String Functions in Oracle

- **LENGTH() Function in Oracle**
- **LOWER() Function in Oracle**
- **UPPER() Function in Oracle**
- **INITCAP() Function in Oracle**
- **LTIRM() Function in Oracle**
- **RTRIM() Function in Oracle**
- **TRIM() Function in Oracle**
- **LPAD() Function in Oracle**
- **RPAD() Function in Oracle**
- **CONCAT() Function in Oracle**
- **REPLACE() Function in Oracle**
- **TRANSLATE() Function in Oracle**
- **SUBSTR() Function in Oracle**
- **INSTR() Function in Oracle**

LENGTH() Function in Oracle:

The String LENGTH function in Oracle is used to return the length of a given string. Here, the given string can be any of the data types such as CHAR, VARCHAR2, NCHAR, NVARCHAR2, CLOB, or NCLOB.

The return value is always going to be of datatype NUMBER. If the given string has data
type CHAR, then the LENGTH function includes all the trailing blanks.

If the given string is null, then the LENGTH function returns null.

LENGTHB uses bytes instead of characters. LENGTHC uses Unicode complete characters. LENGTH2 uses UCS2 code points. LENGTH4 uses UCS4 code points.

```
SELECT LENGTH('HELLO') FROM DUAL;
SELECT LENGTH('GOOD MORNING') FROM DUAL;
SELECT LENGTHB('HELLO') FROM DUAL;
SELECT LENGTHB('GOOD MORNING') FROM DUAL;
```

LOWER() Function in Oracle:

The string LOWER() function in Oracle is used to return a specified character expression in lowercase letters.

```
SELECT LOWER('HELLO') FROM DUAL;
```

UPPER() Function in Oracle:

The string UPPER() function in Oracle is used to return a specified character expression in uppercase letters.

```
SELECT LOWER('hello') FROM DUAL;
```

INITCAP() Function in Oracle:

The string INITCAP() function in Oracle is used to set the first letter of each word in uppercase, and rest all other letters in lowercase.

Words are delimited by white space or characters that are not alphanumeric.
The given string can be of any of the data types such as CHAR, VARCHAR2,NCHAR, or NVARCHAR2. The return value is the same datatype as the givenstring. The database sets the case of the initial characters based on the binary mapping defined for the underlying character set.

```
SELECT INITCAP('hello') FROM DUAL;
```

```
SELECT INITCAP('welcome TO OrCale tuTOrials') INICAP FROM DUAL;
```

LTIRM() Function in Oracle:

The string LTRIM() Function in Oracle is used to remove the unwanted spaces (or) unwanted characters from the left side of the given string.

The string returned is of VARCHAR2 datatype if char is a character datatype and a LOB if char is a LOB datatype.

```
SELECT LTRIM(' ANURAG') TRIML FROM DUAL;
```

```
SELECT LTRIM('XXXXXXANURAG','X') TRIML FROM DUAL;
```

```
SELECT LTRIM('123ANURAG','123') TRIML FROM DUAL;
SELECT LTRIM('123ANURAG',' ANURAG') TRIML FROM DUAL;
```

RTRIM() Function in Oracle:

The string RTRIM() Function in Oracle is used to remove the unwanted spaces (or) unwanted characters from the right-hand side of the given string. The string returned is of VARCHAR2 datatype if char is a character datatype and LOB if char is a LOB datatype.

```
SELECT RTRIM(' ANURAG ') TRIMR FROM DUAL;
SELECT RTRIM('ANURAG ') TRIMR FROM DUAL;
SELECT RTRIM('ANURAGXXXXXX','X') TRIMR FROM DUAL;
SELECT RTRIM('ANURAG123','123') TRIMR FROM DUAL;
SELECT RTRIM('ANURAG123',' ANURAG') TRIMR FROM DUAL;
```

TRIM() Function in Oracle:

The string TRIM function in Oracle is used to remove the leading or trailing characters (or both) from a character string. If trim_character or trim_source is a character literal, then you must enclose it in single quotes. The following is the syntax to use the TRIM function in oracle.

```
SELECT TRIM(' Removing Leading and Trailing White Spaces ') LRTRIM FROM DUAL;
SELECT TRIM(' Removing Leading White Spaces') LRTRIM FROM DUAL;
SELECT TRIM('Removing Trailing White Spaces ') LRTRIM FROM DUAL;
SELECT TRIM(LEADING '6' FROM '660123') LRTRIM FROM DUAL;
SELECT TRIM(TRAILING '5' FROM '123455') LRTRIM FROM DUAL;
```

LPAD() Function in Oracle:

The string LPAD() function in Oracle is used to pad the left side of a string with a specific set of characters. So, in simple words, we can say that this LPAD function is used to fill a string with a specific character on the left side of a given string. This LPAD function is very useful for formatting the output of a query.

```
SELECT LPAD('Hello',10,'+') PADL FROM DUAL;
SELECT LPAD('Hello',4,'@') PADL FROM DUAL;
```

RPAD() Function in Oracle:

The string RPAD() function in Oracle is used to pad the right side of a given string with a specific set of characters. So, in simple words, we can say that the RPAD function is used to fill a string with a specific character on the right side of a given string. This RPAD function is useful for formatting the output of a query.

```
SELECT RPAD('Hello',10,'@') PADR FROM DUAL;
```

```
SELECT RPAD('Hello',4,'@') PADR FROM DUAL;
```

CONCAT() Function in Oracle:

The string CONCAT() function in Oracle is used to return the result (a string) of concatenating two string values. So, in simple words, we can say that this function is used to concatenate two expressions. This function is equivalent to the concatenation operator (||).

```
SELECT CONCAT('Good', 'Morning') ConcatString FROM DUAL;
```

REPLACE() Function in Oracle:

The string REPLACE function in Oracle is used to return a string with every occurrence of search_string replaced with replacement_string. If replacement_string is omitted or null, then all occurrences of search_string are removed. When search_string is null, then the string is returned. So, basically this string REPLACE function in Oracle is used to replace one string with another string.

```
SELECT REPLACE('JACK and JUE','J','BL') "New String" FROM DUAL;
```

```
SELECT REPLACE('MAN and MAT','M','F') "New String" FROM DUAL;
```

TRANSLATE() Function in Oracle:

The string TRANSLATE function in Oracle is used to return a string with all occurrences of each character specified in another string as 2nd argument replaced by its corresponding character specified in the 3rd argument. So, in simple words, we can say that this function is used to translate a single char with another single char. The Oracle string TRANSLATE function does not support the CLOB data type directly. However, the CLOB data can be passed in as arguments through implicit data type conversion.

```
SELECT TRANSLATE('comma|delimited|text', '|', ',') AS "New String" FROM DUAL;
SELECT TRANSLATE('MAP-SSS-PAM-MAN-NUT-TUB', 'AMST', 'BDFL') AS "New String" FROM DUAL;
SELECT TRANSLATE('So What', 'ah', 'e') AS "New String" FROM DUAL;
SELECT TRANSLATE('"Replace Double Quotes "', 'A"', 'A') AS "New String" FROM DUAL;
```

SUBSTR() Function in Oracle:

The SUBSTR function in Oracle is used to return the specified number (substring_length) of characters from a particular position of a given string. While working with Oracle SUBSTR functions, you need to remember the following points.

1. If the position is 0, then it is treated as 1.
2. If the position is positive, then Oracle Database counts from the beginning of char to find the first character.
3. If the position is negative, then Oracle counts backward from the end of char.
4. If substring_length is omitted, then Oracle returns all characters to the end of char. If substring_length is less than 1, then Oracle returns null.

```
SELECT SUBSTR('HELLO',2,3) "New String" FROM DUAL;
SELECT SUBSTR('WELCOME',4,2) "New String" FROM DUAL;
SELECT SUBSTR('WELCOME',-6,3) "New String" FROM DUAL;
SELECT SUBSTRB('ABCDEFG',5,4.2) "New String" FROM DUAL;
```

INSTR() Function in Oracle:

The string INSTR function in Oracle is used to search string for a substring. The function returns an integer indicating the position of the character in the string that is the first character of this occurrence. f a substring that is equal to substring is found, then the function returns an integer indicating the position of the first character of this substring. If no such substring is found, then the function returns zero.

```
SELECT INSTR('HELLO WELCOME','O') "New String" FROM DUAL;
SELECT INSTR('HELLO WELCOME','Z') "New String" FROM DUAL;
SELECT INSTR('HELLO WELCOME','O',1,2) "New String" FROM DUAL;
SELECT INSTR('HELLO WELCOME','E',5,2) "New String" FROM DUAL;
SELECT INSTR('HELLO WELCOME','E',1,4) "New String" FROM DUAL;
SELECT INSTR('HELLO WELCOME','E',-1,3) "New String" FROM DUAL;
SELECT INSTR('HELLO WELCOME','L',-4,3) "New String" FROM DUAL;
SELECT INSTR('HELLO WELCOME','L',-6,3) "New String" FROM DUAL;
```