

EXERCISE-8

Aggregating Data Using Group Functions

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(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;
```

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_People  
FROM employees
```

```
WHERE job_id = :job_title  
GROUP BY 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 min_salary FROM employees WHERE manager_id IS  
NOT NULL GROUP BY manager_id HAVING MIN(salary) > 6000 ORDER BY min_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(CASE WHEN EXTRACT(YEAR FROM hire_date) = 1995 THEN 1 ELSE 0  
END) AS hired_1995,  
  SUM(CASE WHEN EXTRACT(YEAR FROM hire_date) = 1996 THEN 1 ELSE 0  
END) AS hired_1996,  
  SUM(CASE WHEN EXTRACT(YEAR FROM hire_date) = 1997 THEN 1 ELSE 0  
END) AS hired_1997,  
  SUM(CASE WHEN EXTRACT(YEAR FROM hire_date) = 1998 THEN 1 ELSE 0  
END) AS hired_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 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 THEN salary ELSE  
0 END) AS dept_80_salary, SUM(CASE WHEN department_id = 90 THEN salary  
ELSE 0 END) AS dept_90_salary, SUM(salary) AS total_salary FROM employees  
WHERE department_id IN (20, 50, 80, 90) GROUP BY job_id;
```

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 departments d JOIN employees e ON d.department_id =  
e.department_id JOIN locations l ON d.location_id = l.location_id  
GROUP BY d.department_name, l.city;
```

Practice Questions

Date Functions

1. For DJs on Demand, display the number of months between the event_date of the Vigil wedding and today's date. Round to the nearest month.

```
SELECT ROUND(MONTHS_BETWEEN(SYSDATE, (SELECT event_date FROM events WHERE event_name = 'Vigil wedding'))) AS months_difference FROM dual;
```

2. Display the days between the start of last summer's school vacation break and the day school started this year. Assume 30.5 days per month. Name the output "Days."

```
SELECT ROUND((this_year_start - last_summer_start) * 30.5, 0) AS Days FROM dual;
```

3. Display the days between January 1 and December 31.

```
SELECT (TO_DATE('31-DEC', 'DD-MON') - TO_DATE('01-JAN', 'DD-MON')) AS days_difference  
  
FROM dual;
```

4. Using one statement, round today's date to the nearest month and nearest year and truncate it to the nearest month and nearest year. Use an alias for each column.

```
SELECT ROUND(SYSDATE, 'MONTH') AS rounded_month, ROUND(SYSDATE, 'YEAR') AS rounded_year,  
  
TRUNC(SYSDATE, 'MONTH') AS truncated_month, TRUNC(SYSDATE, 'YEAR') AS truncated_year FROM  
  
Dual;
```

5. What is the last day of the month for June 2005? Use an alias for the output.

```
SELECT LAST_DAY(TO_DATE('01-JUN-2005', 'DD-MON-YYYY')) AS last_day_of_june_2005  
  
FROM dual;
```

6. Display the number of years between the Global Fast Foods employee Bob Miller's birthday and today. Round to the nearest year.

```
SELECT ROUND(MONTHS_BETWEEN(SYSDATE, (SELECT birth_date FROM employees WHERE name =  
  
'Bob Miller'))) / 12, 0) AS years_difference  
  
FROM dual;
```

7. Your next appointment with the dentist is six months from today. On what day will you go to the dentist? Name the output, "Appointment."

```
SELECT ADD_MONTHS(SYSDATE, 6) AS Appointment  
  
FROM dual;
```

8.The teacher said you have until the last day of this month to turn in your research paper. What day will this be? Name the output, "Deadline."

```
SELECT LAST_DAY(SYSDATE) AS Deadline FROM dual;
```

9.How many months between your birthday this year and January 1 next year? 10.What's the date of the next Friday after your birthday this year? Name the output, "First Friday."

```
SELECT MONTHS_BETWEEN(TO_DATE('01-JAN-YYYY', 'DD-MON-YYYY') + 1,  
TO_DATE('your_birthday_this_year', 'DD-MON-YYYY')) AS months_difference  
FROM dual;
```

11. Name a date function that will return a number.

```
MONTHS_BETWEEN(date1, date2)
```

12.Name a date function that will return a date.

```
ADD_MONTHS(date, n)
```

13.Give one example of why it is important for businesses to be able to manipulate date data?

It is crucial for businesses to manipulate date data for accurate scheduling, tracking deadlines, forecasting, and generating timely reports which are essential for decision-making and operational efficiency.

Conversion Functions

In each of the following exercises, feel free to use labels for the converted column to make the output more readable.

1. List the last names and birthdays of Global Fast Food Employees. Convert the birth dates to character data in the Month DD, YYYY format. Suppress any leading zeros.

```
SELECT last_name, TO_CHAR(birth_date, 'Month DD, YYYY') AS birth_date_formatted  
  
FROM employees;
```

2. Convert January 3, 04, to the default date format 03-Jan-2004.

```
SELECT TO_DATE('03-Jan-2004', 'DD-MON-YYYY') AS converted_date FROM dual;
```

3. Format a query from the Global Fast Foods f_promotional_menus table to print out the start_date of promotional code 110 as: The promotion began on the tenth of February 2004.

```
SELECT 'The promotion began on the ' || TO_CHAR(start_date, 'DDth of Month YYYY') AS  
  
promotion_start FROM f_promotional_menus WHERE promo_code = 110;
```

4. Convert today's date to a format such as: "Today is the Twentieth of March, Two Thousand Four"

```
SELECT 'Today is the ' || TO_CHAR(SYSDATE, 'DDth of Month, YYYY') AS formatted_today  
  
FROM dual;
```

5. List the ID, name and salary for all Global Fast Foods employees. Display salary with a \$ sign and two decimal places.

```
SELECT id, name, TO_CHAR(salary, '$999,999.99') AS formatted_salary  
  
FROM employees;
```