BASE64ENCODE Function

Contents:

Converts an input value to base64 encoding with optional padding with an equals sign (=). Input can be of any type. Output type is String.

- base64 is a method of representing data in a binary format over text protocols. During encoding, text values are converted to binary values 0-63. Each value is stored as an ASCII character based on a conversion chart.
  - Typically, base64 is used to transmit binary information, such as images, over transfer methods that use text, such as HTTP.

    > **NOTE:** base64 is not an effective method of encryption.

  - For more information on base64, see *https://en.wikipedia.org/wiki/Base64*.

Wrangle vs. SQL: This function is part of Wrangle , a proprietary data transformation language. Wrangle is not SQL. For more information, see *Wrangle Language*.

Basic Usage

Column reference example:

```
base64encode(mySource)
```

Output: Returns the values from the `mySource` column written in base64 format.

String literal example:

```
base64encode('Hello, World. ', true)
```

Output: Returns the string: `GVsbG8sIFdvcmxkLiA=`. Note that the output string is padded with the equals sign at the end of the output value.

Syntax and Arguments

```
base64encode(column_string,bool_padding)
```

Copyright © 2020 Trifacta Inc.

Argument	Required?	Data Type	Description
column_string	Y	string	Name of the column or string literal to be applied to the function
bool_padding	N	Boolean	When `true`, excess padding in the data stream is padded with an equals sign (=) in the output. Default is `true`.

For more information on syntax standards, see *Language Documentation Syntax Notes*.

column_string

Name of the column or string constant to be converted.

- Missing string or column values generate missing string results.
- String constants must be quoted (`'Hello, World'`).
- Multiple columns and wildcards are not supported.

Usage Notes:

Required?	Data Type	Example Value
Yes	String literal or column reference	`myColumn`

bool_padding

Boolean value that determines if spaces are padded with the equals sign.

Base64 represents six-bit values (0-63). These values are represented in encoded values as ASCII characters, which are 8-bit values (0-255).

For any arbitrary input, it is possible that the number of bits required to represent it as a base64 value (number of characters * 6) won't precisely match up ASCII representation. Four sextets of base64 encoding map to three octets of ASCII encoding. If the input string has been fully encoded, but there are extra ASCII octets so that the number of output octets is divisible by four.

When this parameter is set to `true`, the output value is padded with the equals sign (=) to represent output octets that are generated but do not contain any data encoded from the input. The default is `true`.

For more information on base64 padding, see *https://en.wikipedia.org/wiki/Base64*.

Usage Notes:

Required?	Data Type	Example Value
No	Boolean	`false`

Examples

> **Tip:** For additional examples, see *Common Tasks*.

Example - base64 encoding and decoding

This example demonstrates base64 encoding functions in Trifacta Wrangler.

Copyright © 2020 Trifacta Inc.

- BASE64ENCODE - converts an input string to base64 encoding, with optional padding at the end. See *BASE64ENCODE Function*.
- BASE64DECODE - converts an input base64encoded-string back to ASCII text. See *BASE64DECODE Function*.

Source:

The following example contains three columns of different data types:

IntegerField	StringField	ssn
-2082863942	This is a test string.	987654321
2012994989	"Hello, world."	987654322
-1637187918	"Hello, world. Hello, world. Hello, world."	987654323
-1144194035	fyi	987654324
-971872543		987654325
353977583	This is a test string.	987-65-4321
-366583667	"Hello, world."	987-65-4322
-573117553	"Hello, world. Hello, world. Hello, world."	987-65-4323
2051041970	fyi	987-65-4324
522691086		987-65-4325

Transformation - encode:

You can use the following transformation to encode all of the columns in your dataset:

Transformation Name	`Edit column with formula`
Parameter: Columns	`All`
Parameter: Formula	`base64encode($col, true)`

Results - encode:

The transformed dataset now looks like the following. Note the padding (equals signs) at the end of some of the values. Padding is added by default.

IntegerField	StringField	ssn
LTIwODI4NjM5NDI=	VGhpcyBpcyBhIHRlc3Qgc3RyaW5nLg==	OTg3NjU0MzIx
MjAxMjk5NDk4OQ==	IkhlbGxvLCB3b3JsZC4i	OTg3NjU0MzIy
LTE2MzcxODc5MTg=	IkhlbGxvLCB3b3JsZC4gSGVsbG8sIHdvcmxkLiBIZWxsbywgd29ybGQuIg==	OTg3NjU0MzIz
LTExNDQxOTQwMzU=	Znlp	OTg3NjU0MzI0
LTk3MTg3MjU0Mw==		OTg3NjU0MzI1
MzUzOTc3NTgz	VGhpcyBpcyBhIHRlc3Qgc3RyaW5nLg==	OTg3LTY1LTQzMjE=
LTM2NjU4MzY2Nw==	IkhlbGxvLCB3b3JsZC4i	OTg3LTY1LTQzMjI=
LTU3MzExNzU1Mw==	IkhlbGxvLCB3b3JsZC4gSGVsbG8sIHdvcmxkLiBIZWxsbywgd29ybGQuIg==	OTg3LTY1LTQzMjM=
MjA1MTA0MTk3MA==	Znlp	OTg3LTY1LTQzMjQ=
NTIyNjkxMDg2		OTg3LTY1LTQzMjU=

Copyright © 2020 Trifacta Inc.

Transformation - decode:

The following transformation can be used to decode all of the columns:

Transformation Name	`Edit column with formula`
Parameter: Columns	`All`
Parameter: Formula	`base64decode($col)`

Results - decode:

IntegerField	StringField	ssn
-2082863942	This is a test string.	987654321
2012994989	"Hello, world."	987654322
-1637187918	"Hello, world. Hello, world. Hello, world."	987654323
-1144194035	fyi	987654324
-971872543		987654325
353977583	This is a test string.	987-65-4321
-366583667	"Hello, world."	987-65-4322
-573117553	"Hello, world. Hello, world. Hello, world."	987-65-4323
2051041970	fyi	987-65-4324
522691086		987-65-4325

Copyright © 2020 Trifacta Inc.