
       500          1
       300          1
         0          1
```

Filtering Results Of Group By (Having) ===>
==========

* To filter the results of GROUP BY , we can use HAVING clause.

* The purpose of the HAVING clause is to eliminate groups, just as the WHERE clause is used to eliminate rows.

* If a query has a HAVING clause along with a GROUP BY clause, the result set will include only the groups that satisfy the condition specified in the HAVING clause.

WAQ to display number of emp in each dept, consider only dept 10 & 20;
- select deptno, count(*) from emp group by deptno having deptno in (10, 20);

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

WHERE vs HAVING ===>
==========

* WHERE can only be used to filter on the basis of scalar values and not on the basis of group functions.

* WHERE does not allow using GROUP FUNCTIONS as the results of the aggregate function cannot be determined until after the grouping takes place.

Can We Use Both WHERE and HAVING Together ?
==========

* Yes, it is possible to use both WHERE and HAVING in the same query and this sometimes becomes essential too.

* This happens when we want to filter both on the basis of SCALAR VALUE as well as on the basis of a GROUP FUNCTION result.

WAQ to display no of employees in each department who get a comm and where the MINIMUM sal is greater than 1000.
- select deptno, count(*) from emp where comm is not null group by deptno having min(sal) > 1000;

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

WAQ to display number of employees hired every month.
- select to_char(hiredate,'MON') as MONTH, count(*) from emp group by to_char(hiredate,'MON');

```
MONTH          COUNT(*)
------------ ----------
DEC                   3
APR                   2
NOV                   1
SEP                   2
FEB                   2
JUN                   1
MAY                   2
JAN                   1
```

Modify the previous query so that the output doesn’t contain the month of SEP.
- select to_char(hiredate,'MON') as MONTH, count(*) from emp where to_char(hiredate, 'MON') != 'SEP' group by to_char(hiredate,'MON');

```
MONTH          COUNT(*)
------------ ----------
DEC                   3
APR                   2
NOV                   1
FEB                   2
JUN                   1
MAY                   2
JAN                   1
```

Further modify the previous query so that the output doesn’t contain those months where employee count hired is less than 2.
- select to_char(hiredate,'MM') as MONTH, count(*) from emp where to_char(hiredate, 'MM') != 09 group by to_char(hiredate,'MM');

```
MO   COUNT(*)
-- ----------
```

```
04          2
12          3
11          1
01          1
02          2
05          2
06          1

7 rows selected.
```