

BASE64DECODE Function

Converts an input base64 value to text. 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:

```
base64decode(mySource)
```

Output: Decodes the base64 values from the `mySource` column into text.

String literal example:

```
base64decode('GVsbG8sIFdvcmxkLiA=')
```

Output: Decodes the input value to the following text: `Hello, World. .`

Syntax and Arguments

```
base64decode(column_string)
```

| Argument | Required? | Data Type | Description |
|---------------|-----------|-----------|--|
| column_string | Y | string | Name of the column or string literal to be applied to the function |

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 |

Examples

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

Example - base64 encoding and decoding

This example demonstrates base64 encoding functions in Trifacta Wrangler.

- `BASE64ENCODE` - converts an input string to base64 encoding, with optional padding at the end. See *BASE64ENCODE Function*.
- `BASE64DECODE` - converts an input base64-encoded-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 | <code>base64encode(\$col, true)</code> |

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 |
|------------------|---|------------------|
| LTlwODI4NjM5NDI= | VGhpcyBpcyBhIHRlc3Qgc3RyaW5nLg== | OTg3NjU0Mzlx |
| MjAxMjk5NDk4OQ== | IkhlbGxvLCB3b3JsZC4i | OTg3NjU0Mzly |
| LTE2MzcxODc5MTg= | IkhlbGxvLCB3b3JsZC4gSGVsbG8sIHdvcmxkLiBIZWxsbywg29ybGQulG== | OTg3NjU0Mzlz |
| LTExNDQxOTQwMzU= | Znlp | OTg3NjU0MzI0 |
| LTk3MTg3MjU0Mw== | | OTg3NjU0MzI1 |
| MzUzOTc3NTgz | VGhpcyBpcyBhIHRlc3Qgc3RyaW5nLg== | OTg3LTU1LTQzMjE= |
| LTM2NjU4MzY2Nw== | IkhlbGxvLCB3b3JsZC4i | OTg3LTU1LTQzMjI= |
| LTU3MzExNzU1Mw== | IkhlbGxvLCB3b3JsZC4gSGVsbG8sIHdvcmxkLiBIZWxsbywg29ybGQulG== | OTg3LTU1LTQzMjM= |
| MjA1MTA0MTk3MA== | Znlp | OTg3LTU1LTQzMjQ= |
| NTlyNjkxMDg2 | | OTg3LTU1LTQzMjU= |

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 |