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

```
ELECT ROUND(MAX(salary)) AS Maximum,
ROUND(MIN(salary)) AS Minimum, ROUN(SUM(salary))
AS Sum, ROUN(AVG(salary)) AS Average By Job_Id.
```

68

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,
ROUND (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 'Count FROM employees WHERE
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 Lowest_Salary FROM
employees WHERE manager_id IS NOT NULL GROUP BY
Manager_id HAVING MIN (SALARY) > 6000 ORDER BY Lowest_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 (YEAR (Hire_date)
= 1995) AS Hired_1995, SUM (YEAR (Hire_date) = 1996) AS
Hired_1996 , SUM (YEAR (Hire_date) = 97 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 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,
```

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 AS 'Name_Location',
l.city AS LOCATION, COUNT(e.employee_id) AS
'Number of People', ROUND(AVG(e.salary),2) AS
SALARY FROM employees e JOIN departments d ON
e.department_id = d.department_id GROUP BY
d.department_name, l.city ;
```

Evaluation Procedure	Marks awarded
Query(5)	5
Execution (5)	5
Viva(5)	5
Total (15)	15
Faculty Signature	9/9/25