



# Introduction to Regular Expressions

Erland Sommarskog  
Data Platform MVP





# Erland Sommarskog

Independent consultant based in Stockholm

SQL Server MVP since 2001

[esquel@sommarskog.se](mailto:esquel@sommarskog.se)

<http://www.sommarskog.se>

Slides and scripts available on

<http://www.sommarskog.se/present>





## GOLD SPONSORS



Woodler



DATUMO

## SILVER SPONSOR

Xebia

## PARTNERS



Uniwersytet  
Bielsko-Bialski



# Agenda



- Introduction and overview.
- DEMOS!
- Reference-links slide.

**General caveat: SQL 2025 is still in preview. Things can change until release!**

# Why Regular Expressions?



- Advanced string manipulation.
- Example: replace multiple spaces with a single:

```
SELECT regexp_replace(  
  'A      text with      extra  space', ' +', ' ')
```

- Returns `A text with extra space`.
- Doing this in SQL 2022 is very difficult!
- We will see more examples later.

# Caveat about Implementations



- About every environment that has regular expressions has its own implementation.
- =>Just like SQL, there are subtle differences. Basic operations usually work the same, but a more “advanced” regexp may work differently in different environments - or simply be unsupported.
- SQL Server’s implementation is based on the RE2 library from Google.
- There are some irritating deviations.

# Overview of the RegExp Functions



- REGEXP\_LIKE - Does *string* match *regexp*?
- REGEXP\_REPLACE - Like the REPLACE function, but the search string can be a regular expression.
- REGEXP\_COUNT - How many matches of *regexp* are there in *string*?
- REGEXP\_SUBSTR - Returns the *n*-th match of *regexp* in *string*.
- REGEXP\_INSTR - Returns the start/end position of a match.
- REGEXP\_LIKE requires compatibility level 170. Remaining runs on all compat levels.

# Two Table-Valued Functions



- **REGEXP\_MATCHES** - Returns a table with all matches of *regexp* in *string*
- **REGEXP\_SPLIT\_TO\_TABLE** - Returns a table with the fragments of *string* outside matches of *regexp*.
  - This is **STRING\_SPLIT** on steroids!
- Both functions require compatibility level 170.



# Data-Type Support



Regexp functions support the string types in SQL Server with these restrictions:

- For **varchar(MAX)** and **nvarchar(MAX)**, input cannot exceed 2 MB.
- Currently, the table-valued functions do not support MAX types.
- The old types **text** and **ntext** are not supported.

# DEMO

# The Bottom Line



- With regular expressions you can perform advanced string operations that previously were impossible or very cumbersome to write.
- Complex regular expressions can certainly be deterring. But you will get used to it. :-)
- Be warned: You will still hit limits. Not all parsing is fit for regular expressions.
- It still a giant leap forwards.



# THANK YOU and References



- Regular Expressions Overview: <https://learn.microsoft.com/en-us/sql/relational-databases/regular-expressions/overview>.
- Regular Expression Functions: <https://learn.microsoft.com/en-us/sql/t-sql/functions/regular-expressions-functions-transact-sql>.
- Syntax Google RE2: <https://github.com/google/re2/wiki/Syntax>.
- Slides and scripts: <https://www.sommarskog.se/present>.
- Email: [esquel@sommarskog.se](mailto:esquel@sommarskog.se).