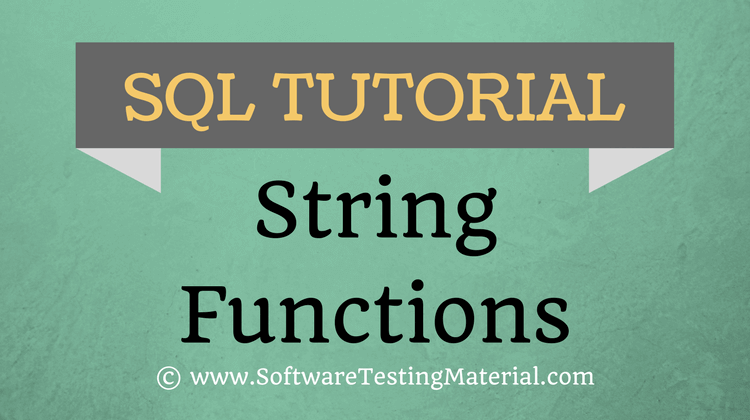
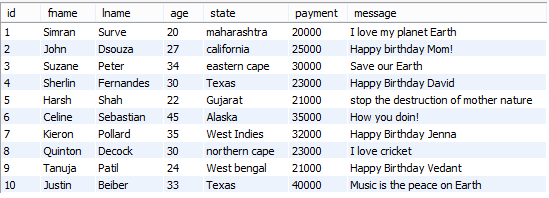
**SQL\_String Transformations**



SQL string functions are used primarily for string manipulation. There are many functions for strings; we will be covering the major functions that are used widely.

**Schema Used: Msg table**

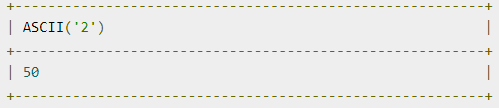


## ASCII(str)

Returns the numeric value of the leftmost character of the string str. Returns 0 if string is the empty string. Returns NULL if string is NULL. ASCII() works for characters with numeric values from 0 to 255.

Example:

SELECT ASCII('2');



**Q1. Select the ASCII value of the fname column.**

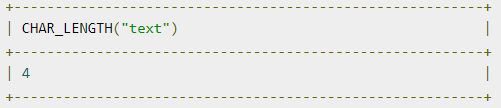
select ascii(fname) from msg;

## CHAR\_LENGTH(str)

Returns the length of the string measured in characters.

Example:

SELECT CHAR\_LENGTH("text");



**Q2. Display the length of the messages sent by the customers.**

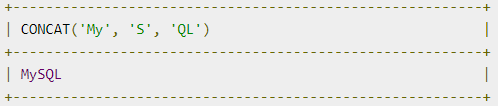
select message,char\_length(message) as length\_of\_message from msg;

## CONCAT(str1,str2,...)

Returns the string that results from concatenating the arguments.

Example:

SELECT CONCAT('My', 'S', 'QL');



**Q3. Concatenate the first name and last name of the customers.**

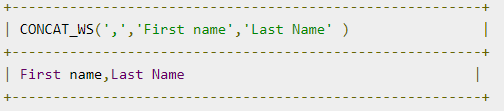
select fname,lname,concat(fname,lname) as full\_name from msg;

## CONCAT\_WS(separator,str1,str2,...)

CONCAT\_WS() stands for Concatenate With Separator

Example:

SELECT CONCAT\_WS(',','First name','Last Name' );



**Q4.Concatenate the first name and last name of the customers**

**with a suitable separator.**

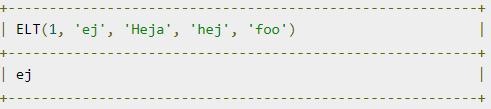
select fname,lname,concat\_ws(' ',fname,lname) as full\_name from msg;

## ELT(N,str1,str2,str3,...)

Returns str1 if N = 1, str2 if N = 2, and so on. Returns NULL if N is less than 1 or greater than the number of arguments.

Example:

SELECT ELT(1, 'ej', 'Heja', 'hej', 'foo');



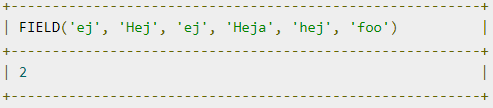
**Q5. Find the age column from id,lname,age,payment.**

select elt(3,id,lname,age,payment) from msg;

## FIELD(str,str1,str2,str3,...)

Returns the index (position starting with 1) of str in the str1, str2, str3, ... list. Returns 0 if str is not found.

SELECT FIELD('ej', 'Hej', 'ej', 'Heja', 'hej', 'foo');



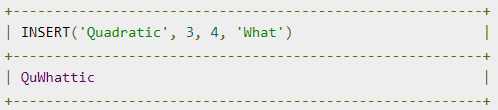
**Q6. Find the position of age in id,lname,age,payment.**

select field(age,id,lname,age,payment) as 'index' from msg;

## INSERT(str,pos,len,newstr)

Returns the string str, with the substring beginning at position pos and len characters long replaced by the string newstr.

SELECT INSERT('Quadratic', 3, 4, 'What');



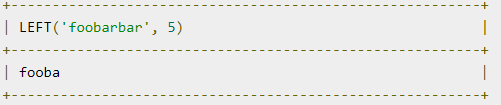
**Q7.Replace the words in state column with fname from the 4th position for 3 characters.**

select insert(state,4,3,fname) as 'new values' from msg;

## LEFT(str,len)

Returns the leftmost len characters from the string str, or NULL if any argument is NULL.

SELECT LEFT('foobarbar', 5);



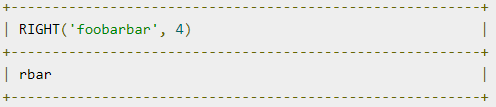
**Q8. Return the 6 left characters of the messages.**

select left(message,6) as 'left characters' from msg;

## RIGHT(str,len)

Returns the rightmost len characters from the string str, or NULL if any argument is NULL.

SELECT RIGHT('foobarbar', 4);



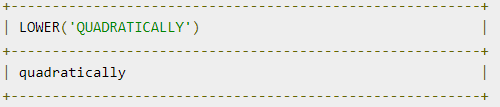
**Q9. Return the 4 right characters of the state.**

select right(state,4) as 'right characters' from msg;

## LOWER(str)

Returns the string str with all characters changed to lowercase.

SELECT LOWER('QUADRATICALLY');



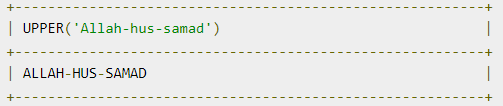
**Q10. Convert the messages to lowercase.**

select lower(message) as 'small message' from msg;

## UPPER(str)

Returns the string str with all characters changed to uppercase.

SELECT UPPER('Allah-hus-samad');



**Q11. Convert the messages to uppercase.**

select upper(message) as 'big message' from msg;

## SPACE(N)

Returns a string consisting of N space characters.

SELECT SPACE(6);



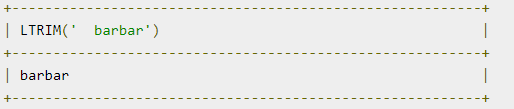
**Q12. Create a string of 15 spaces.**

select space(15) as '15 spaces';

## LTRIM(str)

Returns the string str with leading space characters removed.

SELECT LTRIM(' barbar');



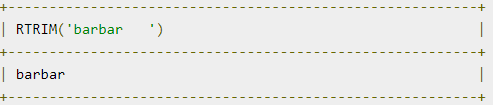
**Q13. Remove the spaces from “ CloudyML”.**

select ltrim(' cloudml');

## RTRIM(str)

Returns the string str with trailing space characters removed.

SELECT RTRIM('barbar ');



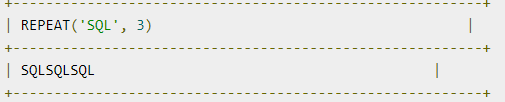
**Q14. Remove the spaces from “CloudyML ”.**

select rtrim('cloudml ');

## REPEAT(str,count)

Returns a string consisting of the string str repeated count times.

SELECT REPEAT('SQL', 3);



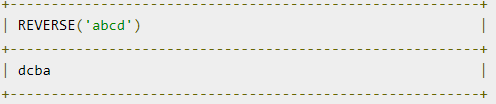
**Q15. Repeat the first name 4 times.**

select repeat(fname,4) as 'multiple names' from msg;

## REVERSE(str)

Returns the string str with the order of the characters reversed.

SELECT REVERSE('abcd');

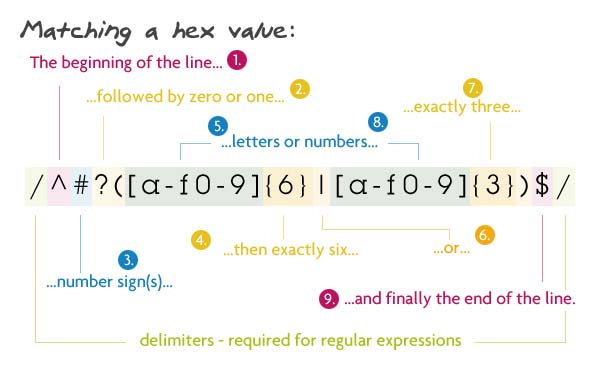


**Q16. Reverse the messages.**

elect reverse(message) as 'reverse message' from msg;

**SQL Regular Expressions**

A Regular Expression is popularly known as RegEx, is a generalized expression that is used to match patterns with various sequences of characters. A RegEx can be a combination of different [data types](https://www.edureka.co/blog/sql-basics/) such as integer, special characters, Strings, images, etc. Generally, these patterns are used in String searching algorithms in order to perform find or find and replace operations on Strings, or for validating the input.

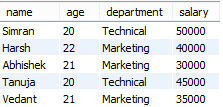


**Syntax for using SQL Regex**

SELECT statements... WHERE field\_name REGEXP 'my\_pattern';

|  |  |
| --- | --- |
| Pattern | Description |
| \* | Matches zero or more instances of the preceding String |
| + | Matches one or more instances of the preceding String |
| . | Matches any single character |
| ? | Matches zero or one instance of the preceding Strings |
| ^ | ^ matches the beginning of a String |
| $ | $ matches the ending of a String |

* **We will be using the wfuncs database which has the ‘employee’ table.**



1. **[abc]:**

Matches any character listed in between the square brackets

**Q1. Use Regex to display customer names of Texas state.**

select fname,state from msg where state regexp 'Texas';

1. **‘^abc’:**

Matches the characters at the beginning of a string

**Q2. Use Regex to display messages beginning with ‘Happy’.**

select message from msg where message regexp '^Happy';

1. **[A-Z]:**

Matches any letter in uppercase

**Q3. Use Regex to display first names matching with ‘[A-Z]’.**

select fname from msg where fname regexp '[A-Z]';

1. **[a-z]:**

Matches any letter in lowercase

**Q4. Use Regex to display first names beginning and ending with ‘[a-z]’.**

select fname from msg where fname regexp '^[a-z]$';

1. **[0-9] :**

Matches any digit between 0-9

**Q5. Use Regex to display the details of customers for which payment matching with ‘[0-9]’.**

select fname,payment from msg where payment regexp '[0-9]';

1. **p1|p2|p3:**

Matches any of the specified pattern

**Q6. Use Regex to display the name and age of customers having age 20 or 30.**

select fname,age from msg where age regexp '20|30';

1. **[abc$]:**

Matches the characters at the ending of a string

**Q7. Use Regex to display the name and message for customers whose message end with ‘Earth’.**

select fname,message from msg where message regexp 'Earth$';

1. **‘.’:**

Matches any number of characters

**Q8. Use Regex to display the first name and last name for which last name have ‘Se’ or “Su’ at the beginning.**

select fname,lname from msg where lname regexp 'S(u|e).';

1. **{n}**:

Matches n instances of the preceding element

**Q9. Use Regex to display the name and payment of customers beginning with 3 and having 5 digits.**

select fname,payment from msg where payment regexp '^3([0-9]{4})';

1. **[^abc]:**

Matches any character not listed in between the square brackets

**Q10. Use Regex to display customer names not beginning with ‘S’.**

select fname from msg where fname regexp '^[^S]';

