

# Funkcje tekstowe

Funkcja/Operator	Opis	Przykład użycia
<code>text \ \  text → text</code>	Łączy dwa ciągi znaków.	<code>'Post' \ \  'greSQL' → PostgreSQL</code>
<code>text \ \  anynonarray → text</code>	Łączy ciąg znaków z innym typem danych, który zostaje przekonwertowany na tekst.	<code>'Value: ' \ \  42 → Value: 42</code>
<code>btrim ( string text [, characters text ] ) → text</code>	Usuwa najdłuższy ciąg zawierający tylko znaki z <b>characters</b> (domyślnie spacje) z początku i końca ciągu.	<code>btrim('xyxtrimyyx', 'xyz') → trim</code>
<code>text IS [NOT] [form] NORMALIZED → boolean</code>	Sprawdza, czy ciąg jest w określonym formacie normalizacji Unicode.	<code>U&amp;'����������������' IS NFD NORMALIZED → t</code>
<code>bit_length ( text ) → integer</code>	Zwraca liczbę bitów w ciągu (8 razy długość w oktetach).	<code>bit_length('jose') → 32</code>
<code>char_length ( text ) → integer</code>	Zwraca liczbę znaków w ciągu.	<code>char_length('jos��') → 4</code>
<code>lower ( text ) → text</code>	Konwertuje ciąg na małe litery, zgodnie z regu��ami lokalizacji bazy danych.	<code>lower('TOM') → tom</code>
<code>lpad ( string text, length integer [, fill text ] ) → text</code>	Rozszerza ciąg do d��ugo��ci <b>length</b> , dodaj��c <b>fill</b> (domy��lnie spacje) z lewej strony. Je��li ciąg jest d��u��szy, zostaje przyci��ty.	<code>lpad('hi', 5, 'xy') → xyxhi</code>
<code>ltrim ( string text [, characters text ] ) → text</code>	Usuwa najd��u��szy ciąg zawieraj��cy tylko znaki z <b>characters</b> (domy��lnie spacje) z pocz��tku ci��gu.	<code>ltrim('zzzytest', 'xyz') → test</code>
<code>normalize ( text [, form ] ) → text</code>	Konwertuje ciąg do okre��lonej formy normalizacji Unicode.	<code>normalize(U&amp;'����������������', NFC) → U&amp;'����������'</code>
<code>octet_length ( text ) → integer</code>	Zwraca liczb�� bajt��w w ci��gu.	<code>octet_length('jos��') → 5 (je��li kodowanie serwera to UTF8)</code>

Funkcja/Operator	Opis	Przykład użycia
<code>octet_length ( character ) → integer</code>	Zwraca liczbę bajtów w ciągu, nie usuwając trailing spaces.	<code>octet_length('abc '::character(4)) → 4</code>
<code>overlay ( string text PLACING newsubstring text FROM start integer [ FOR count integer ] ) → text</code>	Zastępuje część ciągu, zaczynając od <code>start</code> -tej pozycji i długości <code>count</code> , nowym ciągiem.	<code>overlay('Txxxxas' placing 'hom' from 2 for 4) → Thomas</code>
<code>position ( substring text IN string text ) → integer</code>	Zwraca pierwszy indeks wystąpienia <code>substring</code> w <code>string</code> , lub 0, jeśli nie jest obecny.	<code>position('om' in 'Thomas') → 3</code>
<code>rpadd ( string text, length integer [, fill text ] ) → text</code>	Rozszerza ciąg do długości <code>length</code> , dodając <code>fill</code> (domyślnie spacje) z prawej strony. Jeśli ciąg jest dłuższy, zostaje przycięty.	<code>rpadd('hi', 5, 'xy') → hixyx</code>
<code>rtrim ( string text [, characters text ] ) → text</code>	Usuwa najdłuższy ciąg zawierający tylko znaki z <code>characters</code> (domyślnie spacje) z końca ciągu.	<code>rtrim('testxxzx', 'xyz') → test</code>
<code>substring ( string text [ FROM start integer ] [ FOR count integer ] ) → text</code>	Wyciąga podciąg z <code>string</code> , zaczynając od <code>start</code> -tej pozycji i kończąc po <code>count</code> znakach.	<code>substring('Thomas' from 2 for 3) → hom</code>
<code>substring ( string text FROM pattern text ) → text</code>	Wyciąga pierwszy podciąg pasujący do wzorca POSIX regular expression.	<code>substring('Thomas' from '...\$') → mas</code>
<code>substring ( string text SIMILAR pattern text ESCAPE escape text ) → text</code>	Wyciąga pierwszy podciąg pasujący do wzorca SQL regular expression.	<code>substring('Thomas' similar '%#"o_a#"_' escape '#') → oma</code>
<code>trim ( [ LEADING \\ TRAILING \\\ BOTH ] [ characters text ] FROM string text ) → text</code>	Usuwa najdłuższy ciąg znaków z początku, końca lub obu stron <code>string</code> zawierający tylko znaki z <code>characters</code> (domyślnie spacje).	<code>trim(both 'xyz' from 'yxTomxx') → Tom</code>
<code>unicode_assigned ( text ) → boolean</code>	Zwraca <code>true</code> , jeśli wszystkie znaki w ciągu mają przypisane kody Unicode.	<code>unicode_assigned('abc') → true</code>

Funkcja/Operator	Opis	Przykład użycia
<code>upper ( text ) → text</code>	Konwertuje ciąg na wielkie litery, zgodnie z regułami lokalizacji bazy danych.	<code>upper('tom') → TOM</code>