Looker Developer - Qwik Start | Google Cloud Skills Boost

Qwiklabs : 12-15 minutes

GSP891

Overview

Looker is a modern data platform in Google Cloud that lets you analyze and visualize your data interactively. You can use Looker to do in-depth data analysis, integrate insights across different data sources, build actionable data-driven workflows, and create custom data applications.

What is LookML?

LookML (Looker Modeling Language) generates abstracted SQL and provides a modeling layer between the database and user. It is Looker's proprietary language that provides an abstraction layer for SQL databases.

Specifically, 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. It creates the layer between that SQL database and how the business user interacts with it.

As such, it defines many different things, like how to join tables, how to define custom tables, how to define fields from the database, and the logic for new fields. In this lab, you will get hands-on experience with the fundamentals of LookML.

Setup and requirements

Before you click the Start Lab button

Read these instructions. Labs are timed and you cannot pause them. The timer, which starts when you click **Start Lab**, shows how long Google Cloud resources will be made available to you.

This hands-on lab lets you do the lab activities yourself in a real cloud environment, not in a simulation or demo environment. It does so by giving you new, temporary credentials that you use to sign in and access Google Cloud for the duration of the lab.

To complete this lab, you need:

- Access to a standard internet browser (Chrome browser recommended).

Note: Use an Incognito or private browser window to run this lab. This prevents any conflicts between your personal account and the Student account, which may cause extra charges incurred to your personal account.

- Time to complete the lab---remember, once you start, you cannot pause a lab.

Note: If you already have your own personal Google Cloud account or project, do not use it for this lab to avoid extra charges to your account.

How to start your lab and sign in to Looker

1. When ready, click **Start Lab** .

   A new panel will appear with the temporary credentials that you must use for this lab.

   If you need to pay for the lab, a pop-up will open for you to select your payment method.

2. Note your lab credentials in the left pane. You will use them to sign in to the Looker instance for this lab.

   Note: If you use other credentials, you will get **errors or incur charges**.

3. Click **Open Looker**.

4. Enter the provided Username and Password in the Email and Password fields.

   Important: You must use the credentials from the Connection Details panel on this page. Do not use your Google Cloud Skills Boost credentials. If you have your own Looker account, do not use it for this lab.

5. Click **Log In**.

   After a successful login, you will see the Looker instance for this lab.

Overview of LookML structures

The hierarchy of LookML is structured using the following objects:

- **Projects**, which are libraries of LookML code. Because Looker uses Git for version control, a best practice is for each project to map 1:1 with a Git repository.
  - A project is composed of one or more *models*.
- A **model** is a set of Explores by business area or need. An Explore is a set of pre-joined views for business-user analysis.
  - Each model contains one or more *Explores*.
- A **view** in LookML is a database table or a logical representation of one.
  - Each view includes dimensions (which are database columns or logical representations of them) and measures (which are aggregate functions on dimensions, such as a COUNT of customers or a SUM of cost).
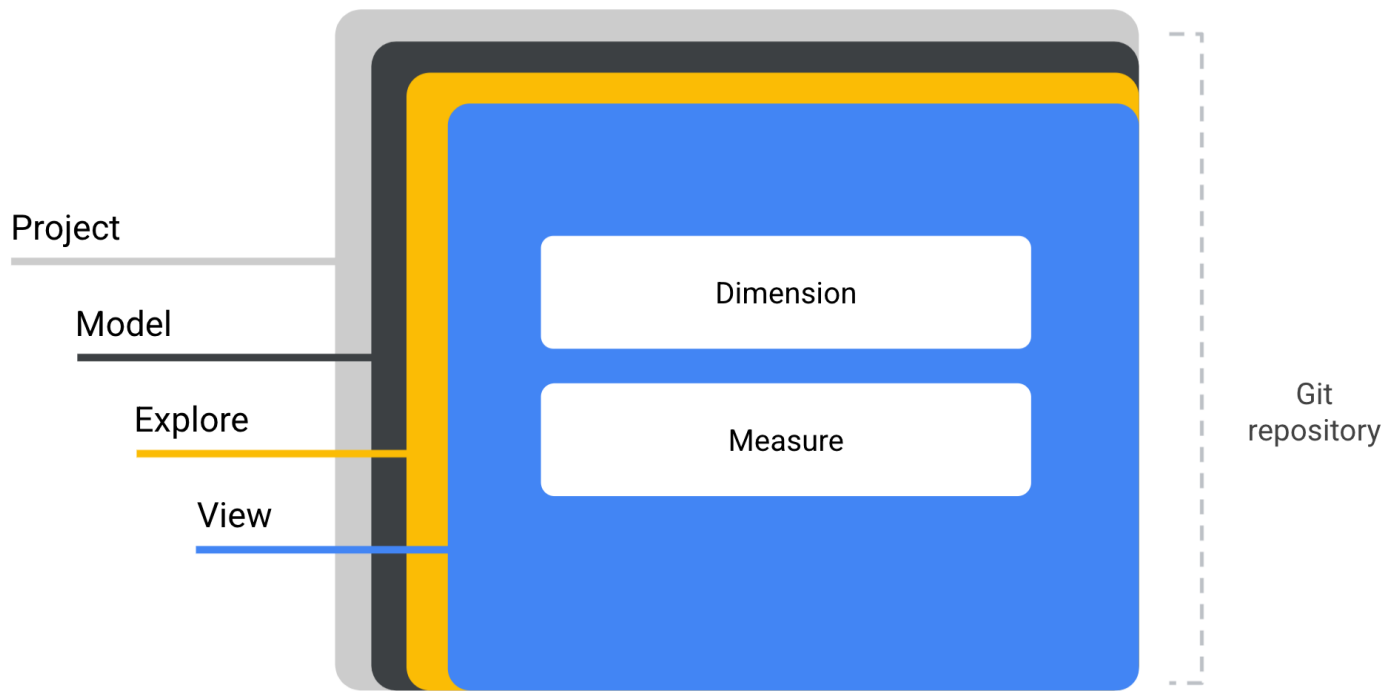
Projects

The highest-level LookML object is the project. A project is essentially a library of code that typically maps 1:1 to a data source or database connection. You can think of each project as an almost independent mini-instance or microcosm of Looker.

Schemas that cannot be joined together usually reside in different projects because there is no relation to be made across the two datasets. This depends on your database dialect and database user permissions.

A key concept to remember is: if it's possible in your SQL dialect, it should be possible in Looker. If you can go to your database console and hand-write a `SELECT` statement that does a thing, you can also code LookML so that Looker does the same thing.

LookML Projects

Project	Models
qwiklabs-ecommerce	training_ecommerce
qwiklabs-flights	ecommerce
	faa
	looker_basics

You can share content from one project to another via a feature called Project Import, if necessary, and if it's enabled for your instance, but this is an advanced approach to setting up your model architecture and not in the scope of this lab.

Models

Models are the next level of hierarchy and contain:

- The database connection you are using, as defined in the image by *line 1*.
- View files that are accessible to this model, as defined by *lines 4, 5, and 6*.
- Definitions of Explores and their join logic.

training_ecommerce.model ▾

```
1    connection: "bigquery_public_data_looker"
2
3    # include all the views
4    include: "/views/*.view"
5    include: "/z_tests/*.lkml"
6    include: "/**/*.dashboard"
7
8 ▾  datagroup: training_ecommerce_default_datagroup {
9      # sql_trigger: SELECT MAX(id) FROM etl_log;;
10     max_cache_age: "1 hour"
11   }
12
13   persist_with: training_ecommerce_default_datagroup
14
15   label: "E-Commerce Training"
16
17 ▾ explore: order_items {
18 ▾   join: users {
19       type: left_outer
20       sql_on: ${order_items.user_id} = ${users.id} ;;
21       relationship: many_to_one
22     }
23
24 ▾   join: inventory_items {
25       type: left_outer
26       sql_on: ${order_items.inventory_item_id} = ${inventory_items.id} ;;
27       relationship: many_to_one
28     }
29
30 ▾   join: products {
31       type: left_outer
32       sql_on: ${inventory_items.product_id} = ${products.id} ;;
33       relationship: many_to_one
34     }
```

Models contain data connection information and definitions of Explores. Models can be used to restrict user access to certain Explores and separate and organize Explores by business area.

Explores

Explores are one or more views joined together, usually to target a specific business question. Explores should be organized around business themes to minimize confusion for users.

Explores are the "drivers" of analysis on the frontend. They include one or more views joined together, and each usually targets a specific business question. Think of an Explore as a predefined set of tables that would frequently be joined for business inquiries and use cases.

Views

Views are where you define dimensions (which are the data attributes) and measures (which are aggregations of dimensions). Think of views as tables that bundle related fields. There are a few different types of views:

- **Standard views**, which abstract what's already in the database tables.
- **Virtual tables**, known as *derived tables*, which are discussed later in this quest.

users.view ▾

```
1 ▾  view: users {
2       sql_table_name: `cloud-training-demos.looker_ecomm.users`
3            ;;
4       drill_fields: [id]
5
6 ▾     dimension: id {
7         primary_key: yes
8         type: number
9         sql: ${TABLE}.id ;;
10      }
11
12 ▾    dimension: age {
13        type: number
14        sql: ${TABLE}.age ;;
15      }
```

Dimensions

The lowest level of a LookML object are **fields**, which can be *dimensions* or *measures*. Dimensions are created for any columns that are already in your database tables when the view files are generated from a table by Looker.

You can also create additional dimensions that would serve as logical representations of table columns. These appear in the `SELECT` and `GROUP BY` clause of a SQL statement. They are the "attributes" that describe your data.

users.view ▾

```
85 ▾    dimension: traffic_source {
86        type: string
87        sql: ${TABLE}.traffic_source ;;
88      }
89
90 ▾    dimension: zip {
91        type: zipcode
92        sql: ${TABLE}.zip ;;
93      }
94
95 ▾    measure: count {
96        type: count
97        drill_fields: [id, last_name, first_name, events.count, order_items.count]
98      }
99    }
```

Measures

Measures are aggregates that do not live explicitly in your database tables. They must be created in LookML. They aggregate dimensions into values like sums or counts.

Note that they do not appear in the GROUP BY statement of the SQL generated by Looker. Instead, they depend on dimensions to determine that grouping.

users.view ▾

```
85 ▾    dimension: traffic_source {
86         type: string
87         sql: ${TABLE}.traffic_source ;;
88       }
89
90 ▾    dimension: zip {
91         type: zipcode
92         sql: ${TABLE}.zip ;;
93       }
94
95 ▾    measure: count {
96         type: count
97         drill_fields: [id, last_name, first_name, events.count, order_items.count]
98       }
99     }
```

LookML hierarchy recap

To recap, a project is a library of code that models a data source and should map 1:1 to a Git repository. Projects contain:

- **Model files**, which define the Explores that should be packaged together and how those Explores work.
- **View files**, which describe database tables or logical representations of them.

Dimensions and measures are defined within *view* files.

Projects can also include dashboards defined in LookML to prevent business users from editing them, maintain version control, and sync them across Looker instances if your company has more than one. LookML dashboards are not in the scope of this training.

There are other types of project files, such as documents and manifests, which are not in the scope of this lab. If you're interested, you can refer to the Understanding other project files documentation.

Task 1. Create a view

In this section, you will create a new view and add some dimensions and measures to it.

1. First, on the bottom left of the Looker User Interface, click the toggle button to enter **Development mode**.

2. Click the **Develop** tab and then select the `qwiklabs-ecommerce` LookML project.

3. To create the file at the project's root level, click the **+** button at the top of the file browser in the Looker IDE.

4. Select **Create View**. Name the file `users_limited`. Click **Create**.

5. After you have created your new view, click the arrow next to the **views** folder to see a list of the existing views for the project.

6. To put your view file into the **views** folder, click and hold the `users_limited` file and drag it into the expanded folder. Your project should resemble the following:

Add some dimensions and measures

Now that you have created a new view file and organized it in your project file browser, you're ready to add some content to it.

1. Start by specifying the view name and the SQL table name you want to connect your view to. For this example, you will be connecting to the dataset used for the `qwiklabs_ecommerce` project. This is the same table for `users.view`. Add the following code on line 2:

sql_table_name: `cloud-training-demos.looker_ecomm.users` ;;

2. Now add a few dimensions. Here you will be adding the user `id`, `country`, `email`, `first_name`, and `last_name`:

dimension: id { primary_key: yes type: number sql: ${TABLE}.id ;; } dimension: country { type: string map_layer_name: countries sql: ${TABLE}.country ;; } dimension: email { type: string sql: ${TABLE}.email ;; } dimension: first_name { type: string sql: ${TABLE}.first_name ;; } dimension: last_name { type: string sql: ${TABLE}.last_name ;; }

3. Next, add a measure. This will be used for counting specific dimensions:

measure: count { type: count drill_fields: [id, last_name, first_name] }

4. Click **Save Changes**. Great! You're all done adding dimensions and measures to your new view. Your file should resemble the following:

File Browser ÷ ☑ Q +

- ▼ models
  - ⊚ training_ecommerce.model
- ▼ views
  - ▦ distribution_centers.view
  - ▦ event_session_facts.view
  - ▦ event_session_funnel.view
  - ▦ events.view
  - ▦ inventory_items.view
  - ▦ order_items.view
  - ▦ products.view
  - ▦ users.view
  - ▦ **users_limited**.view ⬤
  - ▸ z_tests
- ⊡ business_pulse.dashboard

users_limited.view ▾

```
1  view: users_limited {
2      sql_table_name: `cloud-training-demos.looker_ecomm.users` ;;
3
4      dimension: id {
5        primary_key: yes
6        type: number
7        sql: ${TABLE}.id ;;
8      }
9
10     dimension: country {
11       type: string
12       map_layer_name: countries
13       sql: ${TABLE}.country ;;
14     }
15
16     dimension: email {
17       type: string
18       sql: ${TABLE}.email ;;
19     }
20
21     dimension: first_name {
22       type: string
23       sql: ${TABLE}.first_name ;;
24     }
25
26     dimension: last_name {
27       type: string
28       sql: ${TABLE}.last_name ;;
29     }
30
31     measure: count {
32       type: count
33       drill_fields: [id, last_name, first_name]
34     }
35  }
```

Commit changes and deploy to production

1. Click **Validate LookML** and then click **Commit Changes & Push**.

2. Add a commit message and click **Commit**.
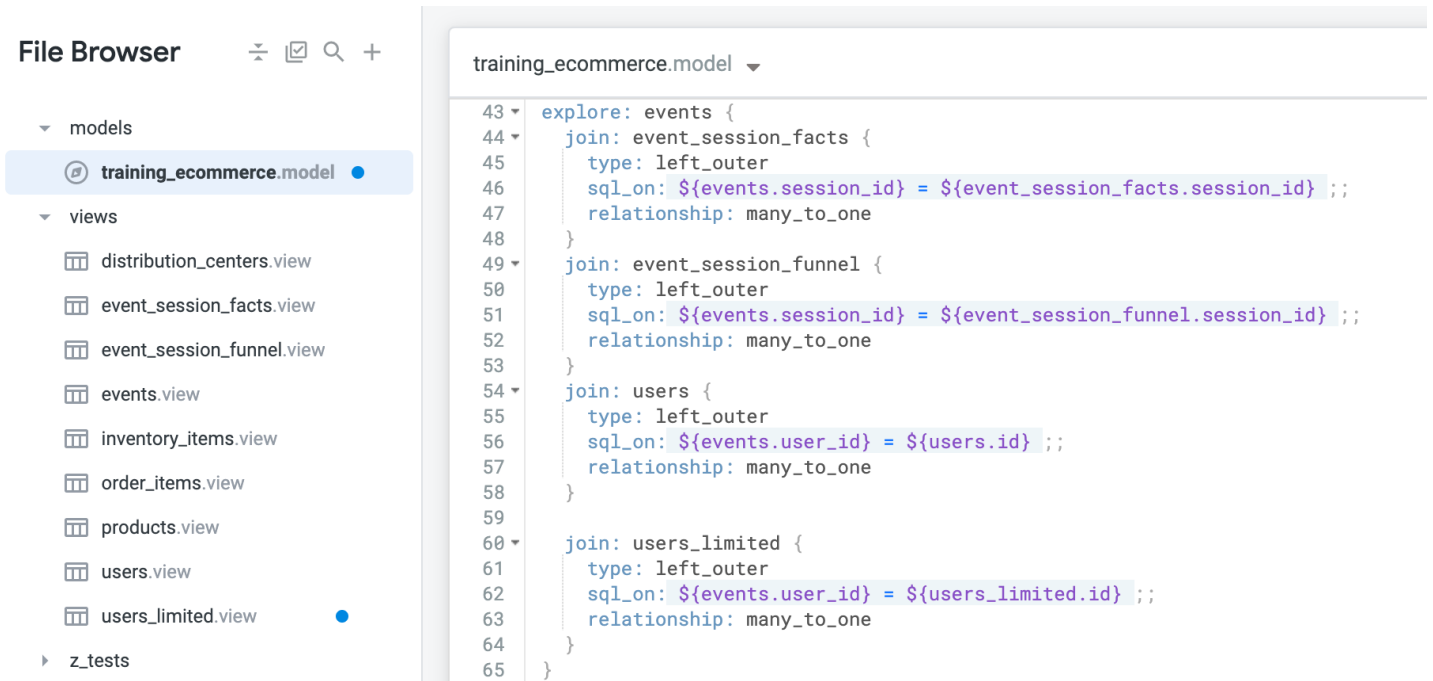
3. Lastly, click **Deploy to Production**.

Click **Check my progress** to verify the objective. Create a view

Task 2. Join a view to an existing explore

1. In the file browser, under the **models** folder, navigate to the `training_ecommerce.model` file.

2. In the `explore: events` definition, add a new line after `join: users,` and paste the following:

join: users_limited { type: left_outer sql_on: ${events.user_id} = ${users_limited.id};; relationship: many_to_one }

3. Click **Save Changes**. Your project file should now resemble the following:

File Browser   ÷ ☑ Q +

- ▼ models
  - ⊘ **training_ecommerce**.model ●
- ▼ views
  - ⊞ distribution_centers.view
  - ⊞ event_session_facts.view
  - ⊞ event_session_funnel.view
  - ⊞ events.view
  - ⊞ inventory_items.view
  - ⊞ order_items.view
  - ⊞ products.view
  - ⊞ users.view
  - ⊞ users_limited.view   ●
  - ▸ z_tests

training_ecommerce.model ▾

```
43 ▾  explore: events {
44 ▾    join: event_session_facts {
45        type: left_outer
46        sql_on: ${events.session_id} = ${event_session_facts.session_id} ;;
47        relationship: many_to_one
48      }
49 ▾    join: event_session_funnel {
50        type: left_outer
51        sql_on: ${events.session_id} = ${event_session_funnel.session_id} ;;
52        relationship: many_to_one
53      }
54 ▾    join: users {
55        type: left_outer
56        sql_on: ${events.user_id} = ${users.id} ;;
57        relationship: many_to_one
58      }
59
60 ▾    join: users_limited {
61        type: left_outer
62        sql_on: ${events.user_id} = ${users_limited.id} ;;
63        relationship: many_to_one
64      }
65    }
```

4. Click the caret next to the file title at the top of the IDE and then select **Explore Events**.

training_ecommerce.model ▼


```
43 ▼    explore: events {
44 ▼       join: event_ses
45            type: left_ou
46            sql_on: ${eve                              cts.session_id} ;;
47            relationship:
48          }
49 ▼       join: event_ses
50            type: left_ou
51            sql_on: ${eve                              nnel.session_id} ;;
52            relationship:
53          }
54 ▼       join: users {
55            type: left_ou
56            sql_on: ${eve
57            relationship:
58          }
59
60 ▼       join: users_lim
61            type: left_ou
62            sql_on: ${events.user_id} = ${users_limited.id} ;;
63            relationship: many_to_one
64          }
65       }
66
67
```

View Uncommitted File Changes

Revert Uncommitted File Changes

Explore Events

Explore Order Items

SQL Runner: bigquery_public_data_looker

Explore Last Query

Fold LookML   ⌥⌘0

Unfold LookML   ⌥⌘=

Keyboard Shortcuts   ?

5. Next, navigate to your new view in the Explore page by selecting **Users Limited**.

▸	Custom Fields	+ Add
▸	Event Session Facts	
▸	Event Session Funnel	
▸	Events	
▸	Users	
▾	Users Limited	

DIMENSIONS

Country

Email

First Name

ID

Last Name

MEASURES

Count

6. Under **Users Limited**, select the **First Name** dimension and the **Count** measure.

7. Click **Run**. Your visualization should resemble the following:

	Users Limited **First Name**	Users Limited **Count** ∨
1	Mary	2,078
2	James	2,034
3	John	2,023
4	Robert	1,990
5	Michael	1,699
6	William	1,508
7	David	1,506
8	Richard	1,118
9	Charles	968
10	Joseph	880

8. Navigate back to the `training_ecommerce.model` file.

Commit changes and deploy to production

1. Click **Validate LookML** and then click **Commit Changes & Push**.

2. Add a commit message and click **Commit**.

3. Lastly, click **Deploy to Production**.

Click **Check my progress** to verify the objective. Join view to an explore

Congratulations!

In this lab you learned how to define and read core LookML terms and concepts. You then learned how to organize and understand the main LookML structures and hierarchy, created a view, added dimensions and measures to it, and joined the view to an existing Explore.

Finish your quest

This self-paced lab is part of the Understanding LookML in Looker quest and Build LookML Objects in Looker skill badge quest. A quest is a series of related labs that form a learning path. Completing a quest earns you a badge to recognize your achievement. You can make your badge or badges public and link to them in your online resume or social media account. Enroll in any quest that contains this lab and get immediate completion credit. See the Google Cloud Skills Boost catalog to see all available quests.

Next steps / Learn more

- LookML quick reference
- LookML terms and concepts
- Join the Looker Community
- Additional LookML basics

Google Cloud training and certification

...helps you make the most of Google Cloud technologies. Our classes include technical skills and best practices to help you get up to speed quickly and continue your learning journey. We offer fundamental to advanced level training, with on-demand, live, and virtual options to suit your busy schedule. Certifications help you validate and prove your skill and expertise in Google Cloud technologies.

Manual Last Updated August 29, 2022

Lab Last Tested October 21, 2021