

**M.MOHANA**

**231901031**

**EX NO:10 AGGREGATING DATA USING GROUP FUNCTIONS**

**30/10/24**

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;
```

Maximum	Minimum	Sum	Average
16000	4600	158500	7925

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;
```

JOB_ID	Maximum	Minimum	Sum	Average
IT_PROG	16000	6000	51600	8600
AC_ACCOUNT	15000	15000	15000	15000
AC_MGR	7100	7100	7100	7100
SA_MAN	7200	7200	7200	7200
SA_REP	13500	5500	30800	7700
HR_REP	12500	4600	35300	7060
ST_CLERK	6200	5300	11500	5750

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 People" FROM employees WHERE job_id =  
'&job_title';
```

Number of People
6

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;
```

Number of Managers
5

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

```
SELECT (MAX(salary) - MIN(salary)) AS "DIFFERENCE" FROM employees;
```

DIFFERENCE
11400

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 MIN(salary) DESC;
```

MANAGER_ID	Lowest Salary
103	13500
101	12500

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 in 1995",  
  SUM(CASE WHEN TO_CHAR(hire_date, 'YYYY') = '1996' THEN 1 ELSE 0 END) AS  
  "Hired in 1996",  
  SUM(CASE WHEN TO_CHAR(hire_date, 'YYYY') = '1997' THEN 1 ELSE 0 END) AS  
  "Hired in 1997",  
  SUM(CASE WHEN TO_CHAR(hire_date, 'YYYY') = '1998' THEN 1 ELSE 0 END) AS  
  "Hired in 1998" FROM employees;
```

Total Employees	Hired in 1995	Hired in 1996	Hired in 1997	Hired in 1998
20	1	1	2	3

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",
       SUM(CASE WHEN department_id = 50 THEN salary ELSE 0 END) AS "Dept 50",
       SUM(CASE WHEN department_id = 80 THEN salary ELSE 0 END) AS "Dept 80",
       SUM(CASE WHEN department_id = 90 THEN salary ELSE 0 END) AS "Dept 90",
       SUM(salary) AS "Total Salary"
FROM employees WHERE department_id IN (20, 50, 80, 90) GROUP BY job_id;
```

JOB_ID	Dept 20	Dept 50	Dept 80	Dept 90	Total Salary
IT_PROG	0	6000	0	0	6000
AC_ACCOUNT	15000	0	0	0	15000
AC_MGR	0	7100	0	0	7100
SA_MAN	0	7200	0	0	7200
SA_REP	0	0	13500	0	13500
HR_REP	4600	0	0	0	4600

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 "Department Name", l.city 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.city;
```

Department Name	Location	Number of People	Average Salary
IT	London	3	6766.67
ST_CLERK	Dubai	1	13500
ST_CLERK	Sydney	1	5300
Customer Service	Mumbai	1	12500
Admin	New York	2	11100
ST_CLERK	San Francisco	2	9800