

EXERCISE-6**Single Row Functions****Objective**

After the completion of will be able to do the

- Describe various in SQL.
- Use character, in SELECT statement.
- Describe the use

Evaluation Procedure	Marks awarded
Practice Evaluation (5)	
Viva(5)	
Total (10)	
Faculty Signature	

this exercise, the students following:
types of functions available
number and date functions
of conversion functions.

Single row functions:

Manipulate data items.

Accept arguments and return one value.

Act on each row returned.

Return one result per row.

May modify the data type.

Can be nested.

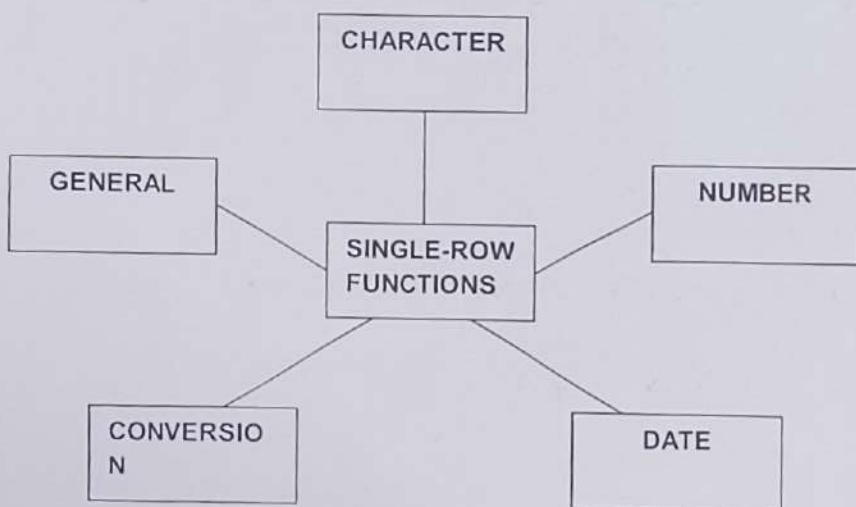
Accept arguments which can be a column or an expression

Syntax

Function_name(arg1,...argn)

An argument can be one of the following

- ✓ User-supplied constant
- ✓ Variable value
- ✓ Column name
- ✓ Expression



- Character Functions: Accept character input and can return both character and number values.

Find the Solution for the following:

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

Select current_date() AS Date;

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 AS 'Employee Number', 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 AS 'Employee Number', last_name,
salary, ROUND(salary * 1.155) AS 'New salary', ROUND
(salary * 1.55) - 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 CONCAT(UCASE(LEFT(last_name, 1)), LCASE(SUBSTRING(
last_name, 2))) AS 'Formatted Last Name', LENGTH(last_name)
AS 'Name Length' FROM employees WHERE last_name
LIKE 'J.%' OR last_name LIKE 'A.%' OR last_name
LIKE 'M.%';

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 CONCAT(UCASE(LEFT(last_name, 1)), LCASE(SUBSTRING(
last_name, 2))) AS 'Formatted Last Name', LENGTH(last_name)
AS 'Name Length' FROM employees WHERE last_name
LIKE CONCAT('?', '.', '%') 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(TIMESTAMPDIFF(MONTH, hire_date, CURRENT_DATE)) AS MONTHS_WORKED FROM
employees ORDER BY MONTHS_WORKED DESC;

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 CONCAT(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(CONCAT('$', salary),
15, '$') 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, DATE_FORMAT(
DATA-ADD(hire_date, INTERVAL (8 - DAYOFWEEK(
DATE-ADD(hire_date, INTERVAL 6 MONTH))) / 7 DAY) '%Y-%m-%d',`

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.

`AS
SELECT last_name, hire_date, DAYNAME(hire_date)
DAYFROM employees ORDER BY field('Monday', 'Tuesday', 'Wednesday', 'Thursday',
'Friday', 'Saturday');`

Evaluation Procedure	Marks awarded
Query(5)	5
Execution (5)	5
Viva(5)	5
Total (15)	15
Faculty Signature	P. M. Galur