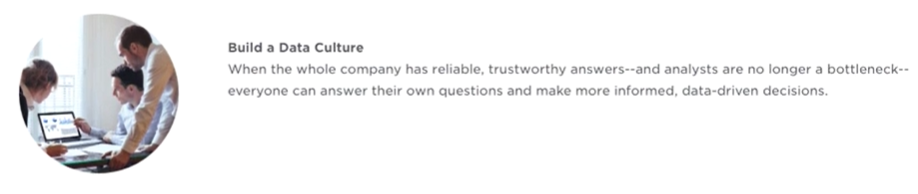
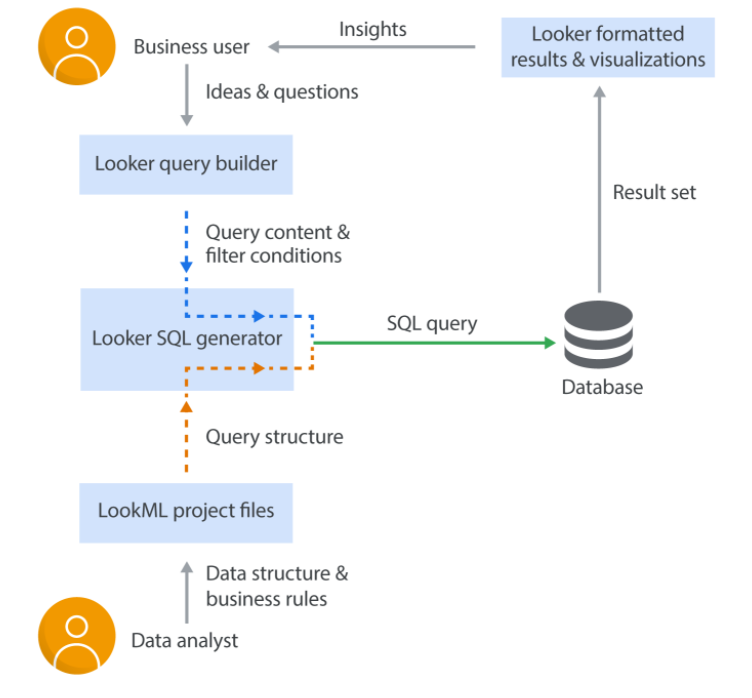
Looker is a complete BI platform, self contained and can act as a data pipeline apart from being a frontend BI tool.



Row limit is to avoid crashing and for optimal user experience

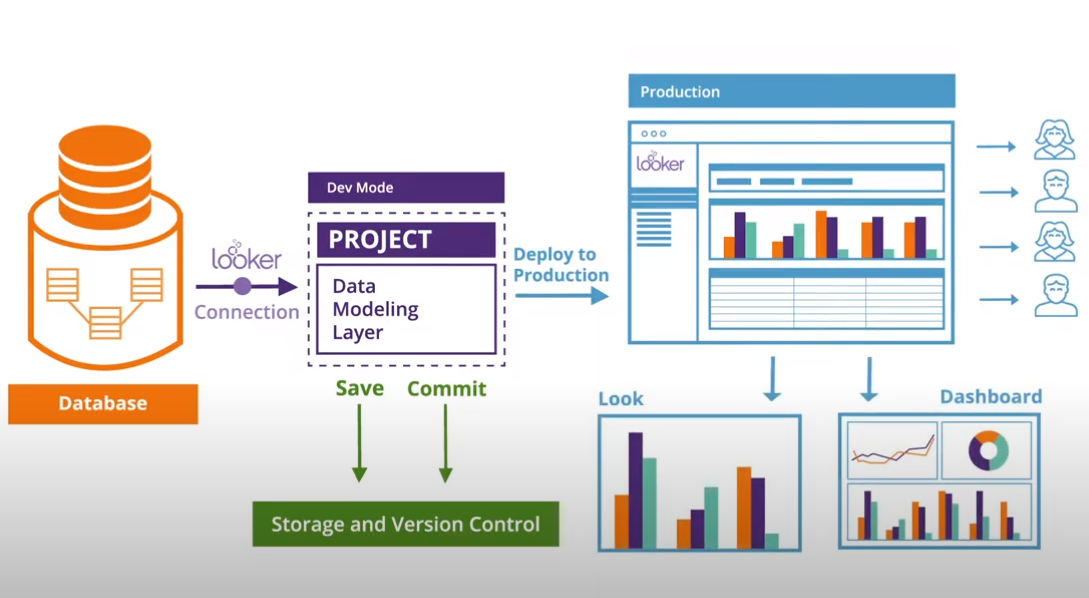


## Looker Project Structure

A Looker/LookML project is a collection of model, view, and dashboard files that are typically version controlled together via a Git repository.

Setting up a project in looker involves following three basic steps :

* Creating a Database connection to Access the data
* Creating & Configuring a Project to hold a data model & setting up version control to manage & setup changes to Data Model
* Creating Explores, Looks & Dashboards.



## Connection

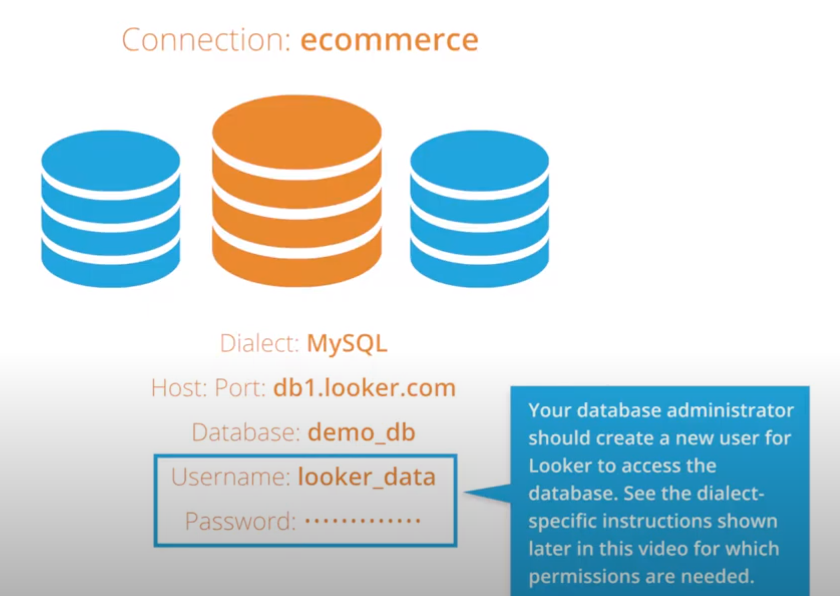
Prerequisites:

[Enabling secure database access](https://docs.looker.com/setup-and-management/enabling-secure-db)

[Database configuration instructions](https://docs.looker.com/setup-and-management/database-config)

Instructions:

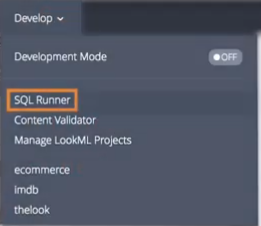
[Connecting Looker to your database](https://docs.looker.com/setup-and-management/connecting-to-db)



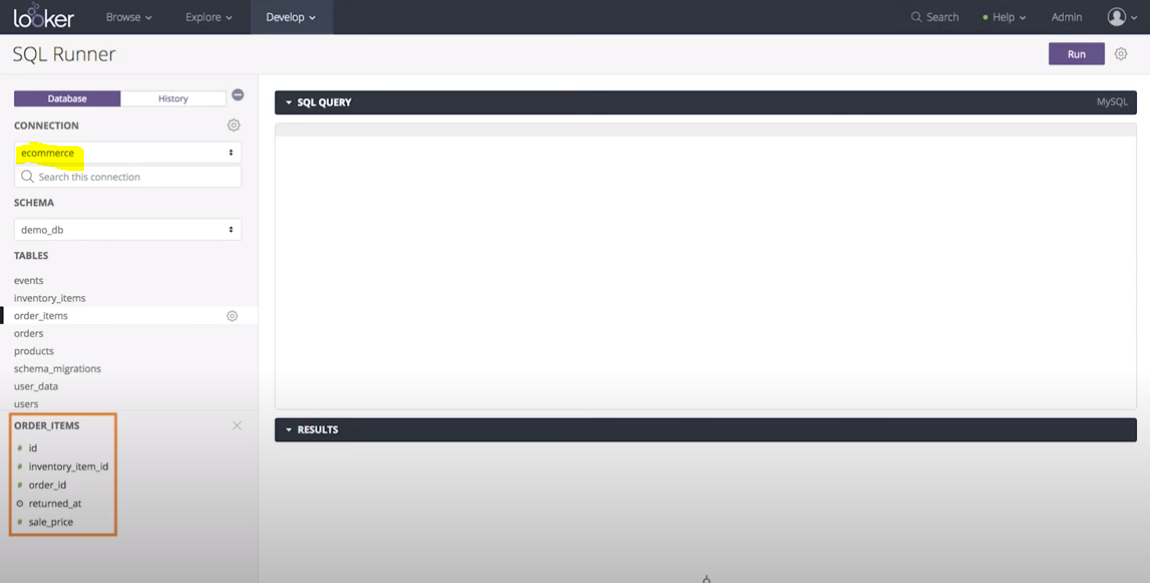
Looker also provides a SQL Runner platform where you can write a query and test the data on the connection that you have made

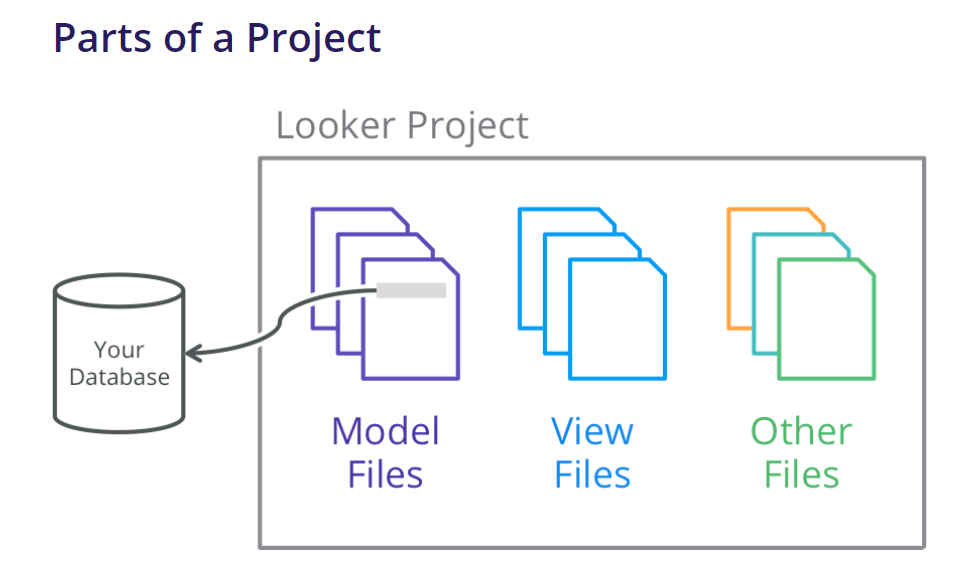
[SQL Runner basics](https://docs.looker.com/data-modeling/learning-lookml/sql-runner) — Learn about SQL Runner and how to use it.

1. Go to Develop menu and click on SQL Runner



1. Choose your Connection, Schema, Tables and start typing a query





A project can contain below type of files (It can have user defined folders too for organizing files):-

* [**Model files**](https://docs.looker.com/data-modeling/getting-started/model-development#model_files) **with extension .model.lkml**
* [**View files**](https://docs.looker.com/data-modeling/getting-started/model-development#view_files) **with extension .view.lkml**
* [Dashboard files](https://docs.looker.com/data-modeling/getting-started/other-project-files#dashboard_files) with extension .dashboard.lookml
* [Data files](https://docs.looker.com/data-modeling/getting-started/other-project-files#data_files) with extension .topojson or .geojson or .json
* [Document files](https://docs.looker.com/data-modeling/getting-started/other-project-files#document_files) with extension .md
* [Project manifest files](https://docs.looker.com/data-modeling/getting-started/other-project-files#project_manifest_files) that are always named manifest.lkml
* [Locale strings files](https://docs.looker.com/data-modeling/getting-started/other-project-files#locale_strings_files) with extension .strings.json
* [Explore files](https://docs.looker.com/data-modeling/getting-started/other-project-files#explore_files) with extension .explore.lkml
* [Data test files](https://docs.looker.com/data-modeling/getting-started/other-project-files#data_test_files) with extension .lkml
* [Refinements files](https://docs.looker.com/data-modeling/getting-started/other-project-files#refinements_files) with extension .lkml
* [Other files](https://docs.looker.com/data-modeling/getting-started/other-project-files#other_files) with any file extension not listed above.

## Development Mode and Production Mode

Your Looker data model exists in two states: Production Mode and Development Mode.

<https://docs.looker.com/data-modeling/getting-started/dev-mode-prod-mode>

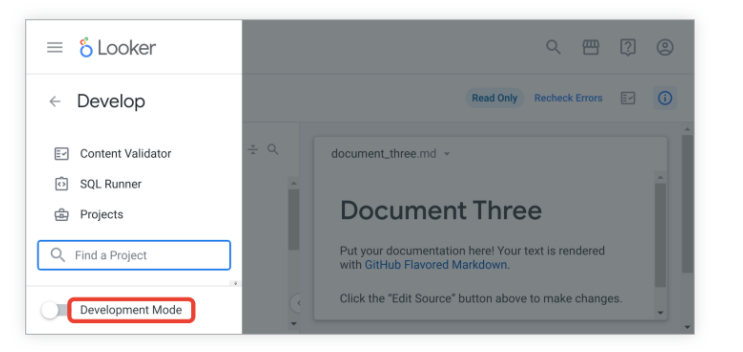
### Production Mode

Production Mode is the production version of Looker. Everyone using a Looker instance in Production Mode accesses their projects in the same state. Project files are read-only in this mode.

### DevelopmentMode

Development Mode lets you make changes to LookML files and to preview how they will affect content on your instance. The changes you make to LookML files in Development Mode do not affect the production environment, until they are pushed to the production environment. (If you are familiar with Git, Development Mode uses a separate branch.)

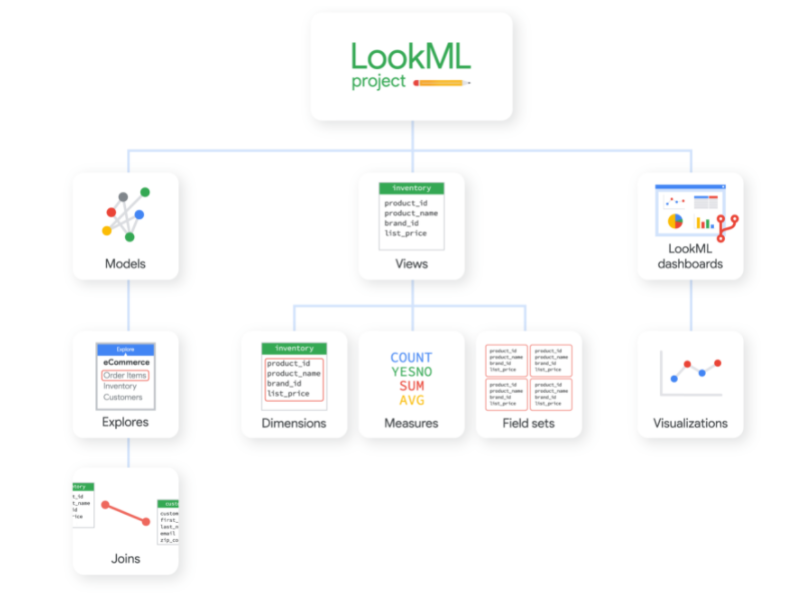
In Development Mode, you can see the effects of your changes to project files in the Explore area of Looker. Once you’re happy with your changes, you can save and merge them into production, where they will then be viewable by everyone.

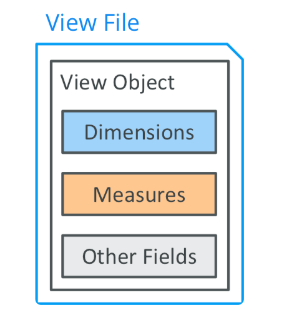


## LookML

* LookML is a language for describing dimensions, aggregates, calculations, and data relationships in a SQL database. Looker uses a model written in LookML to construct SQL queries against a particular database.
* LookML syntax has a structure that is clear and easy to learn.
* LookML is independent of particular SQL dialects, and encapsulates SQL expressions to support any SQL implementation.
* LookML has intellisense which makes writing code very simple.
* LookML separates structure from content.

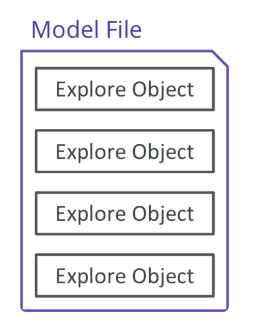
### LookML Components



A [**view**](https://docs.looker.com/data-modeling/getting-started/model-development#view_files) corresponds to a database table or a [derived table](https://docs.looker.com/data-modeling/learning-lookml/derived-tables). Here a developer will define a list of fields from those tables that should appear in the UI for users to build their queries from.

**Fields** - In general, each bit of data that your users are going to choose from will be represented as a LookML field. There are five different types of fields:

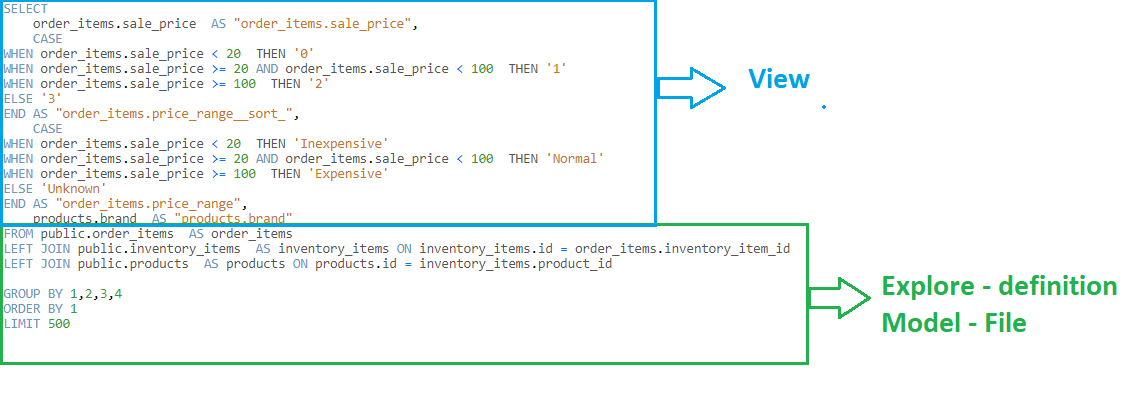
* **Dimensions** represent a column in a table, or a computed value based on some sort of column manipulation or combination
* **Dimension Groups** are only used with time-based data, and enable you to create many time-based dimensions at one time
* **Measures** are similar to aggregate functions in SQL (e.g. COUNT, SUM, AVG) and represent information about multiple rows
* **Filters** create a filter-only field users can use to provide input to a [templated filter](https://docs.looker.com/data-modeling/learning-lookml/templated-filters) or a [conditional join](https://docs.looker.com/reference/explore-params/sql_on#conditional_joins)
* **Parameters** create a filter-only field users can use to provide input to a [Liquid {% parameter %} tag](https://docs.looker.com/reference/liquid-variables)

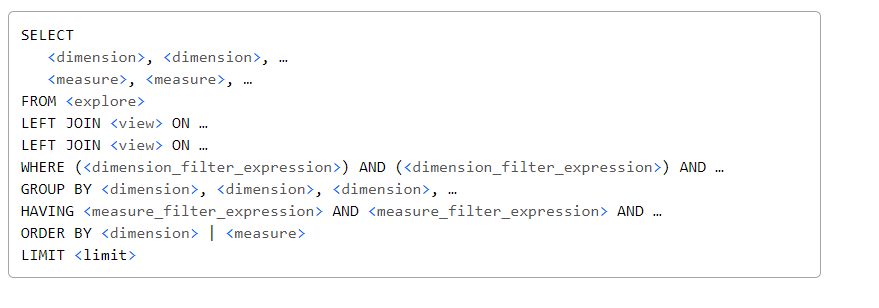


A [**model**](https://docs.looker.com/data-modeling/getting-started/model-development#model_files) specifies a connection to a single database. This is also where a developer will define the model’s **Explores**, as explained below.

In Looker, **Explore** refers to two closely related things. [explore](https://docs.looker.com/reference/explore-reference) (which you’ll see in code font throughout our documentation) is a LookML parameter used within a model file to define query options. In SQL terms, an Explore is the FROM clause of a query.

An [Explore](https://docs.looker.com/exploring-data/exploring-data) (capitalized throughout our documentation) is also an interactive page in Looker where users can build SQL queries by selecting fields, applying filters, and choosing from other options that have been established in the project’s files.





**Model File ( Connection) 🡪 Model File (Explore / From clause) 🡪 Views File ( base and derived columns)🡪 Views Fields**

### LookML Syntax

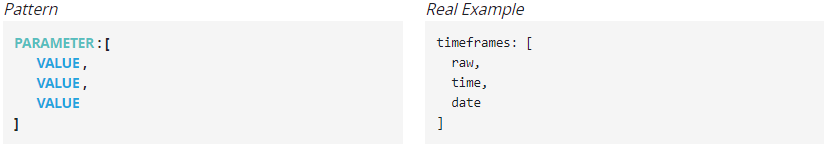
LookML is a declarative language, made up of parameters and values that define data components.

Parameter : Value # Syntax for defining any parameter

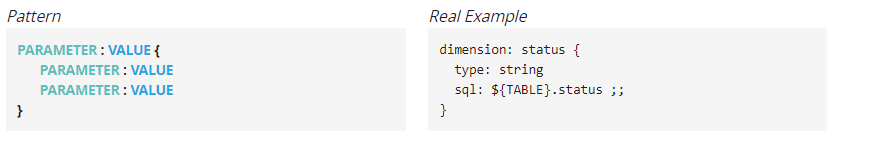
connection: "thelook\_events"

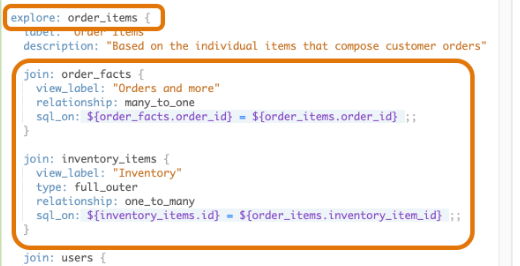
**Rule :- Colon punctuation to assign values to parameters, like a=4 🡪 a:4**

Parameters can also take in multiple values and other parameters as well



**Rule :- Square brackets for defining multiple values in parameter**

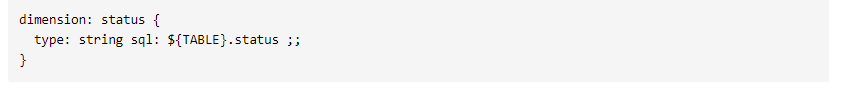




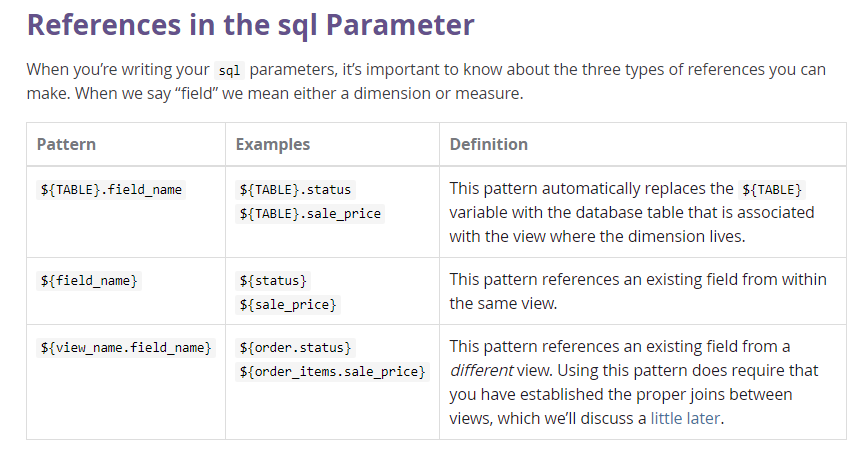
**Rule:- Curly braces for defining any parameter within parameter**

**Rule :- Only values that contain HTML or SQL end in**;;

* Comments can be added as # (it takes the whole row after # as comment)
* Having extra spaces or indentation doesn’t affect the code, its only for readability.
* Make sure every parameter occupies a single line. For ex, below code is wrong:-



**Rule :- $ is a substitution operator in Looker,**



### Views

A view represents a table of data in Looker, whether that table is native to your database or was created using Looker’s [derived table](https://docs.looker.com/data-modeling/learning-lookml/derived-tables) functionality. Views are typically declared in [view files](https://docs.looker.com/data-modeling/getting-started/model-development#view_files), with one view per file.

## View Usually Contains Dimensions and Measures and Filters, field sets

Syntax: view: view\_name { … }

* A view Is Not Accessible by Users until Added Using explore or join
* Two views Must Be Referred to by Different Names within an explore
* Tables Referenced by view Must Be Accessible from the Current Connection
* Only one view per view file should be declared as best practice.

View Reference: <https://docs.looker.com/reference/view-reference>

Field Reference: <https://docs.looker.com/reference/field-reference>

Dimension Reference: <https://docs.looker.com/reference/field-params/dimension>

Measure Reference: <https://docs.looker.com/reference/field-params/measure>

Dimension Group Reference: <https://docs.looker.com/reference/field-params/dimension_group>

Parameter Reference: <https://docs.looker.com/reference/field-params/parameter>

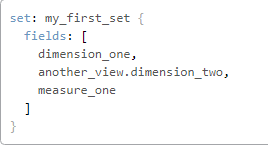
Filter Reference: <https://docs.looker.com/reference/field-params/filter>



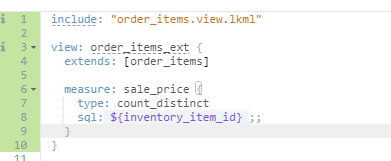
Derived table can be native or SQL Runner based table



#### View Parameters

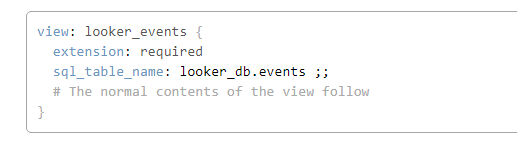
set:  A set can contain any number of dimensions, measures, or filter fields from the current view, including the individual dimensions generated by a dimension group.

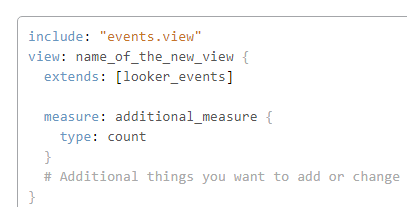
include: The include parameter specifies the LookML files that will be available to a model, a view, or an Explore. If you want to use or reference a LookML file within another file, you must add it with the include parameter.



extends: The extends parameter allows you to build upon the content and settings from another view file, using the other view file as a starting point.

[extension](https://docs.looker.com/reference/view-params/extension-for-view): Specify that the view requires extension and cannot itself be exposed to users.

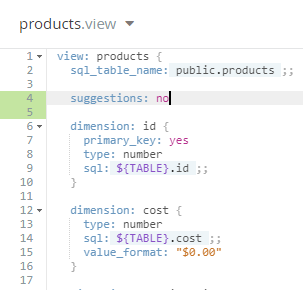




label: Specify how the view name will appear in the explore.



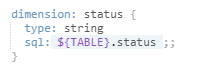
[suggestions](https://docs.looker.com/reference/view-params/suggestions-for-view): suggestions lets you disable suggestions for *all* the type: string dimensions of a view when someone uses *one* of those dimensions to filter a report. The default value of suggestions is yes

.

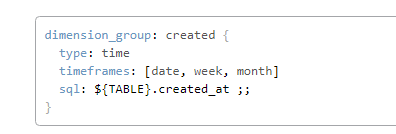
#### Field Parameters

##### dimension: The dimension parameter declares a new dimension and specifies a name for that dimension.

**Dimension Type Reference**: <https://docs.looker.com/reference/field-reference/dimension-type-reference>

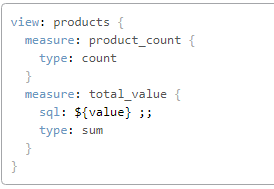


##### dimension\_group: The dimension\_group parameter is used to create a set of time-based dimensions all at once.



##### measure: The measure parameter declares a new measure (aggregation) and specifies a name for that measure.

**Measure Type Reference**<https://docs.looker.com/reference/field-reference/measure-type-reference>



*Note: It is possible to define a measure that is based on another measure. The new measure must be of type: number to avoid nested-aggregation errors*

In order to have measures (aggregations) come through joins, you must define primary keys in all of the views involved in the join.

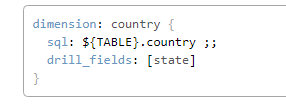
Looker allows you to define measures within other measures.

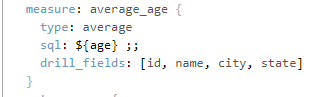
LookML measures are symmetric, no matter what. As long as you define your joins correctly, errors in calculations are eliminated. (<https://discourse.looker.com/t/symmetric-aggregates/261>)

* **Aggregate measures**: Aggregate measure types perform aggregations, such as sum and average. Aggregate measures can reference only dimensions, not other measures. This is the only measure type that works with the [filters](https://docs.looker.com/reference/field-params/filters) parameter.
* **Non-aggregate measures**: Non-aggregate measures are, as the name suggests, measure types that do not perform aggregations, such as number and yesno. These measure types perform simple transformations, and since they do not perform aggregations, can reference only aggregate measures or previously-aggregated dimensions.
* **Post-SQL measures**: Post-SQL measures are special measure types that perform specific calculations after Looker has generated query SQL. They can reference only numeric measures or numeric dimensions. **You cannot use these within other measures. Also, sort order will impact these calculations.**
* **List measure**: list is a specific measure type that does not perform aggregation; rather, it creates a list of distinct values in a dimension.

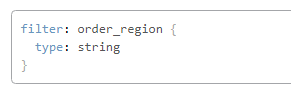
##### drill\_fields: The drill\_fields parameter controls what happens when a user clicks on the value of a table cell while exploring data. When a user does this, they “drill” into the data, allowing them to see the individual records that make up that cell.

The drill\_fields parameter accepts a list of fields, [sets](https://docs.looker.com/reference/view-params/set), or a combination of both fields and sets.



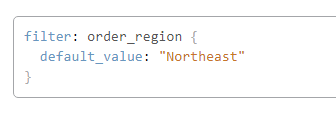


Filter: The filter parameter declares a filter-only field and a name for that filter. A user can add filter-only fields as filters when exploring, but they cannot add them to their result set.





[default\_value](https://docs.looker.com/reference/field-params/default_value): The default\_value parameter is used to specify a default value for [filter fields](https://docs.looker.com/reference/field-params/filter).



link: The link parameter used to add Looker links and other external links to your dimensions and measures.



[description](https://docs.looker.com/reference/field-params/description): Add a description to the field users can see on hover.



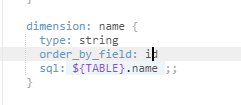
view\_label: This enables the user to display the field in different group other than defined group in model.

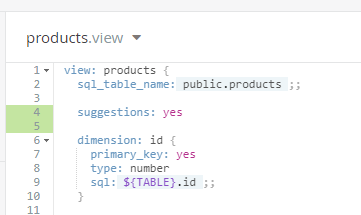
group\_label: The group\_label parameter enables you to combine fields into sub groups, within a given view.

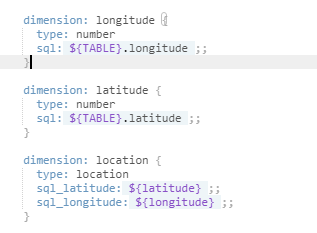
hidden: To hide the field from explore

label: Change the way a field name appears in the explore

[order\_by\_field](https://docs.looker.com/reference/field-params/order_by_field): Sort a field by the values of another field



[primary\_key](https://docs.looker.com/reference/field-params/primary_key): Declare a dimension as the primary key of a view

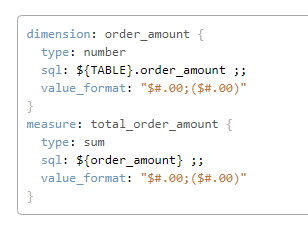
[sql\_latitude](https://docs.looker.com/reference/field-reference/dimension-type-reference#location) & [sql\_longitude](https://docs.looker.com/reference/field-reference/dimension-type-reference#location) : Define the latitude of a [type: location](https://docs.looker.com/reference/field-reference/dimension-type-reference#location) dimension.

value\_format: Format the output of a field using Excel style options.

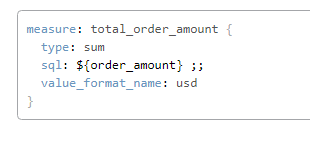
Ex:"$#.00;($#.00)"

"0.00\%" # Display as percent with 2 decimals (1 becomes 1.00%)

"0.000,,\" M\"" # Number in millions with 3 decimals (1.234 M)

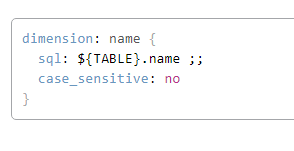
"0.000,\" K\"" # Number in thousands with 3 decimals (1.234 K)

value\_format\_name: Format the output of a field using a built-in or custom format



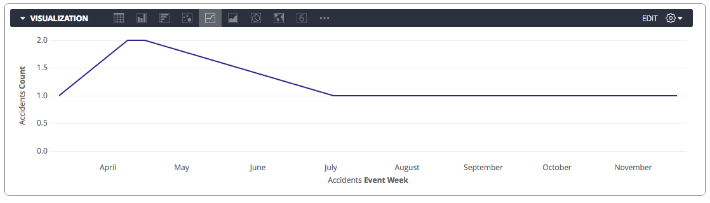
can\_filter: The can\_filter parameter enables you to prohibit a dimension or measure from being used as a filter

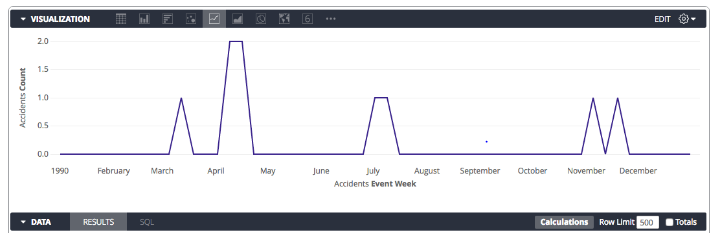
case\_sensitive: When dimension or filter field is used as a filter, you can change its case sensitivity by using the case\_sensitive parameter.

****

**allow\_fill:**

Some datasets have values, such as dates, that follow a predictable pattern. A user might pull data by a time frame and find that some dates, weeks, months, or other date types don’t have any corresponding value. By default, the data table and the visualization will display dates returned from the query and skip any dates that are missing.

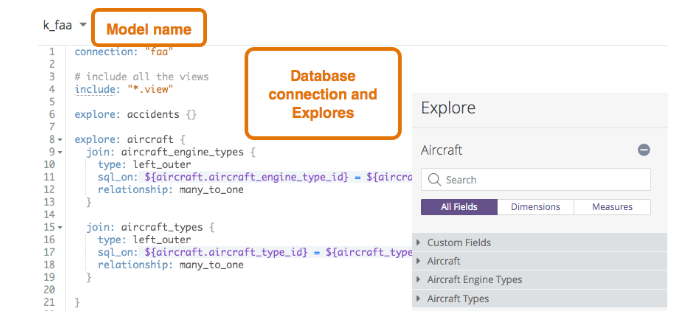
****

****

### 

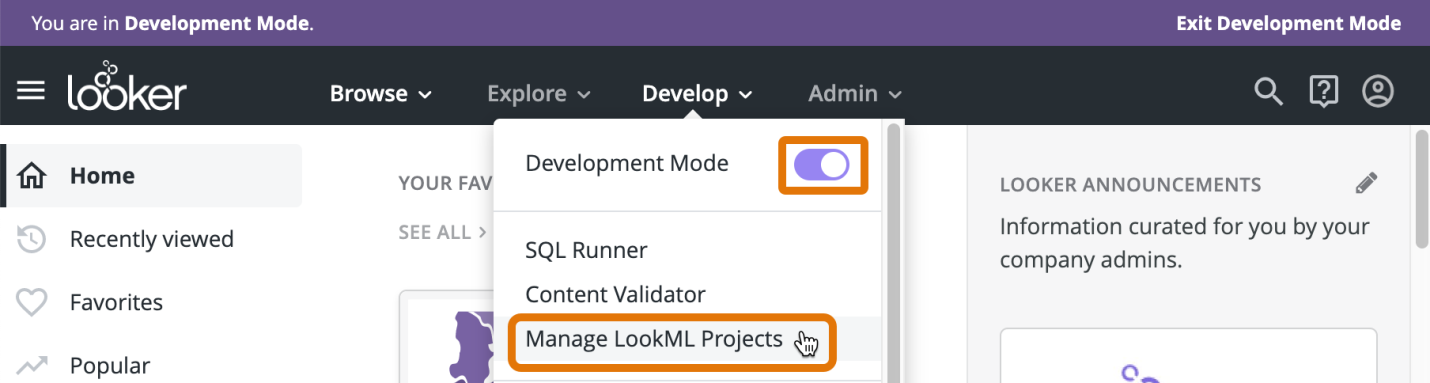
### 

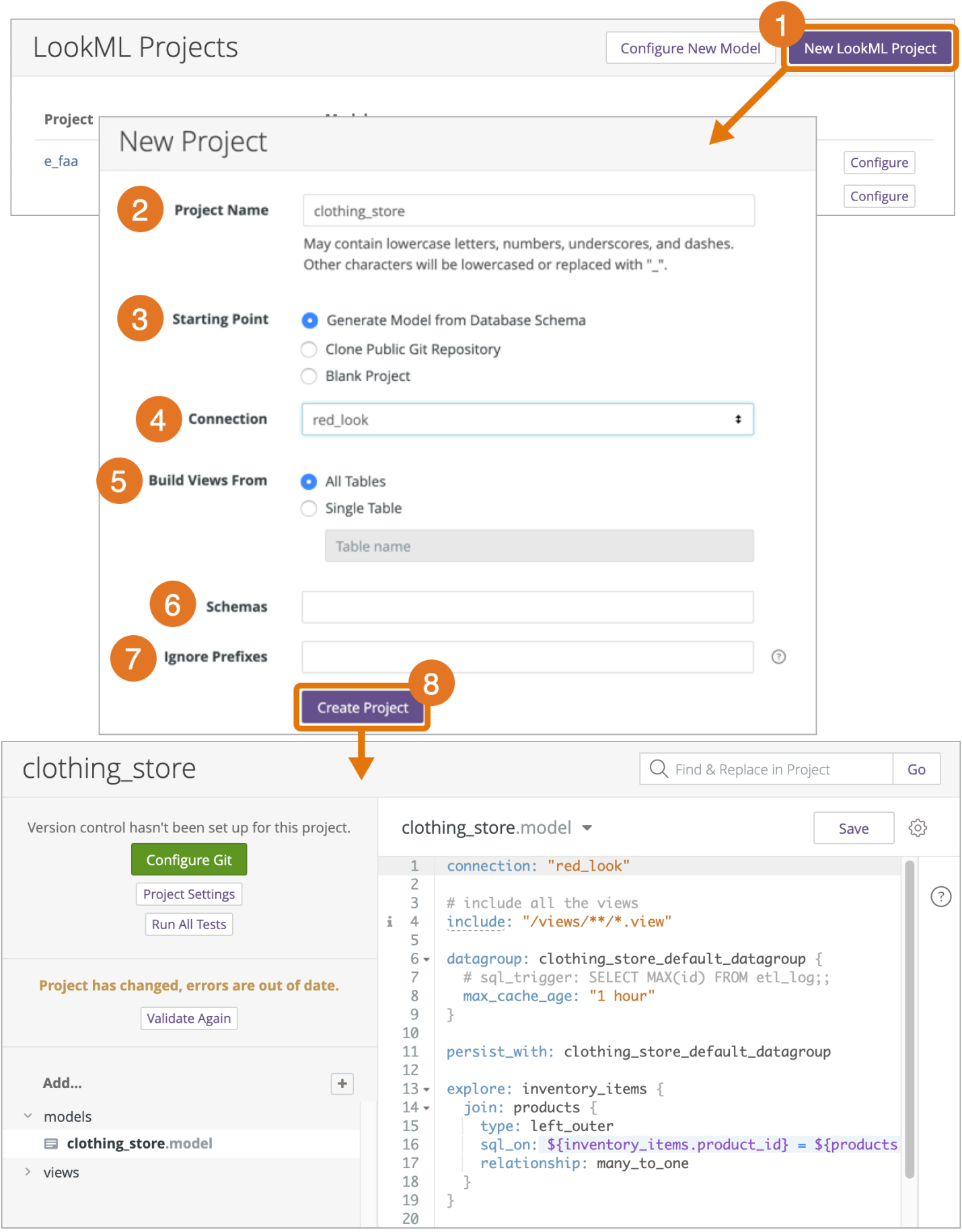
### Models

A model file specifies a database connection, included views and the set of Explores that use that connection. A model file also defines the Explores themselves and their relationships to other views. 

#### Generating a Model

1. Click on Develop in the menu bar and verify that you are in [Development Mode](https://docs.looker.com/data-modeling/getting-started/dev-mode-prod-mode). Then select Manage LookML Projects from the Develop menu and Click on New LookML Project.

****

1. LookML Projects page, you can generate your model and view project files based on the tables in your database. Here’s how: 

#### ModelParameters

**Model Reference**: <https://docs.looker.com/reference/model-reference>

[connection](https://docs.looker.com/reference/model-params/connection-for-model) : Provides the database connection details for a model.

*connection: "connection\_name"*

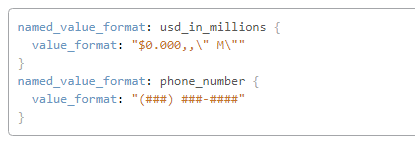
## *connection Must Reference a Connection Name from Looker’s Admin Settings*

[label](https://docs.looker.com/reference/model-params/label-for-model): Changes the way a model appears in the Explore menu.

a[ccess\_grant](https://docs.looker.com/reference/model-params/access_grant): Creates an access grant that limits access of LookML structures to only those users who are assigned an approved [user attribute](https://docs.looker.com/admin-options/settings/user-attributes) value. This parameter has the user\_attribute and allowed\_values subparameters.

[include](https://docs.looker.com/reference/model-params/include): Adds files to a model.

[explore](https://docs.looker.com/reference/explore-params/explore): explore adds an existing view to Looker’s menu of Explores. An Explore should be defined inside of a model file. explore plays an important role in the SQL that Looker generates.

[named\_value\_format](https://docs.looker.com/reference/model-params/named_value_format): The named\_value\_format parameter enables you to create and save a custom format that can be applied to multiple dimensions and measures. 

[fiscal\_month\_offset](https://docs.looker.com/reference/model-params/fiscal_month_offset): If you don’t use the calendar year for your fiscal year, you can specify when your fiscal year begins for each model. Use fiscal\_month\_offset to declare the number of months your fiscal year is offset from the calendar year. 

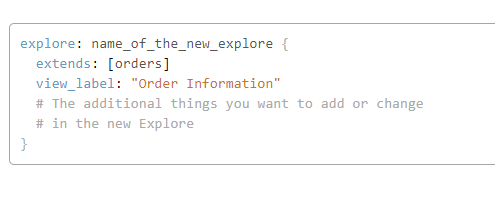
week\_start\_day: monday

#### ExploreParameters

Explore reference: <https://docs.looker.com/reference/explore-reference>

[extends](https://docs.looker.com/reference/explore-params/extends): Specify Explore(s) that will be extended by this Explore

syntax: extends: [explore\_name, explore\_name, …]

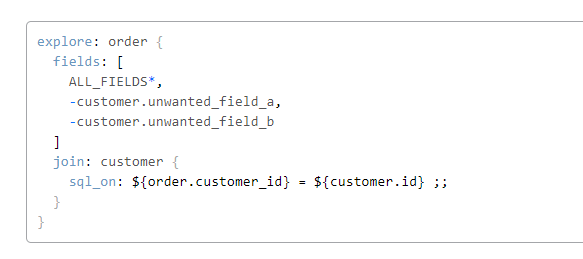


[extension](https://docs.looker.com/reference/explore-params/extension): Specify that the Explore requires extension and cannot itself be exposed to users

extension: required

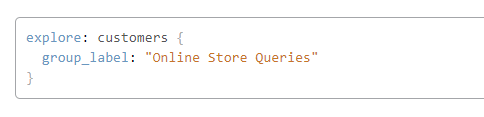
[fields](https://docs.looker.com/reference/explore-params/fields-for-explore):

fields enables you to specify which fields from an Explore are exposed in the Explore UI.



[description](https://docs.looker.com/reference/explore-params/description-for-explore): Add a description for an Explore that appears to users in the UI

[group\_label](https://docs.looker.com/reference/explore-params/group_label): Use group\_label to change the default heading of the Explore menu.



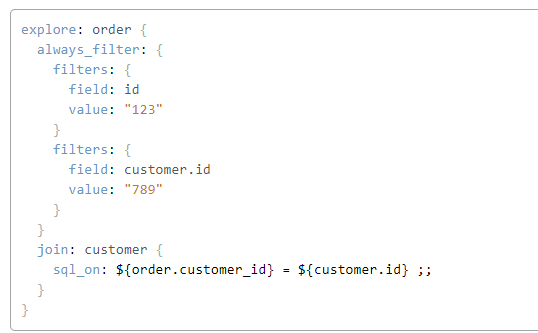
[hidden](https://docs.looker.com/reference/explore-params/hidden-for-explore): Hide an Explore from the Explore menu

[label](https://docs.looker.com/reference/explore-params/label-for-explore):

[view\_label](https://docs.looker.com/reference/explore-params/view_label-for-explore): Specify how a group of fields from the Explore’s base view will be labeled in the field picker

[always\_filter](https://docs.looker.com/reference/explore-params/always_filter):

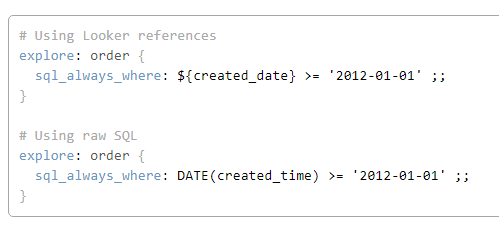
always\_filter enables you to require users to include a certain set of filters that you define.



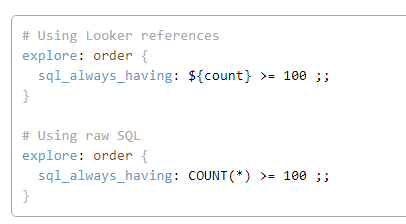
[case\_sensitive](https://docs.looker.com/reference/explore-params/case_sensitive-for-explore):

Specify whether filters are case sensitive for an Explore

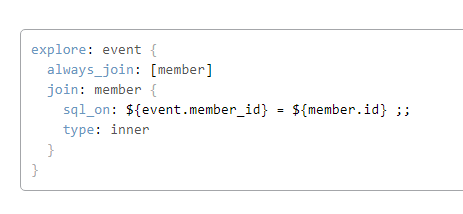
[sql\_always\_where](https://docs.looker.com/reference/explore-params/sql_always_where): Insert conditions into the query’s WHERE clause that a user cannot change or remove for this Explore



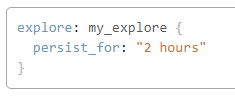
[sql\_always\_having](https://docs.looker.com/reference/explore-params/sql_always_having): Insert conditions into the query’s HAVING clause that a user cannot change or remove for this Explore



[always\_join](https://docs.looker.com/reference/explore-params/always_join): Specify which joins must always be applied to an Explore



[persist\_for](https://docs.looker.com/reference/explore-params/persist_for-for-explore): persist\_for enables you to modify the amount of time that cached query results are used for a given Explore. The default cache length in Looker is 1 hour. Cache results are stored in an encrypted file on your Looker instance.



**Datagroups in Looker:**

datagroup: orders\_dg {label: “Order Item datagroup”description: “Trigger when new Order ID added into system”max\_cache\_age: “24 hours”sql\_trigger: SELECT max(id) FROM orders ;;}

explore: order\_items {  
***persist\_with: orders\_dg***  
join: orders {  
sql\_on: ${orders.id} = ${order\_items.order\_id} ;;  
type: left\_outer  
relationship: many\_to\_one  
}

}

#### Join Parameters

Join Reference: <https://docs.looker.com/reference/explore-params/join>

[view\_label](https://docs.looker.com/reference/explore-params/view_label-for-join): Change the way the join’s view name appears in the Field Picker

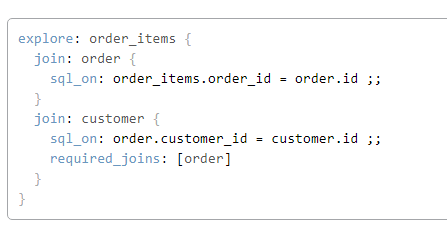
[fields (for join)](https://docs.looker.com/reference/explore-params/fields-for-join): Determine which fields from a join are brought into an Explore

[from (for join)](https://docs.looker.com/reference/explore-params/from-for-join): Specify the view on which a join will be based

[relationship](https://docs.looker.com/reference/explore-params/relationship): Declare a join as having a one-to-one, many-to-one, one-to-many, or many-to-many relationship.

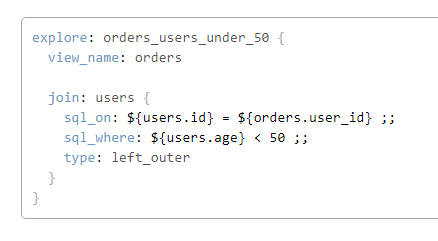
[type (for join)](https://docs.looker.com/reference/explore-params/type-for-join): Declare a join as being a left, full, inner, or cross type

[required\_joins](https://docs.looker.com/reference/explore-params/required_joins): required\_joins forces one or more [joins](https://docs.looker.com/reference/explore-params/join) to be included in the SQL that Looker generates, even if the user has not selected a field from that joined view.



[sql\_on](https://docs.looker.com/reference/explore-params/sql_on): Specify a relationship between an Explore and a join by writing a SQL ON clause

[sql\_where](https://docs.looker.com/reference/explore-params/sql_where): sql\_where enables you to apply a query restriction that users cannot change. The restriction will be inserted into the WHERE clause of the underlying SQL



## Other References

Filter regex reference

<https://docs.looker.com/reference/filter-expressions>

<https://docs.looker.com/data-modeling/learning-lookml/templated-filters>

Liquid language reference

<https://docs.looker.com/reference/liquid-variables>

<https://shopify.github.io/liquid/filters/where/>

<https://www.shopify.com/partners/shopify-cheat-sheet>

Custom Drill Down Reference

<https://discourse.looker.com/t/html-to-link-to-custom-explore/8320>

<https://discourse.looker.com/t/creating-custom-vis-via-html/3735>

<https://info.looker.com/youtube-learning-looker-tutorials-and-product-features/look-learn-drilling-into-your-data>

<https://docs.looker.com/reference/field-params/drill_fields>

<https://help.looker.com/hc/en-us/articles/360023589613--More-Powerful-Data-Drilling>

<https://help.looker.com/hc/en-us/articles/360001288228-Custom-Drilling-Using-HTML-and-Link>