CONVERSION
- ASCII code of the first byte of the argument.
  ASCII ('k') = 120
  ASCII ('á') = 124
- Convert string to ASCII from another encoding (only supports conversion from LATIN1, LATIN2, LATIN5, and WIN1250 encodings)
  TO_ASCII ('Karel') = 'Karel'
- Character with the given code
  CHR (65) = 'A'
  CHR (1234) = 'Ӓ'
  CHR (NULL) = NULL
- Convert string to dest_encoding
  CONVERT ('Text', 'UTF8', 'LATIN1') = 'Text'
  CONVERT_TO ('Text', 'UTF8') = 'Text'
- Encode / Decode binary data into/from textual representation in string
  ENCODE (E'1234', 'base64') = 'MTIzNA=='
  DECODE (E'MTIAEw==', 'base64') = 123
- Convert the first letter of each word to upper case and the rest to lower case
  INITCAP ('hi thomas') = 'Hi Thomas'
  INITCAP ('all-in-all') = 'All-In-All'
  INITCAP ('all in all') = 'All In All'
  INITCAP ('go2bed') = 'go2bed'
- Convert string to lower / upper case
  LOWER ('TOM') = 'ton'
  UPPER ('tom') = 'TOM'
- Calculates the MD5 hash of string, returning the result in hexadecimal
  MDS('abc')='9b08159083cdd4fb0b6d6963f7d2e17f72'
  Current client encoding name
  PG_CLIENT_ENCODING () = 'UTF8'

RETURN THE GIVEN STRING SUITABLY QUOTED TO BE USED AS AN IDENTIFIER IN AN SQL STATEMENT STRING. QUOTES ARE ADDED ONLY IF NECESSARY. EMBEDDED QUOTES ARE PROPERLY DUBBED
- Return the given string quoted properly (two quotes back to back)
  TO_HEX (E'001234567890EF') = '001234567890EF'

MEASUREMENT
- Number of bits in string
  BIT_LENGTH ('3') = 8
  BIT_LENGTH ('0') = 16
  BIT_LENGTH ('jose') = 32
  BIT_LENGTH ('05e') = 48
- Number of characters in string
  CHAR_LENGTH ('jose') = 4
  CHARACTER_LENGTH ('jose') = 4
  LENGTH ('jose') = 4
  LENGTH ('05e', 'UTF8') = 4
- Number of bytes in string
  OCTET_LENGTH ('AB') = 2
  OCTET_LENGTH ('O') = 2
  OCTET_LENGTH ('€') = 4

REGULAR EXPRESSIONS
- Split string on delimiter and return the given field (counting from one)
  SPLIT_PART ('1,2,3', ',', 2) = '2'
- Extract substring
  SUBSTRING ('12345', 2 for 3) = '234'
  SUBSTR ('12345', 3, 2) = '34'
- Extract substring matching POSIX regular expression
  SUBSTRING ('RegEx from .\{3\}$', 'g') = 'gex'
- Extract substring matching SQL regular expression
  SUBSTRING('ABCDE' FROM '%\?B_D\?' FOR 'x') = 'BDC'
  SUBSTRING('ABCDE' FROM '%\?B_D\?' FOR 'x') = NULL
- Replace all occurrences of one substring with other substring
  REPLACE ('A-B-C', '.', '+') = 'A+B+C'

SEARCH & MATCHING
- Location of specified substring
  POSITION ('tom' IN 'Thomas') = 3
  STRPOS ('Thomas', 'tom') = 3
- Return all captured substrings resulting from matching a POSIX regular expression against the string
  REGEXP_MATCHES('12\d', '1\d', '2\d') = {12, 2}
  REGEXP_MATCHES('12\d', '1\d', '2\d') = {12, 2}
- Split string using a POSIX regular expression as the delimiter
  REGEXP_SPLIT_TO_ARRAY ('ABC DEF', E '\ s+') = {'ABC', 'DEF'}
  REGEXP_SPLIT_TO_TABLE ('ABC DEF', E '\ s+') = {'ABC', 'DEF'} (2 rows)
- Return true if the string matches the supplied pattern
  'ABC' LIKE '_B_' = true
  'ABC' NOT LIKE '_B_' = false
  'ABC' SIMILAR TO '[ABC]*' = true
  'ABC' NOT SIMILAR TO '[ABC]*' = false