

# Tame Your Text Data with SQL



# Why Text Analytics?

---

- Unlock insights hidden in free-text survey responses
- Understand customer sentiment from reviews and feedback
- Categorize support tickets automatically for better routing
- Identify emerging trends and topics in large datasets
- Clean messy data for more accurate analysis
- Move beyond simple numerical metrics



# The LIKE Operator

- Use for simple pattern matching in text
- The percent sign (%) matches any sequence of characters
- The underscore (\_) matches any single character
- Example: Find all comments containing the word 'excellent'
- Query: `SELECT * FROM feedback WHERE comment LIKE '%excellent%'`
- Ideal for basic keyword searches



# CHARINDEX Function

- Finds the starting position of a substring within a string
- Returns an integer value for the character position
- Returns 0 if the substring is not found
- Syntax: CHARINDEX('search\_text', 'your\_string')
- Useful for conditional logic based on text presence
- Often used with other functions to extract parts of text



# PATINDEX Power

- Similar to CHARINDEX but uses more powerful patterns
- Allows for wildcard pattern matching like the LIKE operator
- Returns the starting position of the first pattern occurrence
- Syntax: PATINDEX('%pattern%', 'your\_string')
- Example: Find the position of the first number in a string
- More flexible for complex search criteria



# SUBSTRING & LEFT/RIGHT

---

- Extract specific parts of a text string
- SUBSTRING lets you pull text from any position
- LEFT returns a specified number of characters from the start
- RIGHT returns characters from the end of the string
- Often combined with CHARINDEX or PATINDEX for dynamic extraction
- Essential for parsing structured text within free-form fields



# REPLACE & STUFF

---

- **REPLACE** swaps all instances of one substring with another
- Great for correcting common typos or standardizing terms
- **STUFF** deletes a specified length and inserts another string
- Useful for masking sensitive information within text
- Both functions help clean and standardize messy data



# TRIM, LTRIM, RTRIM

---

- Remove unwanted spaces from your text data
- TRIM removes leading and trailing spaces
- LTRIM removes only leading spaces
- RTRIM removes only trailing spaces
- Critical for ensuring data consistency before analysis
- Prevents duplicate records due to spacing differences



# LOWER, UPPER, PROPER

---

- Standardize the case of your text data
- LOWER converts a string to all lowercase letters
- UPPER converts a string to all uppercase letters
- Some SQL dialects have PROPER for title case
- Ensures matching is not case-sensitive
- Makes your data look clean and professional



# LEN & DATALENGTH

---

- **LEN** returns the number of characters in a string
- **DATALength** returns the number of bytes used
- Useful for validating input length or filtering short responses
- Helpful for data quality checks and constraints
- **Note:** LEN does not count trailing spaces



# CONCAT & Concatenation

---

- Combine multiple strings into one
- Use the + operator or the CONCAT function
- CONCAT handles NULL values by treating them as empty strings
- Create full names from separate first and last name fields
- Build dynamic messages or labels from data



# Real Regex Power

---

- Some databases offer full regular expression support
- Functions like REGEXP\_SUBSTR or REGEXP\_REPLACE
- Provide extremely powerful and flexible pattern matching
- Match complex patterns like email addresses or phone numbers
- Currently available in PostgreSQL, Oracle, and BigQuery
- The ultimate tool for sophisticated text parsing



# Keyword Counting Strategy

---

- Use a combination of functions to count keyword frequency
- Create a temporary table or CTE of your target keywords
- Join or cross apply to your main text data
- Use LOWER on both sides for case-insensitive matching
- Count occurrences using string functions
- Aggregate and summarize to see trending topics



# Your Action Plan

---

- Start with basic LIKE searches for quick wins
- Use TRIM and LOWER to standardize your text first
- Practice extracting parts of strings with SUBSTRING
- Build a dictionary of common keywords for your domain
- Automate cleaning steps into your data pipeline
- Explore regex functions if your database supports them



**FOLLOW FOR MORE**



**Like this post? Repost!**

**Share your thoughts in the comments!**

