SQL cmd: SubString & SUBSTRING_INDEX, SPLIT_PART slicing, CONCAT & CONCAT_WS, CAST & TRY_CAST, `TRIM`.

SUBSTRING

The `SUBSTRING()` function extracts a substring from a string (starting at any position).

Note: The `SUBSTR()` and `MID()` functions equals to the `SUBSTRING()` function.

```
SUBSTRING(string, start, length)

SELECT SUBSTRING(CustomerName, 2, 5) AS ExtractString
FROM Customers;

OR

SUBSTR(string, start, length)

SELECT SUBSTR("SQL Tutorial", -5, 5) AS ExtractString;

-- OUTPUT
-- orial

-- if in place of 5 there is -5 then,
SELECT SUBSTR("SQL Tutorial", -5, -5) AS ExtractString;

-- OUTPUT
-- NO VALUE. IT WILL NOT GIVE YOU ERROR, IT WILL GIVE YOU NO VALUE-BLANK
```

SUBSTRING_INDEX() OR SPLIT_PART

The `SUBSTRING_INDEX()` function returns a substring of a string before a specified number of delimiter occurs.

```
SUBSTRING_INDEX(string, delimiter, number)

SELECT SUBSTRING_INDEX("www.w3schools.com", ".", 2);
```

Parameter	Description
string	Required. The original string
delimiter	Required. The delimiter to search for
number	Required. The number of times to search for the *delimiter.* Can be both a positive or negative number. If it is a positive number, this function returns all to the left of the *delimiter*. If it is a negative number, this function returns all to the right of the *delimiter*.

`SPLIT_PART()` - WORKS ON SNOWFLAKE

Splits a given string at a specified character and returns the requested part.

If any parameter is NULL, NULL is returned.

```
SPLIT_PART(<string>, <delimiter>, <partNumber>)

SELECT SPLIT_PART(ORDER_ID, '-', -1) AS ORDER_EXTRACT FROM SALES_DATA;
```

CONCAT

The `CONCAT()` function adds two or more expressions together.

```
SELECT CONCAT("SQL ", "Tutorial ", "is ", "fun!") AS String;
-- OUTPUT
-- SQL Tutorial is fun!
```

`||` - We can use this in place of concat

```
SELECT substr('ADITYA', 1,1) || substr('SINGH', 1,1) AS NAME;
-- OUTPUT
-- AS
```

CAST

The `CAST()` function converts a value (of any type) into the specified datatype.

```
-- CAST(value AS datatype)
SELECT CAST(150 AS CHAR);
```

Parameter	Description	
value	Required. The value to convert	
datatype	Required.	The datatype to convert to. Can be one of the following:
	VALUE	**DESCRIPTION**
	DATE	Converts value to DATE. Format: "YYYY-MM-DD"
	DATETIME	Converts value to DATETIME. Format: "YYYY-MM-DD HH:MM:SS"
	DECIMAL	Converts value to DECIMAL. Use the optional M and D parameters to specify the maximum number of digits (M) and the number of digits following the decimal point (D).
	TIME	Converts value to TIME. Format: "HH:MM:SS"
	CHAR	Converts value to CHAR (a fixed length string)
	NCHAR	Converts value to NCHAR (like CHAR, but produces a string with the national character set)
	SIGNED	Converts value to SIGNED (a signed 64-bit

		integer)
	UNSIGNED	Converts value to UNSIGNED (an unsigned 64-bit integer)
	BINARY	Converts value to BINARY (a binary string)

In snowflake we use `TRY_CAST` if we don't need error instead it will give NULL values

```
SELECT TRY_CAST(150 AS CHAR);
```

TRIM

The `TRIM()` function removes leading and trailing spaces from a string.

```
SELECT TRIM('    SQL Tutorial    ') AS TrimmedString;

-- OUTPUT
-- SQL Tutorial

-- MYSQL
SELECT TRIM("*" FROM "****SQL Tutorial****") AS TrimmedString;

-- SNOWFLAKE
SELECT TRIM('*-*-ABC*-*-', '*-')
-- OUTPUT
-- ABC
```

LTRIM

The `LTRIM()` function removes leading spaces from a string.

```
SELECT LTRIM("     SQL Tutorial") AS LeftTrimmedString;

-- OUTPUT
-- SQL Tutorial
```

RTRIM

The `RTRIM()` function removes trailing spaces from a string.

```
SELECT RTRIM("SQL Tutorial     ") AS RightTrimmedString;

-- OUTPUT
-- SQL Tutorial
```

TO REMOVE VALUES AFTER DECIMAL PLACES USING RTRIM.

```
SELECT RTRIM('$ 125.21', '0.');

-- OUTPUT
-- $125
```