**Dimension, filter, and parameter types**

bookmark\_border

* **On this page**
* [Usage](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#usage)
* [Type definitions](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#type_definitions)
* [bin](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#bin)
* [distance](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#distance)
* [duration](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#duration)

This page refers to the type parameter that is part of a [dimension](https://cloud.google.com/looker/docs/reference/param-field-dimension) or [filter](https://cloud.google.com/looker/docs/reference/param-field-filter).

type can also be used as part of a measure, described on the [Measure types](https://cloud.google.com/looker/docs/reference/param-measure-types) documentation page.

type can also be used as part of a dimension group, described on the [dimension\_group](https://cloud.google.com/looker/docs/reference/param-field-dimension-group) parameter documentation page.

**Usage**

view: view\_name {  
  dimension: field\_name {  
    type: field\_type  
  }  
}

|  |  |
| --- | --- |
| **Hierarchy**  [View File](https://cloud.google.com/looker/docs/reference/param-view)  [view](https://cloud.google.com/looker/docs/reference/param-view-view)  [Field Type](https://cloud.google.com/looker/docs/reference/param-field)  type | **Possible Field Types**  Dimension, Filter, Parameter  **Default Value**  string  **Accepts**  [A dimension, filter, or parameter type](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#type_definitions) |

This page includes additional details about the various types that can be assigned to a [dimension](https://cloud.google.com/looker/docs/reference/param-field-dimension), [filter](https://cloud.google.com/looker/docs/reference/param-field-filter), or [parameter](https://cloud.google.com/looker/docs/reference/param-field-parameter). A dimension, filter, or parameter can only have one type, which defaults to string if no type is specified.

Some types have supporting parameters, which are described within the appropriate section.

**Type definitions**

**D** = Dimension

**DG** = Dimension Group

**F** = Filter

**P** = Parameter

| **Type** | **Description** | **Valid Field Types** |
| --- | --- | --- |
| [bin](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#bin) | ADDED 21.14 For fields that group numerical values into several ranges | **D** |
| [date](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#date_types) | For fields that contain dates | **D** **F** **P** |
| [date\_time](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#time_types) | For fields that contain dates and times | **D** **F** **P** |
| [distance](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#distance) | For fields that calculate the distance of the most direct path ("as the crow flies") between two [type: location](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#location) dimensions | **D** |
| [duration](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#duration) | Used with a [dimension\_group](https://cloud.google.com/looker/docs/reference/param-field-dimension-group) to create several duration-based dimensions from a single table column. For information about dimension groups, see the [dimension\_group](https://cloud.google.com/looker/docs/reference/param-field-dimension-group) parameter documentation page. | **DG** |
| [location](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#location) | For fields that are based on a latitude and longitude and will be used in visualizations | **D** |
| [number](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#number) | For fields that contain numbers | **D** **F** **P** |
| [string](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#string) | For fields that contain letters or special characters | **D** **F** **P** |
| [tier](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#tier) | For fields that group numerical values into several ranges | **D** |
| [time](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#time) | Used with a [dimension\_group](https://cloud.google.com/looker/docs/reference/param-field-dimension-group) to create several time-based dimensions from a single table column. For information about dimension groups, see the [dimension\_group](https://cloud.google.com/looker/docs/reference/param-field-dimension-group) parameter documentation page. | **DG** |
| [unquoted](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#unquoted) | For [parameter](https://cloud.google.com/looker/docs/reference/param-field-parameter) fields whose values will be inserted directly into SQL, and therefore should not be quoted (as they would with [type: string](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#string)) | **P** |
| [yesno](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#yesno) | For fields that show if something is true or false | **D** **F** **P** |
| [zipcode](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#zipcode) | For fields that contain a zip code and will be used in visualizations | **D** |
| [Individual Time and Date Types](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#individual-timeframes) | A rarely used alternative to [type: time](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#time) for creating single, time-based dimensions | **D** **F** |
| [Individual Duration Types](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#individual-durations) | A rarely used alternative to [type: duration](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#duration) for creating single time-based dimensions that calculate time differences | **D** |
| int | REMOVED 5.4 Replaced by [type: number](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#number) | **D** |

bin

type: bin is an alias for [type: tier](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#tier). The two types can be used interchangeably.

type: bin is used in conjunction with the bins parameter to separate a numeric dimension into a set of number ranges. For example, you might bin an age dimension into different age ranges. You can change how the bins appear in the Looker UI with the [style](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#style) parameter.

The usage pattern is:

view: view\_name {  
  dimension: field\_name {  
    type: bin  
    bins: [numeric\_value, numeric\_value, ... ]  
    style: interval  
    sql: ${my\_field\_name} ;;  
  }  
}

The [sql](https://cloud.google.com/looker/docs/reference/param-field-sql) parameter for type: bin dimensions can take any valid SQL expression that results in a number or an integer.

The age example above might look like this:

dimension: age\_bin {

type: bin

bins: [0, 10, 20, 30, 40, 50, 60, 70, 80, 90]

style: interval

sql: ${age} ;;

}

The way that this would appear in the Looker UI is described in the [style](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#style) section for the [type: tier](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#tier) parameter on this page.

The bin type is an alias for [type: tier](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#tier). The two types can be used interchangeably, and the behavior is the same for both:

* The [style](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#style) subparameter is used to customize the appearance of bins in the Looker UI.
* Dimensions of type: bin cannot be used in [custom filters](https://cloud.google.com/looker/docs/filtering-and-limiting#custom_filters).
* Using type: bin in conjunction with [dimension fill](https://cloud.google.com/looker/docs/creating-visualizations#filling_in_missing_dates_and_values) can result in unexpected tier buckets.

distance

type: distance is used to calculate the distance of the most direct path ("as the crow flies") between two [type: location](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#location) dimensions.

The [sql](https://cloud.google.com/looker/docs/reference/param-field-sql) parameter for type: distance dimensions is excluded. Instead, you supply a reference to a [type: location](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#location) dimension in the start\_location\_field and end\_location\_field parameters.

The usage is:

view: view\_name {  
  dimension: field\_name {  
    type: distance  
    start\_location\_field: field\_name\_1  
    end\_location\_field: field\_name\_2  
    units: kilometers  
  }  
}

The unit of distance is determined by the units parameter, which can take the following values:

* feet
* kilometers
* meters
* miles
* nautical\_miles
* yards

For example, you might calculate the distance traveled by a customer to pickup a rental like this:

dimension: distance\_to\_pickup {

type: distance

start\_location\_field: customer.home\_location

end\_location\_field: rental.pickup\_location

units: miles

}

The distance calculated will be the most direct path between the two points, not necessarily the distance traveled by road.

Do not use the [${view\_name.field\_name}](https://cloud.google.com/looker/docs/sql-and-referring-to-lookml#substitution_operator_($)) syntax in the start\_location\_field and end\_location\_field parameters. Instead, use the view name and field name by themselves, like view\_name.field\_name.

duration

type: duration is used in conjunction with a [dimension\_group](https://cloud.google.com/looker/docs/reference/param-field-dimension-group) to create a set of calculated time differences between dimensions and/or SQL expressions.

type: duration works only with a dimension\_group and will *not* work with a regular [dimension](https://cloud.google.com/looker/docs/reference/param-field-dimension). However, you can specify individual duration-based dimensions, as explained [in this section below](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#individual-durations).

For information about dimension groups with type: duration, see the [dimension\_group](https://cloud.google.com/looker/docs/reference/param-field-dimension-group" \l "duration_type) parameter documentation page.

location

type: location is used in conjunction with the sql\_latitude and sql\_longitude parameters to create coordinates that you want to plot on a [Map](https://cloud.google.com/looker/docs/map-options) or [Static Map (Points)](https://cloud.google.com/looker/docs/map-points-options) visualization (use a state or country field for [Static Map (Regions)](https://cloud.google.com/looker/docs/map-regions-options)), or that you want to use in a [type: distance](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#distance) calculation.

The usage pattern is:

view: view\_name {  
  dimension: field\_name {  
    type: location  
    sql\_latitude:${field\_name\_1} ;;  
    sql\_longitude:${field\_name\_2} ;;  
  }  
}

The [sql](https://cloud.google.com/looker/docs/reference/param-field-sql) parameter for type: location dimensions is excluded. Instead, you supply any valid SQL expression that results in a decimal latitude or longitude to the sql\_latitude and sql\_longitude parameters. These are usually references to LookML fields that contain latitude or longitude information, but they can be static values if you wanted to have a location of your headquarters, or something along those lines.

For example, you might create a store\_location dimension like this:

dimension: store\_location {

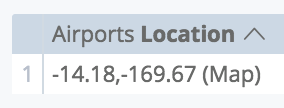
type: location

sql\_latitude: ${store\_latitude} ;;

sql\_longitude: ${store\_longitude} ;;

}

If you do not want to plot the locations or calculate distances, you can use a simpler type such as [type: number](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#number). When you view a location in a table, it will show the value from your database, as well as automatically generate a link to that location in Google Maps:



**Supported database dialects for**location

For Looker to support type: location in your Looker project, your database dialect must also support it. The following table shows which dialects support type: location in the latest release of Looker:

number

type: number is used with numbers or integers.

The [sql](https://cloud.google.com/looker/docs/reference/param-field-sql) parameter for type: number dimensions can take any valid SQL expression that results in a number or an integer.

type: number fields can be formatted by using the [value\_format](https://cloud.google.com/looker/docs/reference/param-field-value-format) or [value\_format\_name](https://cloud.google.com/looker/docs/reference/param-field-value-format-name) parameters.

For example, the following LookML creates a field called profit based on the revenue and cost fields, then displays it in a money format ($1,234.56):

dimension: profit {

type: number

sql: ${revenue} - ${cost} ;;

value\_format\_name: usd

}

A dimension can only perform arithmetic on other dimensions, not measures. Additionally, type: number dimensions will not provide suggestions to users, even if you are using them to display ID numbers.

string

type: string is typically used with fields that contain letters or special characters. It can be used with number fields as well, although Looker has better features for handling numbers if you use [type: number](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#number) instead.

The [sql](https://cloud.google.com/looker/docs/reference/param-field-sql) parameter for type: string dimensions can take any valid SQL expression.

For example, the following LookML creates the field full\_name by combining a field called first\_name and last\_name:

dimension: full\_name {

type: string

sql: CONCAT(${first\_name}, ' ', ${last\_name}) ;;

}

In this example type: string could be omitted, because string is the default value for type.

tier

You can use [type: bin](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#bin) as an alias for type: tier. The two types can be used interchangeably.

type: tier is used in conjunction with the tiers parameter to separate a numeric dimension into a set of number ranges. For example, you might tier an age dimension into different age ranges. You can change how the tiers appear in the Looker UI with the [style](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#style) parameter.

The usage pattern is:

view: view\_name {  
  dimension: field\_name {  
    type: tier  
    tiers: [numeric\_value, numeric\_value, ... ]  
    style: interval  
    sql: ${my\_field\_name} ;;  
  }  
}

The [sql](https://cloud.google.com/looker/docs/reference/param-field-sql) parameter for type: tier dimensions can take any valid SQL expression that results in a number or an integer.

The age example above might look like this:

dimension: age\_tier {

type: tier

tiers: [0, 10, 20, 30, 40, 50, 60, 70, 80]

style: classic # the default value, could be excluded

sql: ${age} ;;

}

The way that this would appear in the Looker UI is described in the [style](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#style) section on this page.

Dimensions of type: tier cannot be used in [custom filters](https://cloud.google.com/looker/docs/filtering-and-limiting#custom_filters).

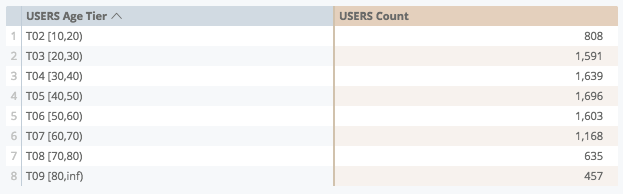
style

The style parameter lets you change the way that tiers appear in the Looker UI. Although not shown in the examples below, if there were negative numbers in the data, there would be a beginning tier that would include all numbers from negative infinity up to but not including 0. There are four possible values:

* [classic](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#style_classic) (the default value)
* [interval](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#style_interval)
* [integer](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#style_integer)
* [relational](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#style_relational)

classic

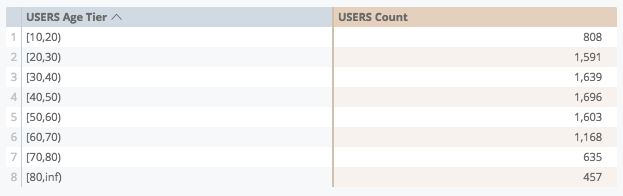
style: classic is the default, and looks like:



* You can interpret this tier notation as follows:
  + **T02 [10,20)** is the range including 10, and up to but not including 20
  + **T09 [80,inf)** is the range including 80, and up to infinity

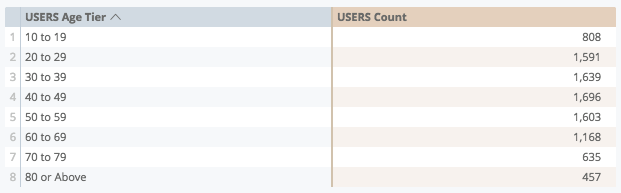
interval

style: interval is similar to style: classic, but it does not have the leading **TXX** labels. It looks like:



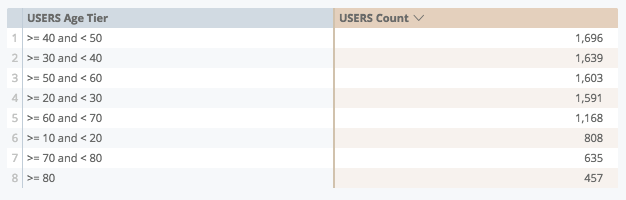
integer

style: integer must be used with discrete integer values (such as age). If you try to use non-integers to define the tiers you will receive an error. This style looks like:



relational

style: relational is best used with continuous numbers (such as dollars) and looks like:



You can also style tiers with [value\_format](https://cloud.google.com/looker/docs/reference/param-field-value-format). For example:

dimension: amount\_tier {

type: tier

tiers: [0, 10, 20, 30, 40, 50, 60, 70, 80]

style: integer

sql: ${amount} ;;

value\_format: "$#,##0"

}

This example would result in tier labels like $10 to $19, $20 to $29, and so on.

**Things to consider**

Using tier in conjunction with [dimension fill](https://cloud.google.com/looker/docs/creating-visualizations#filling_in_missing_dates_and_values) can result in unexpected tier buckets.

For example, a dimension of type: tier, **Age Tier**, will display tier buckets for **Below 0** and **0 to 9** when dimension fill is enabled, although the data does not include age values for those buckets:



When dimension fill is disabled for **Age Tier**, the buckets more accurately reflect the age values that are available in the data:



You can enable or disable dimension fill by hovering over the dimension name in the Explore, clicking the field-level [gear icon](https://cloud.google.com/looker/docs/creating-visualizations#to_use_dimension_fill), and selecting either **Remove Filled in Tier Values** to disable, or **Fill in Missing Tier Values** to enable.

time

type: time is used in conjunction with a [dimension\_group](https://cloud.google.com/looker/docs/reference/param-field-dimension-group) and the [timeframes](https://cloud.google.com/looker/docs/reference/param-field-dimension-group#timeframes) parameter to create a set of time-based dimensions. For example, you could easily create a date, week, and month dimension based on a single timestamp column.

type: time works only with a dimension\_group and will *not* work with a regular [dimension](https://cloud.google.com/looker/docs/reference/param-field-dimension). However, you can specify individual time-based dimensions, as explained in the [Individual time and date types](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#individual-timeframes) section.

For information about dimension groups, see the [dimension\_group](https://cloud.google.com/looker/docs/reference/param-field-dimension-group) parameter documentation page, which also includes information on the [timeframes](https://cloud.google.com/looker/docs/reference/param-field-dimension-group#timeframes), [convert\_tz](https://cloud.google.com/looker/docs/reference/param-field-convert-tz), and [datatype](https://cloud.google.com/looker/docs/reference/param-field-dimension-group#specifying_the_database_datatype) parameters, as well as [common challenges](https://cloud.google.com/looker/docs/reference/param-field-dimension-group#things_to_consider) and caveats to consider when you are using time-based data.

unquoted

type: unquoted is used only with [parameter](https://cloud.google.com/looker/docs/reference/param-field-parameter) fields. The unquoted type is similar to [type: string](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#string), except that when the value of the parameter is inserted into the [{% parameter %} liquid variable](https://cloud.google.com/looker/docs/liquid-variable-reference) it will not be quoted. This is useful when inserting values into SQL, such as column or table names, that cannot be quoted in order to work properly.

Inserting unquoted values directly into SQL could create the possibility of unwanted SQL actions. To address this, parameter values of type: unquoted are restricted to the characters A through Z and 0 through 9 (no spaces or other special characters).

As an example, the following LookML creates a parameter called table\_name that will produce an unquoted value:

parameter: table\_name {

type: unquoted

}

yesno

type: yesno creates a field that indicates if something is true or false. The values appear as **Yes** and **No** in the Explore UI.

The [sql](https://cloud.google.com/looker/docs/reference/param-field-sql) parameter for a type: yesno dimension takes a valid SQL expression that evaluates to TRUE or FALSE. If the condition evaluates to TRUE, **Yes** is displayed to the user; otherwise, **No** is displayed.

The SQL expression for type: yesno dimensions cannot include any aggregations. This means it cannot contain SQL aggregations or any references to LookML measures. If you want to create a yesno field that includes a SQL aggregation or that references a LookML measure, use a *measure* with [type: yesno](https://cloud.google.com/looker/docs/reference/param-measure-types#yesno), not a dimension.

For example, the following LookML creates a field that indicates whether or not an order has been paid, based on the status field:

dimension: is\_order\_paid {

type: yesno

sql: ${status} = 'paid' ;;

}

To reference a type: yesno field in another field, treat the type: yesno field as a boolean (in other words, as if it contains a true or false value already). For example:

dimension: is\_big\_order {

type: yesno

sql: ${order\_size} = 'big' ;;

}

# This is correct

measure: total\_boxes\_needed {

type: number

sql: SUM(CASE WHEN ${is\_big\_order} THEN 2 ELSE 1 END) ;;

}

# This is NOT correct

measure: total\_boxes\_needed {

type: number

sql: SUM(CASE WHEN ${is\_big\_order} = 'Yes' THEN 2 ELSE 1 END) ;;

}

If you use type: yesno with time-based data, the dimension returns **yes** if the datetime has a value, and returns **no** if not.

zipcode

type: zipcode is used with zip code dimensions that you want to plot on a [Static Map (Points)](https://cloud.google.com/looker/docs/map-points-options) visualization (use a state or country field for [Static Map (Regions)](https://cloud.google.com/looker/docs/map-regions-options)). Any dimension of type: zipcode is automatically given the [map\_layer\_name](https://cloud.google.com/looker/docs/reference/param-field-map-layer-name) of us\_zipcode\_tabulation\_areas. If you do not want to plot the zip codes, you can use a simpler type such as [type: number](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#number).

The [sql](https://cloud.google.com/looker/docs/reference/param-field-sql) parameter for type: zipcode dimensions can take any valid SQL expression that results in a five-digit, US zipcode.

For the purposes of filtering on a zipcode dimension, some database dialects require that the database field referenced by the zipcode dimension is a varchar or string type field, not an integer type field.

For example:

dimension: zip {

type: zipcode

sql: ${TABLE}.zipcode ;;

}

**Individual time and date types**

Typically, dates are handled as a [dimension\_group](https://cloud.google.com/looker/docs/reference/param-field-dimension-group) that uses [type: time](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#time).

It is possible to create one [dimension](https://cloud.google.com/looker/docs/reference/param-field-dimension) or [filter field](https://cloud.google.com/looker/docs/reference/param-field-filter) for each individual time frame you want to include, instead of generating all of them in a single [dimension\_group](https://cloud.google.com/looker/docs/reference/param-field-dimension_group). This is generally avoided, unless you have already pre-calculated time columns in your database or want to change Looker's timeframe naming convention (such as having a field named created\_date\_of\_purchase instead of created\_date).

Many individual time and date based types are listed [below](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#individual-time-types).

As an example, for this dimension\_group definition:

dimension\_group: created {

type: time

timeframes: [week, month, year]

sql: ${TABLE}.created\_at ;;

}

You can use this as a logic equivalent:

dimension: created\_week {

type: date\_week

sql: ${TABLE}.created\_at ;;

}

dimension: created\_month {

type: date\_month

sql: ${TABLE}.created\_at ;;

}

dimension: created\_year {

type: date\_year

sql: ${TABLE}.created\_at ;;

}

**Available time-based types**

The following types are used in the type parameter of an individual dimension to create time or date based fields. Do *not* use these types with the timeframe parameter, which is documented on the [dimension\_group](https://cloud.google.com/looker/docs/reference/param-field-dimension-group" \l "timeframe_options) documentation page.

All individual time and date types require a timestamp as input from your database.

**Special types**

| **Type** | **Description** | **Example Output** |
| --- | --- | --- |
| date\_raw | The raw value from your database, without casting or time zone conversion, will not show up on Explore page (typically not needed except in joins or time comparisons) | 2014-09-03 17:15:00 +0000 |

**Time types**

| **Type** | **Description** | **Example Output** |
| --- | --- | --- |
| date\_time | Datetime of the underlying field (some SQL dialects show as much precision as your database contains, while others show only to seconds) | 2014-09-03 17:15:00 |
| date\_time\_of\_day | Time of day | 17:15 |
| date\_hour | Datetime truncated to the nearest hour | 2014-09-03 17 |
| date\_hour\_of\_day | Integer hour of day of the underlying field | 17 |
| date\_hourX | Splits each day into intervals with the specified number of hours. Requires explanation, see [below](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#hourx). | *See Below* |
| date\_minute | Datetime truncated to the nearest minute | 2014-09-03 17:15 |
| date\_minuteX | Splits each hour into intervals with the specified number of minutes. Requires explanation, see [below](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#minutex). | *See Below* |
| date\_second | Datetime truncated to the nearest second | 2014-09-03 17:15:00 |
| date\_millisecond | Datetime truncated to the nearest millisecond (see the [Dialect Support for Milliseconds and Microseconds](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#dialect_support_for_milliseconds_and_microseconds) section for information on dialect support) | 2014-09-03 17:15:00.000 |
| date\_millisecondX | Splits each second into intervals with the specified number of milliseconds (see the [Dialect Support for Milliseconds and Microseconds](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#dialect_support_for_milliseconds_and_microseconds) section for information on dialect support). | 2014-09-01 01:00:00.250 |
| date\_microsecond | Datetime truncated to the nearest microsecond (see the [Dialect Support for Milliseconds and Microseconds](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#dialect_support_for_milliseconds_and_microseconds) section for information on dialect support) | 2014-09-03 17:15:00.000000 |

**Date types**

| **Type** | **Description** | **Example Output** |
| --- | --- | --- |
| date | Date of the underlying field | 2017-09-03 |
| date\_date | REMOVED 4.6 Replaced by date |  |

**Week types**

| **Type** | **Description** | **Example Output** |
| --- | --- | --- |
| date\_week | Date of the week starting on a Monday of the underlying datetime | 2017-09-01 |
| date\_day\_of\_week | Day of week alone | Wednesday |
| date\_day\_of\_week\_index | Day of week index (0 = Monday, 6 = Sunday) | 2 |

Please note that the date\_week, date\_day\_of\_week, and date\_day\_of\_week\_index types depend on the value of [week\_start\_day](https://cloud.google.com/looker/docs/reference/param-model-week-start-day), which defaults to Monday.

**Month types**

| **Type** | **Description** | **Example Output** |
| --- | --- | --- |
| date\_month | Year and month of the underlying datetime | 2017-09 |
| date\_month\_num | Integer number of the month of the underlying datetime | 9 |
| date\_month\_name | Name of the month | September |
| date\_day\_of\_month | Day of month | 3 |
| date\_fiscal\_month\_num | Integer number of the month of the underlying datetime | 9 |

To use date\_fiscal\_month\_num type, the [fiscal\_month\_offset](https://cloud.google.com/looker/docs/reference/param-model-fiscal-month-offset) parameter must be set in the model.

**Quarter types**

| **Type** | **Description** | **Example Output** |
| --- | --- | --- |
| date\_quarter | Year and quarter of the underlying datetime | 2017-Q3 |
| date\_quarter\_of\_year | Quarter of the year preceded by a "Q" | Q3 |
| date\_fiscal\_quarter | Fiscal year and quarter of the underlying datetime | 2017-Q3 |
| date\_fiscal\_quarter\_of\_year | Fiscal quarter of the year preceded by a "Q" | Q3 |

To use date\_fiscal\_quarter and date\_fiscal\_quarter\_of\_year types, the [fiscal\_month\_offset](https://cloud.google.com/looker/docs/reference/param-model-fiscal-month-offset) parameter must be set in the model.

**Year types**

| **Type** | **Description** | **Example Output** |
| --- | --- | --- |
| date\_year | Integer year of the underlying date time | 2017 |
| date\_day\_of\_year | Day of year | 143 |
| date\_week\_of\_year | Week of the year as a number | 17 |
| date\_fiscal\_year | Integer fiscal year of the underlying datetime | 2017 |

To use the date\_fiscal\_year type, the [fiscal\_month\_offset](https://cloud.google.com/looker/docs/reference/param-model-fiscal-month-offset) parameter must be set in the model.

**Using**date\_hourX

In date\_hourX the X is replaced with 2, 3, 4, 6, 8, or 12.

This will split up each day into intervals with the specified number of hours. For example date\_hour6 will split each day into 6 hour segments, which will appear like:

* 2014-09-01 00:00:00
* 2014-09-01 06:00:00
* 2014-09-01 12:00:00
* 2014-09-01 18:00:00

For example, a row with a time of 2014-09-01 08:03:17 would have a date\_hour6 of 2014-09-01 06:00:00.

**Using**date\_minuteX

In date\_minuteX the X is replaced with 2, 3, 5, 10, 15, or 30.

This will split up each hour into intervals with the specified number of minutes. For example date\_minute15 will split each hour into 15 minute segments, which will appear like:

* 2014-09-01 01:00:00
* 2014-09-01 01:15:00
* 2014-09-01 01:30:00
* 2014-09-01 01:45:00

To give an example, a row with a time of 2014-09-01 01:17:35 would have a date\_minute15 of 2014-09-01 01:15:00.

**Time zones and**convert\_tz

In general, time computations (differences, durations, and so on) only work correctly when you operate on time values that are all converted to the same time zone, so it is important to keep time zones in mind when writing LookML.

Looker has various [time zone settings](https://cloud.google.com/looker/docs/using-time-zone-settings) that convert time-based data between different time zones. Looker converts time zones by default. If you do not want Looker to perform a time zone conversion for a particular dimension or dimension group, you can use the convert\_tz parameter described on the [convert\_tz](https://cloud.google.com/looker/docs/reference/param-field-convert-tz) parameter documentation page.

**Dialect support for milliseconds and microseconds**

Looker supports precision to microseconds; however, some databases support precision only to seconds. If a database encounters a time type more precise than it can support, it will round up to seconds.

In the latest release of Looker, the following dialects support milliseconds:

In the latest release of Looker, the following dialects support microseconds:

**Individual duration types**

Typically, durations are handled as a [dimension\_group](https://cloud.google.com/looker/docs/reference/param-field-dimension-group) that uses [type: duration](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#duration).

It is possible to create one [dimension](https://cloud.google.com/looker/docs/reference/param-field-dimension) for each individual duration you want to include, instead of generating all of them in a single [dimension\_group](https://cloud.google.com/looker/docs/reference/param-field-dimension_group). This is generally avoided, unless you want to change Looker's timeframe naming convention (such as having a field named **Number of Days to Delivery** instead of **Duration to Delivery**).

Several individual duration types are listed [below](https://cloud.google.com/looker/docs/reference/param-dimension-filter-parameter-types#individual-duration-types).

When you use a duration type for a dimension, you must also include the sql\_start and sql\_end parameters to provide the start and end times for calculating the time difference.

The sql\_start and sql\_end parameters can take any valid SQL expression that contains data in a timestamp, datetime, date, epoch, or yyyymmdd format. The sql\_start and sql\_end fields can be any of the following:

* A reference to a raw timeframe from an existing dimension group of type: time.
* A reference to a dimension of type: date\_raw.
* A SQL expression that is a timestamp, such as a reference to a SQL column that is a timestamp.
* A SQL expression that pulls a time from your database, using the appropriate expression for your dialect.

As an example, for this dimension\_group definition:

dimension\_group: to\_delivery {

type: duration

intervals: [day, hour]

sql\_start: ${created\_raw} ;;

sql\_end: ${delivered\_raw};;

}

You can use these dimension parameters as a logical equivalent:

dimension: number\_of\_days\_to\_delivery {

type: duration\_day

sql\_start: ${created\_raw} ;;

sql\_end: ${delivered\_raw};;

}

dimension: number\_of\_hours\_to\_delivery {

type: duration\_hour

sql\_start: ${created\_raw} ;;

sql\_end: ${delivered\_raw};;

}

In the Explore UI, this would create dimensions called **Number of Days to Delivery** and **Number of Hours to Delivery**.

**Available duration types**

The following types are used in the type parameter of an individual dimension to create duration based fields. Do *not* use these types with the intervals parameter, which is documented on the [dimension\_group](https://cloud.google.com/looker/docs/reference/param-field-dimension-group" \l "interval_options) documentation page.

All individual duration types require a timestamp as input from your database.

| **Type** | **Description** | **Example Output** |
| --- | --- | --- |
| duration\_day | Calculates a time difference in days | 9 days |
| duration\_hour | Calculates a time difference in hours | 171 hours |
| duration\_minute | Calculates a time difference in minutes | 10,305 minutes |
| duration\_month | Calculates a time difference in months | 3 months |
| duration\_quarter | Calculates a time difference in quarters of the year | 2 quarters |
| duration\_second | Calculates a time difference in seconds | 606,770 seconds |
| duration\_week | Calculates a time difference in weeks | 6 weeks |
| duration\_year | Calculates a time difference in years | 2 years |

Was this helpful?