

Advanced SQL Tools: String Functions

Advanced SQL and Cloud

Objectives

1. Apply common string functions (UPPER, LOWER, TRIM, SUBSTRING, CONCAT, POSITION, SPLIT_PART) to clean and manipulate text data

String Functions

Analysts often work with messy text data.

String functions help clean and extract information from this messy text.

	name text	email text	comment text
1	John SMITH	JOHN.SMITH@EXAMPLE.COM	Loved the Product!!
2	jane.doe	jane.doe@example.com	late delivery, rude support
3	Alice Johnson	ALICE.JOHNSON@EXAMPLE.COM	Great service, fast shipping
4	bob_smith	bob_smith@example.com	Item arrived damaged
5	CAROL KING	CAROL.KING@EXAMPLE.COM	Excellent quality, will buy again

	name text	email text	comment text
1	John SMITH	JOHN.SMITH@EXAMPLE.COM	Loved the Product!!
2	jane.doe	jane.doe@example.com	late delivery, rude support
3	Alice Johnson	ALICE.JOHNSON@EXAMPLE.COM	Great service, fast shipping
4	bob_smith	bob_smith@example.com	Item arrived damaged
5	CAROL KING	CAROL.KING@EXAMPLE.COM	Excellent quality, will buy again

SELECT INITCAP(name) AS clean_name
 FROM users;

	clean_name text
1	John Smith
2	Jane.Doe
3	Alice Johnson
4	Bob_Smith
5	Carol King

	name text	email text	comment text
1	John SMITH	JOHN.SMITH@EXAMPLE.COM	Loved the Product!!
2	jane.doe	jane.doe@example.com	late delivery, rude support
3	Alice Johnson	ALICE.JOHNSON@EXAMPLE.COM	Great service, fast shipping
4	bob_smith	bob_smith@example.com	Item arrived damaged
5	CAROL KING	CAROL.KING@EXAMPLE.COM	Excellent quality, will buy again

```
SELECT LOWER(email) AS clean_email
FROM users;
```

	clean_email text
1	john.smith@example.com
2	jane.doe@example.com
3	alice.johnson@example.com
4	bob_smith@example.com
5	carol.king@example.com

	name text	email text	comment text
1	John SMITH	JOHN.SMITH@EXAMPLE.COM	Loved the Product!!
2	jane.doe	jane.doe@example.com	late delivery, rude support
3	Alice Johnson	ALICE.JOHNSON@EXAMPLE.COM	Great service, fast shipping
4	bob_smith	bob_smith@example.com	Item arrived damaged
5	CAROL KING	CAROL.KING@EXAMPLE.COM	Excellent quality, will buy again

SELECT

```
regexp_split_to_array(comment, '\W+') AS words
FROM users;
```

	words text[]
1	{Loved,the,Product,""}
2	{late,delivery,rude,support}
3	{Great,service,fast,shipping}
4	{Item,arrived,damaged}
5	{Excellent,quality,will,buy,again}

String Functions

There are string transformation functions:

- UPPER, LOWER
 - Converts string to uppercase or lowercase
 - Can be helpful for joining on text columns that may have mismatched case
- INITCAP
 - Capitalize the first letter of each word (title case)
- TRIM
 - Remove whitespace
- CONCAT
 - Concatenate two or more strings

String Functions

Functions for Extracting and Splitting:

- LEFT/RIGHT
 - Extract the first (last) n characters from a string
- SUBSTRING
 - Extracts a substring by specifying a starting position and number of characters
- SPLIT_PART
 - Divide a string based on a delimiter and extract a specified part
- STRING_TO_ARRAY
 - Split a string into an array based on a delimiter

String Functions

Functions for Pattern Matching

- REGEXP_MATCHES
 - Return an array of matches based on a regular expression pattern
- REGEXP_REPLACE
 - Replace substrings that match a regular expression
- REGEXP_SPLIT_TO_ARRAY/REGEXP_SPLIT_TO_TABLE
 - Split to an array or table based on a delimiter specified by a regular expression

String Functions

There are several ways to filter based on strings:

- LIKE and ILIKE for pattern matching with wildcards
 - Useful when you want to match simple patterns or partial text
 - % = any number of characters
 - _ = a single character
 - LIKE is case sensitive, ILIKE ignores case
- SIMILAR TO can do pattern matching with regular expressions
- There are other regular expression-based matching functions:
 - ~ is regex match
 - ~* is case-insensitive regex match
 - !~ and !~* are the negations of the above

String Functions

A complete list of string functions can be found at

<https://www.postgresql.org/docs/current/functions-string.html>