Section 1: Model management

* [How a project works in Looker](https://docs.looker.com/data-modeling/getting-started/how-project-works)

In Looker, a project is a collection of files that describe the objects, database connections, and user interface elements that will be used to carry out [SQL queries](https://cloud.google.com/looker/docs/how-looker-generates-sql).

Parts of lookml project:

Project

Model view lookml dashbords

Explorer

joins dimension measure field set visualization

The most common way to create LookML files is to [generate a LookML project from your database](https://cloud.google.com/looker/docs/generating-a-model#automatically_generating_a_model_from_your_database). You can also [create a blank project](https://cloud.google.com/looker/docs/create-projects#creating_a_blank_project) and manually create its LookML files, or you can [create a project by cloning an existing Git repository](https://cloud.google.com/looker/docs/create-projects#cloning_a_public_git_repository).

* [Development Mode and Production Mode](https://docs.looker.com/data-modeling/getting-started/dev-mode-prod-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.

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.

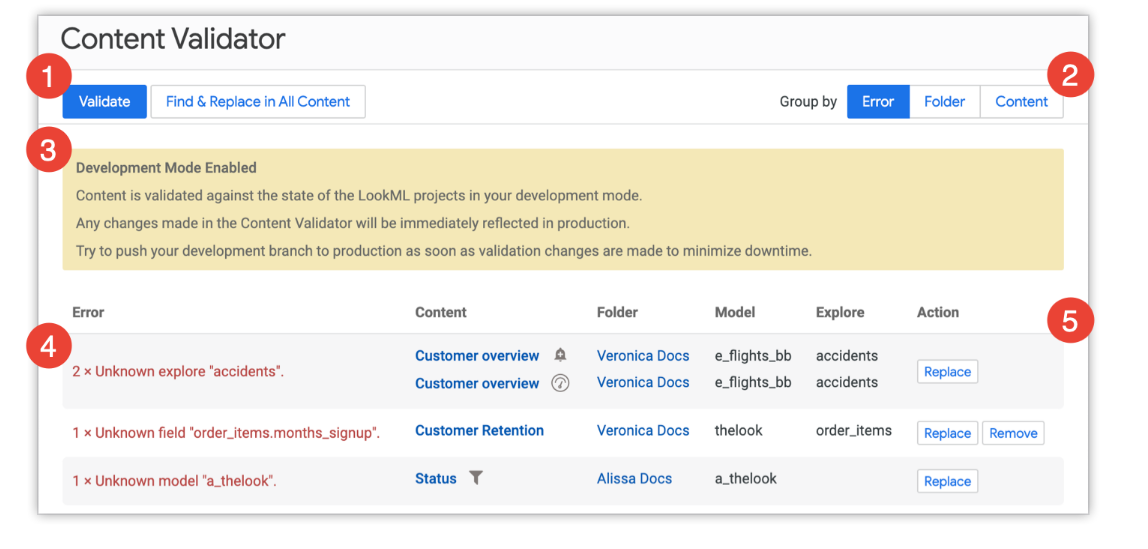
* [LookML project files](https://docs.looker.com/data-modeling/getting-started/file-types-in-project)

A Looker [project](https://cloud.google.com/looker/docs/lookml-terms-and-concepts#lookml_project) is a collection of [LookML](https://cloud.google.com/looker/docs/what-is-lookml) files that tell Looker how to connect to your database, how to query your data, and how to control the user interface's behavior.

Select the following links to get more information about each of the types of files that can be used in a LookML project:

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

The Content Validator searches your LookML for model, Explore, and field names referenced in your Looker content. It's a super useful tool for validating your LookML references, for fixing errors in your LookML after making changes, and for updating the names of your LookML elements across your project.



* [Access\_grants](https://docs.looker.com/reference/model-params/access_grant)

An access grant is a LookML structure defined in a model file that controls access to other LookML structures, specifically Explores, joins, views, and fields. The access\_grant parameter defines an access grant.

For example, the LookML below creates an access grant called can\_view\_financial\_data, which is based on the department user attribute. Only those users who are assigned the values "finance" or "executive" in the department user attribute are given access to the can\_view\_financial\_data access grant:

access\_grant: can\_view\_financial\_data {

user\_attribute: department

allowed\_values: [ "finance", "executive" ]

}

* [Required\_access\_grants](https://docs.looker.com/reference/field-params/required_access_grants-for-field)

required\_access\_grants pairs with the model-level [access\_grant](https://cloud.google.com/looker/docs/reference/param-model-access-grant) parameter to limit access for a field to only those users who have a specific [user attribute](https://cloud.google.com/looker/docs/admin-panel-users-user-attributes) value assigned to them.

For example, the following LookML requires that users have access to both the can\_view\_financial\_data and the view\_payroll access grants to see the salary dimension:

dimension: salary {

...

required\_access\_grants: [can\_view\_financial\_data, view\_payroll]

}

* [Access\_filter](https://docs.looker.com/reference/explore-params/access_filter)

access\_filter lets you apply user-specific data restrictions. Unlike most LookML parameters, it needs to be used in conjunction with other settings in Looker in order to work properly. An access\_filter parameter is specific to a single [Explore](https://cloud.google.com/looker/docs/reference/param-explore-explore), so you need to make sure you apply an access\_filter parameter to each Explore that needs a restriction.

explore: customer {

access\_filter: {

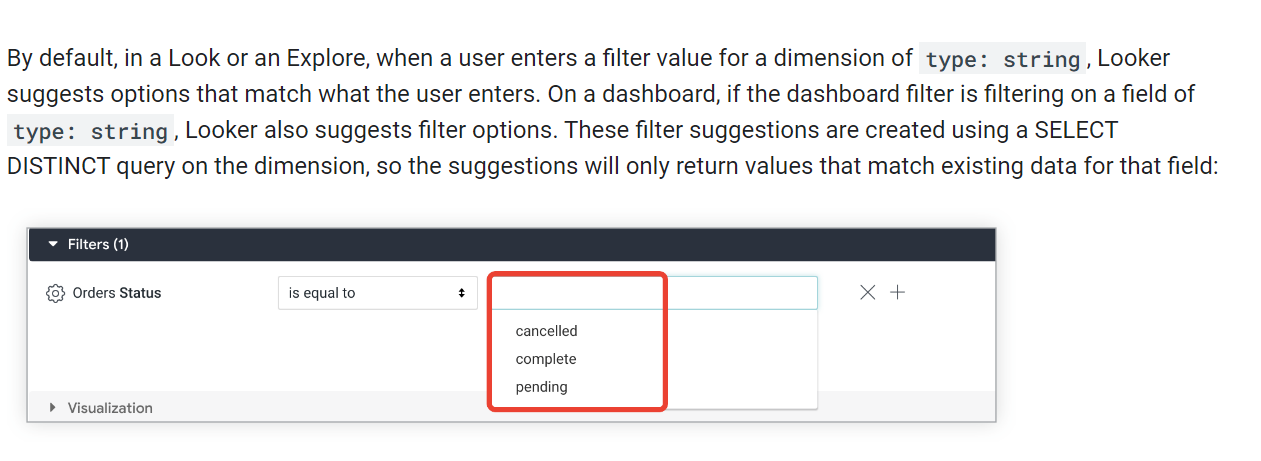
field: customer.name

user\_attribute: allowed\_customers

}

}

[Changing filter suggestions](https://docs.looker.com/data-modeling/learning-lookml/filter-suggestions)



### **Example:**

### **allowed\_value** (field)

[allowed\_value](https://cloud.google.com/looker/docs/reference/param-field-parameter#specifying_allowed_values) works with the parameter LookML parameter. If a Looker developer provides a [parameter filter-only field](https://cloud.google.com/looker/docs/templated-filters#basic_usage), the values entered or selected by users can be referenced using a [{% parameter %} Liquid variable](https://cloud.google.com/looker/docs/liquid-variable-reference). Optionally, you can use allowed\_value to define a list of values a user can choose for that parameter field.

parameter: order\_amount {  
  type: string  
  allowed\_value: {  
    label: "Less than $50"  
    value:"< 50"  
  }  
  allowed\_value: {  
    label: "Between $50 and $100"  
    value:"<= 100"  
  }  
  allowed\_value: {  
    label: "Over $100"  
    value:"> 100"  
  }  
}

**Suggestions:**

This parameter lets you hard-code the list of suggested values for a [filter](https://cloud.google.com/looker/docs/reference/param-field-filter) field or a [dimension](https://cloud.google.com/looker/docs/reference/param-field-dimension) of type: string that will appear when someone uses that field to filter a query.

dimension: colors {

type: string

sql: ${TABLE}.colors ;;

suggestions: ["red", "yellow", "blue"]

}

#### Section 2: Customization

* [What is LookML?](https://docs.looker.com/data-modeling/learning-lookml/what-is-lookml)

LookML stands for Looker Modeling Language; it's the language that is used in Looker to create semantic data models. You can use LookML to describe dimensions, aggregates, calculations, and data relationships in your SQL database. Looker uses a model that is written in LookML to construct SQL queries against a particular database.

* [LookML terms and concepts](https://docs.looker.com/data-modeling/learning-lookml/lookml-terms-and-concepts)
* [Editing and validating LookML](https://docs.looker.com/data-modeling/getting-started/lookml-validation)
* [Additional LookML basics](https://docs.looker.com/data-modeling/learning-lookml/advanced-lookml-concepts)
* Difference between [from](https://docs.looker.com/reference/explore-params/from-for-explore) and [view\_name](https://docs.looker.com/reference/explore-params/view_name#:~:text=view_name%20determines%20the%20view%20that,Explores%20from%20the%20same%20view.)

view\_name determines the view that will define the fields of an Explore. If view\_name is omitted, Looker assumes that the underlying view name is the same as the Explore name. Typically view\_name is only used to create multiple Explores from the same view.

view\_name, [from](https://cloud.google.com/looker/docs/reference/param-explore-from) and [label](https://cloud.google.com/looker/docs/reference/param-explore-label) parameters have similar, but different effects.

#### Using view\_name

Use view\_name to create multiple Explores from the same view, and reference fields the same way for each Explore:

explore: customer {

view\_name: users

}

#### Using from

Use from to create multiple Explores from the same view, and reference fields differently for each Explore:

explore: customer {

from: users}

#### Using label

Use label if you don't need to create multiple Explores from the same view, and you want the Explore's name to appear differently in the Explore menu:

explore: users {

label: "Customer"

}

* [Dimension, filter, and parameter types](https://docs.looker.com/reference/field-reference/dimension-type-reference)
* [Dimension\_group](https://docs.looker.com/reference/field-params/dimension_group)
* [Measure types](https://docs.looker.com/reference/field-reference/measure-type-reference)
* [Sql\_always\_where](https://docs.looker.com/reference/explore-params/sql_always_where)
* [Always\_filter](https://docs.looker.com/reference/explore-params/always_filter)
* [Conditionally\_filter](https://docs.looker.com/reference/explore-params/conditionally_filter)
* [Best-Practice-LookML-Dos-and-Don-ts](https://help.looker.com/hc/en-us/articles/360001784747-Best-Practice-LookML-Dos-and-Don-ts)
* [Best-Practice-Writing-Sustainable-Maintainable-LookML](https://help.looker.com/hc/en-us/articles/360001784587-Best-Practice-Writing-Sustainable-Maintainable-LookML)
* [Understanding model and view files](https://docs.looker.com/data-modeling/getting-started/model-development)
* Hidden
* [LookML quick reference](https://docs.looker.com/reference/lookml-quick-reference)
* [Working with joins in LookML](https://docs.looker.com/data-modeling/learning-lookml/working-with-joins)
* [Changing the Explore menu and field picker](https://docs.looker.com/data-modeling/learning-lookml/explore-menu-and-field-picker)
* Link
* Dril\_fields
* Yesno Dimension