Lab 04: SQL Functions

Objective(s):

To learn the use of Multiple Row Functions in SELECT statement.

Multiple Row or Group Functions

1. Group Functions operate on set of rows to give one result per group.

2. All group functions except COUNT(*) ignore null values in the column.

3. **WHERE** clause comes **before GROUP BY** clause.

4. **HAVING** clause can come **after GROUP BY** clause.

5. **Alias names** cannot be used in **GROUP BY** clause.

6. **GROUP BY** columns are not necessarily be in the **SELECT** clause (example 3).

7. **GROUP BY** clause can be applied on more than one columns (example 3).

Syntax:

Select [column,] group_function(column) from table

[where condition] [group by column] [order by column];

Example:

1)Select avg(sal),max(sal),sum(sal) from emp where job like “SALES%”;

2)Select deptno, sum(sal) from emp group by deptno;

3)Select sum(sal) from emp group by deptno,job;

1) AVG([Distinct | All] n):

Average of *n values*, ignoring null values.

Example:

Select avg(sal) from emp;

2) COUNT({* | [distinct | all] exp }):

1. Count the number of rows, where the expression evaluates to something other than null.

2. Count(*) returns the total number of row including duplicate and rows with null.

Example:

1) Select count(*) from emp where deptno = 30;

2) Select count(comm) from emp where deptno = 30;

3) Select count(distinct(deptno)) from emp;

3) MAX ([distinct | all] exp):

Maximum value of expression, ignoring null values.

4) MIN ([distinct | all] exp):

Minimum value of expression, ignoring null values.

5) SUM([Distinct | All] n):

Sum values of n, ignoring null values.

Guidelines:

1. **DISTINCT** returns the nonduplicate values.

2. **All** returns the duplicate values.

3. All group functions except **COUNT(*)** ignore null values. To substitute a value for null values, use the NVL,NVL2 or COALESCE functions.

Grouping by more than one column:

Group By clause can be applied on more than one columns. Lets say, we want to display the total salary being paid to each job title, within each department.

Group By clause can be used without using a group function in the Select list.

Example:

Select deptno,job,sum(sal) from emp group by deptno,job;

HAVING Clause:

1. You cannot use the **WHERE** clause to restrict groups.

2. You use the **HAVING** clause to restrict groups.

Syntax:

Select column,group_function **From** table

[**Where** condition] [**Group By** column]

[**Having** group condition] [**Order By** column];

Example:

WRONG:

SELECT deptno, AVG(sal)

FROM emp

WHERE AVG(sal) > 2000

GROUP BY deptno;

CORRECT:

SELECT deptno, AVG(sal)

FROM emp

GROUP BY deptno

HAVING AVG(sal) > 2000

Exercise

Now by using the EMP table, formulate the following queries in SQL

1. Display the dname and average salary of those departments whose average salary is greater than 2500.

2. Display the ename, sal, deptno and average salary of the department of those employees who earn more than the average salary of their own departments.

3. Display the total number of employees who have no commission.

4. Write a query to display the number of employees with the same job.

5. Display the manager number and the salary of the lowest paid employee of that manager. Exclude anyone whose manager is not known. Exclude any groups where the minimum salary is 2000. Sort the output is descending order of the salary.