

Ex.No.: 10

## AGGREGATING DATA USING GROUP FUNCTIONS

Date:

### Objectives

After the completion of this exercise, the students be will be able to do the following:

- Identify the available group functions
- Describe the use of group functions
- Group data by using the GROUP BY clause
- Include or exclude grouped rows by using the HAVING clause

### What Are Group Functions?

Group functions operate on sets of rows to give one result per group

### Types of Group Functions

- AVG
- COUNT
- MAX
- MIN
- STDDEV
- SUM
- VARIANCE

Each of the functions accepts an argument. The following table identifies the options that you can use in the syntax:

Function	Description
AVG ( [DISTINCT   <u>ALL</u> ] n )	Average value of n, ignoring null values
COUNT ( { *   [DISTINCT   <u>ALL</u> ] expr } )	Number of rows, where expr evaluates to something other than null (count all selected rows using *, including duplicates and rows with nulls)
MAX ( [DISTINCT   <u>ALL</u> ] expr )	Maximum value of expr, ignoring null values
MIN ( [DISTINCT   <u>ALL</u> ] expr )	Minimum value of expr, ignoring null values
STDDEV ( [DISTINCT   <u>ALL</u> ] x )	Standard deviation of n, ignoring null values
SUM ( [DISTINCT   <u>ALL</u> ] n )	Sum values of n, ignoring null values
VARIANCE ( [DISTINCT   <u>ALL</u> ] x )	Variance of n, ignoring null values

### Group Functions: Syntax

```
SELECT [column,] group_function(column), ...  
FROM table  
[WHERE condition]
```

Group functions can be nested to a depth of two. The slide example displays the maximum average salary.

```
SELECT MAX(AVG(salary)) FROM employees GROUP BY department_id;
```

#### Summary

In this exercise, students should have learned how to:

- Use the group functions COUNT, MAX, MIN, and AVG
- Write queries that use the GROUP BY clause
- Write queries that use the HAVING clause

```
SELECT column, group_function  
FROM table  
[WHERE condition]  
[GROUP BY group_by_expression]  
[HAVING group_condition]  
[ORDER BY column];
```

#### Find the Solution for the following:

Determine the validity of the following three statements. Circle either True or False.

1. Group functions work across many rows to produce one result per group.

☒ True/False

2. Group functions include nulls in calculations.

True/☒ False

3. The WHERE clause restricts rows prior to inclusion in a group calculation.

☒ True/False

#### The HR department needs the following reports:

4. Find the highest, lowest, sum, and average salary of all employees. Label the columns Maximum, Minimum, Sum, and Average, respectively. Round your results to the nearest whole number

Select ROUND(MAX(salary)) AS maximum, ROUND(MIN(salary)) AS minimum  
ROUND(SUM(salary)) AS sum, ROUND(AVG(salary)) AS Average from employees

5. Modify the above query to display the minimum, maximum, sum, and average salary for each job type.

Select Job id ROUND(MAX(salary)) AS maximum, ROUND(MIN(salary)) AS  
minimum, ROUND(SUM(salary)) AS sum, ROUND(AVG(salary)) AS Average  
from employees group by job-id;

6. Write a query to display the number of people with the same job. Generalize the query so that the user in the HR department is prompted for a job title.

select job\_id, count(\*) as number\_of\_employees from employees where  
job\_id = 'job-title' group by job\_id;

7. Determine the number of managers without listing them. Label the column Number of Managers. Hint: Use the MANAGER\_ID column to determine the number of managers.

select count(DISTINCT manager\_id) as 'Numbers of Managers' from  
employees where manager\_id is NOT NULL;

8. Find the difference between the highest and lowest salaries. Label the column DIFFERENCE.

select MAX(salary) - MIN(salary) as DIFFERENCE from employees;

9. Create a report to display the manager number and the salary of the lowest-paid employee for that manager. Exclude anyone whose manager is not known. Exclude any groups where the minimum salary is \$6,000 or less. Sort the output in descending order of salary.

select manager\_id, min(salary) as minimum\_salary from employee where  
manager\_id is NOT NULL group by manager\_id having min(salary) >  
6000 order by minimum\_salary DESC;

10. Create a query to display the total number of employees and, of that total, the number of employees hired in 1995, 1996, 1997, and 1998. Create appropriate column headings.

select count(\*) as total\_employees, sum(case when TO\_CHAR(HIRE\_DATE  
'yyyy') = '1995' then 1 else 0 end) as hired\_1995, sum(case when  
TO\_CHAR(HIRE\_DATE, 'yyyy') = '1996' then 1 else 0  
'1997' then 1 else 0 end) as hired\_1997, sum(case when  
TO\_CHAR(HIRE\_DATE, 'yyyy') = '1998' then 1 else 0 end) as hired\_1998  
as hired\_1998, sum(case when TO\_CHAR(HIRE\_DATE, 'yyyy') = '1999' then 1 else 0 end) as hired\_1999  
as hired\_1999 from employees;

sum(case when TO\_CHAR (Hire\_date, 'yy44') = '1998' then 1 else 0  
END) AS Hired\_1998 from employees;

11. select Job-id, sum(case when department-id = 20 then salary else 0 END)  
AS DEPT-20-salary, sum(case when department-id = 80 then salary else 0 END)  
AS DEPT-80-salary, sum(case when department-id = 90 then salary  
else 0 END) AS dept-90-salary, sum(salary) as Total salary from

11. Create a matrix query to display the job, the salary for that job based on department  
number, and the total salary for that job, for departments 20, 50, 80, and 90, giving each  
column an appropriate heading.

employees where department-id IN (20, 50, 80, 90) group by  
Job-id ;

12. Write a query to display each department's name, location, number of employees, and the  
average salary for all the employees in that department. Label the column name-Location,  
Number of people, and salary respectively. Round the average salary to two decimal places.

select d.deptname as department\_name, l.location-id as location count  
(e.employee-id) as number of people, ROUND(avg(e.salary), 2) as Average  
Salary from employee e JOIN department d ON e.department-id =  
d.department-id JOIN location l ON d.location-id = l.location-id group  
by d.department\_name, l.location-id

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	