EXERCISE-8

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 answer:TRUE

2. Group functions include nulls in calculations.

True/False answer:FALSE

3. The WHERE clause restricts rows prior to inclusion in a group calculation.

True/False answer:TRUE

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;
```

Output

Maximum	Minimum	Sum	Average
12000	3000	55000	6111

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)) AS Average
FROM employees
GROUP BY job_id;
```

Output

job_id	Minimum	Maximum	Sum	Average
AD_PRES	12000	12000	12000	12000
HR_REP	3000	3200	6200	3100
IT_PROG	5800	6000	11800	5900
MK_MAN	8000	8000	8000	8000
MK_REP	7000	7000	7000	7000
SA_MAN	5500	5500	5500	5500
SA_REP	4500	4500	4500	4500

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 = 'IT_PROG'
GROUP BY job_id;
```

Output

job_id	Number of Employees
IT_PROG	2

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;
```

Output

Number of Managers
4

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

```sql
SELECT MAX(salary) - MIN(salary) AS Difference
FROM employees;
```

Output

Difference
9000

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.

```sql
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;
```

Output

manager_id	Lowest Salary
107	7000

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.

```sql
SELECT
    COUNT(*) AS "Total",
    SUM(CASE WHEN hire_date LIKE '1995%' THEN 1 ELSE 0 END) AS "Hired_1995",
    SUM(CASE WHEN hire_date LIKE '1996%' THEN 1 ELSE 0 END) AS "Hired_1996",
    SUM(CASE WHEN hire_date LIKE '1997%' THEN 1 ELSE 0 END) AS "Hired_1997",
    SUM(CASE WHEN hire_date LIKE '1998%' THEN 1 ELSE 0 END) AS "Hired_1998"
FROM employees;
```

Output

Total	Hired_1995	Hired_1996	Hired_1997	Hired_1998
9	2	3	1	2

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
FROM employees
WHERE department_id IN (20, 50, 80, 90)
GROUP BY job_id;
```

Output

job_id	Dept_20	Dept_50	Dept_80	Dept_90	Total
AD_PRES	0	0	0	12000	12000
IT_PROG	11800	0	0	0	11800
MK_MAN	0	0	8000	0	8000
MK_REP	0	0	7000	0	7000
SA_MAN	0	5500	0	0	5500
SA_REP	0	4500	0	0	4500

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(e.salary), 2) AS "Salary"
FROM departments d
JOIN locations l ON d.location_id = l.location_id
LEFT JOIN employees e ON d.department_id = e.department_id
GROUP BY d.department_id;
```

Output

Name-Location	Number of People	Salary
HR-New York	2	3100
IT-Toronto	2	5900
Sales-New York	2	5000
Marketing-New York	2	7500
Admin-Toronto	1	12000