

# TIME Function

## Contents:

- *Basic Usage*
- *Syntax and Arguments*
  - *hour\_integer\_col*
  - *minute\_integer\_col*
  - *second\_integer\_col*
  - *time\_format\_string*
- *Examples*
  - *Example - date and time functions*

---

Generates time values from three inputs of Integer type: hour, minute, and second.

- Source values can be Integer literals or column references to values that can be inferred as Integers.
- If any of the source values are invalid or out of range, a missing value is returned.
- This function must be nested within another function that accepts date values as arguments. See the example below.

**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

### Integer literal values:

```
dateformat(time(23,58,59),'HH mm ss')
```

**Output:** Returns the following map:

```
23 58 59
```

### Column reference values:

```
dateformat(time(myHour, myMinute, mySecond), 'hh-mm-ss')
```

**Output:** Generates a column of values where:

- *hh* = values from *myHour* column
- *mm* = values from *myMinute* column
- *ss* = values from *mySecond* column

## Syntax and Arguments

```
dateformat(time(hour_integer_col,minute_integer_col,second_integer_col), 'time_format_string')
```

Argument	Required?	Data Type	Description
hour_integer_col	Y	integer	Name of column or Integer literal representing the hour value to apply to the function

minute_integer_col	Y	integer	Name of column or Integer literal representing the minute value to apply to the function
second_integer_col	Y	integer	Name of column or Integer literal representing the second value to apply to the function
time_format_string	Y	string	String literal identifying the time format to apply to the value

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

### hour\_integer\_col

Integer literal or name of the column containing integer values for the hour. Values must integers between 0 and 23, inclusive.

- Missing values for this function in the source data result in missing values in the output.
- Multiple columns and wildcards are not supported.

#### Usage Notes:

Required?	Data Type	Example Value
Yes	Integer (literal or column reference)	15

### minute\_integer\_col

Integer literal or name of the column containing integer values for the minutes. Values must integers between 0 and 59, inclusive.

- Missing values for this function in the source data result in missing values in the output.
- Multiple columns and wildcards are not supported.

#### Usage Notes:

Required?	Data Type	Example Value
Yes	Integer (literal or column reference)	23

### second\_integer\_col

Integer literal or name of the column containing integer values for the second. Values must integers between 0 and 59, inclusive.

- Missing values for this function in the source data result in missing values in the output.
- Multiple columns and wildcards are not supported.

#### Usage Notes:

Required?	Data Type	Example Value
Yes	Integer (literal or column reference)	45

### time\_format\_string

For more information on supported time formatting strings, see *Supported Data Types*.

For more information, see *DATEFORMAT Function*.

## Examples

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

### Example - date and time functions

This example illustrates how the `DATE` and `TIME` functions operate. Both functions require that their outputs be formatted properly using the `DATEFORMAT` function.

- `DATE` - Generates valid Datetime values from three integer inputs: year, month, and day. See *DATE Function*.
- `TIME` - Generates valid Datetime values from three integer inputs: hour, minute, and second. See *TIME Function*.
- `DATETIME` - Generates valid Datetime values from six integer inputs: year, month, day, hour, minute, and second. See *DATETIME Function*.
- `DATEFORMAT` - Formats valid Datetime values according to the provided formatting string. See *DATEFORMAT Function*.

#### Source:

year	month	day	hour	minute	second
2016	10	11	2	3	0
2015	11	20	15	22	30
2014	12	25	18	30	45

#### Transformation:

<b>Transformation Name</b>	New formula
<b>Parameter: Formula type</b>	Single row formula
<b>Parameter: Formula</b>	<code>DATEFORMAT(DATE (year, month, day), 'yyyy-MM-dd')</code>
<b>Parameter: New column name</b>	'fctn_date'

<b>Transformation Name</b>	New formula
<b>Parameter: Formula type</b>	Single row formula
<b>Parameter: Formula</b>	<code>DATEFORMAT(TIME (hour, minute, second), 'HH-mm-ss')</code>
<b>Parameter: New column name</b>	'fctn_time'

<b>Transformation Name</b>	New formula
<b>Parameter: Formula type</b>	Single row formula
<b>Parameter: Formula</b>	<code>DATEFORMAT(DATETIME (year, month, day, hour, minute, second), 'yyyy-MM-dd-HH:mm:ss')</code>

Parameter: New column name	'fctn_datetime'
----------------------------	-----------------

**Results:**

**NOTE:** All inputs must be inferred as Integer type and must be valid values for the specified input. For example, month values must be integers between 1 and 12, inclusive.

year	month	day	hour	minute	second	fctn_date	fctn_time	fctn_datetime
2016	10	11	2	3	0	2016-10-11	02-03-00	2016-10-11-02:03:00
2015	11	20	15	22	30	2015-11-20	15-22-30	2015-11-20-15:22:30
2014	12	25	18	30	45	2014-12-25	18-30-45	2014-12-25-18:30:45

You can apply other date and time functions to the generated columns. For an example, see *YEAR Function*.