DBMS SQL

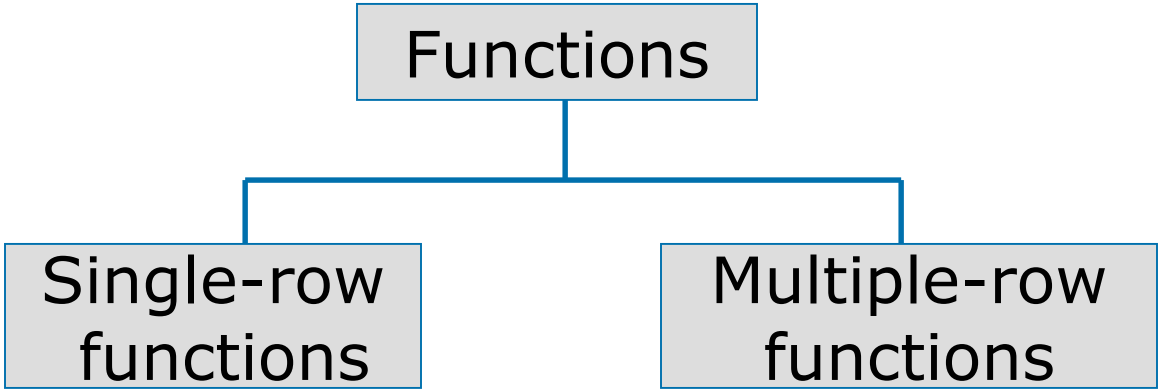
Lesson 07: Aggregate (GROUP) Functions

Lesson Objectives

- To understand the following topics:
  - Aggregate Functions
  - The GROUP BY Clause
  - HAVING Clause
  - ROLLUP Operation
  - CUBE Operation

Types of SQL Functions

- Single row functions :
  - Operate on single rows only and return one result per row
- Multiple row functions:
  - Manipulates groups of rows to give one result per group of rows. Also called as group functions



Functions

Single-row functions

Multiple-row functions

7.1 Aggregate functions

The Group Functions

- The Group functions are built-in SQL functions that operate on “groups of rows”, and return one value for the entire group.
- The results are also based on groups of rows.
- For Example, Group function called “SUM” will help you find the total marks, even if the database stores only individual subject marks.

Syntax : GROUP BY & HAVING clause

- Syntax

```
SELECT          [column, ] aggregate function(column), ……
FROM        table
[WHERE          condition]
[GROUP BY       column]
[HAVING         condition]
[ORDER BY column] ;
```

Listing of Group Functions

- Given below is a list of Group functions supported by SQL:

Function	**Value returned**
SUM (expr)	Sum value of expr, ignoring NULL values.
AVG (expr)	Average value of expr, ignoring NULL values.
COUNT (expr)	Number of rows where expr evaluates to something other than NULL. COUNT(*) counts all selected rows, including duplicates and rows with NULLs.
MIN (expr)	Minimum value of expr.
MAX (expr)	Maximum value of expr.

Examples of using Group Functions

- Example 1: Display the total number of records from student_marks.

```
SELECT COUNT( * ) FROM Student_Marks;
```

- Example 2: Display average marks from each subject.

```
SELECT  AVG(Student_sub1), AVG(Student_sub2), AVG(Student_sub3)
FROM Student_Marks;
```

The GROUP BY clause

- GROUP BY clause is used along with the Group functions to retrieve data that is grouped according to one or more columns.
  - For example: Displays the average staff salary based on every department. The values are grouped based on dept_code

```
SELECT Dept_Code, AVG(Staff_sal)
        FROM Staff_Master
       GROUP BY  Dept_Code;
```

The HAVING clause

- HAVING clause is used to filter data based on the Group functions.
  - HAVING clause is similar to WHERE condition. However, it is used with Group functions.
- Group functions cannot be used in WHERE clause. However, they can be used in HAVING clause.

Examples – GROUP BY and HAVING clause

- For example: Display all department numbers having more than five employees.

```
SELECT Department_Code, Count(*)
        FROM Staff_Master
        GROUP BY Department_Code
        HAVING Count(*)> 5;
```

ROLLUP

➢ROLLUP :

- ROLLUP enables a SELECT statement to calculate multiple levels of subtotals across a specified group of dimensions.
- It also calculates a grand total.
- The ROLLUP extension is highly efficient, adding minimal overhead to a query.

- Syntax:
- SELECT … GROUP BY

```
ROLLUP(grouping_column_reference_list)
```

- Example:

```
SELECT deptno,job,sum(sal)
FROM emp
GROUP BY ROLLUP(deptno,job)
ORDER BY 1,2
```

CUBE

➢CUBE:

- CUBE enables a SELECT statement to calculate subtotals for all possible combinations of a group of dimensions.
- It also calculates a grand total.
- CUBE can calculate a cross-tabular report with a single SELECT statement. CUBE appears in the GROUP BY clause in a SELECT statement. Its form is:

- Syntax:
- SELECT … GROUP BY

```
CUBE (grouping_column_reference_list)
```

- Example:

```
SELECT deptno,job,sum(sal)
FROM emp
GROUP BY CUBE(deptno,job)
ORDER BY 1,2
```

7.6 : Tips and Tricks on using Group Functions GROUP BY & HAVING clause

Quick Guidelines

- All group functions except COUNT(*) ignores NULL values.
- To substitute a value for NULL values use NVL functions.
- DISTINCT clause makes the function consider only non duplicate values.
- The AVG and SUM are used with numerica data.
- The MIN and MAX functions used with any data type.

Quick Guidelines

- All individual columns included in the SELECT clause other than group functions must be specified in the GROUP BY clause.
- Any column other than selected column can also be placed in GROUP BY clause.
- By default rows are sorted by ascending order of the column included in the GROUP BY list.
- WHERE clause specifies the rows to be considered for grouping.

Quick Guidelines

- Suppose your SELECT statement contains a HAVING clause. Then write your query such that the WHERE clause does most of the work (removing undesired rows) instead of the HAVING clause doing the work of removing undesired rows.
- Use the GROUP BY clause only with an Aggregate function, and not otherwise.
- Since in other cases, you can accomplish the same end result by using the DISTINCT option instead, and it is faster.

Summary

- In this lesson, you have learnt about:
  - Aggregate (Group functions)
    - GROUP BY clause
    - HAVING clause

Review – Questions

- Question 1: Identify the various group functions from the list given below:
  - Option 1: maximum
  - Option 2: sum
  - Option 3: count
  - Option 4: minimum

- Question 2: The AVG function ignores NULL values in the column.
  - True / False

- Question 3: Count(*) returns the number of rows in the table, including duplicates and those with NULLs.
  - True / False