

EXERCISE-6

Single Row Functions

Find the Solution for the following:

1. Write a query to display the current date. Label the column Date.

```
SELECT SYSDATE AS "Date" FROM DUAL;
```

2. The HR department needs a report to display the employee number, last name, salary, and increased by 15.5% (expressed as a whole number) for each employee. Label the column New Salary.

```
SELECT employee_id, last_name, salary, ROUND(salary * 1.155) AS "New Salary" FROM employees;
```

3. Modify your query lab_03_02.sql to add a column that subtracts the old salary from the new salary. Label the column Increase.

```
SELECT employee_id, last_name, salary, ROUND(salary * 1.155) AS "New Salary",  
ROUND(salary * 1.155) - salary AS "Increase" FROM employees;
```

4. Write a query that displays the last name (with the first letter uppercase and all other letters lowercase) and the length of the last name for all employees whose name starts with the letters J, A, or M. Give each column an appropriate label. Sort the results by the employees' last names.

```
SELECT INITCAP(last_name) AS "Last Name", LENGTH(last_name) AS "Length" FROM  
employees WHERE last_name LIKE 'J%' OR last_name LIKE 'A%' OR last_name LIKE 'M%'  
ORDER BY last_name;
```

5. Rewrite the query so that the user is prompted to enter a letter that starts the last name. For example, if the user enters H when prompted for a letter, then the output should show all employees whose last name starts with the letter H.

```
SELECT INITCAP(last_name) AS "Last Name", LENGTH(last_name) AS "Length" FROM  
employees WHERE last_name LIKE '&enter_letter%' ORDER BY last_name;
```

6. The HR department wants to find the length of employment for each employee. For each employee, display the last name and calculate the number of months between today and the date on which the employee was hired. Label the column MONTHS_WORKED. Order your results by the number of months employed. Round the number of months up to the closest whole number.

```
SELECT last_name, CEIL(MONTHS_BETWEEN(SYSDATE, hire_date)) AS  
"MONTHS_WORKED" FROM employees ORDER BY "MONTHS_WORKED";
```

Note: Your results will differ.

7. Create a report that produces the following for each employee:

<employee last name> earns <salary> monthly but wants <3 times salary>. Label the column Dream Salaries.

```
SELECT last_name || ' earns ' || salary || ' monthly but wants ' || salary * 3 AS "Dream Salaries"  
FROM employees;
```

8. Create a query to display the last name and salary for all employees. Format the salary to be 15 characters long, left-padded with the \$ symbol. Label the column SALARY.

```
SELECT last_name, LPAD(salary, 15, '$') AS "SALARY" FROM employees;
```

9. Display each employee's last name, hire date, and salary review date, which is the first Monday after six months of service. Label the column REVIEW. Format the dates to appear in the format similar to —Monday, the Thirty-First of July, 2000.¶

```
SELECT last_name, hire_date, TO_CHAR(NEXT_DAY(ADD_MONTHS(hire_date, 6),  
'MONDAY'), 'Day, "the" DD "of" Month, YYYY') AS "REVIEW" FROM employees;
```

10. Display the last name, hire date, and day of the week on which the employee started. Label the column DAY. Order the results by the day of the week, starting with Monday.

```
SELECT last_name, hire_date, TO_CHAR(hire_date, 'Day') AS "DAY" FROM employees  
ORDER BY TO_CHAR(hire_date, 'D') ASC;
```

Practice Questions

Introduction to Functions

1. For each task, choose whether a single-row or multiple row function would be most appropriate:
- a. Showing all of the email addresses in upper case letters
 - b. Determining the average salary for the employees in the sales department
 - c. Showing hire dates with the month spelled out (*September 1, 2004*)
 - d. Finding out the employees in each department that had the most seniority (the earliest hire date)
 - e. Displaying the employees' salaries rounded to the hundreds place
 - f. Substituting zeros for null values when displaying employee commissions.

- a. Showing all of the email addresses in upper case letters: Single-row function
- b. Determining the average salary for the employees in the sales department: Multiple-row function
- c. Showing hire dates with the month spelled out (September 1, 2004): Single-row function
- d. Finding out the employees in each department that had the most seniority (the earliest hire date): Multiple-row function
- e. Displaying the employees' salaries rounded to the hundreds place: Single-row function
- f. Substituting zeros for null values when displaying employee commissions: Single-row function

2. The most common multiple-row functions are: AVG, COUNT, MAX, MIN, and SUM. Give your own definition for each of these functions.

AVG: Calculates the average value of a numeric column.

COUNT: Counts the number of rows in a column or table.

MAX: Finds the maximum value in a column.

MIN: Finds the minimum value in a column.

SUM: Calculates the total sum of a numeric column.

3. Test your definitions by substituting each of the multiple-row functions in this query.

SELECT FUNCTION(salary) FROM employees

Write out each query and its results.

SELECT AVG(salary) FROM employees;

Result: Average salary of all employees.

SELECT COUNT(salary) FROM employees;

Result: Number of employees.

SELECT MAX(salary) FROM employees;

Result: Highest salary of all employees.

SELECT MIN(salary) FROM employees;

Result: Lowest salary of all employees.

SELECT SUM(salary) FROM employees;

Result: Total sum of all employees' salaries.

Case and Character Manipulation

1. Using the three separate words “Oracle,” “Internet,” and “Academy,” use one command to produce the following output:
The Best Class Oracle Internet Academy

```
SELECT 'The Best Class ' || 'Oracle' || ' ' || 'Internet' || ' ' || 'Academy' FROM DUAL;
```

2. Use the string “Oracle Internet Academy” to produce the following output: The Net net

```
SELECT 'The Net ' || SUBSTR('Oracle Internet Academy', 8, 3) || ' ' || LOWER(SUBSTR('Oracle Internet Academy', 8, 3)) FROM DUAL;
```

3. What is the length of the string “Oracle Internet Academy”?

```
SELECT LENGTH('Oracle Internet Academy') FROM DUAL;
```

4. What’s the position of “l” in “Oracle Internet Academy”?

```
SELECT INSTR('Oracle Internet Academy', 'l') FROM DUAL;
```

5. Starting with the string “Oracle Internet Academy”, pad the string to create

****Oracle****Internet****Academy****

```
SELECT RPAD('****' || 'Oracle', 11, '*') || RPAD('****' || 'Internet', 20, '*') || RPAD('****' || 'Academy', 30, '*') FROM DUAL;
```

Number Functions

1. Display Oracle database employee last_name and salary for employee_ids between 100 and 102. Include a third column that divides each salary by 1.55 and rounds the result to two decimal places.

```
SELECT last_name, salary, ROUND(salary / 1.55, 2) FROM employees WHERE employee_id  
BETWEEN 100 AND 102;
```

2. Display employee last_name and salary for those employees who work in department 80. Give each of them a raise of 5.333% and truncate the result to two decimal places.

```
SELECT last_name, TRUNC(salary * 1.05333, 2) AS salary FROM employees WHERE department_id =  
80;
```

3. Use a MOD number function to determine whether 38873 is an even number or an odd number.

```
SELECT MOD(38873, 2) FROM DUAL;
```

4. Use the DUAL table to process the following numbers:

845.553 - round to one decimal place

30695.348 - round to two decimal places

30695.348 - round to -2 decimal Places

2.3454 - truncate the 454 from the decimal place

```
SELECT ROUND(845.553, 1) AS rounded_one, ROUND(30695.348, 2) AS rounded_two,  
ROUND(30695.348, -2) AS rounded_negative_two, TRUNC(2.3454, 2) AS truncated FROM DUAL;
```

5. Divide each employee's salary by 3. Display only those employees' last names and salaries who earn a salary that is a multiple of 3.

```
SELECT last_name, salary FROM employees WHERE MOD(salary, 3) = 0;
```

6. Divide 34 by 8. Show only the remainder of the division. Name the output as EXAMPLE.

```
SELECT MOD(34, 8) AS EXAMPLE FROM DUAL;
```

7. How would you like your paycheck – rounded or truncated? What if your paycheck was calculated to be \$565.784 for the week, but you noticed that it was issued for \$565.78. The loss of .004 cent would probably make very little difference to you. However, what if this was done to a thousand people, a 100,000 people, or a million people! Would it make a difference then? How much difference?

```
SELECT ROUND(565.784, 2) AS rounded_pay FROM DUAL;
```

```
SELECT TRUNC(565.784, 2) AS truncated_pay FROM DUAL;
```

```
SELECT 0.004 * 1000000 AS total_loss FROM DUAL;
```