

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 "Number 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 employees 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  
END) AS Hired_1996, SUM(CASE WHEN TO_CHAR(hire_date, 'YYYY') =  
'1997' THEN 1 ELSE 0 END) AS Hired_1997,
```


Sum(Case WHEN TO_CHAR(hire-date, 'YY') = '98' THEN 1 ELSE 0 END) AS Hire-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 = 50 THEN salary ELSE 0 END) AS Dept-50-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

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.

Total-salary from 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.dept-name 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 employees e JOIN departments d on e-department-id = d-department-id JOIN locations l on d-location-id = l.location-id group by d-department-name, l.location-id;

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