

| | |
|------------------|--|
| Ex.No.: 10 | AGGREGATING DATA USING GROUP FUNCTIONS |
| Date: 12/09/2024 | |

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 - **FALSE**

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;
```

| MAXIMUM | MINIMUM | SUM | AVERAGE |
|---------|---------|--------|---------|
| 15500 | 1100 | 254300 | 7706 |

1 rows returned in 0.02 seconds [Download](#)

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

```
SELECT ROUND(MAX(salary)) AS Maximum, ROUND(MIN(salary)) AS Minimum,
ROUND(SUM(salary)) AS Sum, ROUND(AVG(salary)) AS Average
FROM employees
join department
on department.dept_id = employees.department_id
group by dept_name;
```

| MAXIMUM | MINIMUM | SUM | AVERAGE |
|---------|---------|-------|---------|
| 4000 | 2500 | 6500 | 3250 |
| 13500 | 13500 | 13500 | 13500 |
| 7800 | 4500 | 12300 | 6150 |
| 13500 | 5200 | 26700 | 8900 |
| 7000 | 1100 | 8100 | 4050 |
| 6500 | 5500 | 12000 | 6000 |
| 13500 | 6000 | 19500 | 9750 |
| 13500 | 13500 | 13500 | 13500 |
| 13500 | 3500 | 40500 | 8100 |

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 d.dept_name , COUNT(*) AS NumberOfEmployees
FROM Employees e
join department d on e.department_id = d.dept_id
group by d.dept_name;
```

| DEPT_NAME | NUMBEROFEMPLOYEES |
|------------------|-------------------|
| accounts manager | 2 |
| IT support | 1 |
| admin manager | 2 |
| HR | 3 |
| stock clerk | 2 |
| sales manager | 2 |
| manager | 2 |
| developer | 1 |
| executive | 5 |
| data analyst | 3 |

10 rows returned in 0.01 seconds [Download](#)

7) Determine the number of managers without listing them. Label the column Number of Managers

```
SELECT COUNT(DISTINCT MANAGER_ID) AS "Number of Managers"
FROM Employees
WHERE MANAGER_ID IS NOT NULL;
```

| Number of Managers |
|--|
| 75 |
| 1 rows returned in 0.01 seconds Download |

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

```
select max(salary) - min(salary) as "DIFFERENCE"
from employees;
```

| DIFFERENCE |
|--|
| 12400 |
| 1 rows returned in 0.01 seconds Download |

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;
```

| MANAGER_ID | Lowest Salary |
|--|---------------|
| 350 | 8000 |
| 150 | 7700 |
| 500 | 7500 |
| 800 | 7300 |
| 600 | 6900 |
| 550 | 6500 |
| 6 rows returned in 0.01 seconds Download | |

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 EXTRACT(YEAR FROM hire_date) AS "yearly wise employment", COUNT(*)
FROM employees
GROUP BY EXTRACT(YEAR FROM hire_date)
HAVING EXTRACT(YEAR FROM hire_date) IN (1995, 1996, 1997, 1998);
```

| yearly wise employment | COUNT(*) |
|------------------------|----------|
| 1996 | 2 |
| 1995 | 1 |

2 rows returned in 0.01 seconds [Download](#)

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 d.dept_name , sum(e.salary)
from employees e
join department d on e.department_id = d.dept_id
where department_id in (20,50,80,90)
group by d.dept_name;
```

| DEPT_NAME | SUM(E.SALARY) |
|-------------|---------------|
| stock clerk | 8100 |
| manager | 19500 |

2 rows returned in 0.02 seconds [Download](#)

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 "Name", d.Location_id AS "Location",
COUNT(e.department_id) AS "Number of People", ROUND(AVG(e.Salary), 2) AS
"Salary"
FROM department d
JOIN employees e ON d.dept_id = e.department_id
```

GROUP BY d.dept_name, d.location_id;

| Name | Location | Number of People | Salary |
|--|----------|------------------|---------|
| sales manager | 7 | 2 | 6000 |
| data analyst | 1700 | 3 | 9733.33 |
| stock clerk | 19 | 2 | 4050 |
| HR | 2 | 3 | 8900 |
| admin manager | 16 | 2 | 6850 |
| manager | 10 | 2 | 9750 |
| accounts manager | 7 | 2 | 3250 |
| executive | 4 | 3 | 6333.33 |
| developer | 1 | 1 | 13500 |
| executive | 10 | 2 | 10750 |
| More than 10 rows available. Increase rows selector to view more rows. | | | |
| 10 rows returned in 0.03 seconds Download | | | |