**Name:=Purva Tagde PRN:=23070521111 SUB:=DBMS LAB**

**String Functions in SQL\*Plus (Oracle) & MySQL**

String functions allow you to **manipulate and process text data** in SQL. Below is a detailed comparison of **SQL\*Plus (Oracle)** and **MySQL** string functions, including examples.

# 1. String Functions in SQL\*Plus (Oracle)

## 1.1 CONCAT – String Concatenation

SELECT CONCAT('Hello', ' World') FROM dual; -- Result: Hello World SELECT 'Hello' || ' World' FROM dual; -- Alternative method using ||

**DONE**

**1.2 LENGTH – String Length**

SELECT LENGTH('Oracle Database') FROM dual; -- Result: 16

**DONE**

## 1.3 SUBSTR – Extract Substring

SELECT SUBSTR('Oracle Database', 8, 3) FROM dual; -- Extracts 'Dat' (Start from 8, length 3)

**DONE**

## 1.4 INSTR – Find Position of a Substring

SELECT INSTR('Oracle Database', 'D') FROM dual; -- Finds position of 'D' (Result: 8)

**DONE**

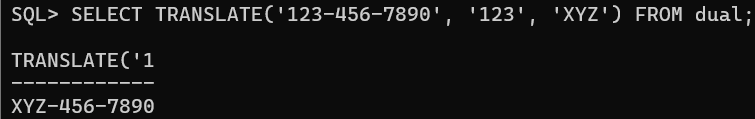
## 1.5 REPLACE – Replace a Substring

SELECT REPLACE('Oracle Database', 'Database', 'SQL') FROM dual; -- Result: Oracle SQL

**DONE**

## 1.6 TRANSLATE – Replace Multiple Characters

SELECT TRANSLATE('123-456-7890', '123', 'XYZ') FROM dual; -- Result: XYZ-456-7890



## 1.7 TRIM – Remove Spaces or Characters

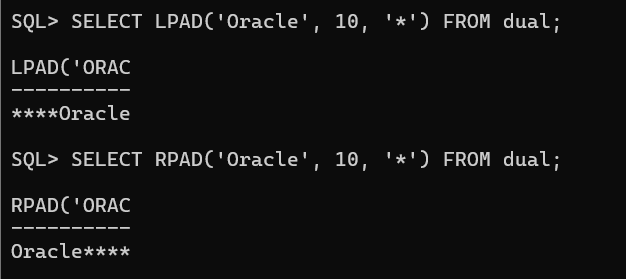
SELECT TRIM(' Oracle ') FROM dual; -- Removes leading and trailing spaces

SELECT TRIM('O' FROM 'Oracle') FROM dual; -- Removes 'O' from both ends

**DONE**

## 1.8 LPAD & RPAD – Padding Strings

SELECT LPAD('Oracle', 10, '\*') FROM dual; -- Result: \*\*\*\*Oracle SELECT RPAD('Oracle', 10, '\*') FROM dual; -- Result: Oracle\*\*\*\*



## 1.9 LOWER, UPPER, INITCAP – Case Conversion

SELECT LOWER('Oracle Database') FROM dual; -- Result: oracle database

SELECT UPPER('Oracle Database') FROM dual; -- Result: ORACLE DATABASE

SELECT INITCAP('oracle database') FROM dual; -- Result: Oracle Database

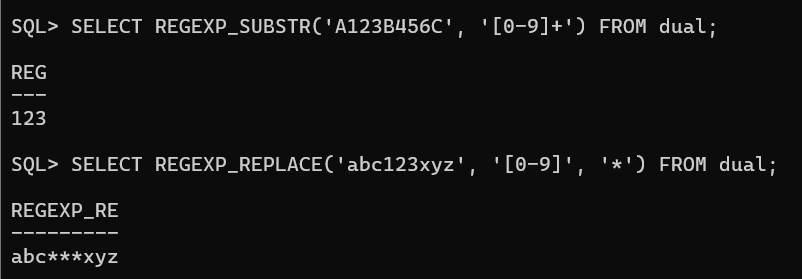
**DONE**

## 1.10 REGEXP Functions – Regular Expressions

SELECT REGEXP\_SUBSTR('A123B456C', '[0-9]+') FROM dual; -- Extracts

first number (Result: 123)

SELECT REGEXP\_REPLACE('abc123xyz', '[0-9]', '\*') FROM dual; -- Replaces digits with '\*' (Result: abc\*\*\*xyz)



# 2. String Functions in MySQL

**2.1 CONCAT – String Concatenation**

SELECT CONCAT('Hello', ' World'); -- Result: Hello World

**2.2 LENGTH – String Length**

SELECT LENGTH('MySQL Database'); -- Result: 15

## 2.3 SUBSTRING – Extract Substring

SELECT SUBSTRING('MySQL Database', 8, 3); -- Extracts 'Dat' (Start from 8, length 3)

## 2.4 LOCATE & INSTR – Find Position of a Substring

SELECT LOCATE('D', 'MySQL Database'); -- Result: 8

SELECT INSTR('MySQL Database', 'D'); -- Result: 8

## 2.5 REPLACE – Replace a Substring

SELECT REPLACE('MySQL Database', 'Database', 'Server'); -- Result: MySQL Server

## 2.6 TRIM – Remove Spaces or Characters

SELECT TRIM(' MySQL '); -- Removes leading and trailing spaces

SELECT TRIM('M' FROM 'MySQL'); -- Removes 'M' from both ends

## 2.7 LPAD & RPAD – Padding Strings

SELECT LPAD('MySQL', 10, '\*'); -- Result: \*\*\*\*\*MySQL

SELECT RPAD('MySQL', 10, '\*'); -- Result: MySQL\*\*\*\*\*

## 2.8 LOWER, UPPER – Case Conversion

SELECT LOWER('MySQL Database'); -- Result: my database

SELECT UPPER('MySQL Database'); -- Result: MYSQL DATABASE

## 2.9 REGEXP Functions – Regular Expressions

SELECT REGEXP\_SUBSTR('abc123xyz', '[0-9]+'); -- Extracts first number (Result: 123)

SELECT REGEXP\_REPLACE('abc123xyz', '[0-9]', '\*'); -- Replaces digits with '\*' (Result: abc\*\*\*xyz)

# 3. Key Differences Between SQL\*Plus (Oracle) and MySQL String Functions

|  |  |  |
| --- | --- | --- |
| **Function** | **Oracle (SQL\*Plus)** | **MySQL** |
| Concatenation | CONCAT(str1, str2) or `' |  |
| Substring | SUBSTR(str, start, length) | SUBSTRING(str, start, length) |
| Find Position | INSTR(str, substring) | LOCATE(substring, str) or INSTR(str, substring) |
| Replace  Substring | REPLACE(str, old, new) | REPLACE(str, old, new) |
| Trim Spaces | TRIM(str) | TRIM(str) |
| Padding | LPAD(str, length, pad\_char), RPAD(str, length, pad\_char) | LPAD(str, length, pad\_char), RPAD(str, length, pad\_char) |
| Case  Conversion | UPPER(str), LOWER(str), INITCAP(str) | UPPER(str), LOWER(str) |
| Regular  Expressions | REGEXP\_SUBSTR(), REGEXP\_REPLACE() | REGEXP\_SUBSTR(), REGEXP\_REPLACE() |

# 4. Special Notes

* Oracle has **INITCAP()**, which capitalizes the first letter of each word, whereas MySQL does **not**.
* **CONCAT()** in Oracle only takes **two** arguments, while in MySQL it can take **multiple**.
* Regular expressions (REGEXP\_...) are available in both, but Oracle has more advanced capabilities.