SELECT department_id, AVG(salary) FROM employees GROUP BY department_id
HAVING max(salary)>10000;

Example displays the job ID and total monthly salary for each job that has a total payroll exceeding
$13,000. The example excludes sales representatives and sorts the list by the total monthly salary.

SELECT job_id, SUM(salary) PAYROLL FROM employees WHERE job_id NOT LIKE
'%REP%'
GROUP BY job_id HAVING SUM(salary) > 13000 ORDER BY SUM(salary);

Nesting Group Functions

Display the maximum average salary:
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   *True*

2. Group functions include nulls in calculations.
True/False   *False*

3. The WHERE clause restricts rows prior to inclusion in a group calculation.
True/False   *True*

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 (min (salary)) as minimum, around(max (salary))
as maximum , round (sum (salary)) as sum , round (
avg (salary));

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 count (*) as "number of employees" from employees.
where job-id = ' a job Title ';

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 manager"
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 employee;

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 "lowest salary" from
employees where manager-id is NOT NULL GROUP BY
manager -id having min (salary > 6000) ORDER BY
"lowest salary " DES;

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, 'x x xx ') = '1995', then (ELSE END)
as "hired in 1995" sum (case when to char (hire-dt;
x 'xxxx') = '1996' then < ELSE END as
"hired in 1996 "; sum (case when to char (hire-date,
'xxxx ') = '1997' then < +4. Eyn) as "hired in
1999 ' sum (case when to char (hire-date, 'xxxx' =

'1998', then (ELSE.END) as "hired in 1998"
from employees;

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.

select job_id, sum (case when department_id = 20 then salary else . END)
as cdept = 20 , sum (case when department_id = 50 then salary else . END)
as dept-50 , sum (case when department_id = 80 then salary else cond)
as dept_80, sum(salary) as total salary;

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.department_name || ' - ' || l.city as "name-location",
count (e.employee_id) as "number of people", round (
AVG (u.salary) , 2) as salary from department ,
d join location on d.department_id = e.department_id
group by d.department_name , l.city;

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