Agenda

- GROUP BY clause
- HAVING clause
- Group By With Rollup

Limitations of Group Functions

```
SELECT ename, MAX(sal) FROM emp;
-- error - cannot select column with group function

SELECT LOWER(ename),MAX(sal) FROM emp;
--error - cannot use single row functions  with group functions

SELECT * FROM emp WHERE sal = MAX(sal);
--error - cannot use group functions in where clause

SELECT SUM(MAX(sal)) FROM emp;
-- error - cannot nest group functions
```

Group By Clause

- Group functions work on group of rows
- with group by clause we can use group functions on specified group of rows

Having Clause

- It must be used with Group By Clause only
- Used to apply condition on aggegrate values
- Having vs WHERE
    - Where clause evaluates for every row
    - Having clause evaluates for group
    - Where can be used with columns, single row function but not with group functions
    - Having is used with group columns but not with other columns

Group by with Roll up

- It provides the super aggegrate summary of the group operations.

```
-- display deptwise count of employees
SELECT deptno, COUNT(empno) FROM emp GROUP BY deptno;

-- display deptwise count of employees and also display count of total
employees in organization
-- deptno      COUNT(empno)
-- 10             3
-- 20             5
```

```
-- 30             6
-- NULL          14
SELECT deptno, COUNT(empno) FROM emp GROUP BY deptno;
SELECT COUNT(empno) FROM emp;

SELECT deptno, COUNT(empno) FROM emp GROUP BY deptno
UNION
SELECT NULL,COUNT(empno) FROM emp;

SELECT deptno, COUNT(empno) FROM emp GROUP BY deptno WITH ROLLUP;


-- Display deptwise total sal along with total sal of all employees.
SELECT deptno, SUM(sal) FROM emp GROUP BY deptno WITH ROLLUP;
```

Grouping

- GROUP BY queries that include a WITH ROLLUP modifier produces super-aggregate output rows where NULL represents the set of all values.
- The GROUPING() function enables you to distinguish NULL values for super-aggregate rows from NULL values in regular grouped rows.
- The GROUPING() function returns 1 indicating sub/grand aggregation on that column

```
INSERT INTO emp(empno,ename,job,sal) VALUES(1,"e1","CLERK",1100);

-- display deptwise count of employees and the total count of employees in
all depts
SELECT deptno, COUNT(empno) FROM emp GROUP BY deptno WITH ROLLUP;

-- In above query give an alis to the super aggerate row as total
SELECT IFNULL(deptno,"Total") AS deptno, COUNT(empno) AS count FROM emp
GROUP BY deptno WITH ROLLUP;
-- we cannot differentatiate between super aggegrate rows and normal rows
with null values,to differentiate we can use GROUPING()

SELECT deptno, COUNT(empno), GROUPING(deptno) FROM emp GROUP BY deptno WITH
ROLLUP;

SELECT IF(GROUPING(deptno)=1,"Total",deptno) AS deptno, COUNT(empno) AS
count FROM emp GROUP BY deptno WITH ROLLUP;
```