String Functions

In this session, you will learn:

- List of String Functions with example queries

CONCAT()

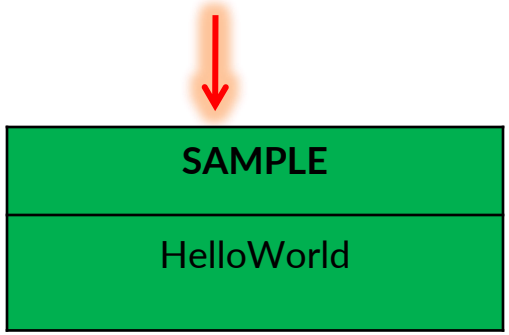
- allows you to concatenate two strings.

Syntax

CONCAT(string1,string2);

Example

Select CONCAT(‘Hello‘,‘World‘) AS SAMPLE FROM dual;



INITCAP()

- sets the first character in each word to uppercase and the rest to lowercase.

Syntax

INITCAP(string1);

Example

Select INITCAP('welcome you all') AS SAMPLE FROM dual;

SAMPLE
Welcome You All

LENGTH()

- This function returns the length of the specified string.

Syntax

```
LENGTH(str);
```

Example

```
SELECT LENGTH('Hello World') AS Length FROM dual;
```

Length
11

TRIM()

- This function to remove leading, trailing, or both leading and trailing unwanted characters from a string.

Syntax - LTRIM

LTRIM(str)

Example

```
SELECT
LTRIM ('      Hello World')
AS Result FROM dual;
```

Result
Hello World

Syntax - RTRIM

RTRIM(str)

Example

```
SELECT
RTRIM('Hello World        ')
AS Result FROM dual;
```

Result
Hello World

LPAD() & RPAD()

- These function pads the left-side and right-side of a string with specific set of characters.

Syntax - LPAD

LPAD(str,padded_len[,pad_str])

Example

```
SELECT
LPAD ('Hello',8,'H')
AS Result FROM dual;
```

Result
HHHHHello

Syntax - RPAD

RPAD(str,padded_len[,pad_str])

Example

```
SELECT
RPAD('Hello',8,'o')
AS Result FROM dual;
```

Result
Helloooo

LOWER() &UPPER()

- These function converts all letters in a string to uppercase or lowercase.

Syntax - LOWER

LOWER(string1)

Example

```
SELECT
LOWER ('HELLO WORLD')
AS Result FROM dual;
```

Result
hello world

Syntax - UPPER

UPPER(string1)

Example

```
SELECT
UPPER('Hello world')
AS Result FROM dual;
```

Result
HELLO WORLD

INSTR()

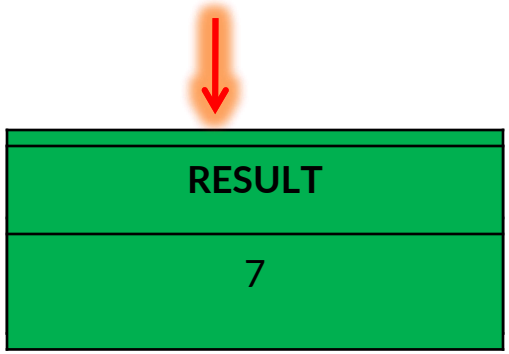
- This function returns the location of a substring in a string.

Syntax

INSTR(string, substring [, start_position [,nth_appearance]])

Example

Select INSTR('Welcome you all','e',1,2) AS RESULT FROM dual;



Select INSTR('Welcome you all','e') AS RESULT FROM dual;

RESULT
2

Metacharacters are special characters that have a special meaning, such as a wild card character, a repeating character, a nonmatching character, or a range of characters.

Symbol	Description
*	Matches zero or more occurrences
|	Alternation operator for specifying alternative matches
^/$	Matches the start of line and the end of line
[]	Bracket expression for a matching list matching any one of the expressions represented in the list
[^exp]	If the caret is inside the bracket, it negates the expression.
{m}	Matches exactly *m* times
{m,n}	Matches at least *m* times but no more than *n* times
[: :]	Specifies a character class and matches any character in that class
\	Can have four different meanings: (1) stand for itself; (2) quote the next character; (3) introduce an operator; (4) do nothing
+	Matches one or more occurrences
?	Matches zero or one occurrence
.	Matches any character in the supported character set (except NULL)
()	Grouping expression (treated as a single subexpression)
\n	Backreference expression
[==]	Specifies equivalence classes
[..]	Specifies one collation element (such as a multicharacter element)

Operator	Description
\d	Match a digit character
\D	Match a non-digit character
\w	Match a word character
\W	Match a non-word character
\s	Match a whitespace character
\S	Match a non-whitespace character
\A	Match only at beginning of string
\Z	Match only at end of string, or before newline at the end
\z	Match only at end of string
*?	Match 0 or more times (non-greedy)
+?	Match 1 or more times (non-greedy)
??	Match 0 or 1 time (non-greedy)
{n}?	Match exactly n times (non-greedy)
{n,}?	Match at least n times (non-greedy)
{n,m}?	Match at least n but not more than m times (non-greedy)

```
SELECT REGEXP_SUBSTR('example', '[^p]+') AS last_letter FROM dual;
```

LAST_LETTER
exam

[^p]: This part of the pattern matches any character that is not 'p'.

+: This part of the pattern matches one or more occurrences of the preceding character class (in this case, any character that is not 'p').

```
SELECT name FROM your_table WHERE SUBSTR(name, -1) = 'c';
```

SUBSTR(name, -1) function is used to extract the last character of the 'name' column,

REGEXP_INSTR()

- This function returns the location of a regular expression pattern in a string.

Syntax

```
REGEXP_INSTR( expression,pattern,
position, occurrence,
return_option, match_type  )
```

Example

```
Select REGEXP_INSTR('This returns','t') AS RESULT FROM dual;
```

RESULT
8

REPLACE()

- This function allows you to replace a string in a column of a table by a new string.

Syntax

```
REPLACE(string1,string_to_replace[,replacement_string]);
```

Example

```
SELECT REPLACE('HelloWorld','Hello','Hi') AS Result
FROM dual;
```

Result
HiWorld

REGEXP_REPLACE()

- This function allows you to replace a sequence of characters in a string with another set of characters using regular expression pattern matching.

Syntax

```
REGEXP_REPLACE( string,
pattern [, replacement_string [, start_position
[, nth_appearance
[, match_parameter ] ] ] ] )
```

Example

```
Select REGEXP_REPLACE('INDIA', '(.)', '\1 ') AS RESULT FROM dual;
```

(.) matches any single character and captures it.

\1 is a backreference that refers to the value captured

RESULT
INDIA

```
Select REGEXP_REPLACE  ('91105434563452345623',
'(^[[:digit:]]{4})(.*)([[:digit:]]{4}$)',
'\1**********\3')
AS card_number from dual;
```

CARD_NUMBER
9110**********5623

(^[[:digit:]]{4}): This is the first capturing group that matches the first four digits at the beginning of the string.

(.*): This is the second capturing group that matches any characters in between.

([[:digit:]]{4}$): This is the third capturing group that matches the last four digits at the end of the string.

\1 and \3 as backreferences to refer to the content captured by the first and third capturing groups in the regular expression.

SUBSTR()

- This function returns a substring with a given length from a string starting at a specific position.

Syntax

SUBSTR(str,position)

Example

```
SELECT
SUBSTR('Hello World',7)
AS Result FROM dual;
```

Result
World

Syntax

SUBSTR(str,position,length)

Example

```
SELECT
SUBSTR('Hello World',1,4)
AS Result FROM dual;
```

Result
Hell

REGEXP_SUBSTR()

- This function allows you to extract a substring from a string using regular expression pattern matching

Syntax

REGEXP_SUBSTR(string, pattern
[, start_position [, nth_appearance
[, match_parameter [, sub_expression]]]])

Example

Select REGEXP_SUBSTR('Welcome you all', '(\S*)(\s)',1,2)

as result FROM dual;

\s	Match a whitespace character
\S	Match a non-whitespace character

RESULT
you

```
Select REGEXP_SUBSTR('Welcome you all', '(\S*)(\s)',1,1) as result
FROM dual;
```

Outut: Welcome

```
Select REGEXP_SUBSTR('Welcome you all', '(\S*)(\s)',1,2)
as result FROM dual;
```

\S*: This part matches any non-whitespace character (\S) zero or more times (*).

\s: This part matches a single whitespace character.

1:specifies the position in the input string to start searching.

2: specifies the occurrence of the substring to return.

THANKS