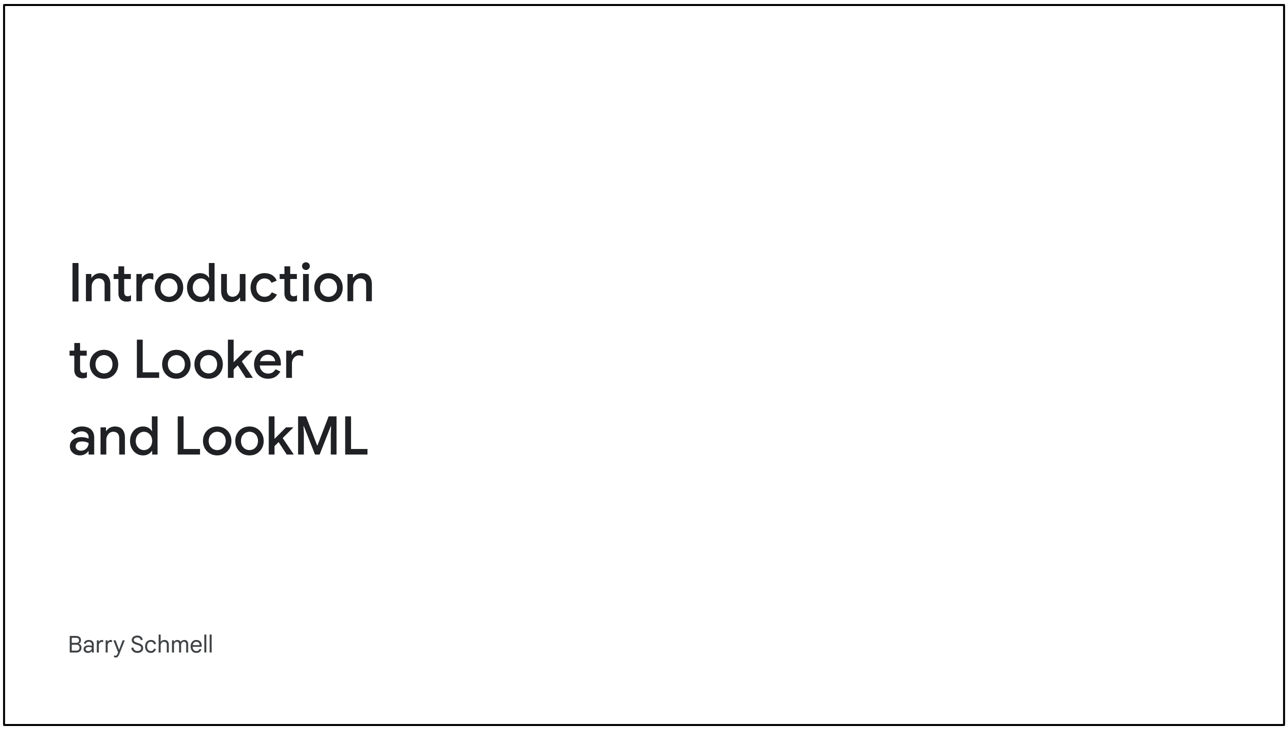
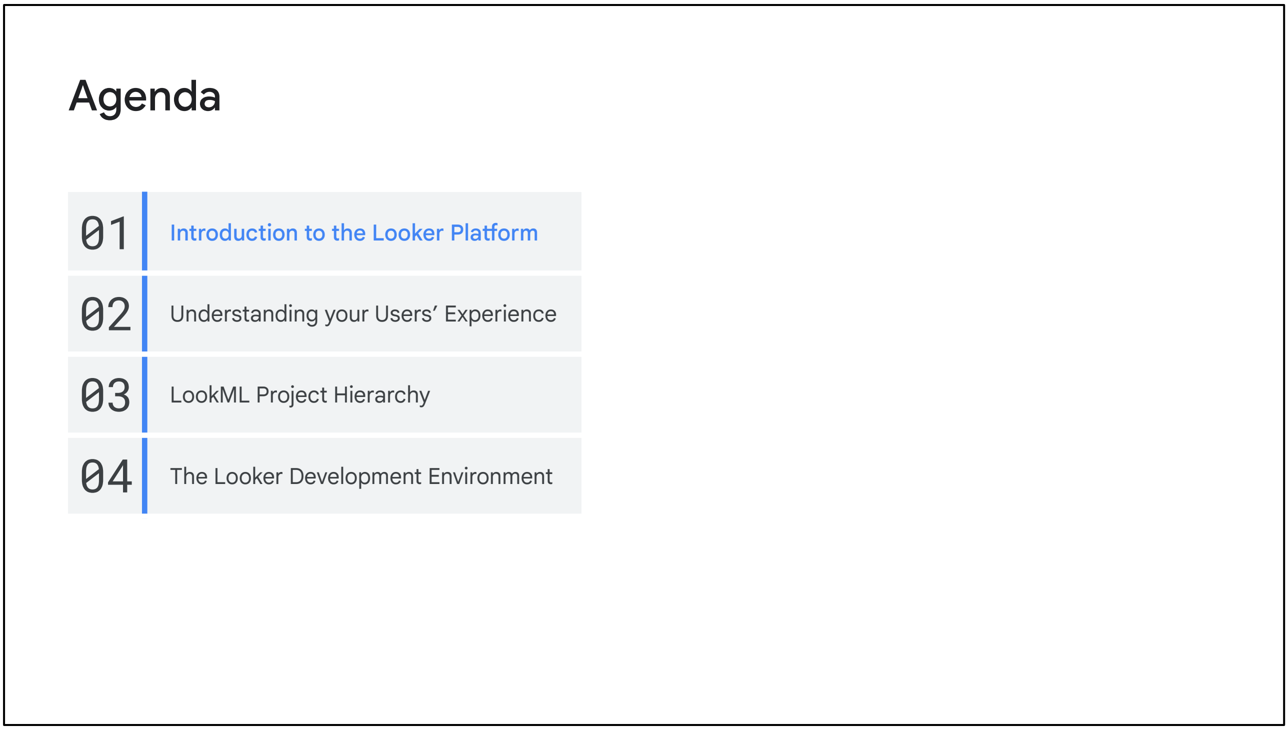


Introduction to Looker and LookML

Barry Schmell

Hi, welcome to the module on Introduction to Looker and LookML! My name is Barry Schmell, and I am a trainer here at Google.

In this module, we will explore the Looker platform as LookML developers and review the key terminology and tools for writing LookML code.



Agenda

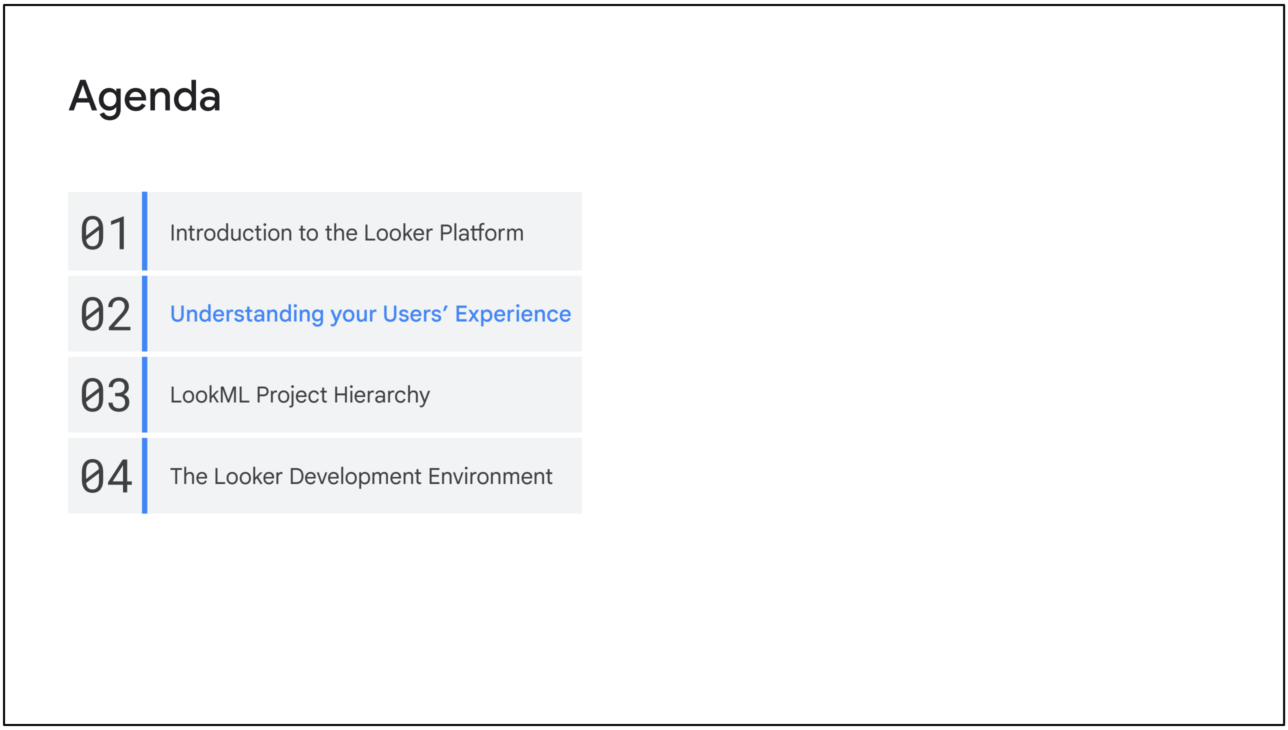
01 Introduction to the Looker Platform

02 Understanding your Users’ Experience

03 LookML Project Hierarchy

04 The Looker Development Environment

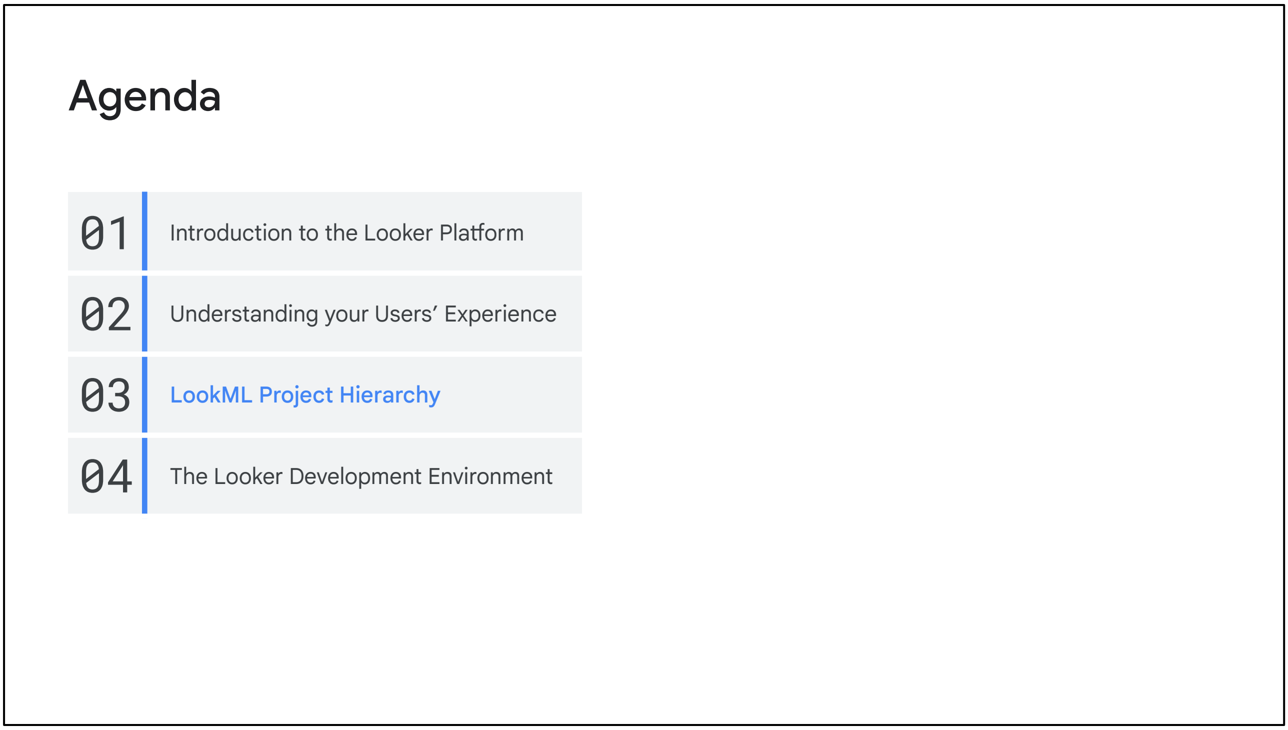
We will begin with an overview of the Looker platform and architecture to understand how Looker can play a key role in your organization’s data workflows.



Agenda

- 01 Introduction to the Looker Platform
- 02 **Understanding your Users’ Experience**
- 03 LookML Project Hierarchy
- 04 The Looker Development Environment

Next, we will complete a walkthrough of the Looker User Interface to understand how your business users leverage Looker to explore and analyze data.



Agenda

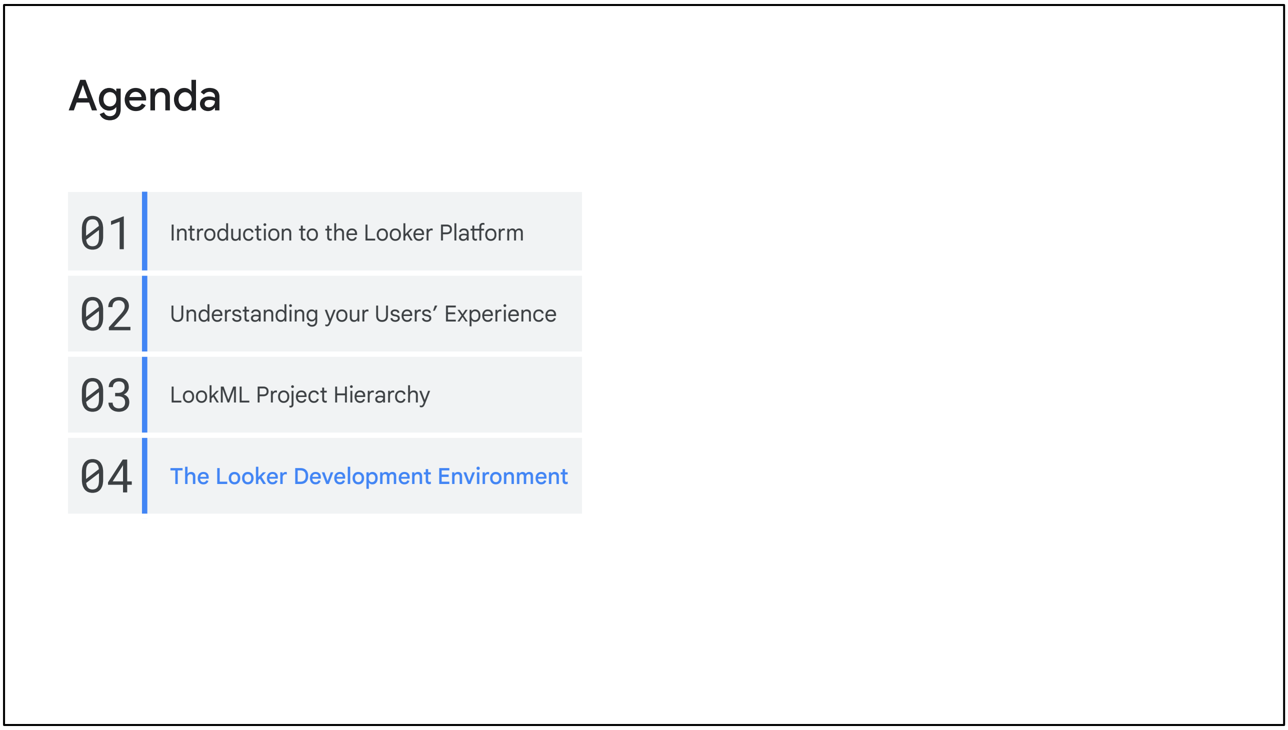
01 Introduction to the Looker Platform

02 Understanding your Users’ Experience

03 LookML Project Hierarchy

04 The Looker Development Environment

Then, we will review the LookML project hierarchy to understand its key components and the role that each plays in the overall LookML structure.



Agenda

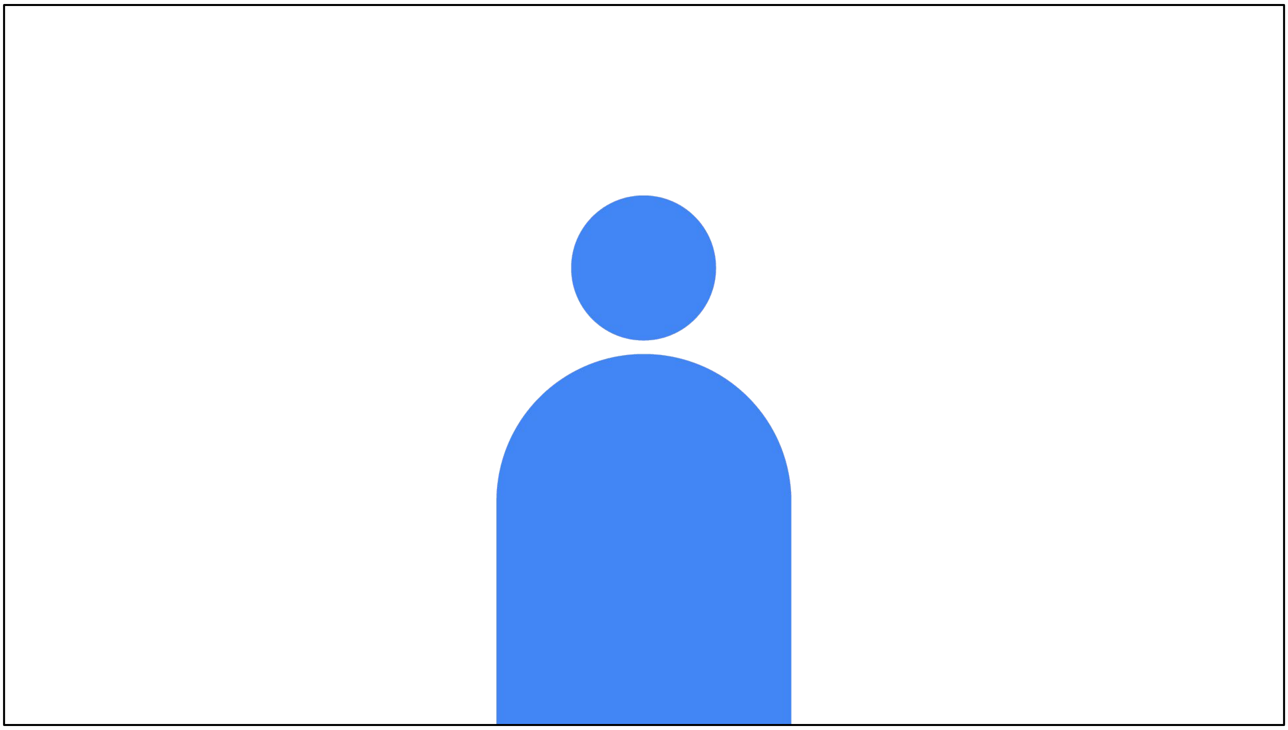
01 Introduction to the Looker Platform

02 Understanding your Users’ Experience

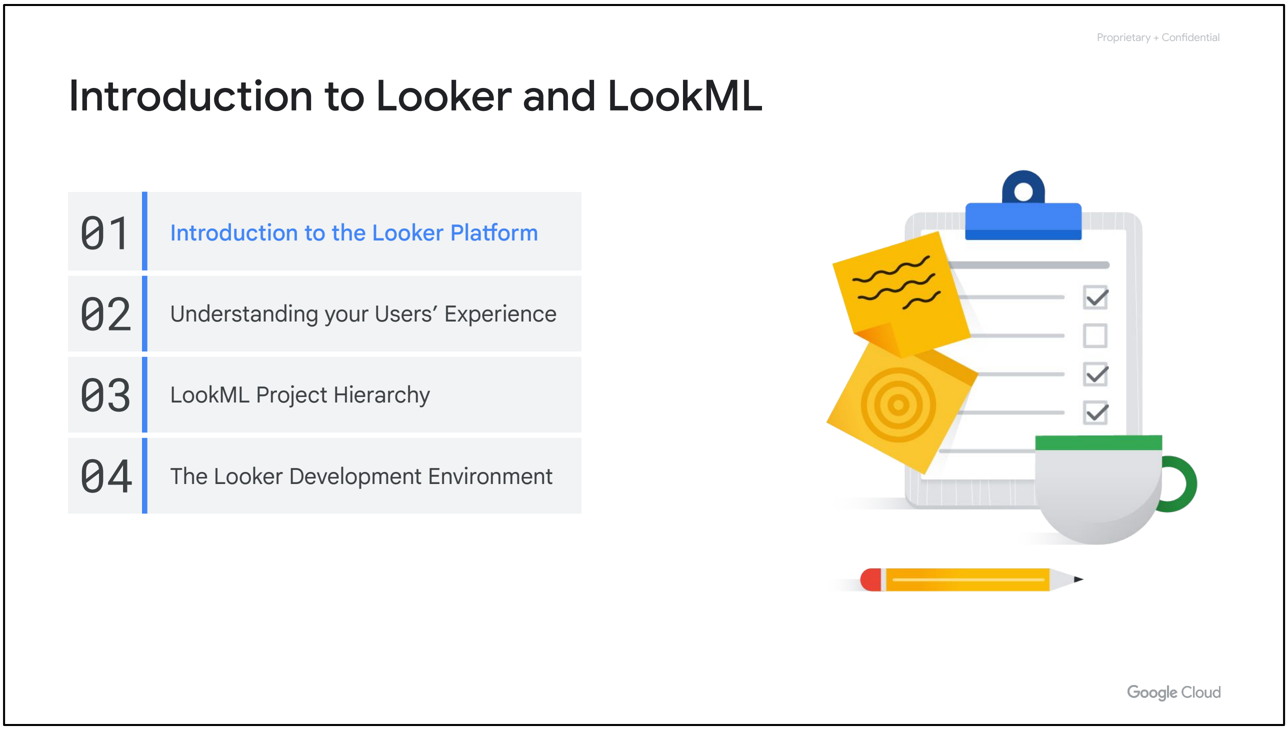
03 LookML Project Hierarchy

04 The Looker Development Environment

Last, we will explore the Looker development environment and review its key features for your workflow as a LookML developer.



Let’s get started!



Proprietary + Confidential

Introduction to Looker and LookML

01 Introduction to the Looker Platform

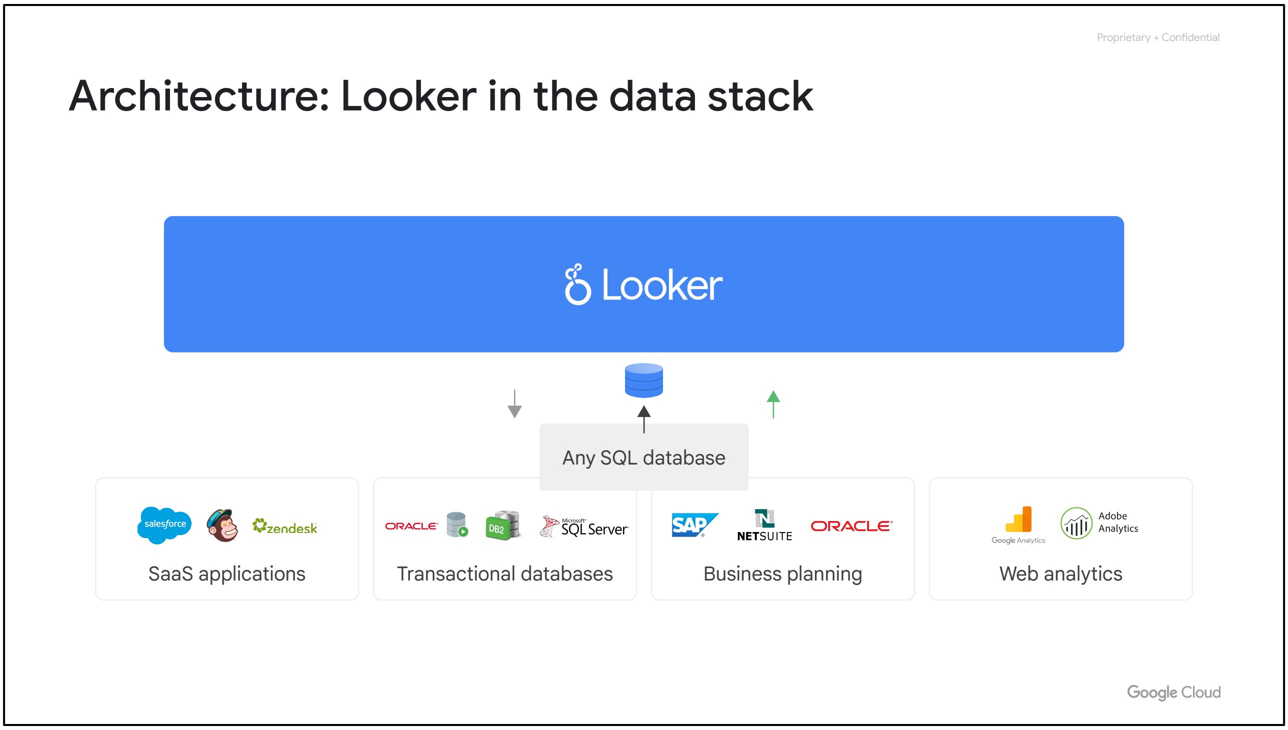
02 Understanding your Users’ Experience

03 LookML Project Hierarchy

04 The Looker Development Environment

Google Cloud

Looker is a Business Intelligence (BI) software and big data analytics platform that helps business users to explore, analyze and share real-time data analytics easily.



Proprietary + Confidential

Architecture: Looker in the data stack

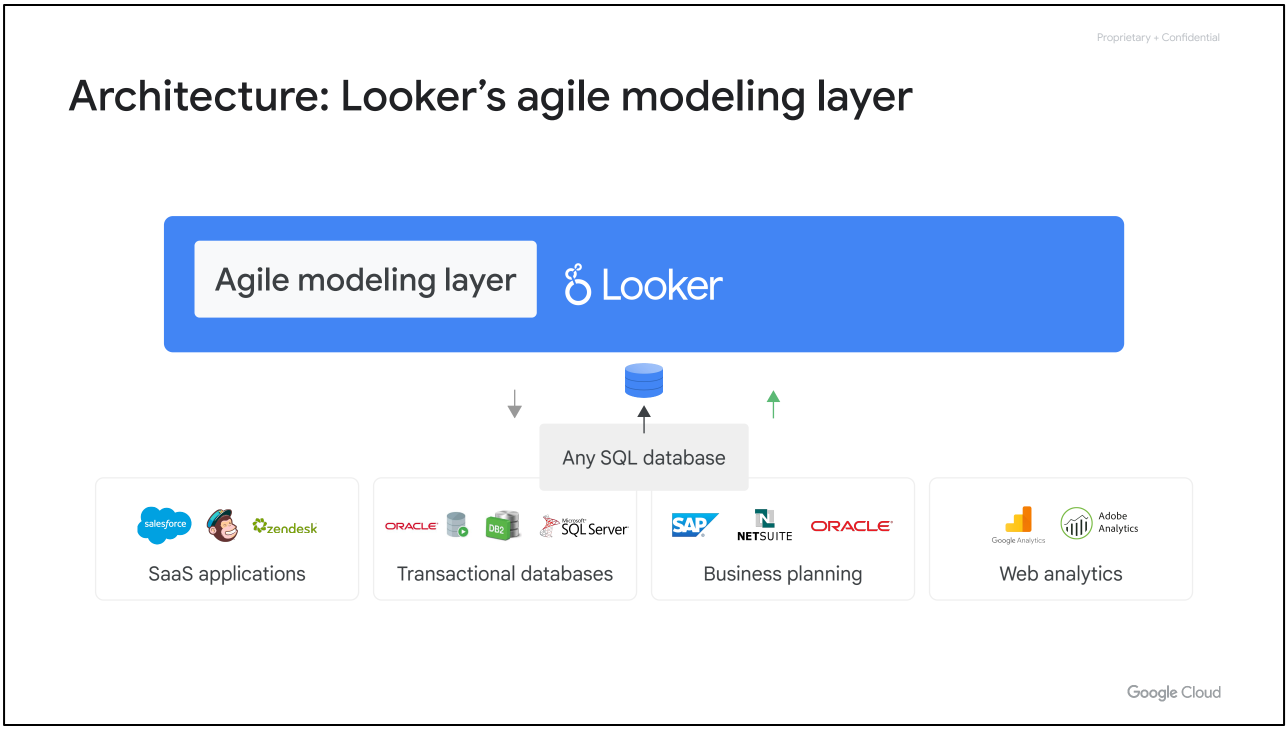
Google Cloud

As a browser-based, Software as a Service (SaaS) platform, Looker connects directly to SQL databases.

For example, you can connect Looker to:

- Other SaaS applications such as Salesforce, Mailchimp, and Zendesk.
- Heavy read-write operations in transactional databases such as Oracle, IBM Db2, and Microsoft SQL Server.
- Business planning tools such as SAP, NetSuite, and Oracle.
- And web analytics products such as Google Analytics or Adobe Analytics.

These are just a few examples. Looker is multi-cloud and supports over 65 database dialects.



Proprietary + Confidential

Architecture: Looker's agile modeling layer

Agile modeling layer

Looker

Any SQL database

SaaS applications

Transactional databases

Business planning

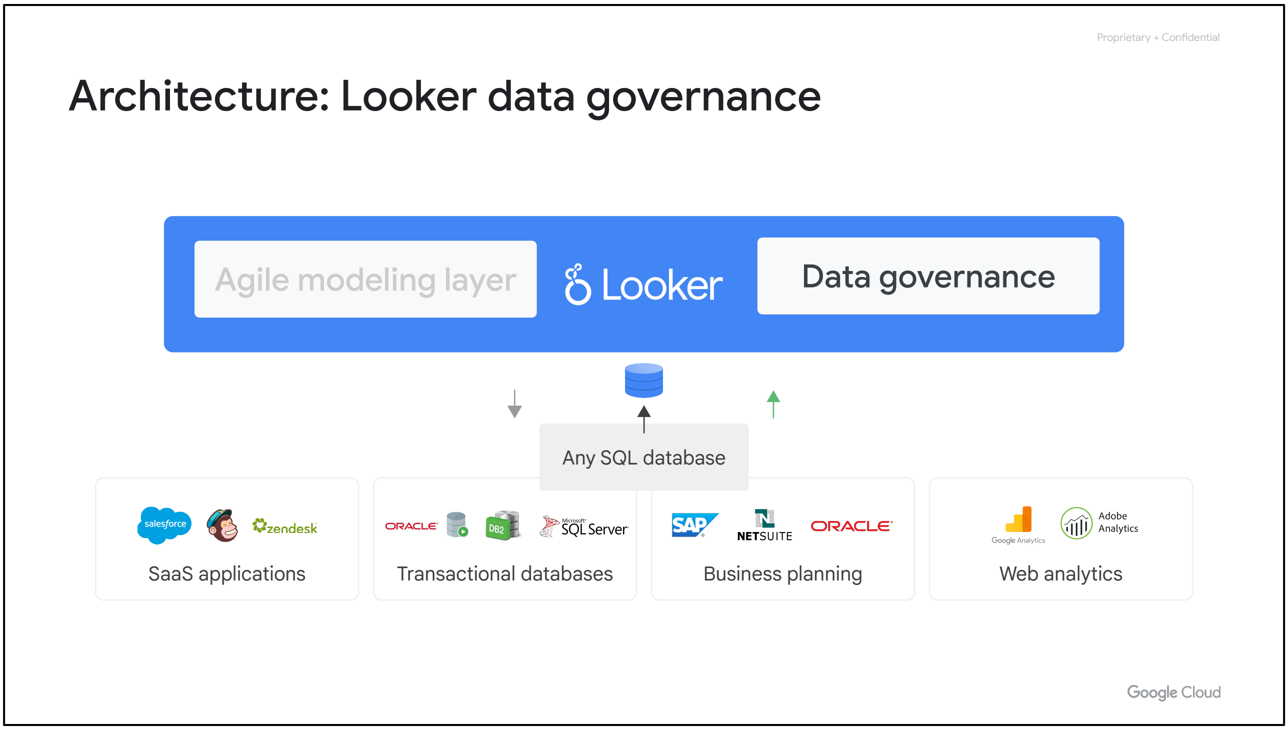
Web analytics

Google Cloud

One major benefit of Looker is its agile modeling layer, which can save data teams and business analysts time that would otherwise be spent manually writing and editing SQL queries. Looker's agile modeling layer allows developers to define, through Looker Modeling Language or LookML, how the database is structured and how the tables and columns relate to each other.

A useful way to think about LookML is that it is an abstraction layer for SQL that developers use to tell Looker what data to use from the connected database and how it should interpret that data.

As users explore and analyze the data, Looker uses the defined LookML models to automatically generate SQL **SELECT** queries to send to the database and return the appropriate results.



Proprietary + Confidential

Architecture: Looker data governance

Agile modeling layer

Looker

Data governance

Any SQL database

SaaS applications

Transactional databases

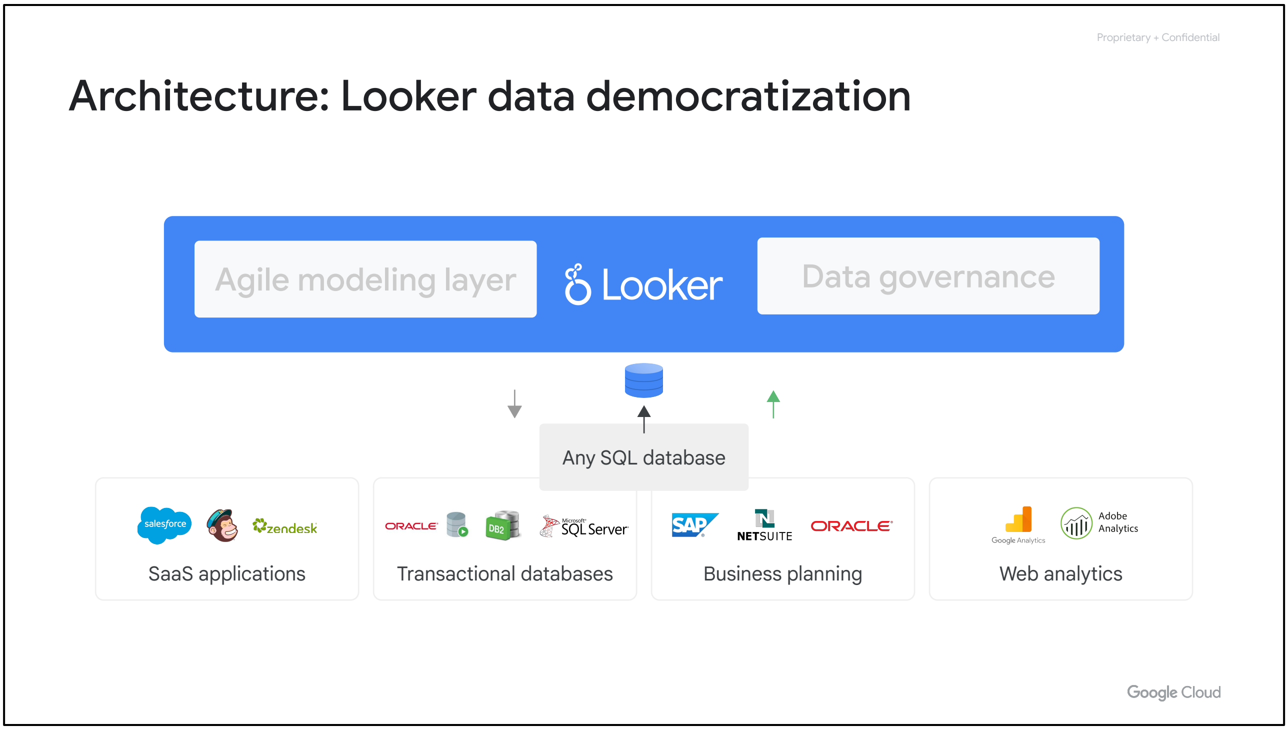
Business planning

Web analytics

Google Cloud

Another benefit of Looker is data governance, which means that you can define a single source of truth for data that everyone in the organization can understand and trust.

In Looker, you can enforce various types of data security and governance through the Looker User Interface (UI), such as assigning specific user roles, as well as through LookML, such as providing access to specific fields or rows of data.



Proprietary + Confidential

Architecture: Looker data democratization

Agile modeling layer

Looker

Data governance

Any SQL database

SaaS applications

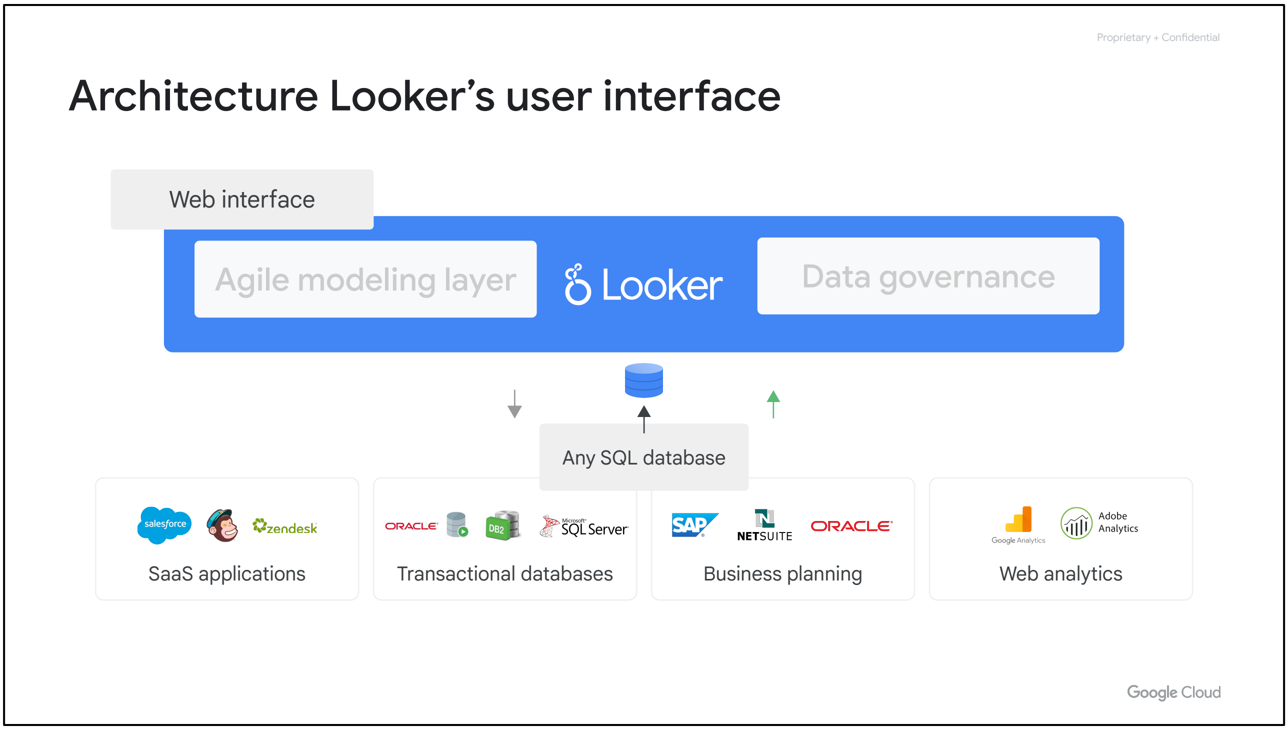
Transactional databases

Business planning

Web analytics

Google Cloud

To help organizations disseminate data, Looker can surface and expose query results in several ways.



Proprietary + Confidential

Architecture Looker's user interface

Web interface

Agile modeling layer

Looker

Data governance

Any SQL database

SaaS applications

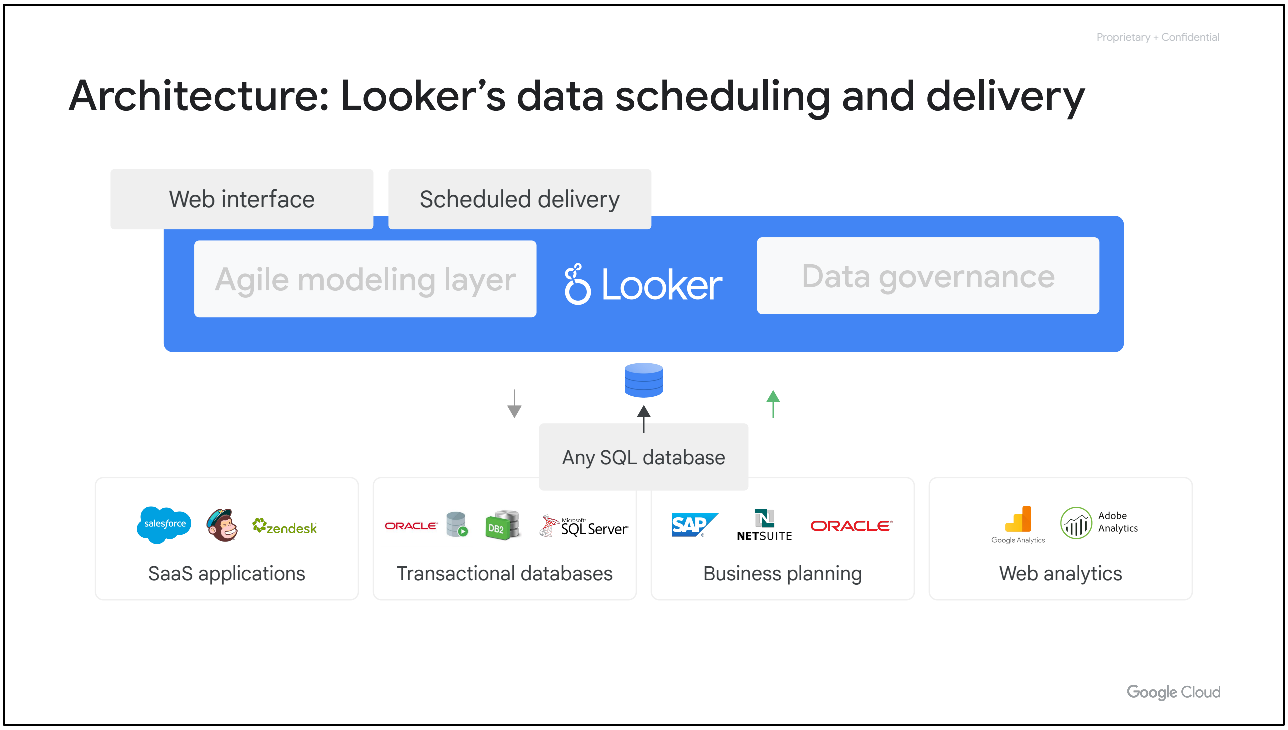
Transactional databases

Business planning

Web analytics

Google Cloud

The first is through the web interface. This can be in the form of Explores (which are report-builder interfaces), Looks (which are standalone reports or visualizations), and dashboards (which contain multiple visualizations).



Proprietary + Confidential

Architecture: Looker's data scheduling and delivery

Web interface

Scheduled delivery

Agile modeling layer

Looker

Data governance

Any SQL database

SaaS applications

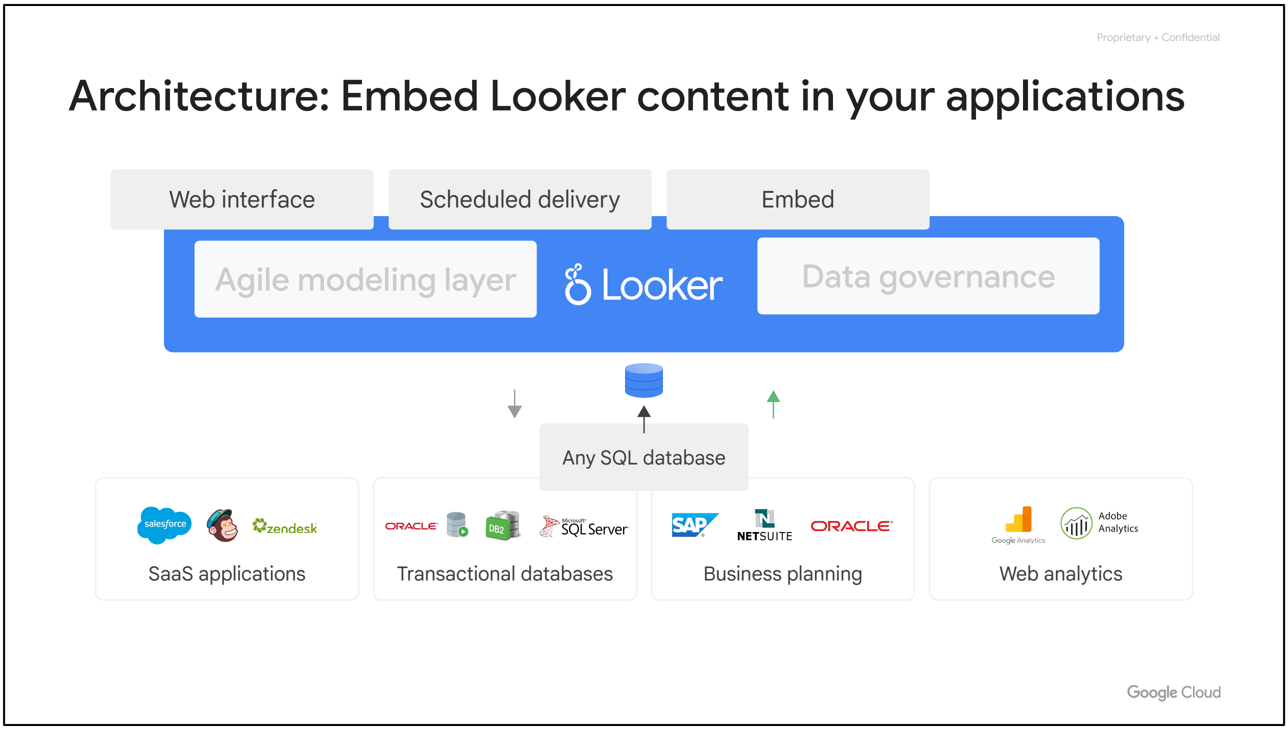
Transactional databases

Business planning

Web analytics

Google Cloud

Another way is through scheduled data deliveries, such as sending Looks and dashboards to specific email addresses or Cloud Storage buckets on a one-time or recurring basis.



Proprietary + Confidential

Architecture: Embed Looker content in your applications

Web interface

Scheduled delivery

Embed

Agile modeling layer

Looker

Data governance

Any SQL database

SaaS applications

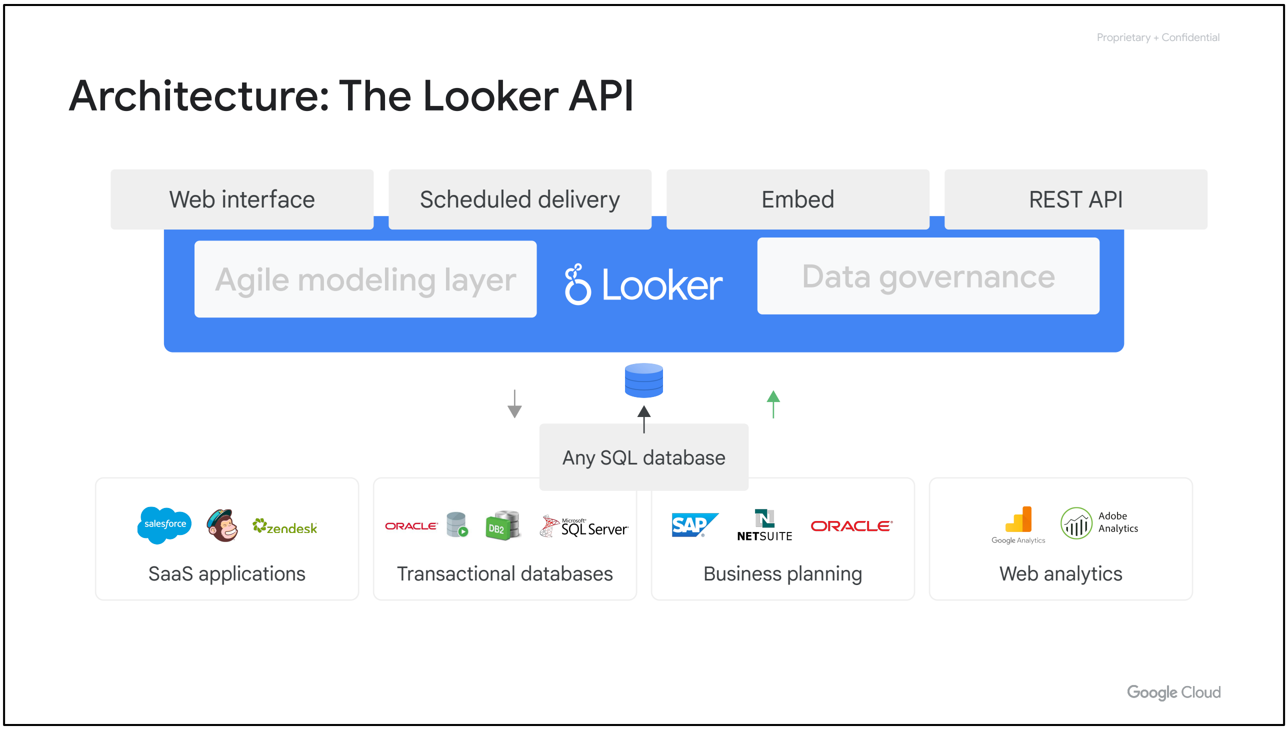
Transactional databases

Business planning

Web analytics

Google Cloud

Explores, dashboards, and Looks can also be embedded within other websites or applications.



Proprietary + Confidential

Architecture: The Looker API

Web interface

Scheduled delivery

Embed

REST API

Agile modeling layer

Looker

Data governance

Any SQL database

SaaS applications

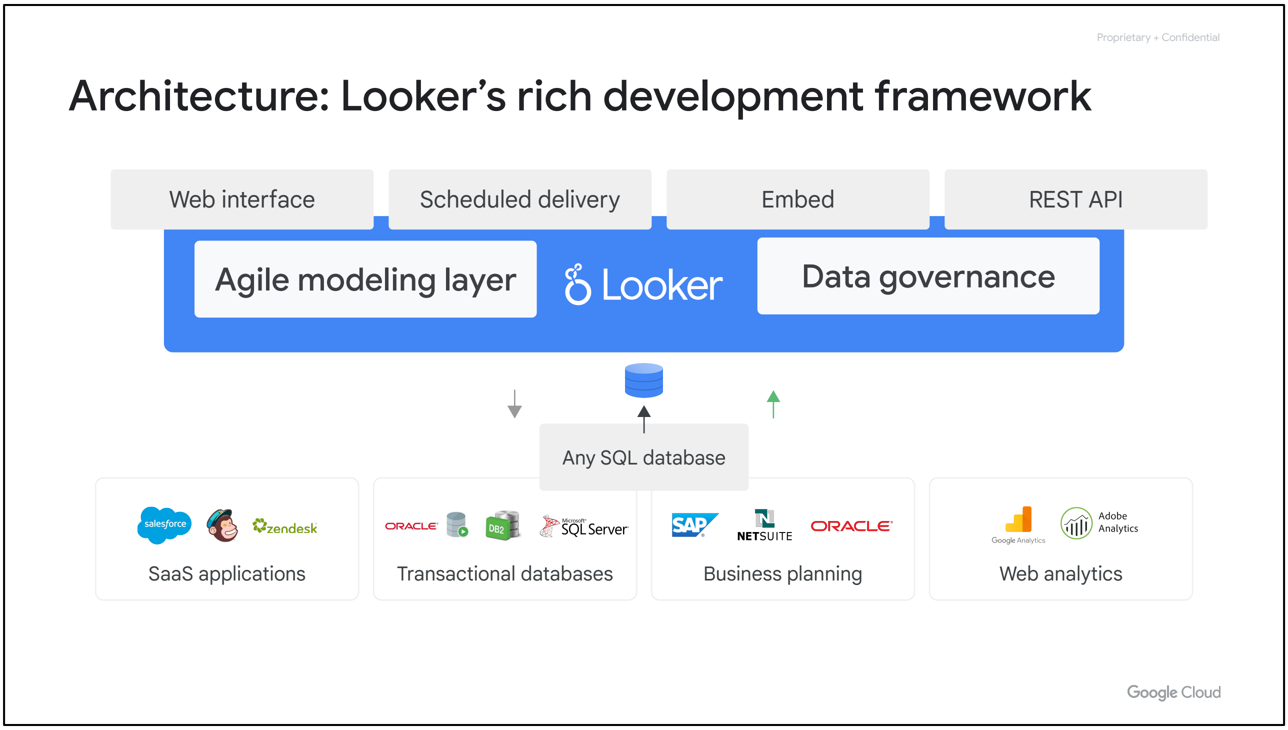
Transactional databases

Business planning

Web analytics

Google Cloud

Last, Looker provides a REST API that allows you to retrieve, analyze, and transform data and metadata directly from the Looker platform.



Proprietary + Confidential

Architecture: Looker's rich development framework

Web interface

Scheduled delivery

Embed

REST API

Agile modeling layer

Looker

Data governance

Any SQL database

SaaS applications

Transactional databases

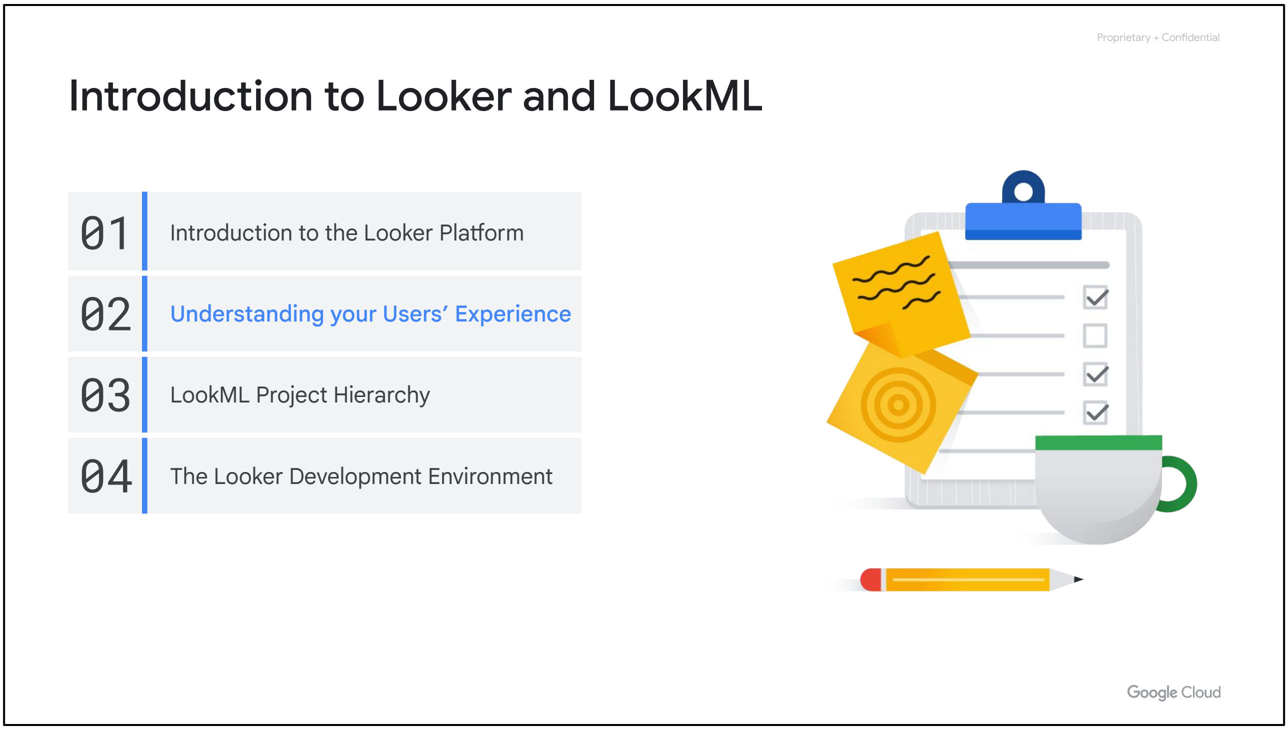
Business planning

Web analytics

Google Cloud

Looker's unique architecture provides a rich development framework that is built to support enterprise-grade workflows and help your users and tools access the most accurate and up-to-date version of your organization's data.

With this unified view into your organization's data, you, as a LookML developer, can curate data experiences to ensure that both people and systems get the data they need, how and when they need it.



Proprietary + Confidential

Introduction to Looker and LookML

01 Introduction to the Looker Platform

02 Understanding your Users’ Experience

03 LookML Project Hierarchy

04 The Looker Development Environment

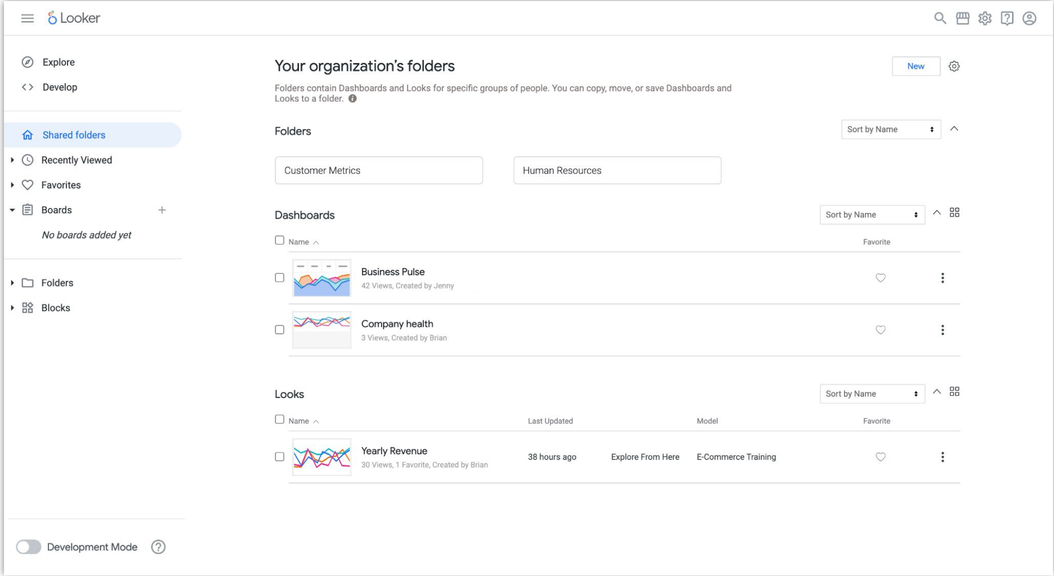
Google Cloud

To understand how you as a LookML developer can support business users, it’s important to become familiar with the business user experience and how they use the Looker platform on a daily basis to answer data-driven questions.

Furthermore, the ability to navigate the Looker platform as a business user will help you to to fully test your changes to LookML code by reviewing how results appear when accessed by business users.

Let’s explore the business user experience by walking through an example Looker instance.

Proprietary + Confidential



Your organization's folders

Folders contain Dashboards and Looks for specific groups of people. You can copy, move, or save Dashboards and Looks to a folder.

Folders — Sort by Name

Customer Metrics

Human Resources

Dashboards — Sort by Name

Name	Favorite
Business Pulse 42 Views, Created by Jenny	♡
Company health 3 Views, Created by Brian	♡

Looks — Sort by Name

Name	Last Updated		Model	Favorite
Yearly Revenue 30 Views, 1 Favorite, Created by Brian	38 hours ago	Explore From Here	E-Commerce Training	♡

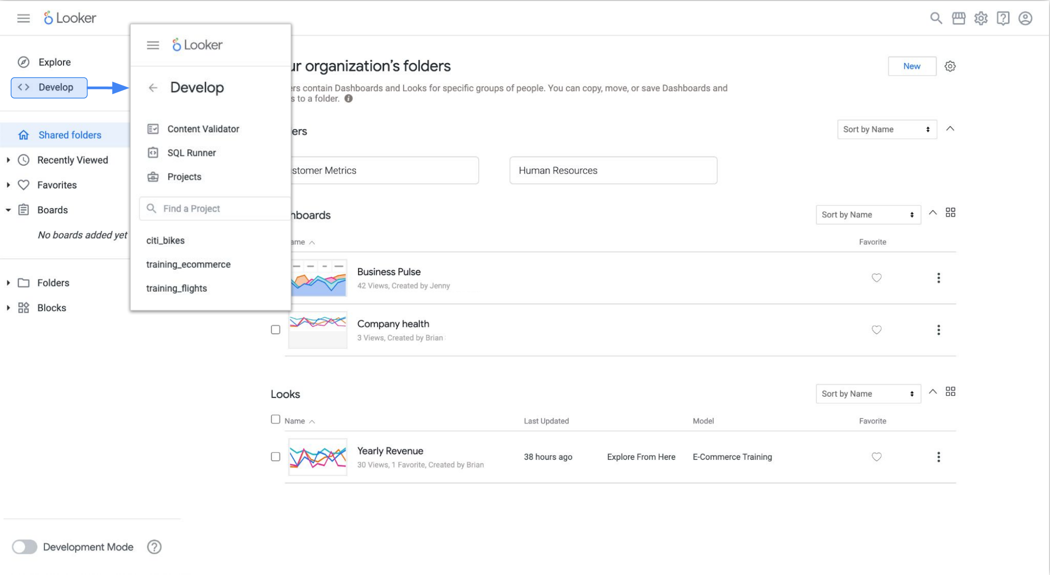
Examples are based on fictional data.

Google Cloud

When you first log in to the Looker platform, your home page may vary depending on what your company's Looker administrator has configured.

In this example instance, we will begin on the **Shared** folders page. Folders in Looker are where content lives, just as files in your computer or Google Drive are stored in folders.

Proprietary + Confidential



ır organization's folders

rs contain Dashboards and Looks for specific groups of people. You can copy, move, or save Dashboards and
s to a folder.

Name	Last Updated		Model	Favorite
Yearly Revenue 30 Views, 1 Favorite, Created by Brian	38 hours ago	Explore From Here	E-Commerce Training	♡

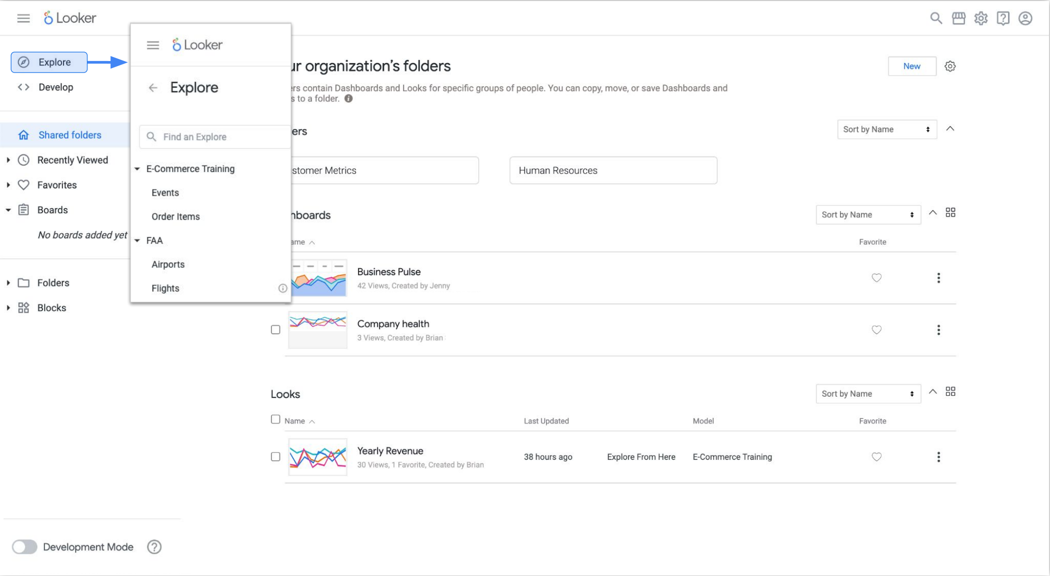
Examples are based on fictional data.

Google Cloud

As you may know, Looker has two primary categories of users: business users and developers.

As a LookML developer, you use the **Develop** environment to curate report-builder interfaces called Explores that are used by business users and to configure other aspects in your Looker instance such as rules for caching and data security.

Proprietary + Confidential



rs contain Dashboards and Looks for specific groups of people. You can copy, move, or save Dashboards and s to a folder.

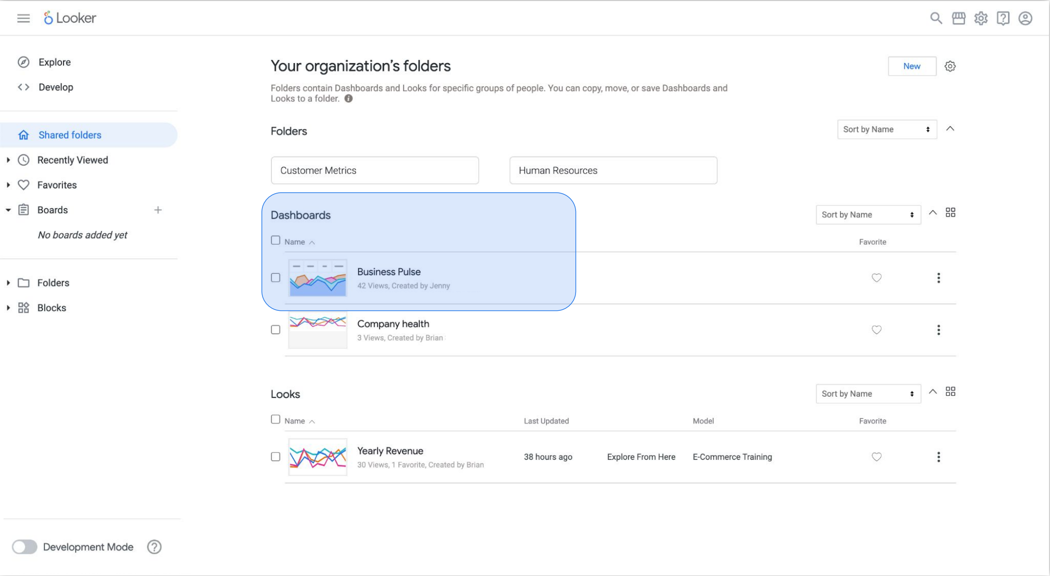
Examples are based on fictional data.

Google Cloud

Business users can click on Explore to see a list of the custom Explores that LookML developers have modeled for them. Within the Explores, they can analyze and visualize data to answer business questions and save their results as visualizations and reports.

This example Looker instance has many Explores including one for Order Items under the E-Commerce Training header and another for Flights under the FAA header.

Proprietary + Confidential



Your organization's folders

Folders contain Dashboards and Looks for specific groups of people. You can copy, move, or save Dashboards and Looks to a folder.

Folders

Customer Metrics

Human Resources

Dashboards

Name	Favorite
Business Pulse 42 Views, Created by Jenny	
Company health 3 Views, Created by Brian	

Looks

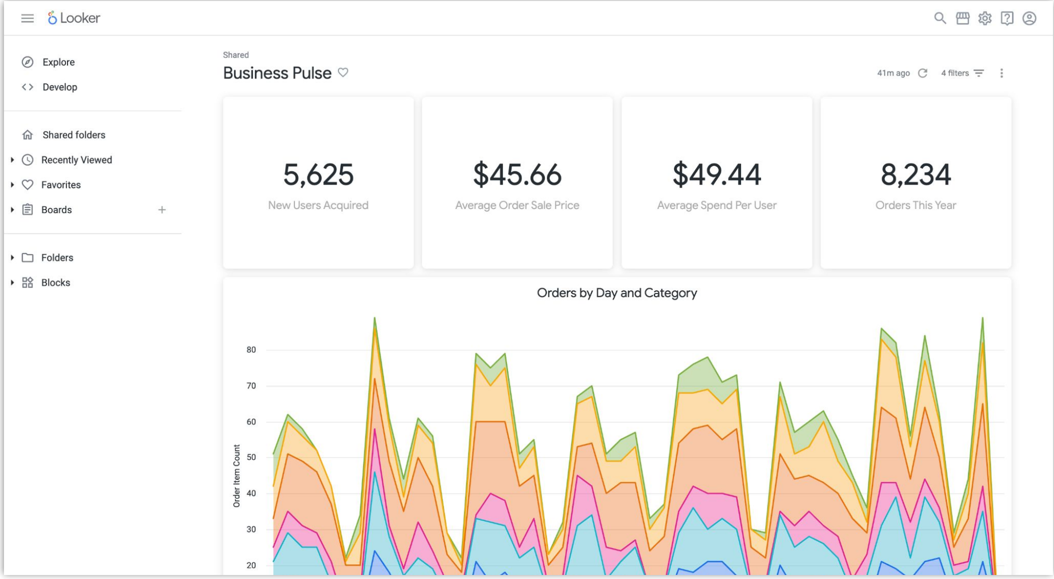
Name	Last Updated		Model	Favorite
Yearly Revenue 30 Views, 1 Favorite, Created by Brian	38 hours ago	Explore From Here	E-Commerce Training	

Examples are based on fictional data.

Google Cloud

As a business user working at a hypothetical e-commerce company, you can also navigate through the various folders of the Looker instance to find content that has already been built using the available Explores, such as the Business Pulse dashboard.

Proprietary + Confidential



Business Pulse

5,625
New Users Acquired

$45.66
Average Order Sale Price

$49.44
Average Spend Per User

8,234
Orders This Year

Orders by Day and Category

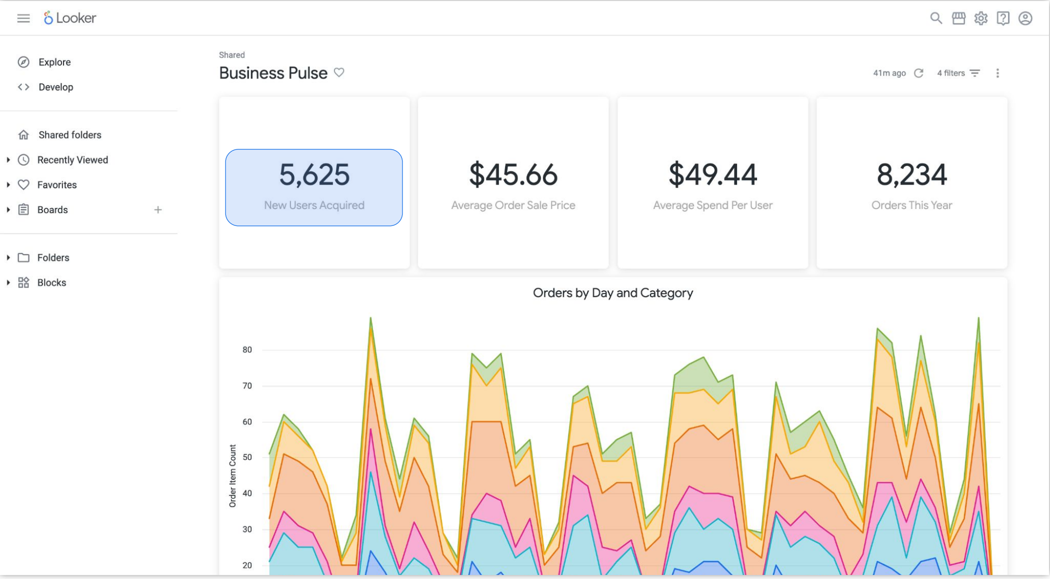
Examples are based on fictional data.

For business users, this example Business Pulse dashboard provides some high-level key performance indicators (KPIs) that a typical ecommerce company might care about, such as the number of new users acquired and the average order amount.

In Looker, these are called single value visualizations.

Scrolling down, you can see other visualizations such as the number of orders by category over time in an area chart.

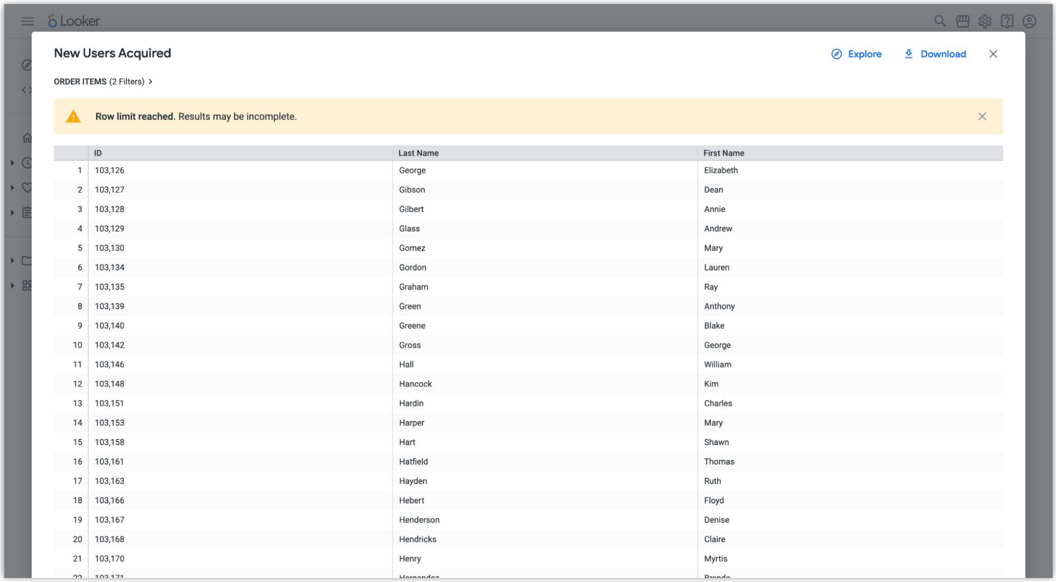
Proprietary + Confidential



Examples are based on fictional data.

Let’s say you want to learn more about new users your company has acquired. You can click on the number provided in the New Users Acquired tile to drill down to the underlying data.

Proprietary + Confidential



New Users Acquired

ORDER ITEMS (2 Filters) >

⚠ **Row limit reached.** Results may be incomplete.

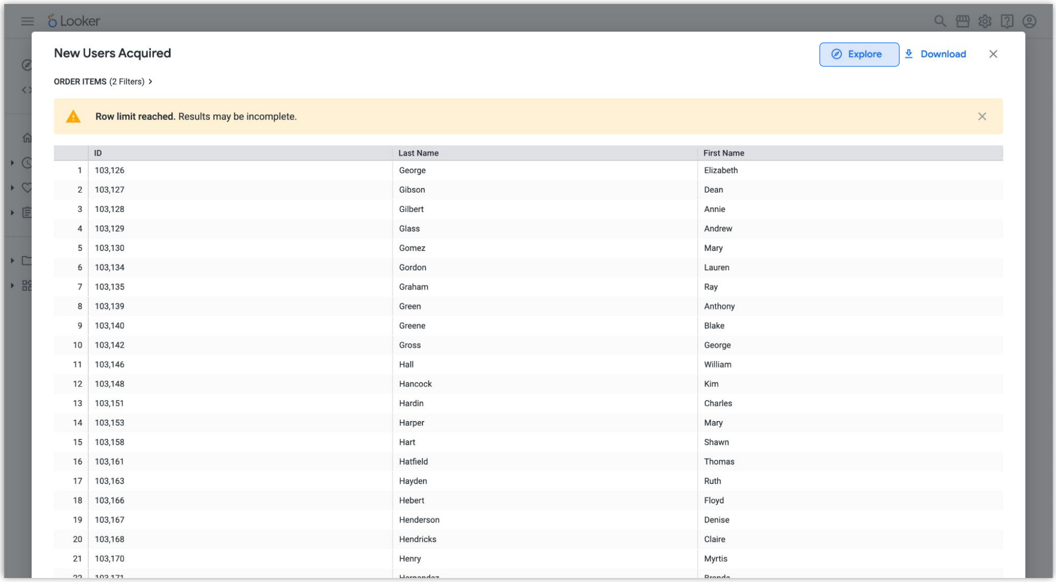
	ID	Last Name	First Name
1	103,126	George	Elizabeth
2	103,127	Gibson	Dean
3	103,128	Gilbert	Annie
4	103,129	Glass	Andrew
5	103,130	Gomez	Mary
6	103,134	Gordon	Lauren
7	103,135	Graham	Ray
8	103,139	Green	Anthony
9	103,140	Greene	Blake
10	103,142	Gross	George
11	103,146	Hall	William
12	103,148	Hancock	Kim
13	103,151	Hardin	Charles
14	103,153	Harper	Mary
15	103,158	Hart	Shawn
16	103,161	Hatfield	Thomas
17	103,163	Hayden	Ruth
18	103,166	Hebert	Floyd
19	103,167	Henderson	Denise
20	103,168	Hendricks	Claire
21	103,170	Henry	Myrtis
22	103,171	Hernandez	Brenda

Examples are based on fictional data.

After drilling down on the New Users Acquired tile, you can view the granular rows of data that constitute the overall number of new users acquired.

For example, for this tile, you can see each user's ID, first name, and last name. In Looker, these data attributes are called dimensions.

Proprietary + Confidential



New Users Acquired

ORDER ITEMS (2 Filters) >

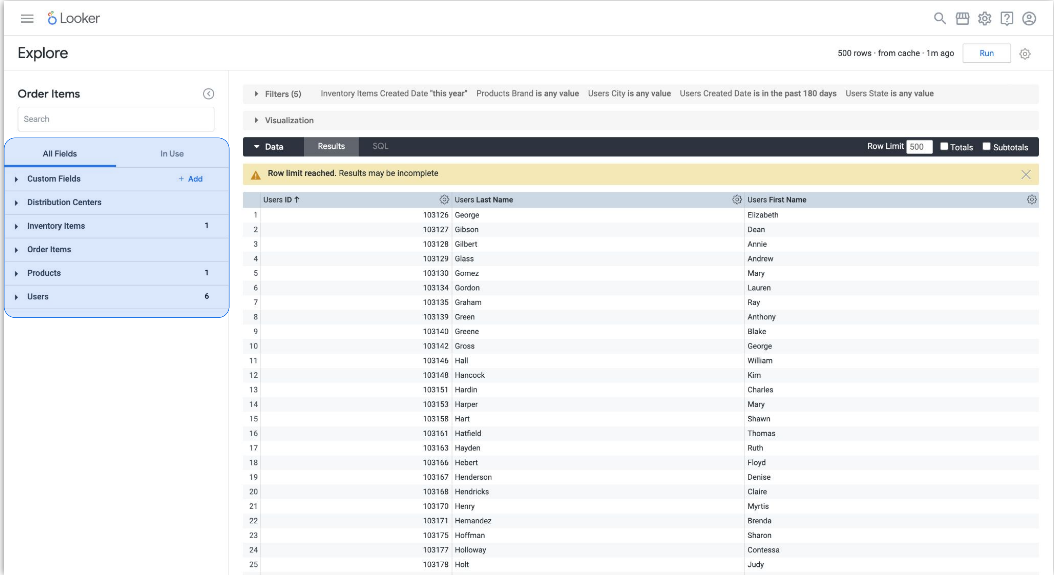
⚠ **Row limit reached.** Results may be incomplete.

	ID	Last Name	First Name
1	103,126	George	Elizabeth
2	103,127	Gibson	Dean
3	103,128	Gilbert	Annie
4	103,129	Glass	Andrew
5	103,130	Gomez	Mary
6	103,134	Gordon	Lauren
7	103,135	Graham	Ray
8	103,139	Green	Anthony
9	103,140	Greene	Blake
10	103,142	Gross	George
11	103,146	Hall	William
12	103,148	Hancock	Kim
13	103,151	Hardin	Charles
14	103,153	Harper	Mary
15	103,158	Hart	Shawn
16	103,161	Hatfield	Thomas
17	103,163	Hayden	Ruth
18	103,166	Hebert	Floyd
19	103,167	Henderson	Denise
20	103,168	Hendricks	Claire
21	103,170	Henry	Myrtis
22	103,171	Hernandez	Brenda

Examples are based on fictional data.

If you want to learn even more about these users, you can click on the **Explore** link.

Proprietary + Confidential



Filters (5) Inventory Items Created Date "this year" Products Brand is any value Users City is any value Users Created Date is in the past 180 days Users State is any value

Row limit reached. Results may be incomplete

	Users ID ↑	Users Last Name	Users First Name
1	103126	George	Elizabeth
2	103127	Gibson	Dean
3	103128	Gilbert	Annie
4	103129	Glass	Andrew
5	103130	Gomez	Mary
6	103134	Gordon	Lauren
7	103135	Graham	Ray
8	103139	Green	Anthony
9	103140	Greene	Blake
10	103142	Gross	George
11	103146	Hall	William
12	103148	Hancock	Kim
13	103151	Hardin	Charles
14	103153	Harper	Mary
15	103158	Hart	Shawn
16	103161	Hatfield	Thomas
17	103163	Hayden	Ruth
18	103166	Hebert	Floyd
19	103167	Henderson	Denise
20	103168	Hendricks	Claire
21	103170	Henry	Myrtis
22	103171	Hernandez	Brenda
23	103175	Hoffman	Sharon
24	103177	Holloway	Contessa
25	103178	Holt	Judy

Examples are based on fictional data.

Google Cloud

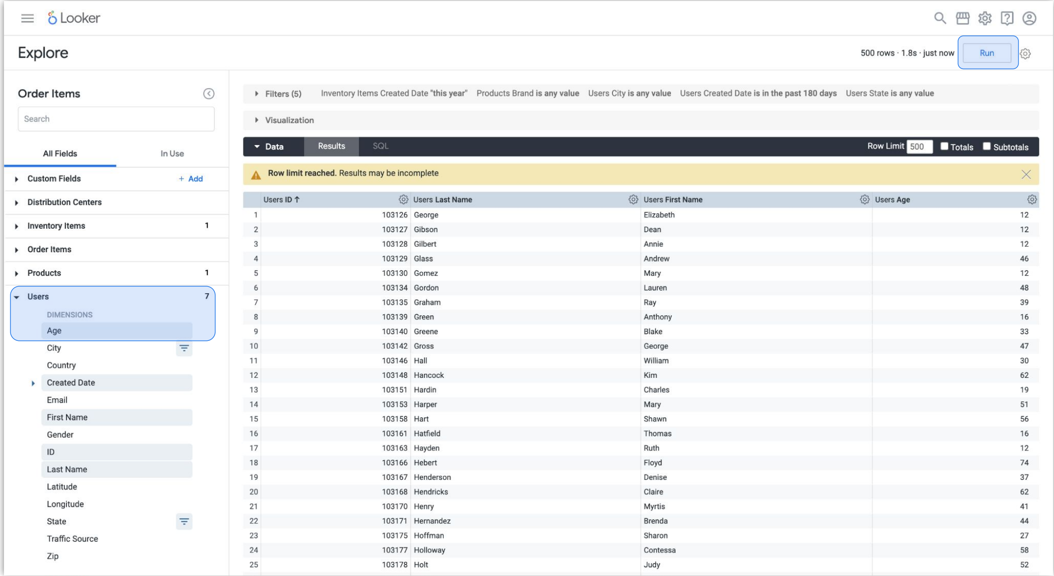
This takes you out of the dashboard and into the Explore, which is the report-builder interface that has been curated by a LookML developer.

In this example, the overall Explore is called **Order Items**, but there are several expandable groups of fields in the field picker found in the left-side panel. These are called views.

An Explore is composed of one or more views. For example, when you want to analyze order information, you also might want to include fields from other related views such as **Users**.

In SQL terms, each view represents a database table, and the tables are pre-joined in the LookML model file to define the overall Explore, like this **Order Items** Explore.

Proprietary + Confidential



500 rows · 1.8s · just now

Filters (5) Inventory Items Created Date "this year" Products Brand is any value Users City is any value Users Created Date is in the past 180 days Users State is any value

Row Limit 500 □ Totals □ Subtotals

Row limit reached. Results may be incomplete

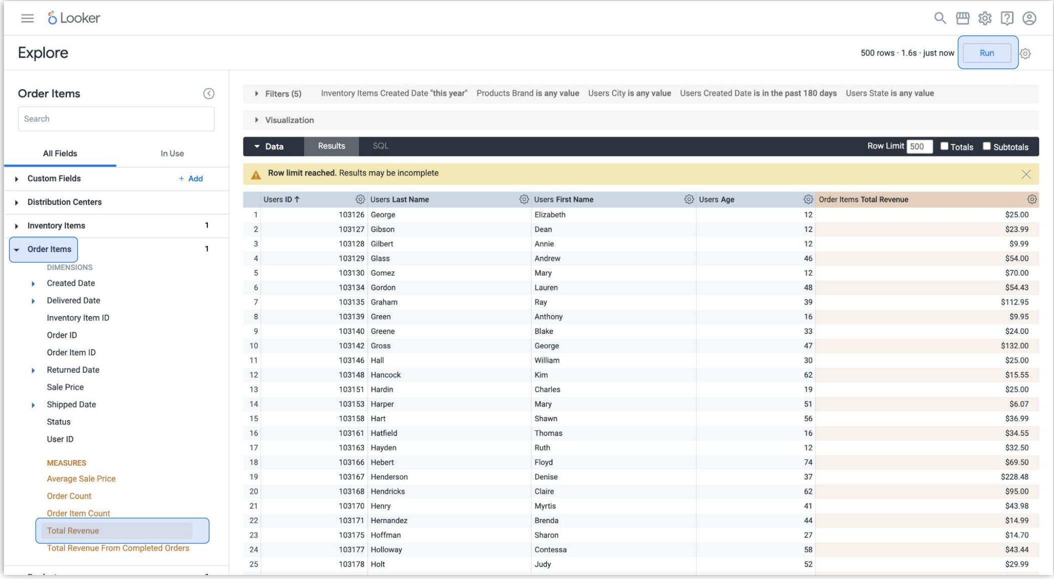
	Users ID ↑	Users Last Name	Users First Name	Users Age
1	103126	George	Elizabeth	12
2	103127	Gibson	Dean	12
3	103128	Gilbert	Annie	12
4	103129	Glass	Andrew	46
5	103130	Gomez	Mary	12
6	103134	Gordon	Lauren	48
7	103135	Graham	Ray	39
8	103139	Green	Anthony	16
9	103140	Greene	Blake	33
10	103142	Gross	George	47
11	103146	Hall	William	30
12	103148	Hancock	Kim	62
13	103151	Hardin	Charles	19
14	103153	Harper	Mary	51
15	103158	Hart	Shawn	56
16	103161	Hatfield	Thomas	16
17	103163	Hayden	Ruth	12
18	103166	Hebert	Floyd	74
19	103167	Henderson	Denise	37
20	103168	Hendricks	Claire	62
21	103170	Henry	Myrtis	41
22	103171	Hernandez	Brenda	44
23	103175	Hoffman	Sharon	27
24	103177	Holloway	Contessa	58
25	103178	Holt	Judy	52

Examples are based on fictional data.

Using this Explore, you can ask questions about the newly acquired users. For example, maybe you are curious about the age of each of these users.

To see age information about these users, you can expand the **Users** view, and click on the **Age** dimension to add it to your results set. Then, you can click **Run** to view the results.

Proprietary + Confidential



Filters (5) Inventory Items Created Date "this year" Products Brand **is any value** Users City **is any value** Users Created Date **is in the past 180 days** Users State **is any value**

Row Limit 500 □ Totals □ Subtotals

Row limit reached. Results may be incomplete

	Users **ID** ↑	Users **Last Name**	Users **First Name**	Users **Age**	Order Items **Total Revenue**
1	103126	George	Elizabeth	12	$25.00
2	103127	Gibson	Dean	12	$23.99
3	103128	Gilbert	Annie	12	$9.99
4	103129	Glass	Andrew	46	$54.00
5	103130	Gomez	Mary	12	$70.00
6	103134	Gordon	Lauren	48	$54.43
7	103135	Graham	Ray	39	$112.95
8	103139	Green	Anthony	16	$9.95
9	103140	Greene	Blake	33	$24.00
10	103142	Gross	George	47	$132.00
11	103146	Hall	William	30	$25.00
12	103148	Hancock	Kim	62	$15.55
13	103151	Hardin	Charles	19	$25.00
14	103153	Harper	Mary	51	$6.07
15	103158	Hart	Shawn	56	$36.99
16	103161	Hatfield	Thomas	16	$34.55
17	103163	Hayden	Ruth	12	$32.50
18	103166	Hebert	Floyd	74	$69.50
19	103167	Henderson	Denise	37	$228.48
20	103168	Hendricks	Claire	62	$95.00
21	103170	Henry	Myrtis	41	$43.98
22	103171	Hernandez	Brenda	44	$14.99
23	103175	Hoffman	Sharon	27	$14.70
24	103177	Holloway	Contessa	58	$43.44
25	103178	Holt	Judy	52	$29.99

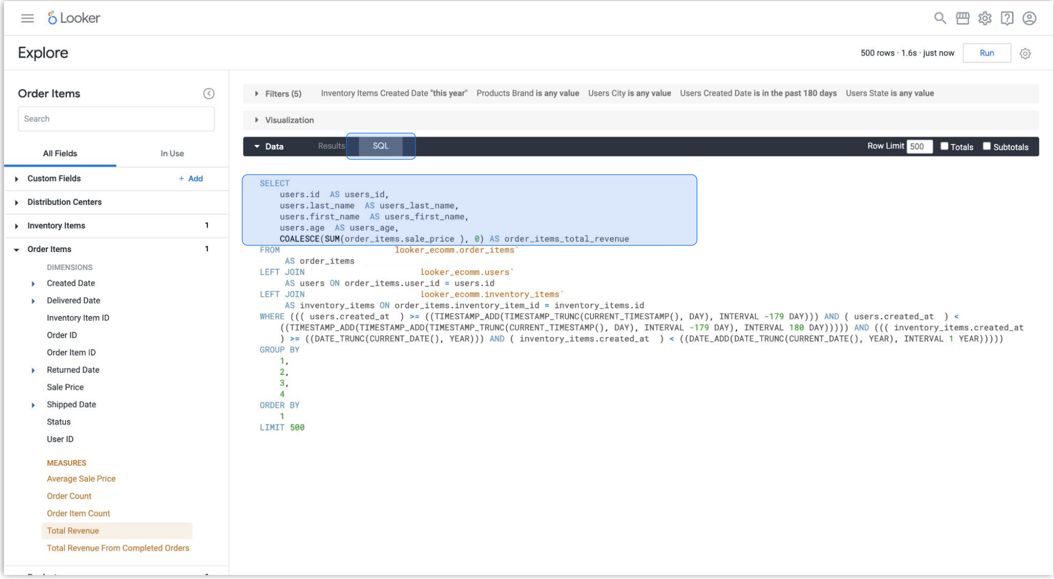
Examples are based on fictional data.

Google Cloud

If you want to know how much each of these new users has already spent, you can expand the **Order Items** view, and click on the **Total Revenue** measure, which aggregates all of the purchases for each user.

Once again, you need to click **Run** to view the new results.

Proprietary + Confidential



Filters (5) Inventory Items Created Date "this year" Products Brand is any value Users City is any value Users Created Date is in the past 180 days Users State is any value

Row Limit 500 □ Totals □ Subtotals

```sql
SELECT
    users.id  AS users_id,
    users.last_name  AS users_last_name,
    users.first_name  AS users_first_name,
    users.age  AS users_age,
    COALESCE(SUM(order_items.sale_price ), 0) AS order_items_total_revenue
FROM `                 looker_ecomm.order_items`
     AS order_items
LEFT JOIN `                 looker_ecomm.users`
     AS users ON order_items.user_id = users.id
LEFT JOIN `                 looker_ecomm.inventory_items`
     AS inventory_items ON order_items.inventory_item_id = inventory_items.id
WHERE ((( users.created_at  ) >= ((TIMESTAMP_ADD(TIMESTAMP_TRUNC(CURRENT_TIMESTAMP(), DAY), INTERVAL -179 DAY))) AND ( users.created_at  ) <
    ((TIMESTAMP_ADD(TIMESTAMP_ADD(TIMESTAMP_TRUNC(CURRENT_TIMESTAMP(), DAY), INTERVAL -179 DAY), INTERVAL 180 DAY))))) AND ((( inventory_items.created_at
    ) >= ((DATE_TRUNC(CURRENT_DATE(), YEAR))) AND ( inventory_items.created_at  ) < ((DATE_ADD(DATE_TRUNC(CURRENT_DATE(), YEAR), INTERVAL 1 YEAR)))))
GROUP BY
    1,
    2,
    3,
    4
ORDER BY
    1
LIMIT 500
```

Examples are based on fictional data.

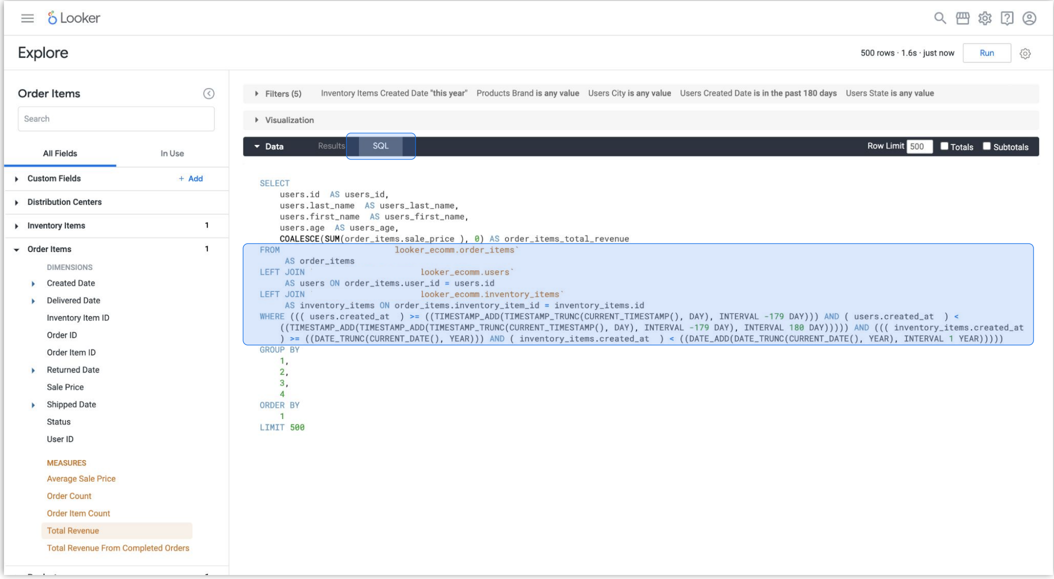
Google Cloud

Behind the scenes, Looker has automatically generated a SQL query for the drill-down results and is updating it with the new selections.

As a LookML developer, you can click on the SQL tab—typically hidden from business users—to see that Looker is **SELECT**ing **users.id**, **users.first_name**, **users.last_name**, and **users.age**, which are the dimensions selected in the Explore.

Looker is also executing a **SUM** of **order_items.sale_price** for **total_revenue,** which is the measure selected to aggregate the purchase data for each user.

Proprietary + Confidential



Filters (5) Inventory Items Created Date "this year" Products Brand is any value Users City is any value Users Created Date is in the past 180 days Users State is any value

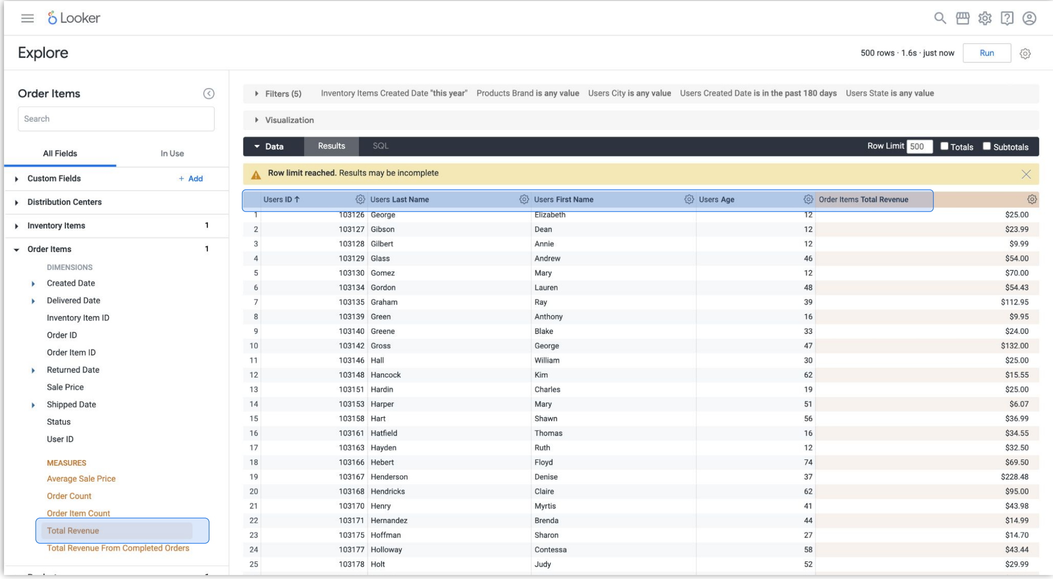
```sql
SELECT
    users.id  AS users_id,
    users.last_name  AS users_last_name,
    users.first_name  AS users_first_name,
    users.age  AS users_age,
    COALESCE(SUM(order_items.sale_price ), 0) AS order_items_total_revenue
FROM                    looker_ecomm.order_items`
     AS order_items
LEFT JOIN `                    looker_ecomm.users`
     AS users ON order_items.user_id = users.id
LEFT JOIN `                    looker_ecomm.inventory_items`
     AS inventory_items ON order_items.inventory_item_id = inventory_items.id
WHERE ((( users.created_at  ) >= ((TIMESTAMP_ADD(TIMESTAMP_TRUNC(CURRENT_TIMESTAMP(), DAY), INTERVAL -179 DAY))) AND ( users.created_at  ) <
    ((TIMESTAMP_ADD(TIMESTAMP_ADD(TIMESTAMP_TRUNC(CURRENT_TIMESTAMP(), DAY), INTERVAL -179 DAY), INTERVAL 180 DAY))))) AND ((( inventory_items.created_at
    ) >= ((DATE_TRUNC(CURRENT_DATE(), YEAR))) AND ( inventory_items.created_at  ) < ((DATE_ADD(DATE_TRUNC(CURRENT_DATE(), YEAR), INTERVAL 1 YEAR)))))
GROUP BY
    1,
    2,
    3,
    4
ORDER BY
    1
LIMIT 500
```

Examples are based on fictional data.

You can also see that the data comes **FROM** a table called **looker_ecomm.order_items**, with a **LEFT JOIN** to **looker_ecomm.users**.

Finally, there's a big **WHERE** condition that is identifying only the new users from the past 180 days.

Proprietary + Confidential



Inventory Items Created Date "this year" Products Brand is any value Users City is any value Users Created Date is in the past 180 days Users State is any value

Row limit reached. Results may be incomplete

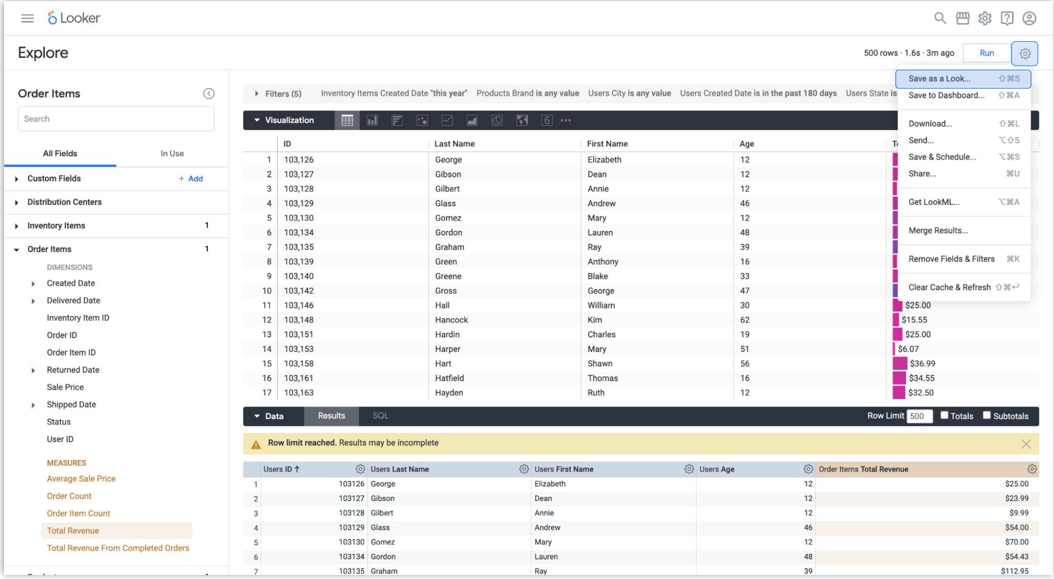
	Users ID ↑	Users Last Name	Users First Name	Users Age	Order Items Total Revenue
1	103126	George	Elizabeth	12	$25.00
2	103127	Gibson	Dean	12	$23.99
3	103128	Gilbert	Annie	12	$9.99
4	103129	Glass	Andrew	46	$54.00
5	103130	Gomez	Mary	12	$70.00
6	103134	Gordon	Lauren	48	$54.43
7	103135	Graham	Ray	39	$112.95
8	103139	Green	Anthony	16	$9.95
9	103140	Greene	Blake	33	$24.00
10	103142	Gross	George	47	$132.00
11	103146	Hall	William	30	$25.00
12	103148	Hancock	Kim	62	$15.55
13	103151	Hardin	Charles	19	$25.00
14	103153	Harper	Mary	51	$6.07
15	103158	Hart	Shawn	56	$36.99
16	103161	Hatfield	Thomas	16	$34.55
17	103163	Hayden	Ruth	12	$32.50
18	103166	Hebert	Floyd	74	$69.50
19	103167	Henderson	Denise	37	$228.48
20	103168	Hendricks	Claire	62	$95.00
21	103170	Henry	Myrtis	41	$43.98
22	103171	Hernandez	Brenda	44	$14.99
23	103175	Hoffman	Sharon	27	$14.70
24	103177	Holloway	Contessa	58	$43.44
25	103178	Holt	Judy	52	$29.99

Examples are based on fictional data.

Google Cloud

Rather than writing this SQL query manually, it was so easy for the business user to click on some fields and get this output automatically. This is all because of the agile modeling layer in LookML.

A LookML developer or team of developers had already created this **Order Items** Explore, specified which views should be in the Explore, specified which dimensions and measures should be in each view, and defined the SQL logic for each dimension and measure.



500 rows · 1.6s · 3m ago

Filters (5) Inventory Items Created Date "this year" Products Brand is any value Users City is any value Users Created Date is in the past 180 days Users State is

	ID	Last Name	First Name	Age	T
1	103,126	George	Elizabeth	12	
2	103,127	Gibson	Dean	12	
3	103,128	Gilbert	Annie	12	
4	103,129	Glass	Andrew	46	
5	103,130	Gomez	Mary	12	
6	103,134	Gordon	Lauren	48	
7	103,135	Graham	Ray	39	
8	103,139	Green	Anthony	16	
9	103,140	Greene	Blake	33	
10	103,142	Gross	George	47	
11	103,146	Hall	William	30	$25.00
12	103,148	Hancock	Kim	62	$15.55
13	103,151	Hardin	Charles	19	$25.00
14	103,153	Harper	Mary	51	$6.07
15	103,158	Hart	Shawn	56	$36.99
16	103,161	Hatfield	Thomas	16	$34.55
17	103,163	Hayden	Ruth	12	$32.50

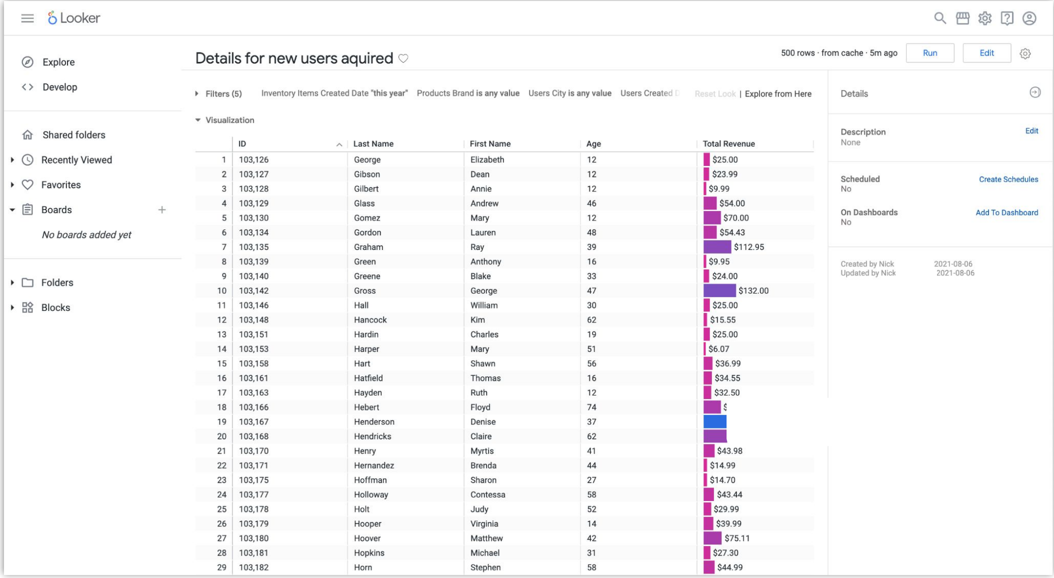
Row limit reached. Results may be incomplete

	Users ID ↑	Users **Last Name**	Users **First Name**	Users **Age**	Order Items **Total Revenue**
1	103126	George	Elizabeth	12	$25.00
2	103127	Gibson	Dean	12	$23.99
3	103128	Gilbert	Annie	12	$9.99
4	103129	Glass	Andrew	46	$54.00
5	103130	Gomez	Mary	12	$70.00
6	103134	Gordon	Lauren	48	$54.43
7	103135	Graham	Ray	39	$112.95

Examples are based on fictional data.

Now, you can save these results as a standalone report, or Look, for ongoing reference by clicking on the gear icon in the top right, and selecting **Save as a Look**.

Proprietary + Confidential



Details for new users aquired

500 rows · from cache · 5m ago

Filters (5) Inventory Items Created Date "this year" Products Brand is any value Users City is any value Users Created D

	ID	Last Name	First Name	Age	Total Revenue
1	103,126	George	Elizabeth	12	$25.00
2	103,127	Gibson	Dean	12	$23.99
3	103,128	Gilbert	Annie	12	$9.99
4	103,129	Glass	Andrew	46	$54.00
5	103,130	Gomez	Mary	12	$70.00
6	103,134	Gordon	Lauren	48	$54.43
7	103,135	Graham	Ray	39	$112.95
8	103,139	Green	Anthony	16	$9.95
9	103,140	Greene	Blake	33	$24.00
10	103,142	Gross	George	47	$132.00
11	103,146	Hall	William	30	$25.00
12	103,148	Hancock	Kim	62	$15.55
13	103,151	Hardin	Charles	19	$25.00
14	103,153	Harper	Mary	51	$6.07
15	103,158	Hart	Shawn	56	$36.99
16	103,161	Hatfield	Thomas	16	$34.55
17	103,163	Hayden	Ruth	12	$32.50
18	103,166	Hebert	Floyd	74	$
19	103,167	Henderson	Denise	37	
20	103,168	Hendricks	Claire	62	
21	103,170	Henry	Myrtis	41	$43.98
22	103,171	Hernandez	Brenda	44	$14.99
23	103,175	Hoffman	Sharon	27	$14.70
24	103,177	Holloway	Contessa	58	$43.44
25	103,178	Holt	Judy	52	$29.99
26	103,179	Hooper	Virginia	14	$39.99
27	103,180	Hoover	Matthew	42	$75.11
28	103,181	Hopkins	Michael	31	$27.30
29	103,182	Horn	Stephen	58	$44.99

Details

Description
None

Scheduled
No

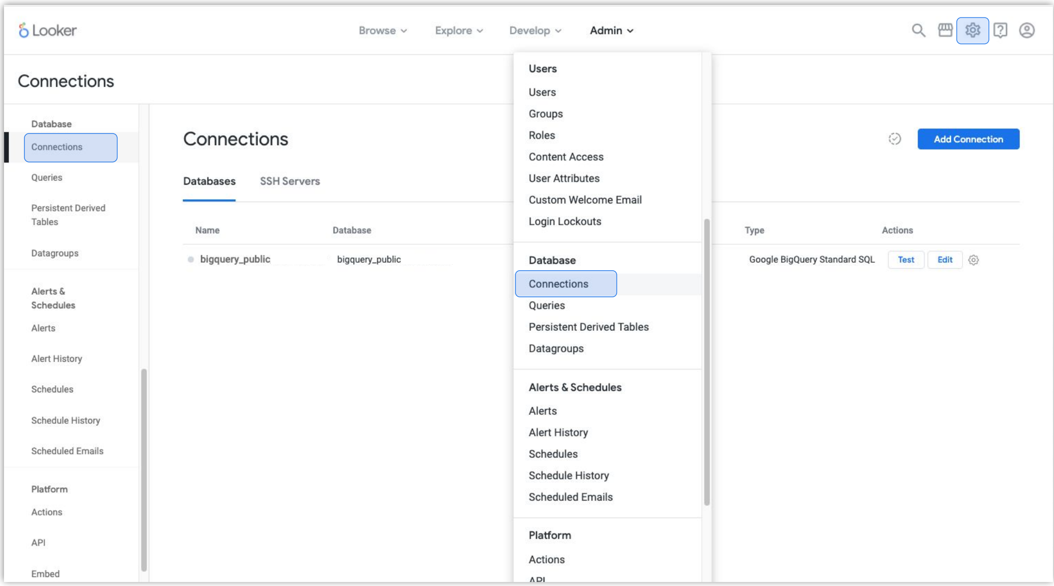
On Dashboards
No

Created by Nick 2021-08-06
Updated by Nick 2021-08-06

Examples are based on fictional data.

That was a quick demonstration of the significance and value of LookML for empowering business users to explore and analyze data.

Proprietary + Confidential



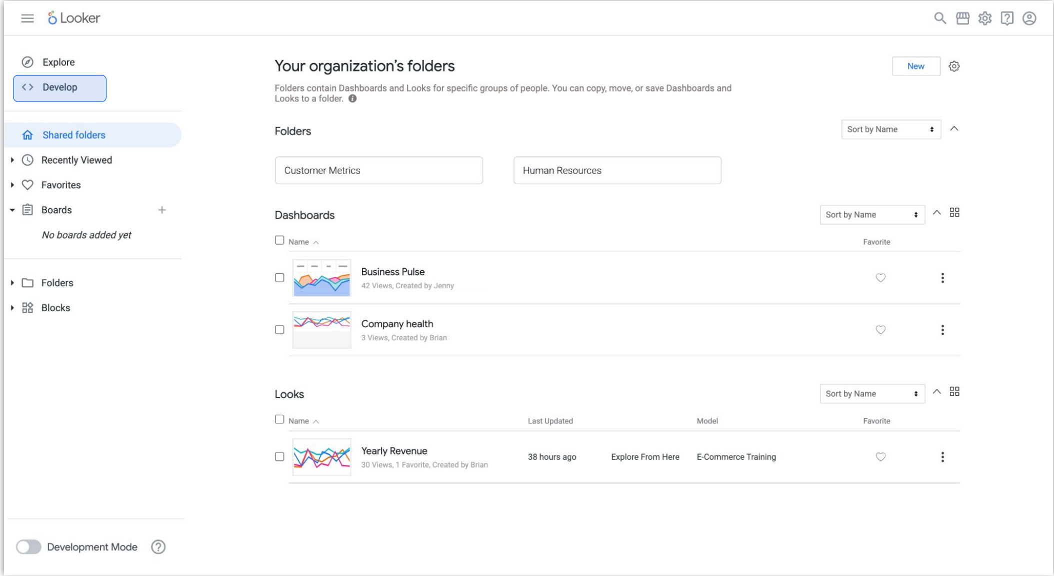
Name	Database	Type	Actions
bigquery_public	bigquery_public	Google BigQuery Standard SQL	Test Edit

Examples are based on fictional data.

If you’re completely new to Looker, there’s a fundamental prerequisite to be aware of regarding the underlying data that is used by business users.

A Looker administrator needs to create a connection to a specific data source such as your company’s primary database, in order for LookML developers to access and model that data to curate Explores for business users.

Admins can find setup and configuration details in the Database configuration details section of Looker’s dialects documentation page.



Your organization's folders

Folders contain Dashboards and Looks for specific groups of people. You can copy, move, or save Dashboards and Looks to a folder.

Folders

Customer Metrics

Human Resources

Dashboards

Name	Favorite
Business Pulse 42 Views, Created by Jenny	
Company health 3 Views, Created by Brian	

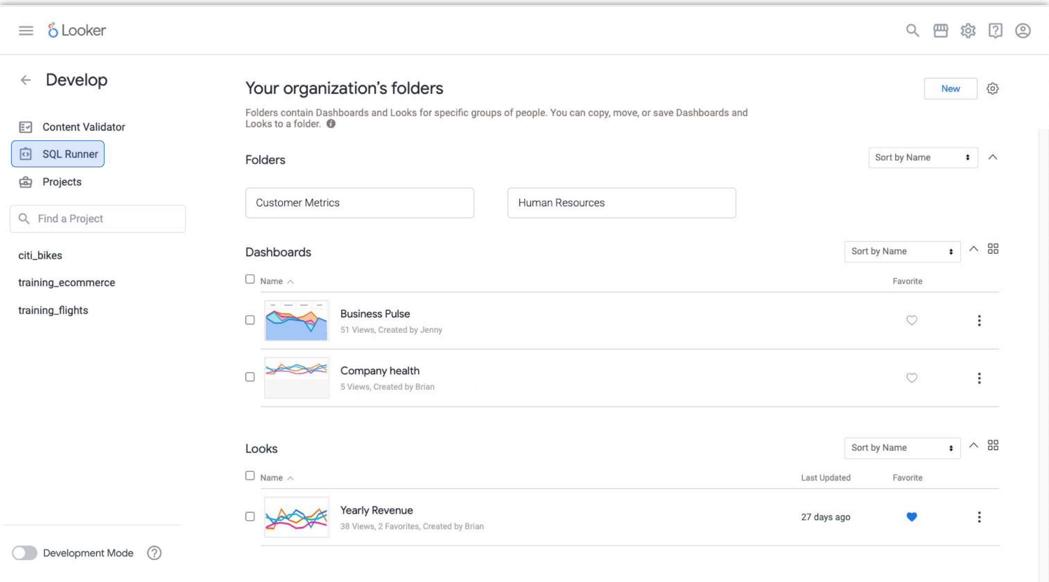
Looks

Name	Last Updated		Model	Favorite
Yearly Revenue 30 Views, 1 Favorite, Created by Brian	38 hours ago	Explore From Here	E-Commerce Training	

Examples are based on fictional data.

As a LookML developer, you can see available database connections in your organization's Looker instance by clicking on the **Develop** tab in left-side navigation panel of the Looker UI..

Proprietary + Confidential



Your organization's folders

Folders contain Dashboards and Looks for specific groups of people. You can copy, move, or save Dashboards and Looks to a folder.

Folders

Customer Metrics

Human Resources

Dashboards

Name	Favorite
Business Pulse 51 Views, Created by Jenny	♡
Company health 5 Views, Created by Brian	♡

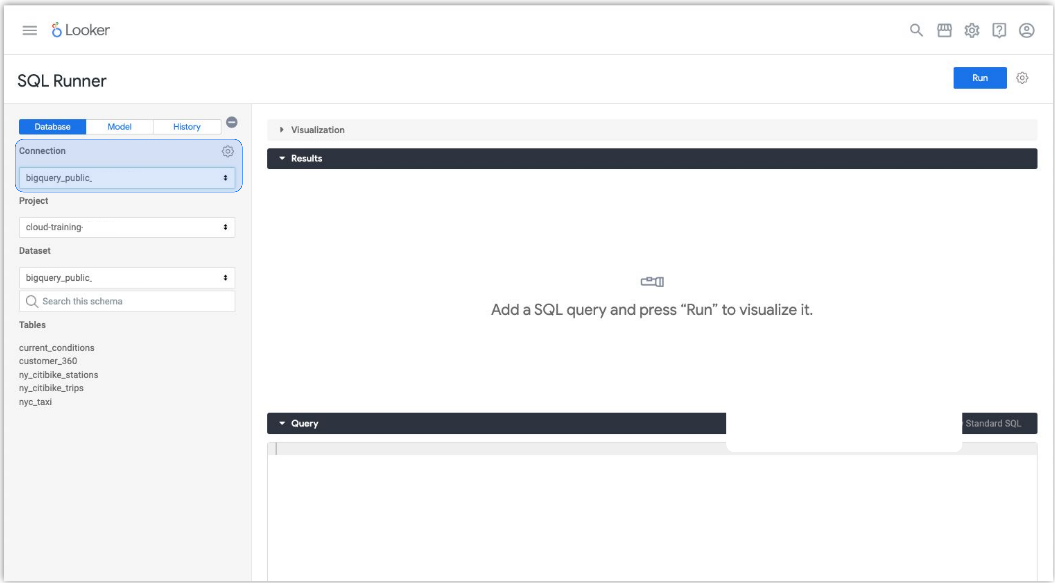
Looks

Name	Last Updated	Favorite
Yearly Revenue 38 Views, 2 Favorites, Created by Brian	27 days ago	♥

Examples are based on fictional data.

... then, selecting **SQL Runner**.

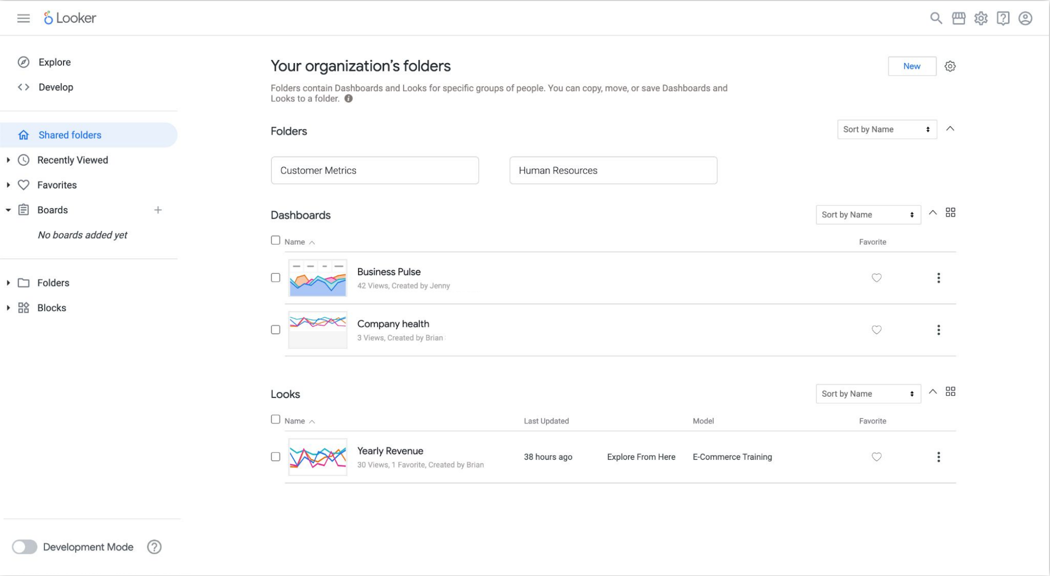
Proprietary + Confidential



Examples are based on fictional data.

SQL Runner is a console or portal to the databases that have been connected to your organization’s Looker instance. You can see all of the database connection options under **Connection**.

Proprietary + Confidential



Your organization's folders

Folders contain Dashboards and Looks for specific groups of people. You can copy, move, or save Dashboards and Looks to a folder.

Folders

Customer Metrics

Human Resources

Dashboards

Name	Favorite
Business Pulse 42 Views, Created by Jenny	
Company health 3 Views, Created by Brian	

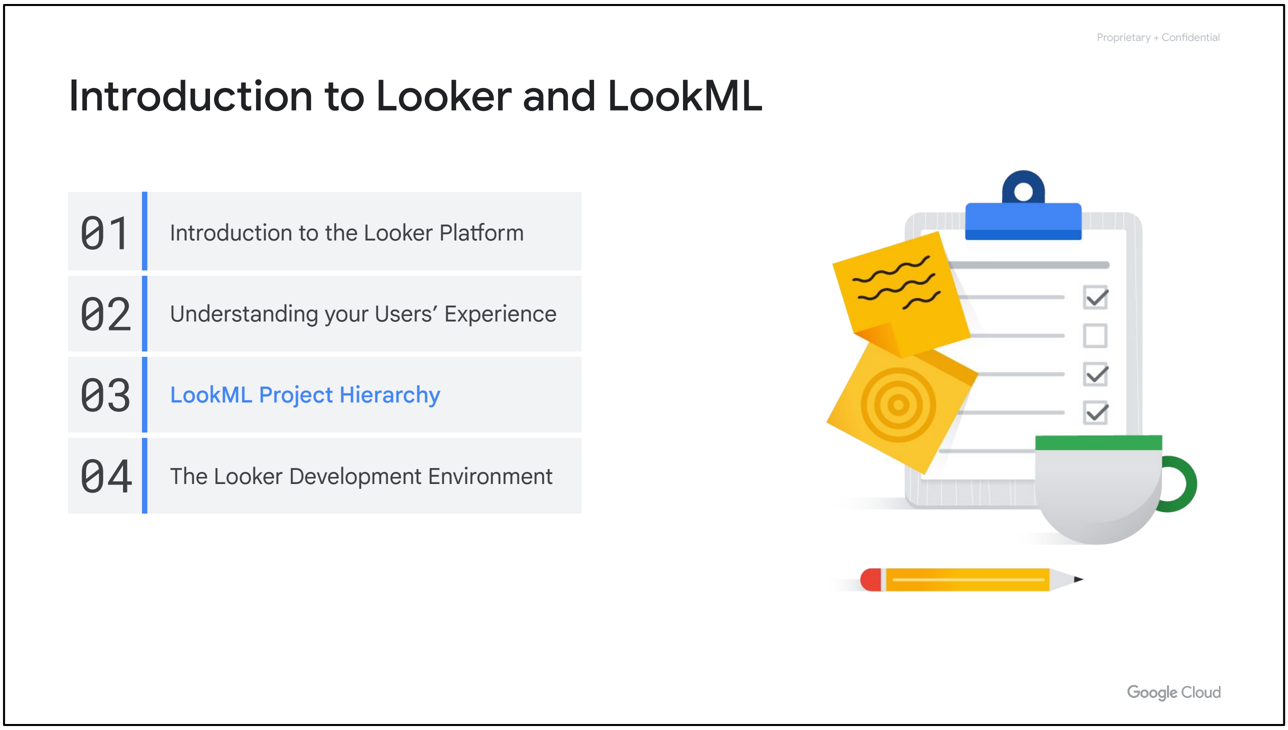
Looks

Name	Last Updated		Model	Favorite
Yearly Revenue 30 Views, 1 Favorite, Created by Brian	38 hours ago	Explore From Here	E-Commerce Training	

Examples are based on fictional data.

Google Cloud

Now, after this brief overview of the Looker business user experience, we hope that you are excited to explore Looker further as a LookML developer.



Proprietary + Confidential

Introduction to Looker and LookML

01 Introduction to the Looker Platform

02 Understanding your Users’ Experience

03 LookML Project Hierarchy

04 The Looker Development Environment

To prepare yourself for writing LookML code, there are some key terms and overall project structure that you need to know.

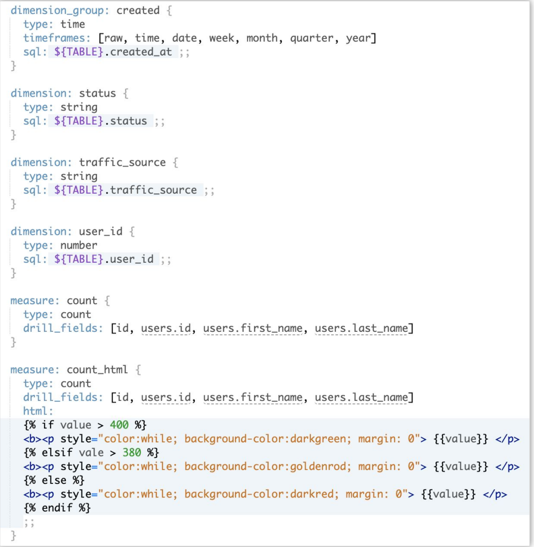
Let’s begin with an easy one - what does the “ML” in LookML stand for?

Proprietary + Confidential

LookML

LookML (Looker Modeling Language) is an abstraction layer for SQL used by developers to tell Looker how to interpret data.

Looker uses the LookML written by developers to generate abstracted SQL queries to the connected database.



```
dimension_group: created {
  type: time
  timeframes: [raw, time, date, week, month, quarter, year]
  sql: ${TABLE}.created_at ;;
}

dimension: status {
  type: string
  sql: ${TABLE}.status ;;
}

dimension: traffic_source {
  type: string
  sql: ${TABLE}.traffic_source ;;
}

dimension: user_id {
  type: number
  sql: ${TABLE}.user_id ;;
}

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

measure: count_html {
  type: count
  drill_fields: [id, users.id, users.first_name, users.last_name]
  html:
  {% if value > 400 %}
  <b><p style="color:while; background-color:darkgreen; margin: 0"> {{value}} </p>
  {% elsif vale > 380 %}
  <b><p style="color:while; background-color:goldenrod; margin: 0"> {{value}} </p>
  {% else %}
  <b><p style="color:while; background-color:darkred; margin: 0"> {{value}} </p>
  {% endif %}
  ;;
}
```

Google Cloud

LookML stands for Looker Modeling Language. It is Looker's proprietary language that establishes an abstraction layer for SQL. Developers use LookML to tell Looker what data to use from the connected database and how it should interpret that data.

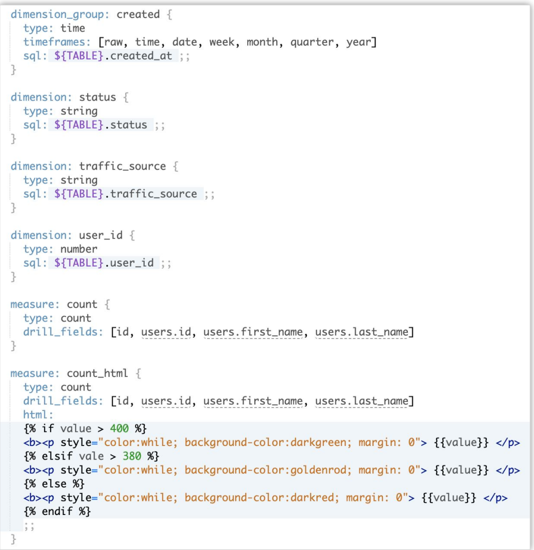
Specifically, LookML acts as the modeling layer between the connected SQL database and your business users. Looker uses the LookML code written by developers to define how business users interact with a connected database and to construct SQL queries against that particular database.

Proprietary + Confidential

LookML

Developers use LookML to define:

- Dimensions (data attributes)
- Measures (aggregations)
- Data relationships (e.g. how to join tables)
- Custom tables and fields

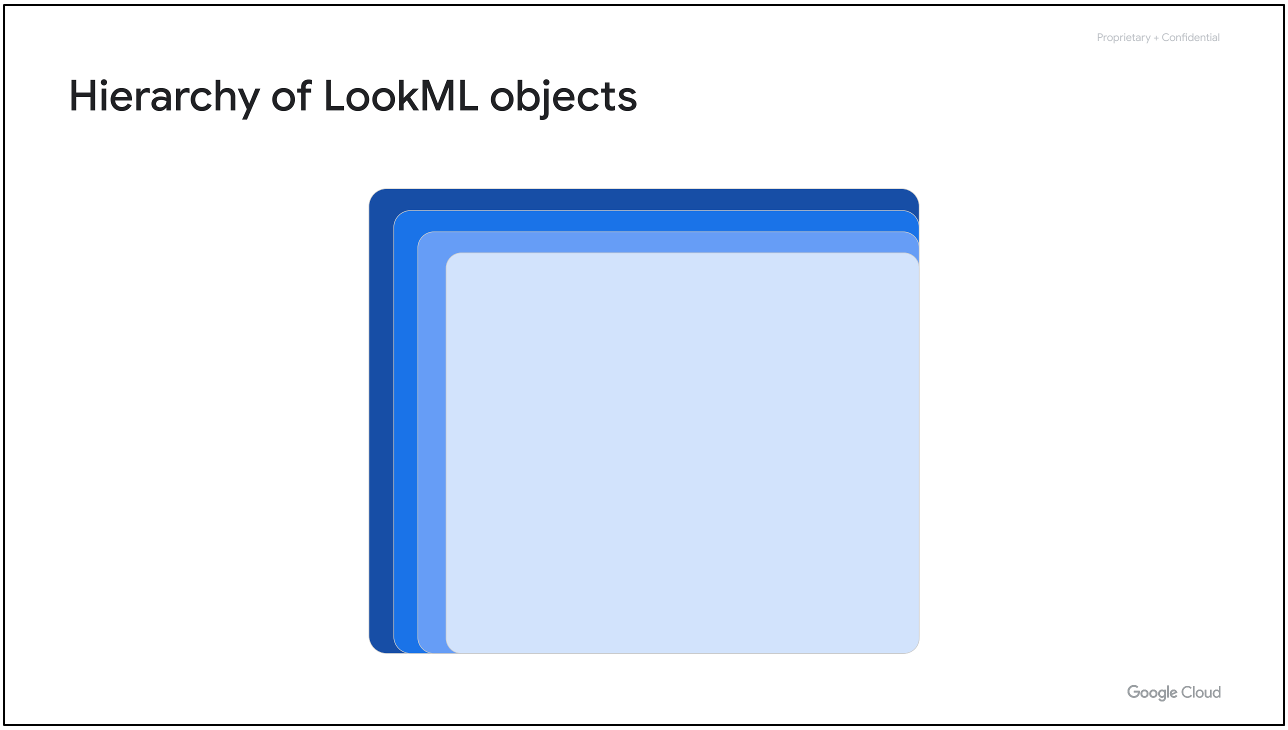


```
dimension_group: created {
  type: time
  timeframes: [raw, time, date, week, month, quarter, year]
  sql: ${TABLE}.created_at ;;
}

dimension: status {
  type: string
  sql: ${TABLE}.status ;;
}

dimension: traffic_source {
  type: string
  sql: ${TABLE}.traffic_source ;;
}

dimension: user_id {
  type: number
  sql: ${TABLE}.user_id ;;
}

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

measure: count_html {
  type: count
  drill_fields: [id, users.id, users.first_name, users.last_name]
  html:
  {% if value > 400 %}
  <b><p style="color:while; background-color:darkgreen; margin: 0"> {{value}} </p>
  {% elsif vale > 380 %}
  <b><p style="color:while; background-color:goldenrod; margin: 0"> {{value}} </p>
  {% else %}
  <b><p style="color:while; background-color:darkred; margin: 0"> {{value}} </p>
  {% endif %}
  ;;
}
```

Google Cloud

Developers use LookML to define many items from the connected SQL database including data attributes called dimensions, aggregates of dimensions called measures, data relationships such as how to join tables, and custom tables and fields.

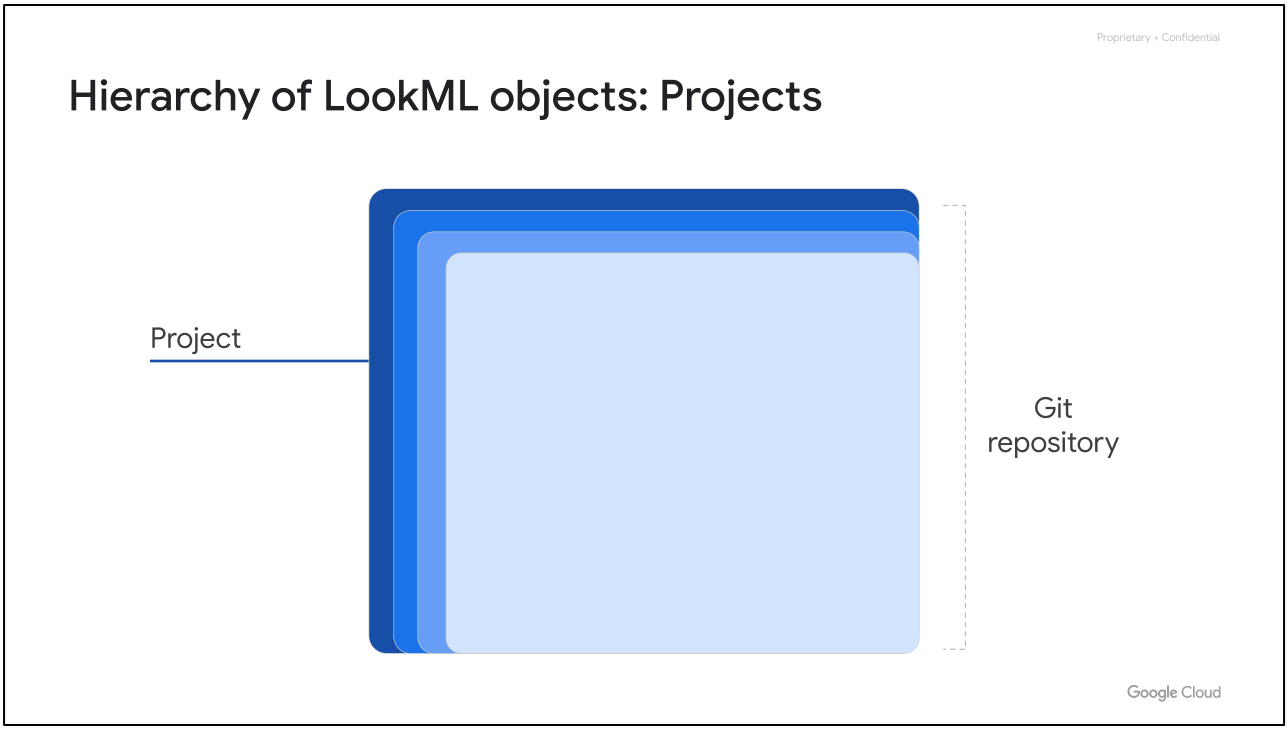
A key concept of LookML to remember is: If it's possible in your SQL dialect, it should be possible in Looker. If you can go to your database console and hand-write a **SELECT** statement that does something in the database, you can also code LookML that Looker can use to accomplish the same task.



Proprietary + Confidential

Hierarchy of LookML objects

For full comprehension of the key LookML terms, developers need to understand where each object fits into the overall hierarchy of a LookML project.



Proprietary + Confidential

Hierarchy of LookML objects: Projects

Project

Git repository

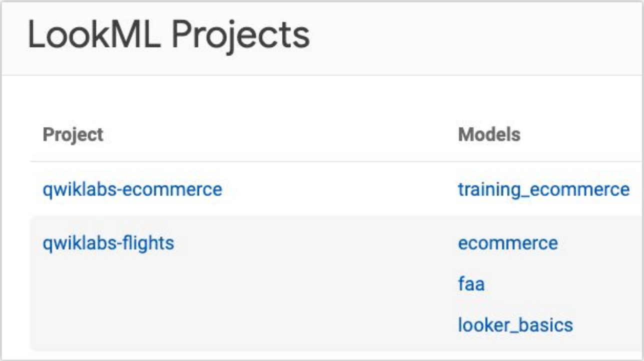
Google Cloud

The highest level is the LookML project itself, which is a library of self-contained LookML code. Because Looker uses Git for version control, a best practice is for each project to map one-to-one with a dedicated Git repository.

Projects

A project is the highest-level LookML object and is essentially a standalone library of code containing one or more data models.

Projects should map 1:1 to a Git repository.

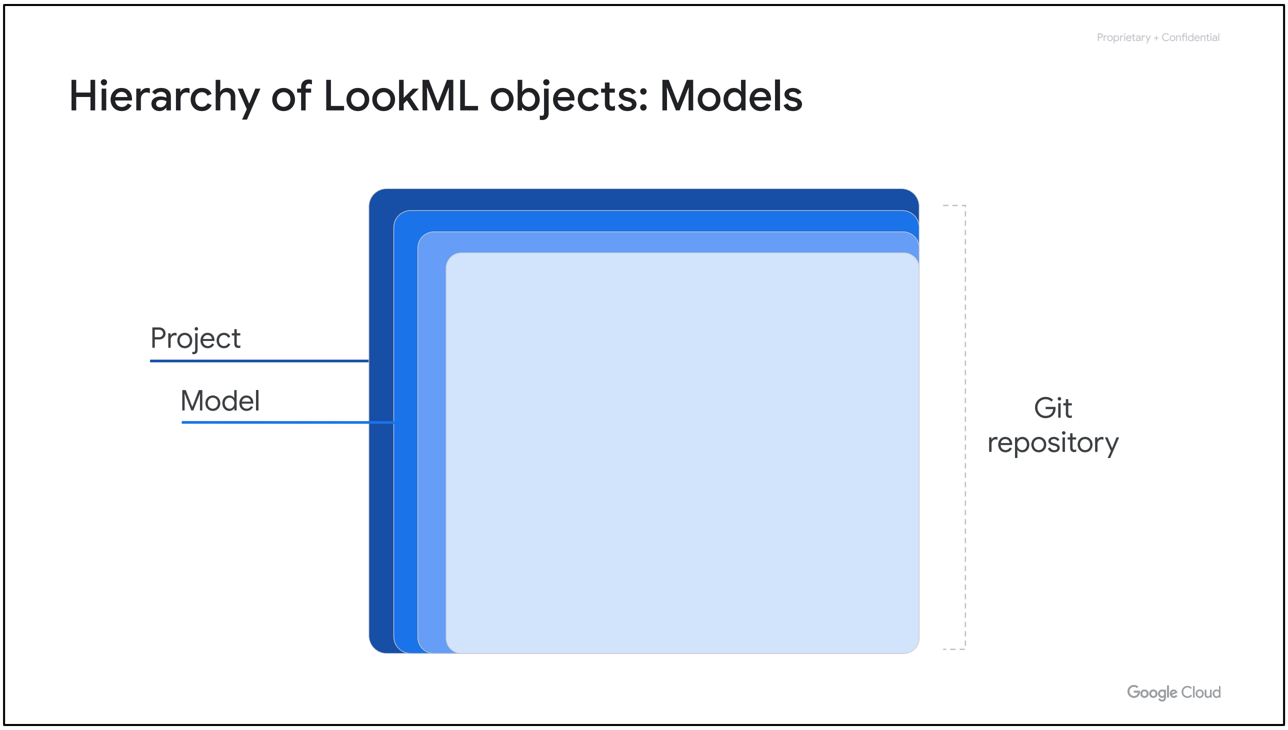


LookML Projects

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

A project is essentially a library of code for a specific data source or database connection and contains one or more data models. You can think of each project as an semi-independent or mini-instance of Looker, and each project should map one-to-one to a Git repository for version control.

Data that cannot be joined together should be separated into different projects because there is no relation to be made across the data.



Proprietary + Confidential

Hierarchy of LookML objects: Models

Project

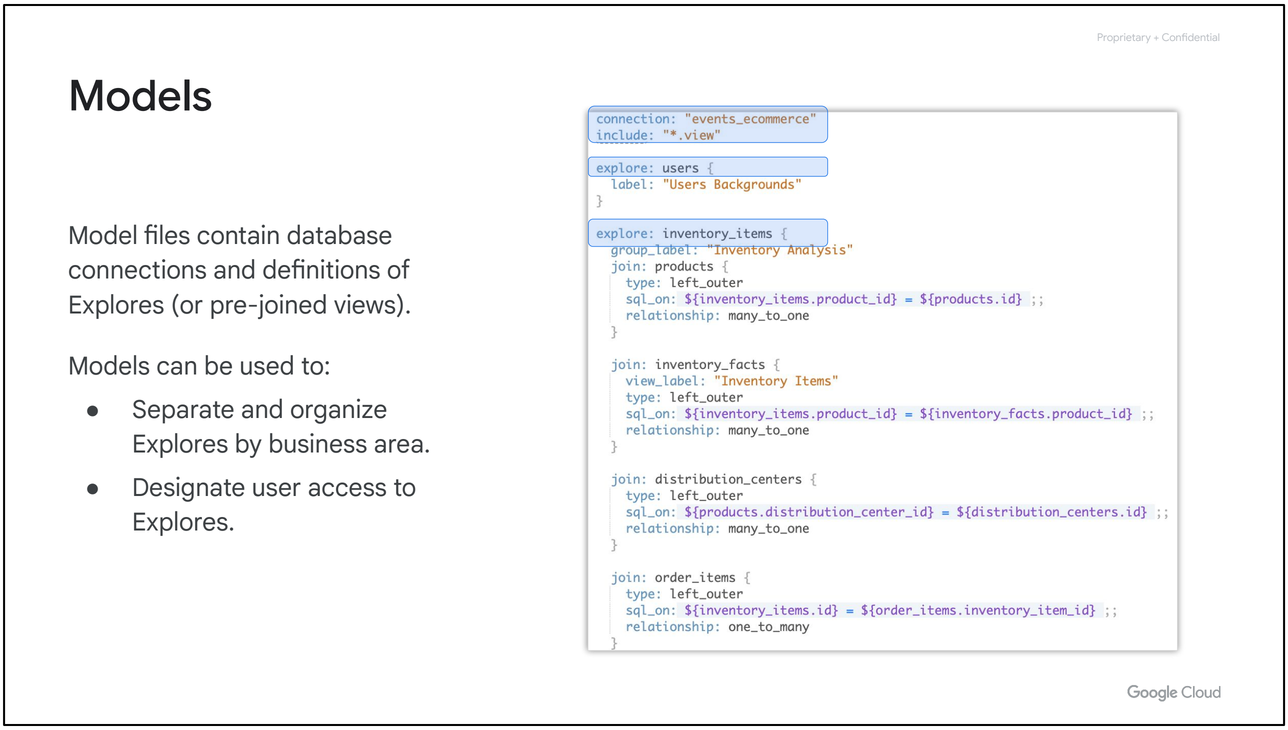
Model

Git repository

Google Cloud

The next level in the hierarchy are models. As previously mentioned, a LookML project is composed of one or more models.

A model specifies a database connection and the data views that utilize that connection.



Proprietary + Confidential

Models

Model files contain database connections and definitions of Explores (or pre-joined views).

Models can be used to:

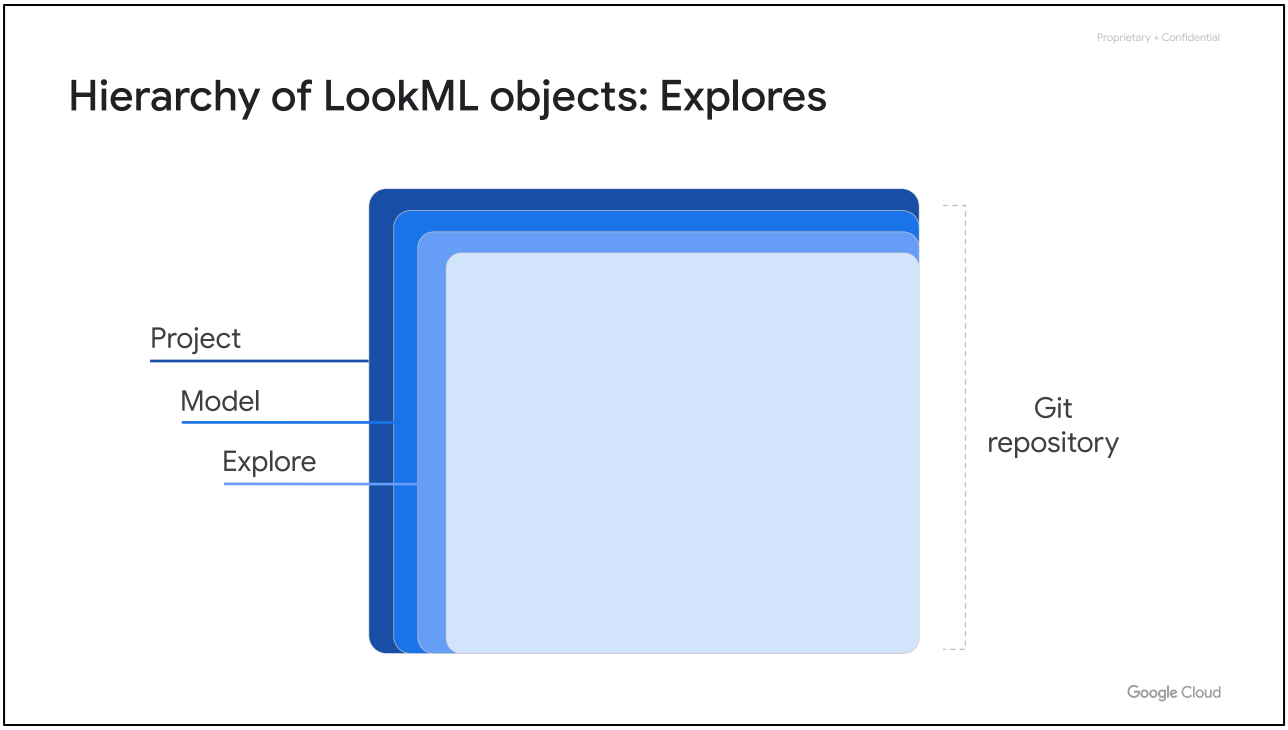
- Separate and organize Explores by business area.
- Designate user access to Explores.

```
connection: "events_ecommerce"
include: "*.view"

explore: users {
  label: "Users Backgrounds"
}

explore: inventory_items {
  group_label: "Inventory Analysis"
  join: products {
    type: left_outer
    sql_on: ${inventory_items.product_id} = ${products.id} ;;
    relationship: many_to_one
  }

  join: inventory_facts {
    view_label: "Inventory Items"
    type: left_outer
    sql_on: ${inventory_items.product_id} = ${inventory_facts.product_id} ;;
    relationship: many_to_one
  }

  join: distribution_centers {
    type: left_outer
    sql_on: ${products.distribution_center_id} = ${distribution_centers.id} ;;
    relationship: many_to_one
  }

  join: order_items {
    type: left_outer
    sql_on: ${inventory_items.id} = ${order_items.inventory_item_id} ;;
    relationship: one_to_many
  }
```

Google Cloud

Specifically, a model file is used to define:

- The database connection.
- The view files that are accessible to this model.
- The Explores (which are pre-joined views) and their join logic.



Proprietary + Confidential

Hierarchy of LookML objects: Explores

Project

Model

Explore

Git repository

Google Cloud

Next on the hierarchy are Explores, which are sets of pre-joined views organized by business area defined within the model files.

For example, you might create a model file containing multiple Explores pertaining to customer purchases.

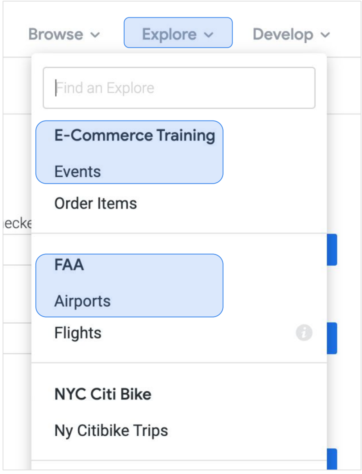
Proprietary + Confidential

Explores

Explores are composed of one or more views joined together to target a specific business question.

Example Explores:

- Events under the E-commerce Training heading
- Airports under the FAA heading

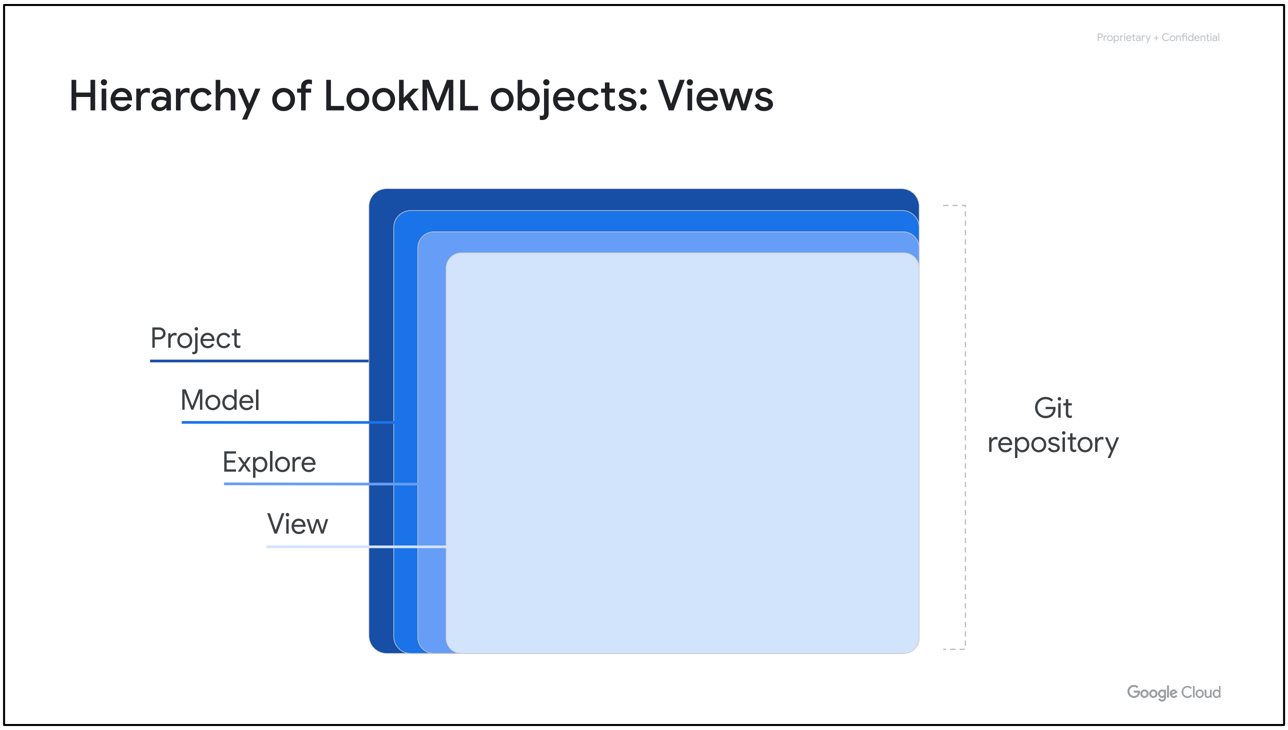


Google Cloud

Explores are the central component of Looker that allow business and data analysts to conduct self-serve data exploration, analysis, and visualization.

Within a model file, developers define Explores that join one or more views together to target specific questions that business users may have. Business users then use the predefined Explores to run queries and create reports and visualizations to answer their questions.

So you can think of an Explore as a predefined set of tables that would frequently be joined for business inquiries and use cases.



Proprietary + Confidential

Hierarchy of LookML objects: Views

Project

Model

Explore

View

Git repository

Google Cloud

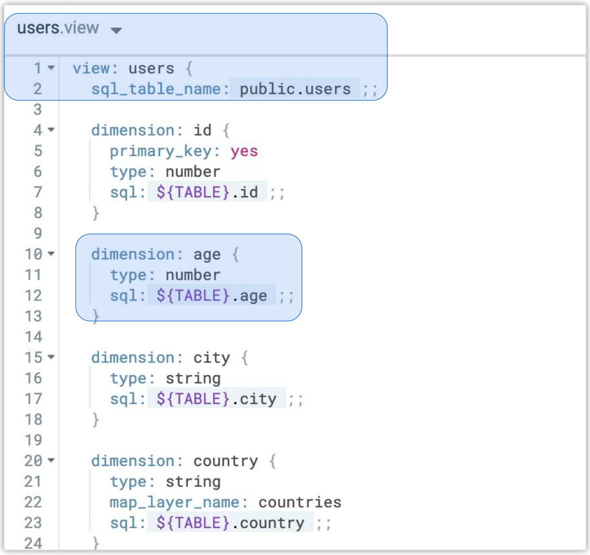
Next, the views that are joined in the Explores represent the underlying database tables.

Proprietary + Confidential

Views

Views represent the underlying database tables and are the building blocks of Explores defined in the model file.

View files contain definitions for dimensions and measures.

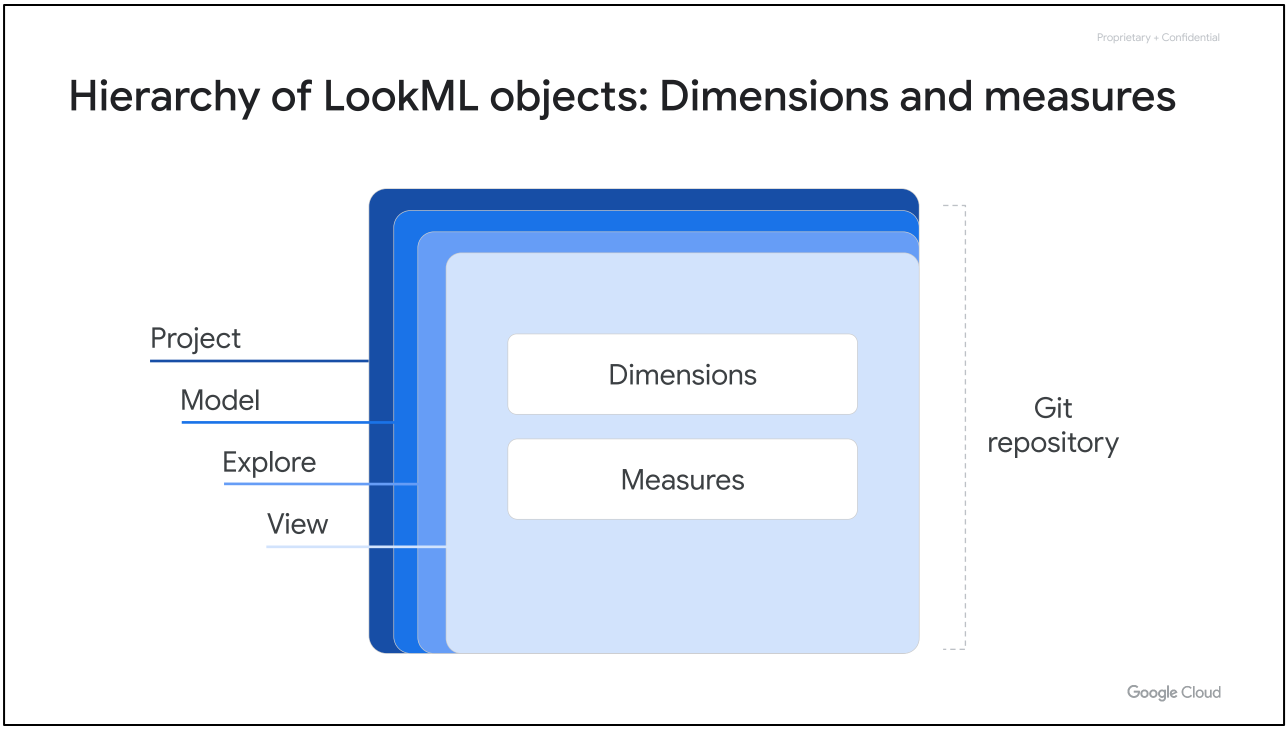


```
view: users {
  sql_table_name: public.users ;;

  dimension: id {
    primary_key: yes
    type: number
    sql: ${TABLE}.id ;;
  }

  dimension: age {
    type: number
    sql: ${TABLE}.age ;;
  }

  dimension: city {
    type: string
    sql: ${TABLE}.city ;;
  }

  dimension: country {
    type: string
    map_layer_name: countries
    sql: ${TABLE}.country ;;
  }
```

Views are the building blocks of Explores used by business users.

Within view files, developers can define the dimensions (or data attributes) and measures (or aggregations of attributes) to provide to business users in the Explore interface.

The view names that are joined to create an Explore are also the headers that business users see in the Explore. For example, in an Explore called **Order Items**, business users may see the **Users** view, which contains dimensions on users, such as their age and city of residence, and measures, such as their total purchases.



Proprietary + Confidential

Hierarchy of LookML objects: Dimensions and measures

Google Cloud

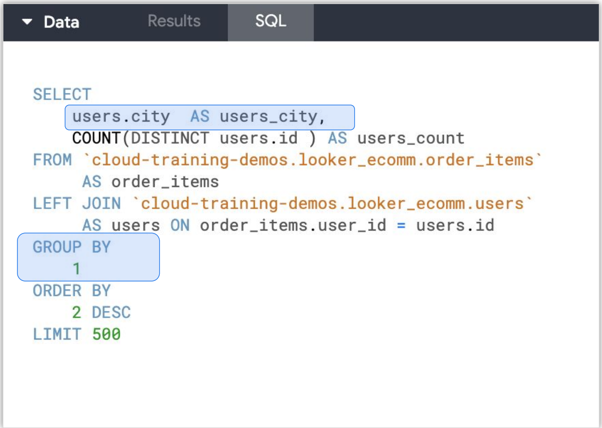
Last, but not least on the hierarchy, are the dimensions and measures defined in the view files.

Proprietary + Confidential

Dimensions

Dimensions represent the fields or columns of a database table and are defined in view files.

They are always in the **GROUP BY** statement of any SQL query generated by Looker.



```
SELECT
    users.city  AS users_city,
    COUNT(DISTINCT users.id ) AS users_count
FROM `cloud-training-demos.looker_ecomm.order_items`
     AS order_items
LEFT JOIN `cloud-training-demos.looker_ecomm.users`
     AS users ON order_items.user_id = users.id
GROUP BY
    1
ORDER BY
    2 DESC
LIMIT 500
```

Dimensions are data attributes and represent the fields or column of a database table.

When the view files are generated from a table by Looker, dimensions are automatically created for any columns that already exist within your database tables. You can also create additional dimensions that would serve as logical representations of table columns.

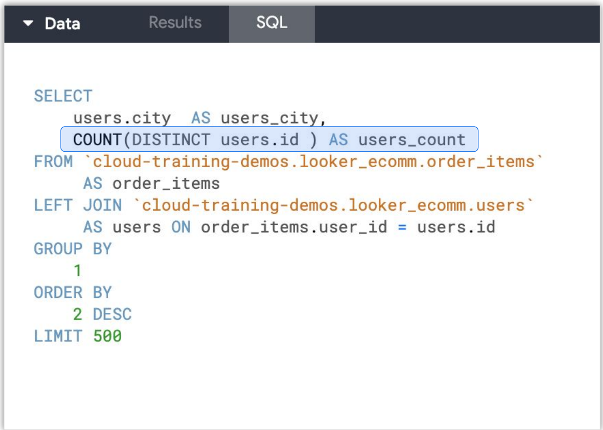
All dimensions appear in the **SELECT** and **GROUP BY** clause of a SQL statement that is generated by Looker based on the business user’s selections in the Explore.

Proprietary + Confidential

Measures

Measures are aggregates of dimensions and are defined in view files.

They appear as aggregate functions in the **SELECT** statement of any SQL query generated by Looker.

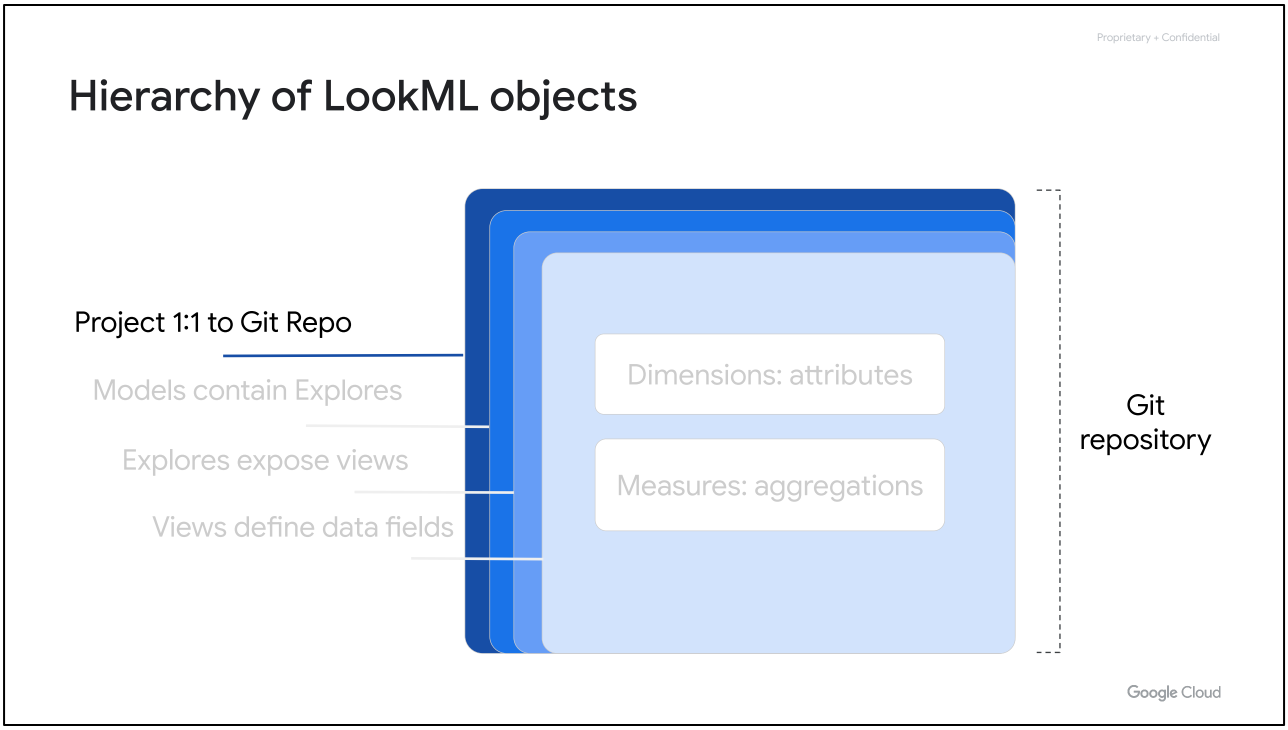


```
SELECT
    users.city  AS users_city,
    COUNT(DISTINCT users.id ) AS users_count
FROM `cloud-training-demos.looker_ecomm.order_items`
     AS order_items
LEFT JOIN `cloud-training-demos.looker_ecomm.users`
     AS users ON order_items.user_id = users.id
GROUP BY
    1
ORDER BY
    2 DESC
LIMIT 500
```

Google Cloud

Measures are aggregates of dimensions and do not live explicitly in your database tables. They must be created using 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, so they appear as aggregate functions in the **SELECT** statement of the SQL query.



Proprietary + Confidential

Hierarchy of LookML objects

Project 1:1 to Git Repo

Models contain Explores

Explores expose views

Views define data fields

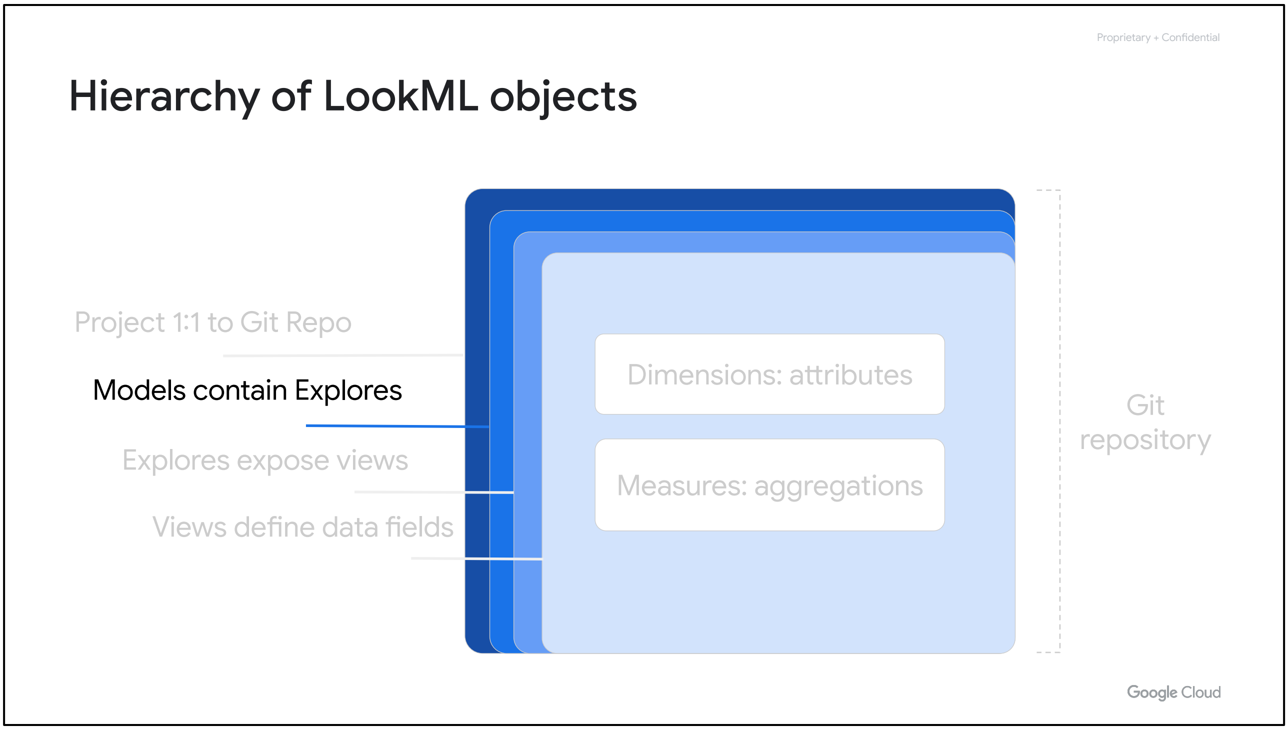
Dimensions: attributes

Measures: aggregations

Git repository

Google Cloud

To recap, a LookML project is a library of code that models a data source and should map 1:1 to a Git repository for version control.



Proprietary + Confidential

Hierarchy of LookML objects

Project 1:1 to Git Repo

Models contain Explores

Explores expose views

Views define data fields

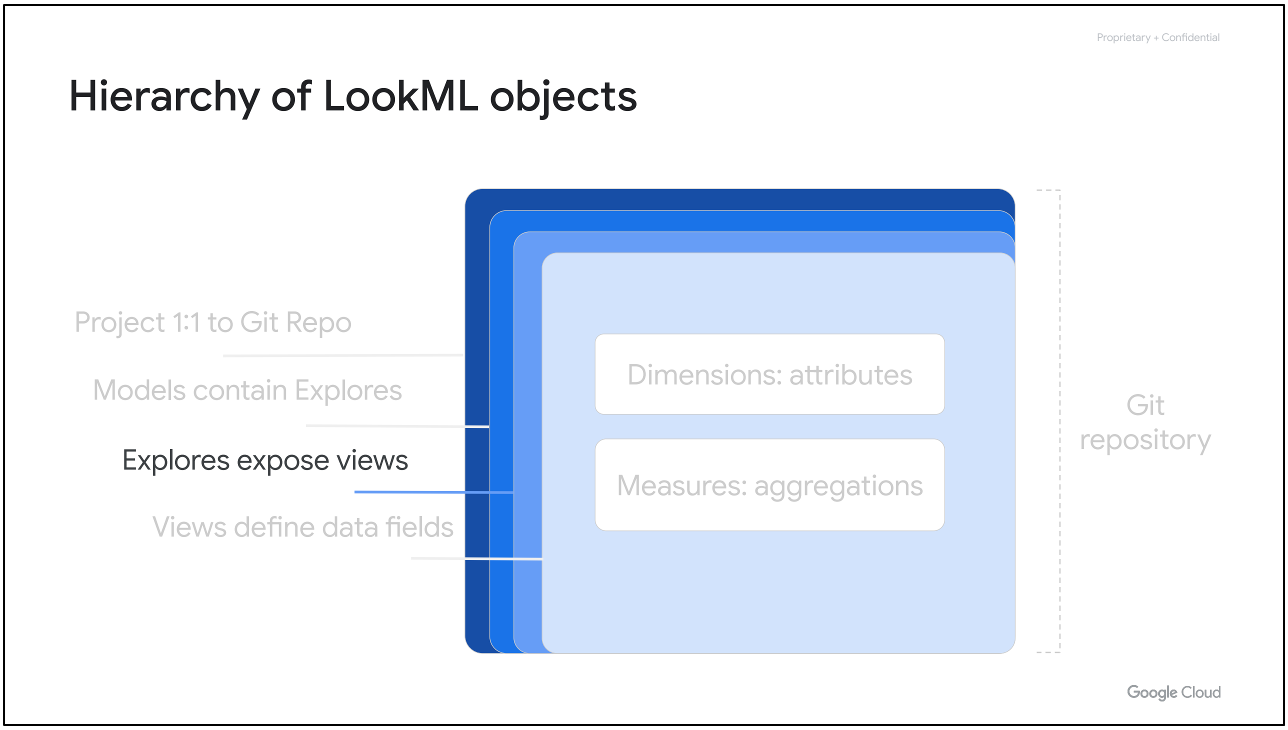
Dimensions: attributes

Measures: aggregations

Git repository

Google Cloud

Projects contain model files, which define the Explores that should be packaged together, how those Explores work, and which views are joined in which Explores.



Proprietary + Confidential

Hierarchy of LookML objects

Project 1:1 to Git Repo

Models contain Explores

Explores expose views

Views define data fields

Dimensions: attributes

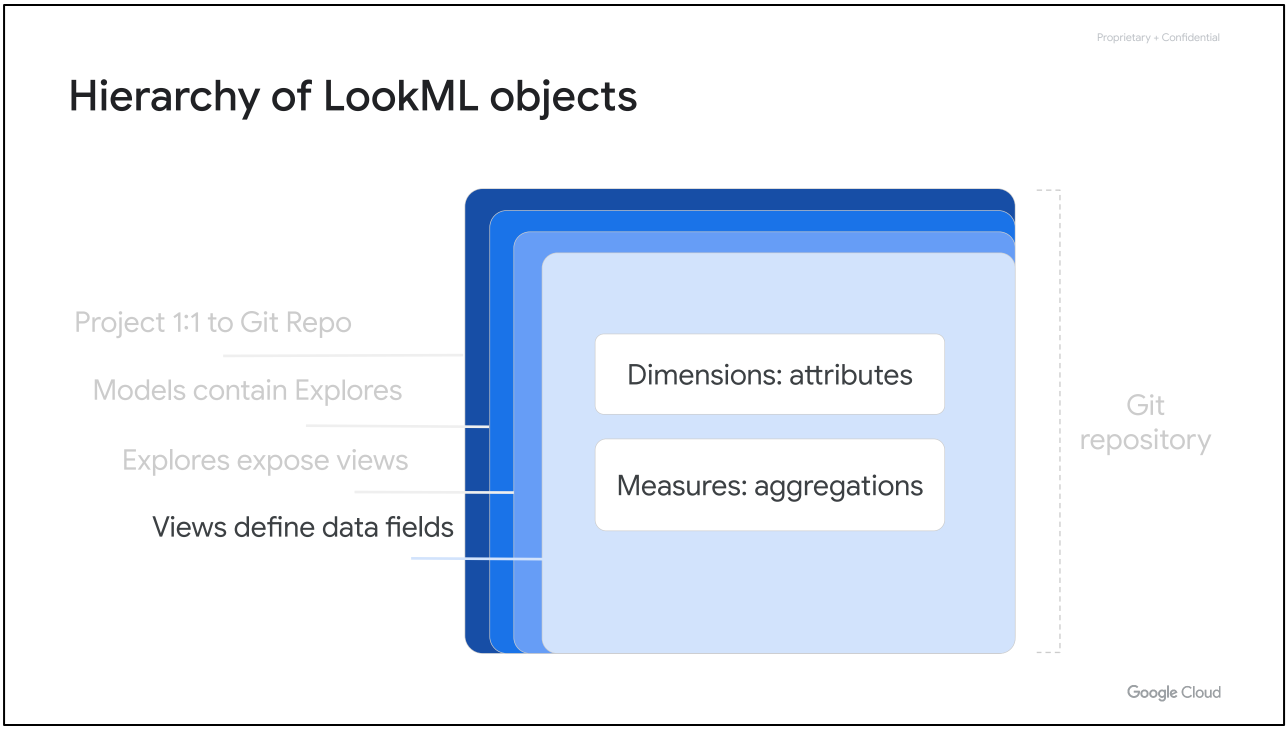
Measures: aggregations

Git repository

Google Cloud

View files describe database tables or logical representations of them and are joined together to define Explores in the model files.

Views are then accessed by business users through the Explore to query data and create reports and visualizations.



Proprietary + Confidential

Hierarchy of LookML objects

Project 1:1 to Git Repo

Models contain Explores

Explores expose views

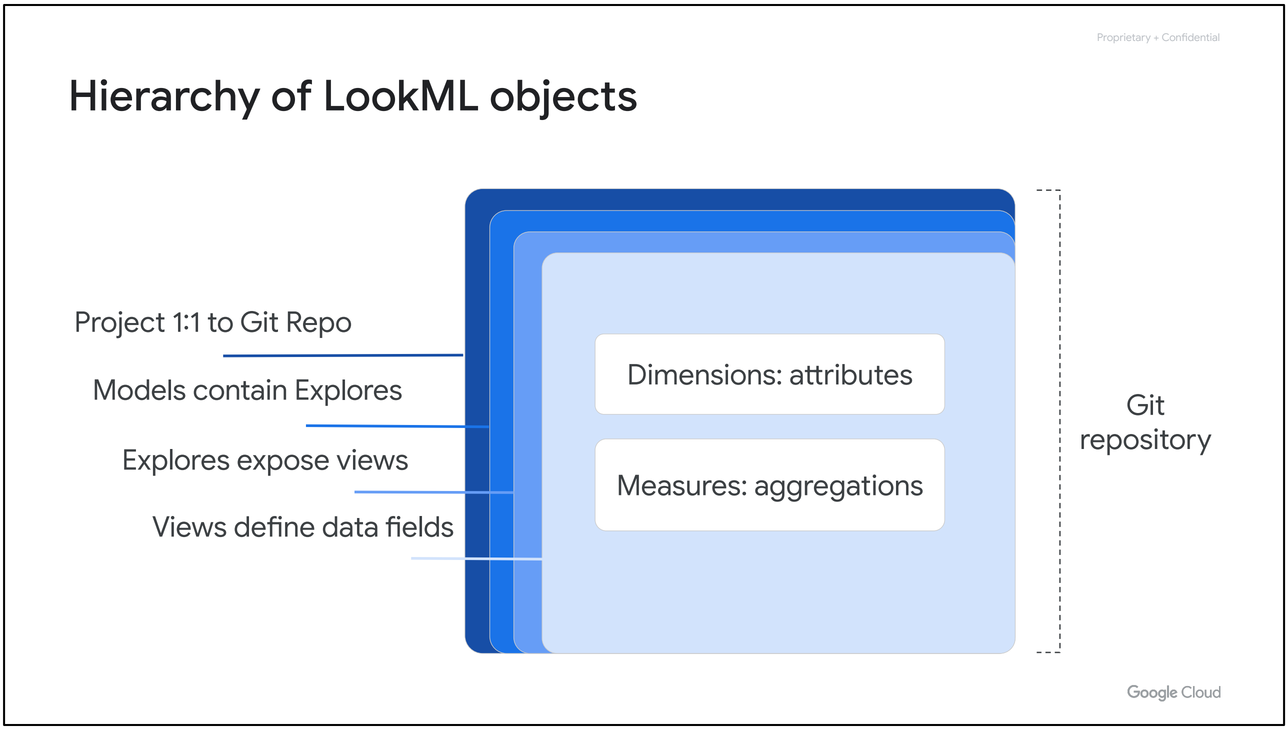
Views define data fields

Dimensions: attributes

Measures: aggregations

Git repository

Dimensions and measures are defined within view files. Dimensions are attributes of data and represent fields or columns in the database tables, while measures are aggregates of dimensions such as a count or sum.



Proprietary + Confidential

Hierarchy of LookML objects

Project 1:1 to Git Repo

Models contain Explores

Explores expose views

Views define data fields

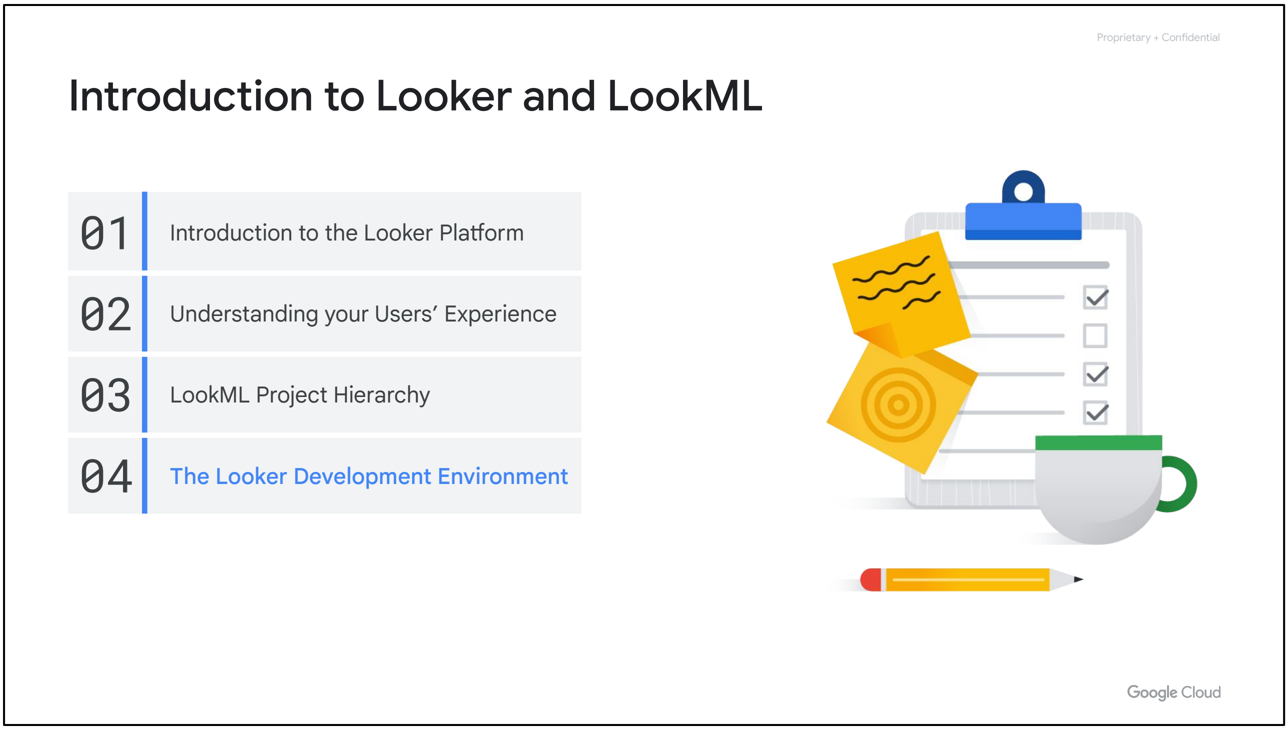
Dimensions: attributes

Measures: aggregations

Git repository

Google Cloud

After this introduction to the LookML key terms and hierarchy, you are now ready to start exploring your organization’s Looker instance as a LookML developer.



Proprietary + Confidential

Introduction to Looker and LookML

01 Introduction to the Looker Platform

02 Understanding your Users’ Experience

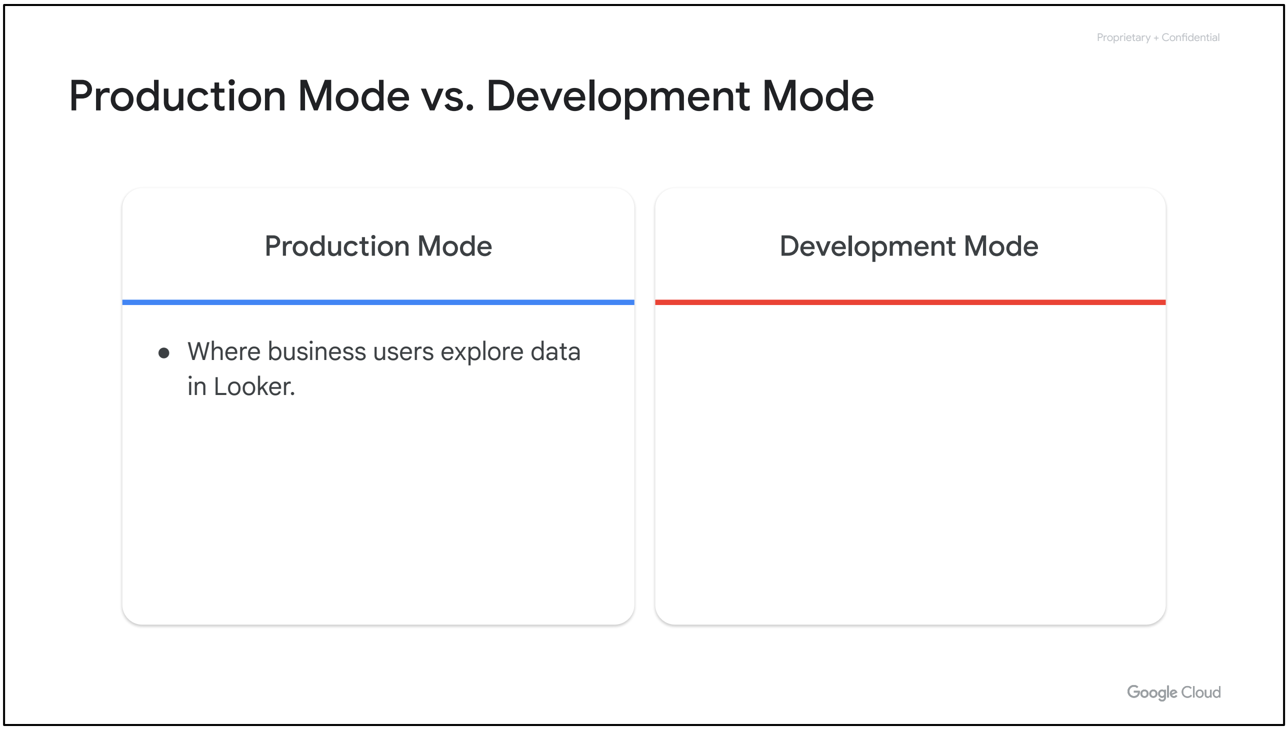
03 LookML Project Hierarchy

04 The Looker Development Environment

Google Cloud

Looker provides an integrated development environment (IDE) that developers can use to modify existing and to model new dimensions, measures and Explores for business users to employ in their data analysis approaches.

In this section, we will explore the overall Looker development environment and review the key components for your workflow as a LookML developer.



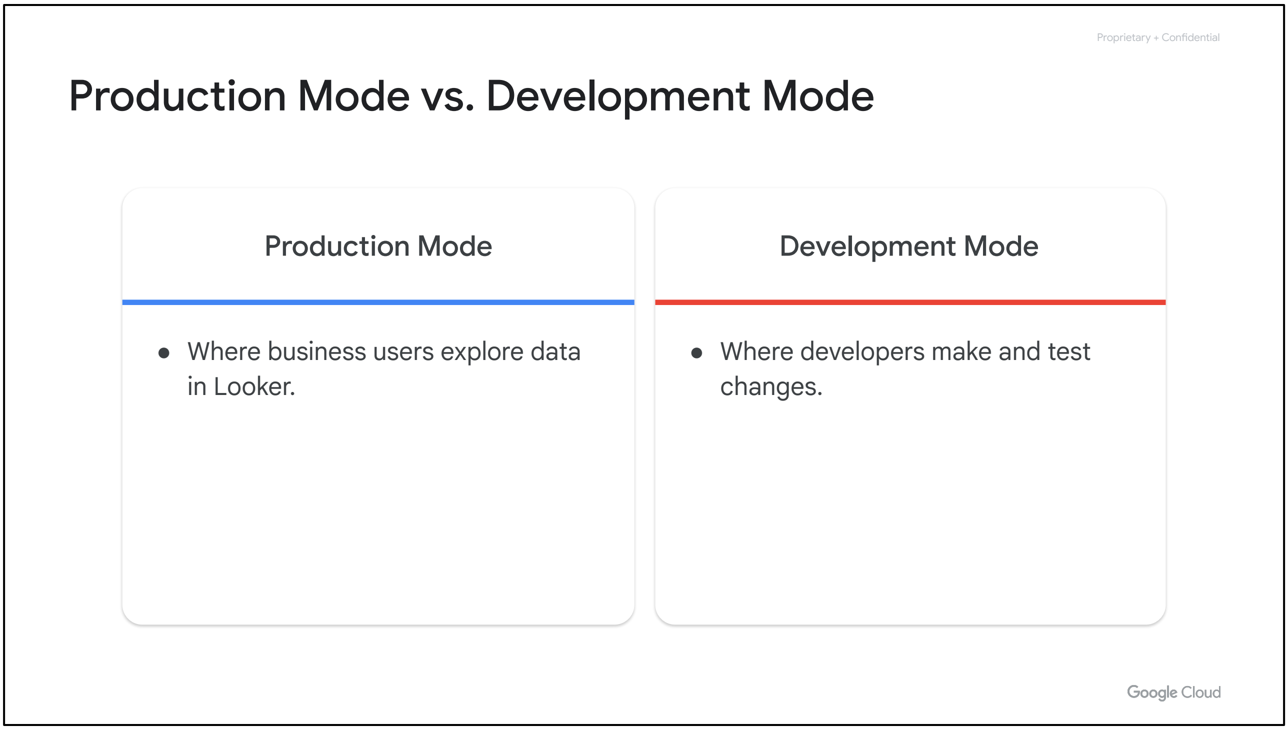
Proprietary + Confidential

Production Mode vs. Development Mode

Production Mode	Development Mode
Where business users explore data in Looker.	

Google Cloud

In Looker, business users interact with and experience the Looker instance in production mode...



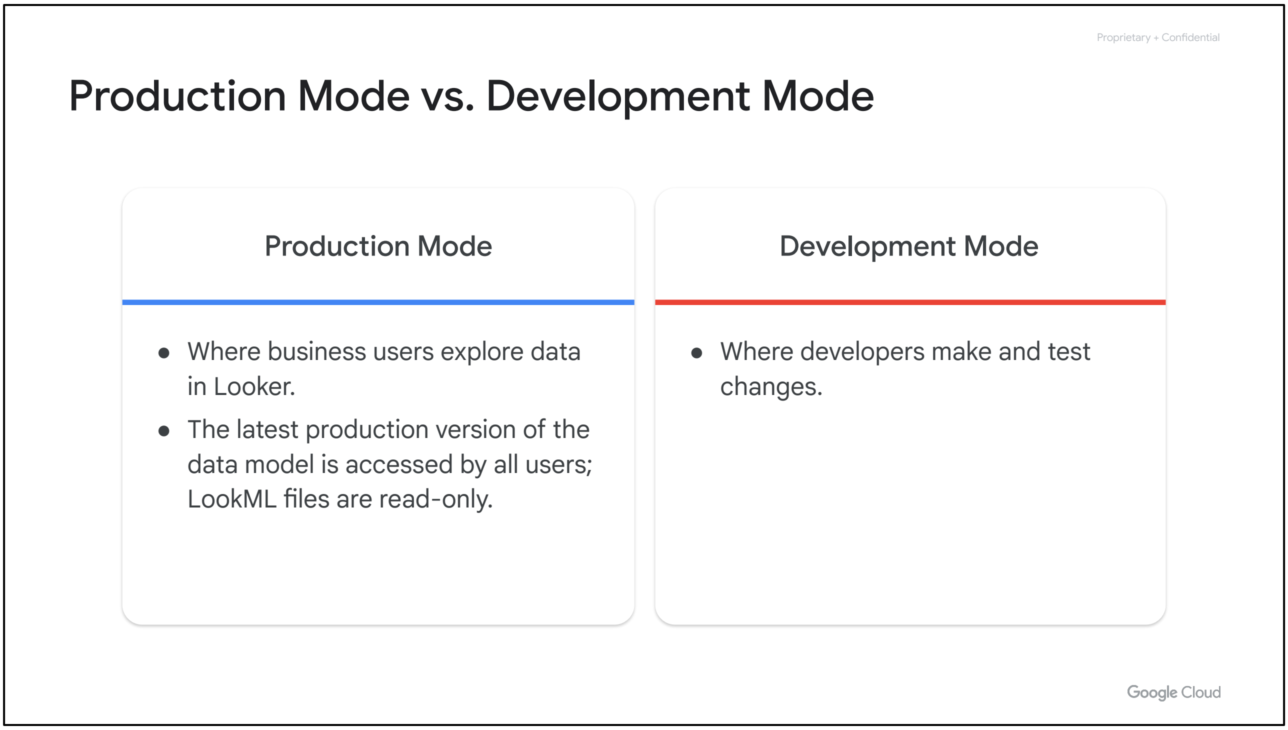
Proprietary + Confidential

Production Mode vs. Development Mode

Production Mode	Development Mode
• Where business users explore data in Looker.	• Where developers make and test changes.

Google Cloud

... while developers make changes and test new features through development mode.



Proprietary + Confidential

Production Mode vs. Development Mode

Production Mode

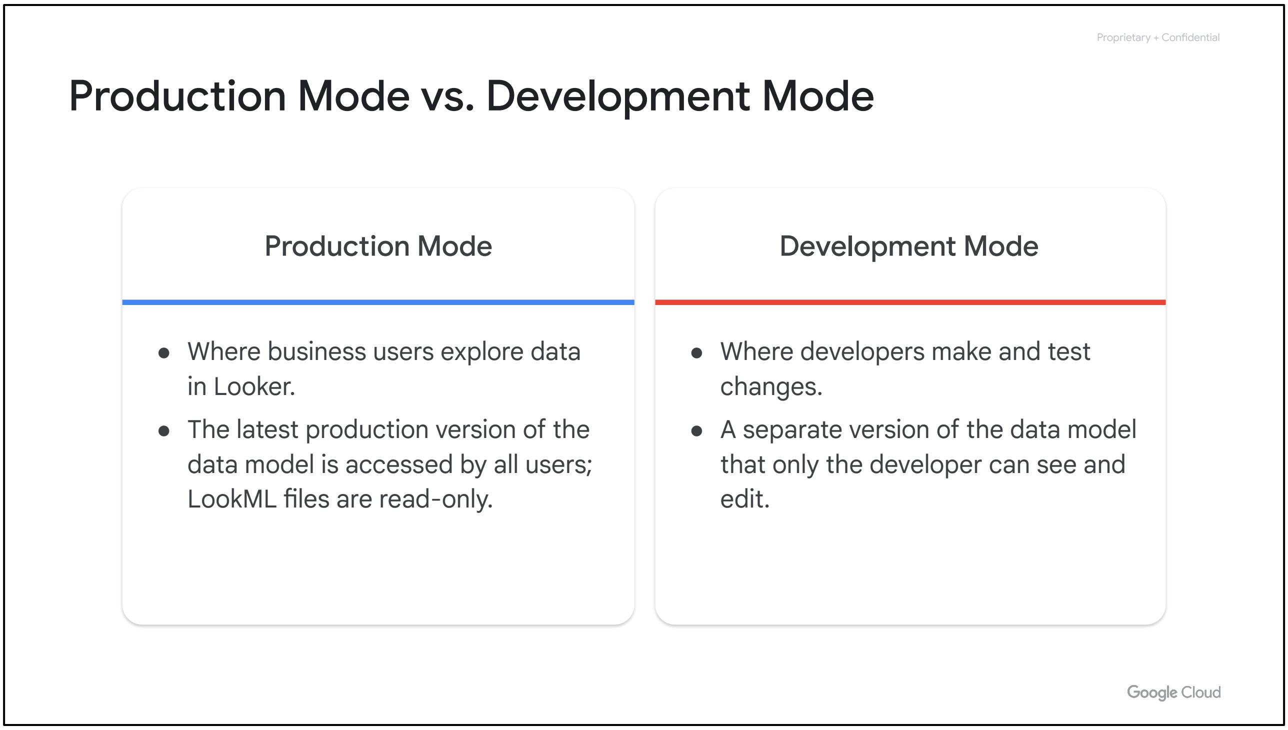
- Where business users explore data in Looker.
- The latest production version of the data model is accessed by all users; LookML files are read-only.

Development Mode

- Where developers make and test changes.

Production mode uses the latest production version of Looker. Everyone using a Looker instance in production mode accesses its projects, Explores, and content in the same state.

LookML project files are read-only in this mode.

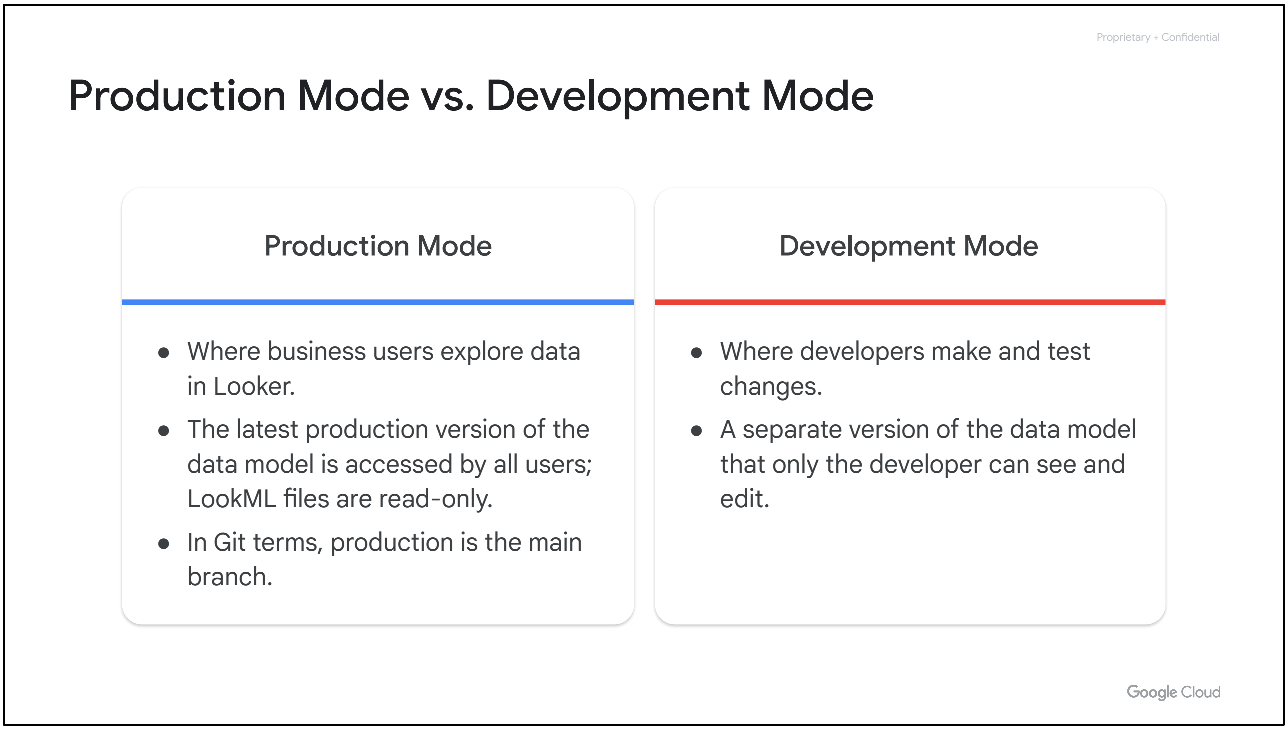


Production Mode vs. Development Mode

Production Mode	Development Mode
• Where business users explore data in Looker. • The latest production version of the data model is accessed by all users; LookML files are read-only.	• Where developers make and test changes. • A separate version of the data model that only the developer can see and edit.

As a LookML developer, you can use development mode to make changes to projects without affecting anyone else.

Development mode accesses a completely separate version of your project files that only you can see and edit.



Proprietary + Confidential

Production Mode vs. Development Mode

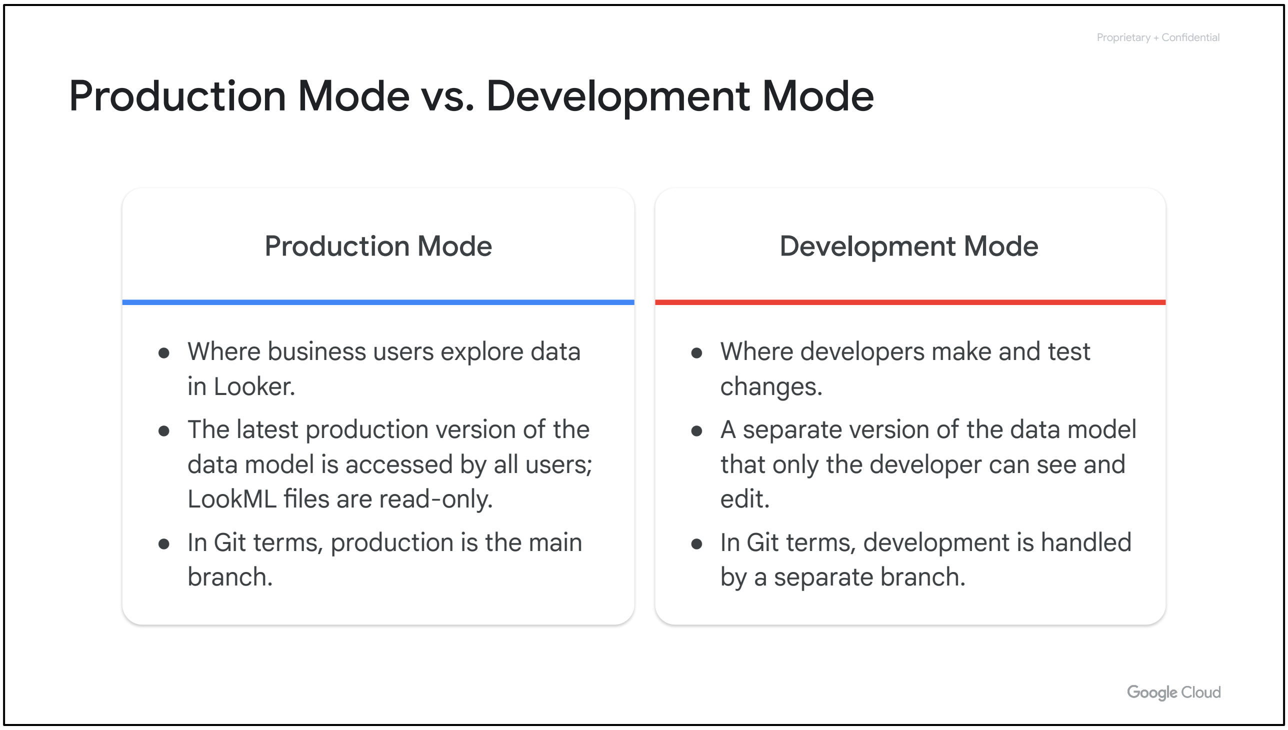
Production Mode

- Where business users explore data in Looker.
- The latest production version of the data model is accessed by all users; LookML files are read-only.
- In Git terms, production is the main branch.

Development Mode

- Where developers make and test changes.
- A separate version of the data model that only the developer can see and edit.

If you are familiar with Git, production mode uses the main branch of the Git repository of the LookML project...



Proprietary + Confidential

Production Mode vs. Development Mode

Production Mode

- Where business users explore data in Looker.
- The latest production version of the data model is accessed by all users; LookML files are read-only.
- In Git terms, production is the main branch.

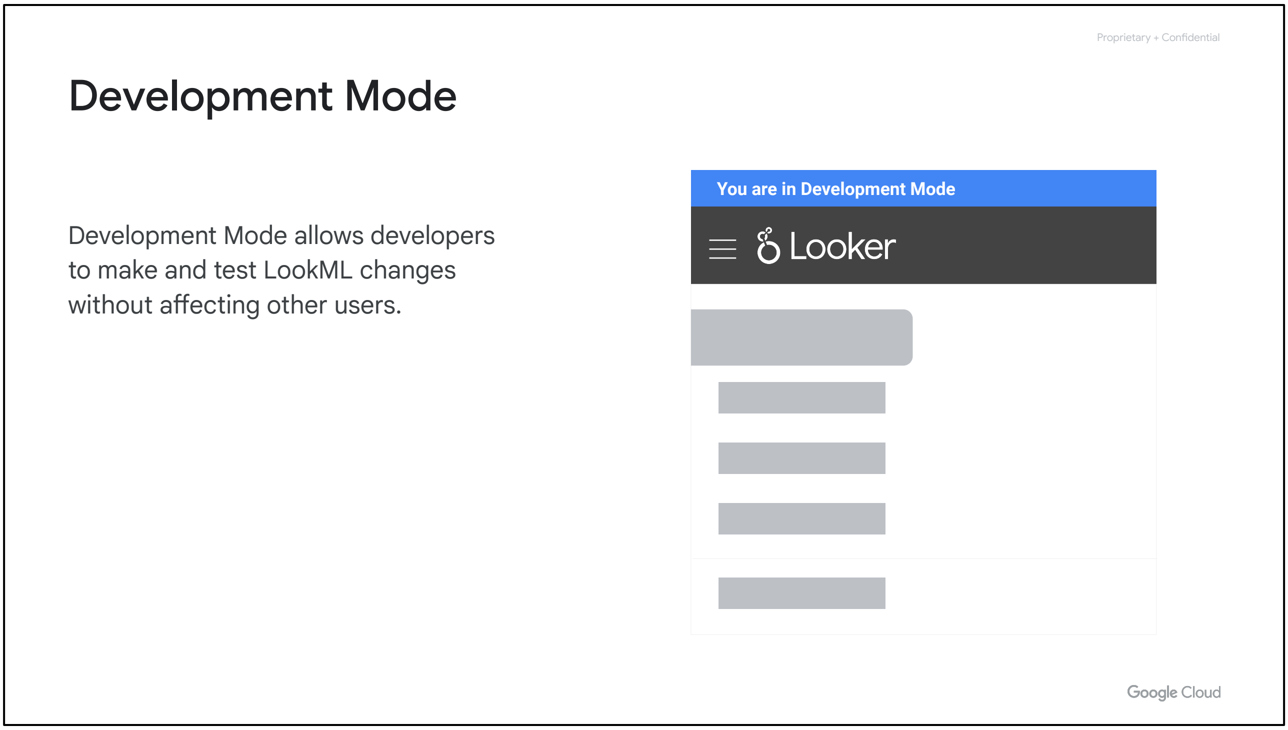
Development Mode

- Where developers make and test changes.
- A separate version of the data model that only the developer can see and edit.
- In Git terms, development is handled by a separate branch.

Google Cloud

… while development mode uses a separate branch created from the main branch.

After testing, changes made by LookML developers in the separate branch can be merged into production, so that all users of the Looker instance can access the updates.



Proprietary + Confidential

Development Mode

Development Mode allows developers to make and test LookML changes without affecting other users.

Google Cloud

By default, LookML projects open in Production Mode. This means that they show the latest version of the project that has been merged to production.

To make changes in your LookML projects, you need to enable Development Mode, which is frequently referred to as “dev mode.”

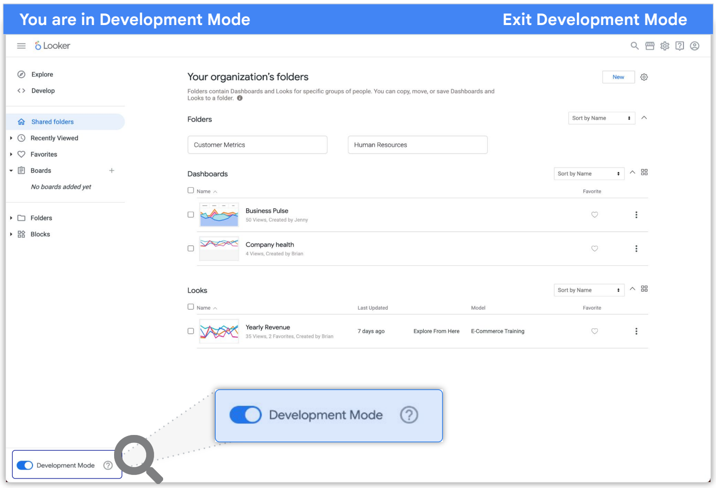
When you are in working in development mode, Looker will display a blue banner at top of the Looker User Interface (or UI) with a message that you are in development mode.

Proprietary + Confidential

Enter Development Mode

Developers can use the toggle button at the bottom left-side of the Looker User Interface to enter Development Mode.

Keyboard shortcut: Control+Shift+D



As a LookML developer, you have several options to enter development mode.

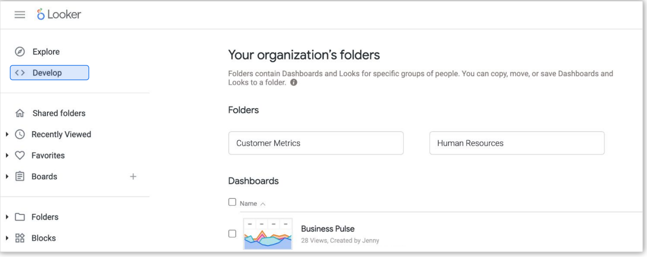
One option is to use the toggle button at the bottom left-side of the Looker UI to enter **Development Mode.** You can also use the same toggle button to exit **Development Mode** and return to production mode.

There is also a keyboard shortcut: Control+Shift+D.

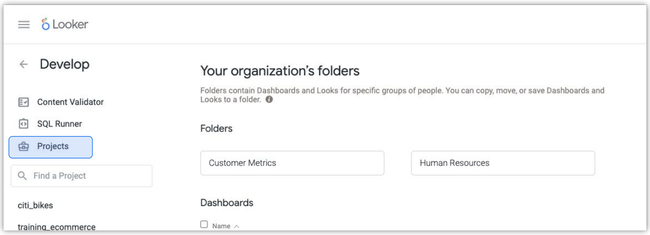
Proprietary + Confidential

Enter Development Mode

On the top left-side of the Looker User Interface, click on Develop, and then select Projects.



Folders contain Dashboards and Looks for specific groups of people. You can copy, move, or save Dashboards and Looks to a folder.

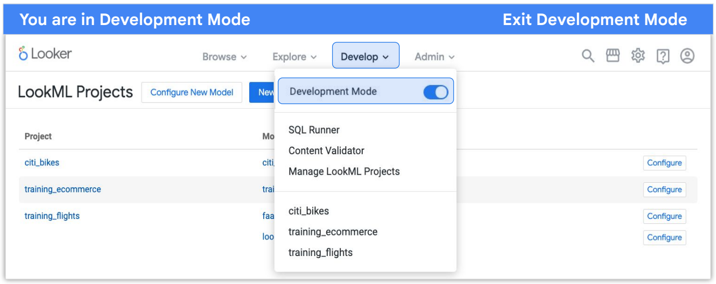


Folders contain Dashboards and Looks for specific groups of people. You can copy, move, or save Dashboards and Looks to a folder.

Alternatively, at the top left-side of the Looker User Interface, you can click on **Develop** menu to see the options, and then select **Projects.**

Enter Development Mode

On the Projects page, click on Develop from the top menu and use the toggle button for Development Mode.

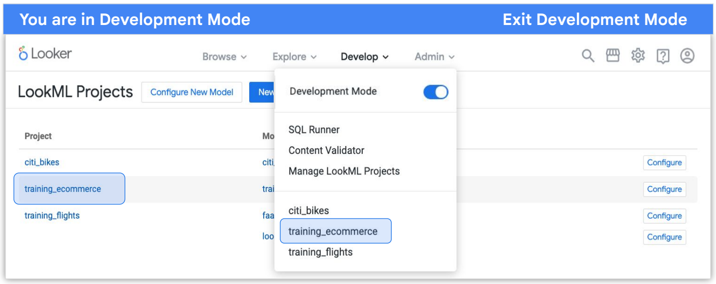


Then, once you are on the **Projects** page, you can turn on Development Mode by clicking on the **Develop** option from the top menu bar and using the toggle button for **Development Mode**.

Proprietary + Confidential

Integrated development environment (IDE)

On the Projects page, you can also click on a project name to open it within the Looker Integrated development environment (IDE).

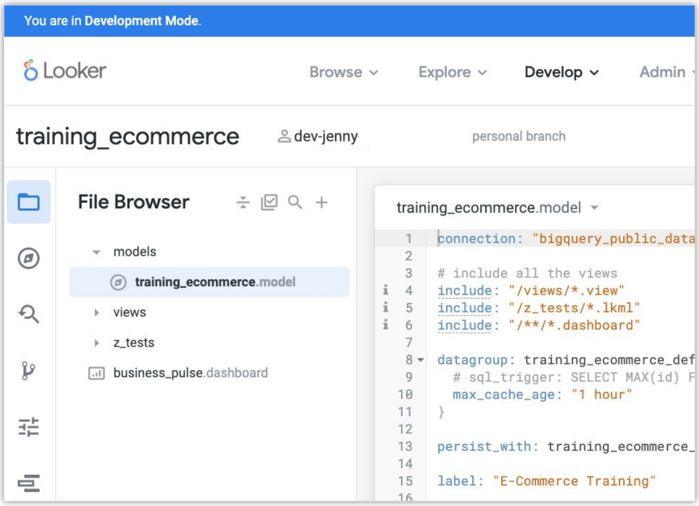


From the **Projects** page, you can also click on a specific project name such as **training_ecommerce** to open it within the Looker Integrated development environment (IDE).

Proprietary + Confidential

Integrated development environment (IDE)

Development in Looker takes place inside the integrated development environment (IDE).



```
connection: "bigquery_public_data

# include all the views
include: "/views/*.view"
include: "/z_tests/*.lkml"
include: "/**/*.dashboard"

datagroup: training_ecommerce_def
  # sql_trigger: SELECT MAX(id) F
  max_cache_age: "1 hour"
}

persist_with: training_ecommerce_

label: "E-Commerce Training"
```

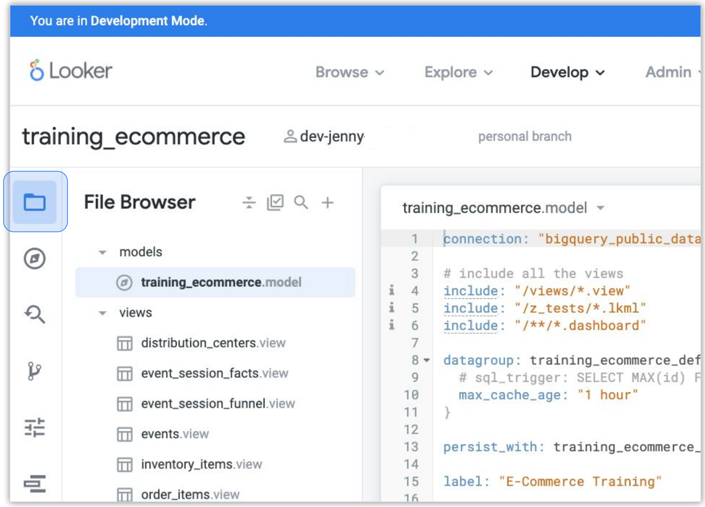
An IDE is an industry term for an application in which software engineers or developers can write and test their code.

Looker has its own built-in IDE in which developers can write LookML.

The Looker IDE displays six navigation options on the left-side panel. We'll walk through each option to understand the functionality that it provides for LookML developers.

Integrated development environment (IDE): File browser

The file browser lists the project files in the folder hierarchy and can be used to navigate to a specific file.



```
connection: "bigquery_public_data

# include all the views
include: "/views/*.view"
include: "/z_tests/*.lkml"
include: "/**/*.dashboard"

datagroup: training_ecommerce_def
  # sql_trigger: SELECT MAX(id) F
  max_cache_age: "1 hour"
}

persist_with: training_ecommerce_

label: "E-Commerce Training"
```

The first option, represented by the folder icon, is the file browser. This displays all the LookML project files in the folder hierarchy.

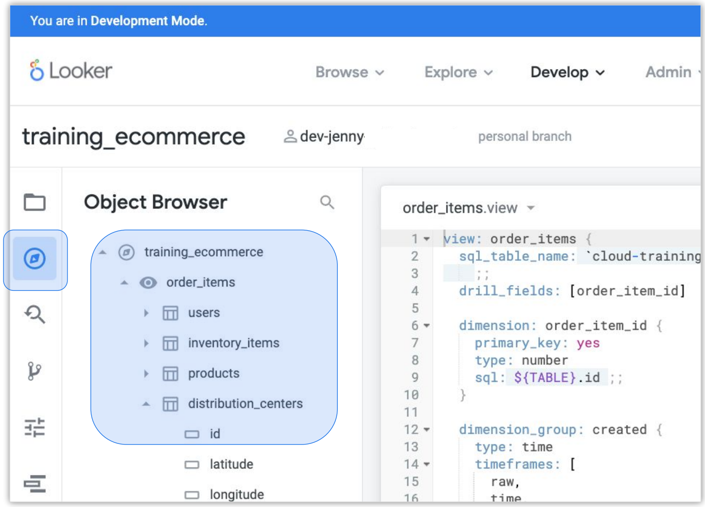
Depending on the complexity of the project, you could see dozens or even hundreds of files, within many folders and levels of subfolders.

At your company, with your own development team, you can decide on any kind of folder structure that would be the most intuitive for your workflows.

Proprietary + Confidential

Integrated development environment (IDE): Object browser

The object browser displays all of the objects in the project and can be used to navigate to the LookML for a specific object.



```
view: order_items {
  sql_table_name: `cloud-training
    ;;
  drill_fields: [order_item_id]

  dimension: order_item_id {
    primary_key: yes
    type: number
    sql: ${TABLE}.id ;;
  }

  dimension_group: created {
    type: time
    timeframes: [
      raw,
      time
```

The second option, represented by the compass icon, is the object browser. The object browser organizes project files by LookML object type in the project.

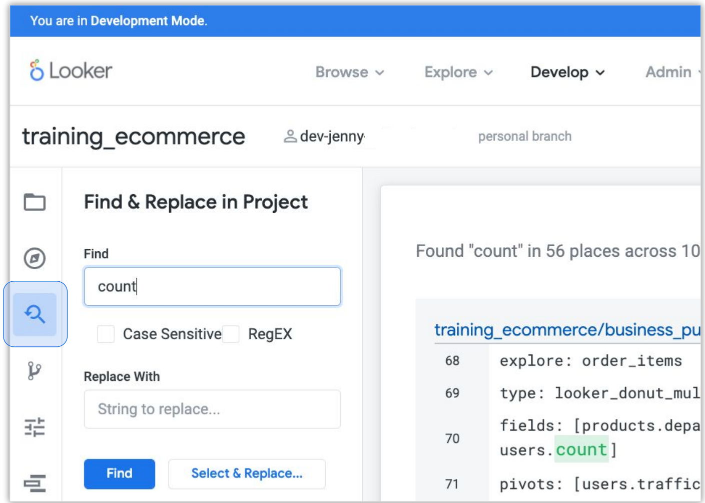
Each model can be expanded to show the Explores within. Furthermore, each Explore can be expanded to reveal the views that are joined together, and each view can be expanded to display the dimensions and measures.

This is really useful when you need to find a specific field or view being used within an Explore, especially if you have many similarly named objects in the model. You may also need to “visualize,” in a sense, in how many Explores the same view is being used, or which models have the same or similar Explores.

Proprietary + Confidential

Integrated development environment (IDE): Find & replace

Use the find-and-replace function to search for or replace text in all of your project files.



You are in Development Mode.

training_ecommerce dev-jenny personal branch

Find & Replace in Project

Find: count

□ Case Sensitive □ RegEX

Replace With: String to replace...

Found "count" in 56 places across 10

training_ecommerce/business_pu

```
68  explore: order_items
69  type: looker_donut_mul
70  fields: [products.depa
    users.count]
71  pivots: [users.traffic
```

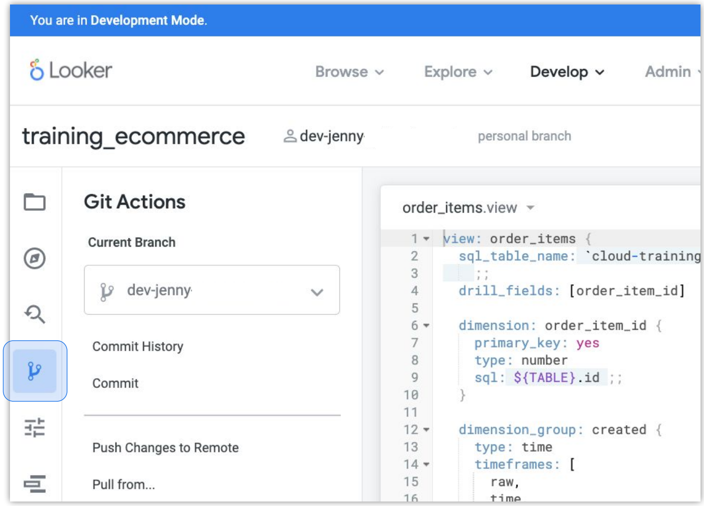
Next is the find-and-replace option. You can search for a specific word like “count” and see where and how many times this term appears throughout all the files in your entire project.

You can also batch-replace all instances of a text string with something else.

Proprietary + Confidential

Integrated development environment (IDE): Git actions

All Git actions available to you within a project are listed under Git Actions.



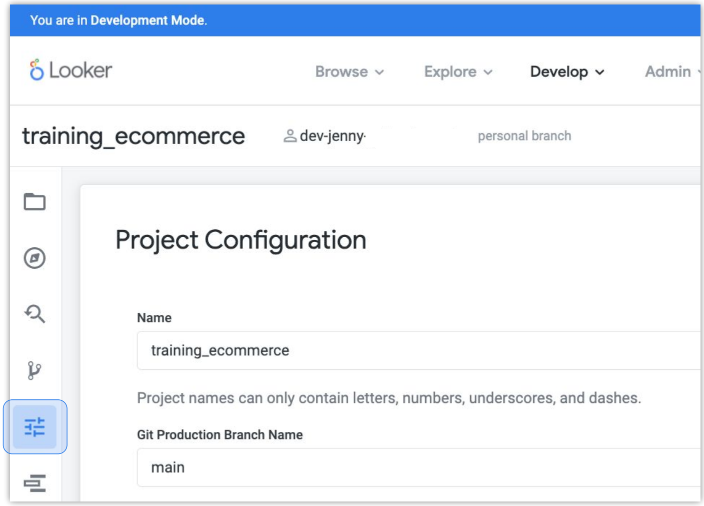
```
view: order_items {
  sql_table_name: `cloud-training
    ;;
  drill_fields: [order_item_id]

  dimension: order_item_id {
    primary_key: yes
    type: number
    sql: ${TABLE}.id ;;
  }

  dimension_group: created {
    type: time
    timeframes: [
      raw,
      time
```

The fourth option in the IDE shows the available Git actions.

These options allow you to switch between Git branches, view past commits by yourself and fellow developers, view the project on GitHub, or whichever Git provider you use at your company, and more.

Integrated development environment (IDE): Project settings

Under Project settings, you can view the configuration of a project. Only Looker admins can change these settings.



Project Configuration

Name

training_ecommerce

Project names can only contain letters, numbers, underscores, and dashes.

Git Production Branch Name

main

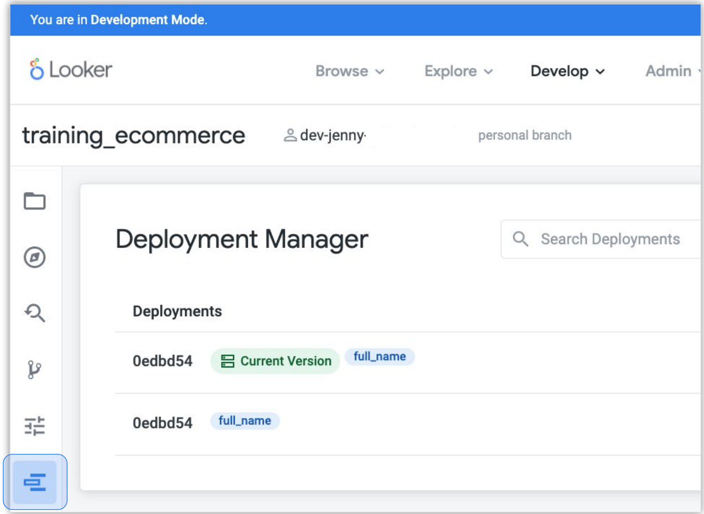
The next option displays the project settings. As a LookML developer, you can see the configurations, but only a Looker admin can change them.

For example, admins can specify what is or isn't required to commit code, enable pull requests if your team prefers to do code reviews before deploying to production, change the Git connection, and more.

Proprietary + Confidential

Integrated development environment (IDE): Deployment manager

Recent changes that have been deployed to the production environment are listed under Deployment manager.



You are in Development Mode.

Deployment Manager

Deployments

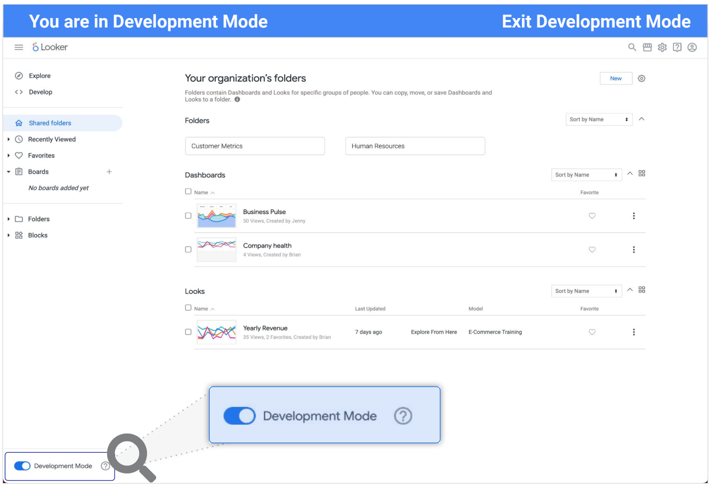
0edbd54 Current Version full_name

0edbd54 full_name

The last option is the deployment manager.

This page lets you see the recent changes that have been deployed to the production environment. Additional details about the changes, such as the user and date associated with the change, are also available on this page.

Production Mode vs. Development Mode



You are in Development Mode

Exit Development Mode

Looker

Your organization's folders

Folders contain Dashboards and Looks for specific groups of people. You can copy, move, or save Dashboards and Looks to a folder.

Folders

Customer Metrics

Human Resources

Dashboards

Name	Favorite
Business Pulse 50 Views, Created by Jenny	♡
Company health 4 Views, Created by Brian	♡

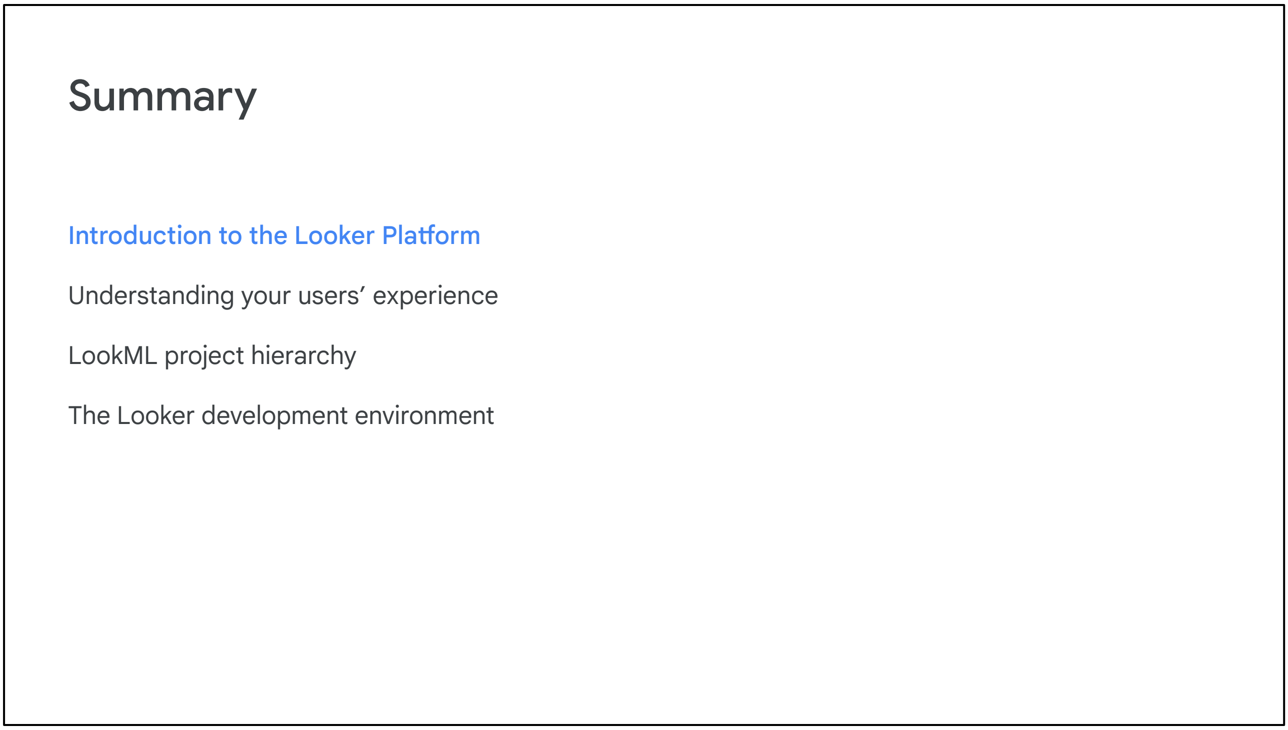
Looks

Name	Last Updated		Model	Favorite
Yearly Revenue 35 Views, 2 Favorites, Created by Brian	7 days ago	Explore From Here	E-Commerce Training	♡

In summary, as a LookML developer, you can make and test changes in your organization's Looker instance in development mode, without impacting the production environment.

When working in your organization's Looker instance, be mindful of whether you are currently in production mode or development mode. The Looker IDE and Explore menus will offer a few different features and options in production vs. development mode. Depending on your permissions and local code differences, you may even see different lists of projects, project files, and Explores.

With this understanding of the differences between production and development mode, we hope that you have fun exploring the LookML projects in your organization's Looker instance!



Summary

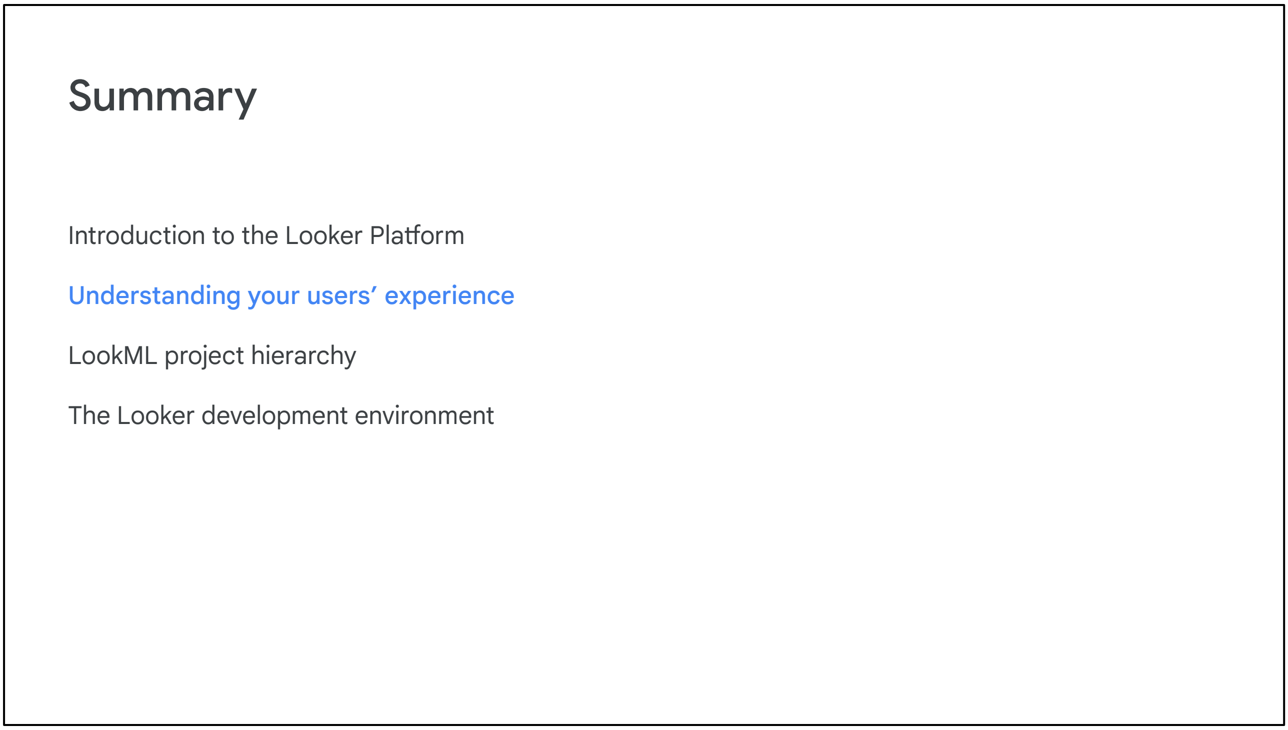
Introduction to the Looker Platform

Understanding your users’ experience

LookML project hierarchy

The Looker development environment

In this module, we started with an overview of the Looker platform and architecture to understand how Looker can play a key role in your organization’s data workflows.



Summary

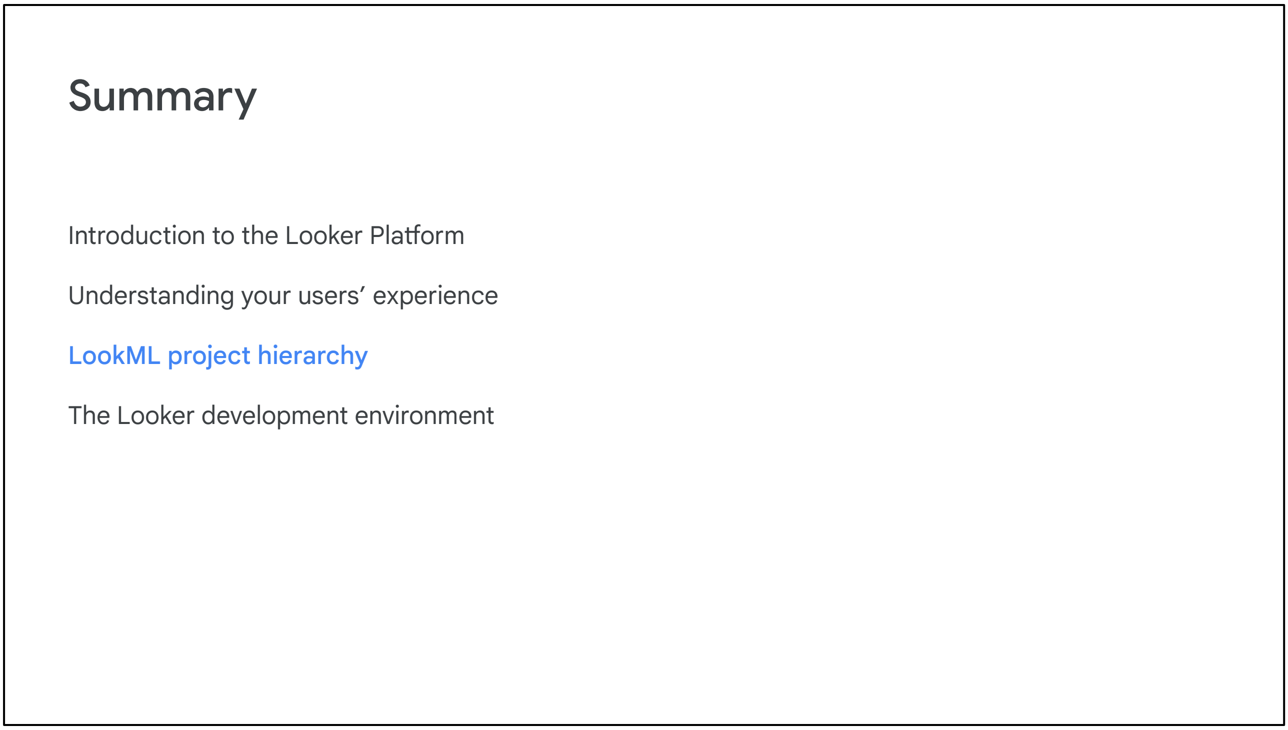
Introduction to the Looker Platform

Understanding your users’ experience

LookML project hierarchy

The Looker development environment

Then, we completed a walkthrough of the Looker User Interface to help you understand how your business users leverage Looker to explore and analyze data.



Summary

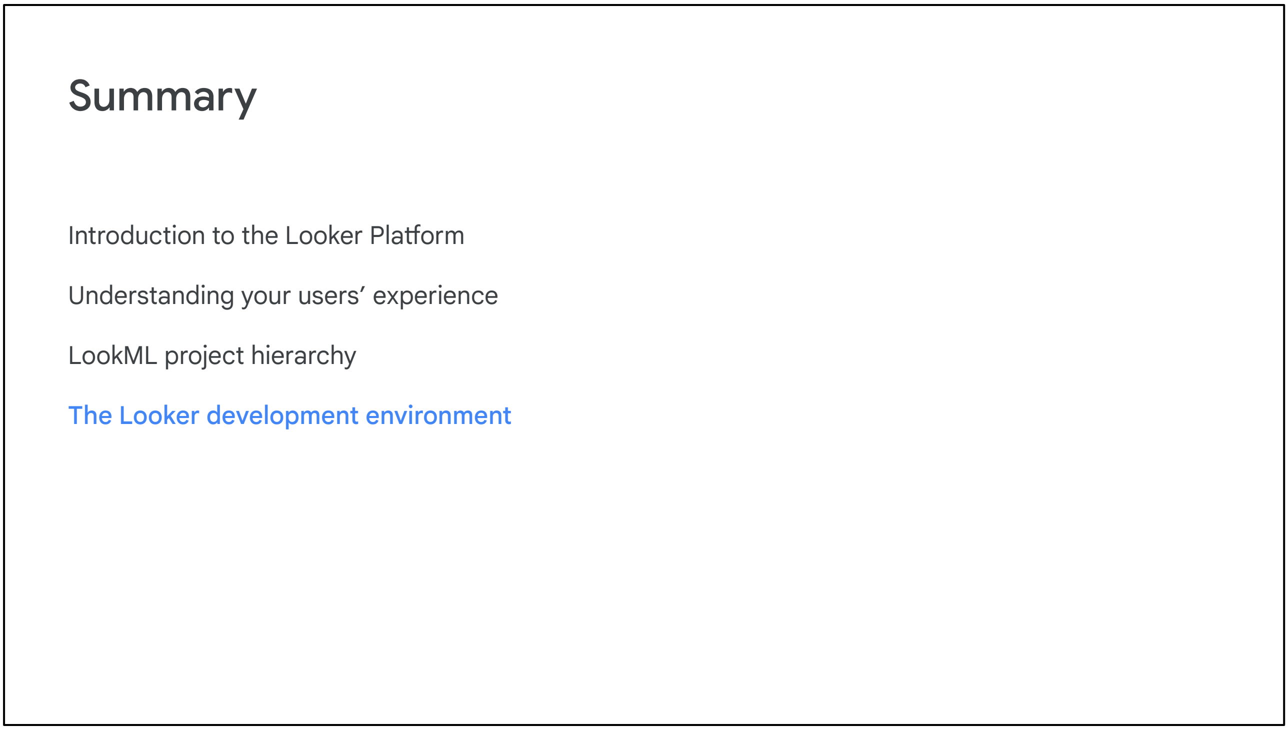
Introduction to the Looker Platform

Understanding your users’ experience

LookML project hierarchy

The Looker development environment

Next, we reviewed the LookML project hierarchy and discussed the role that each key component plays in the overall LookML structure.



Summary

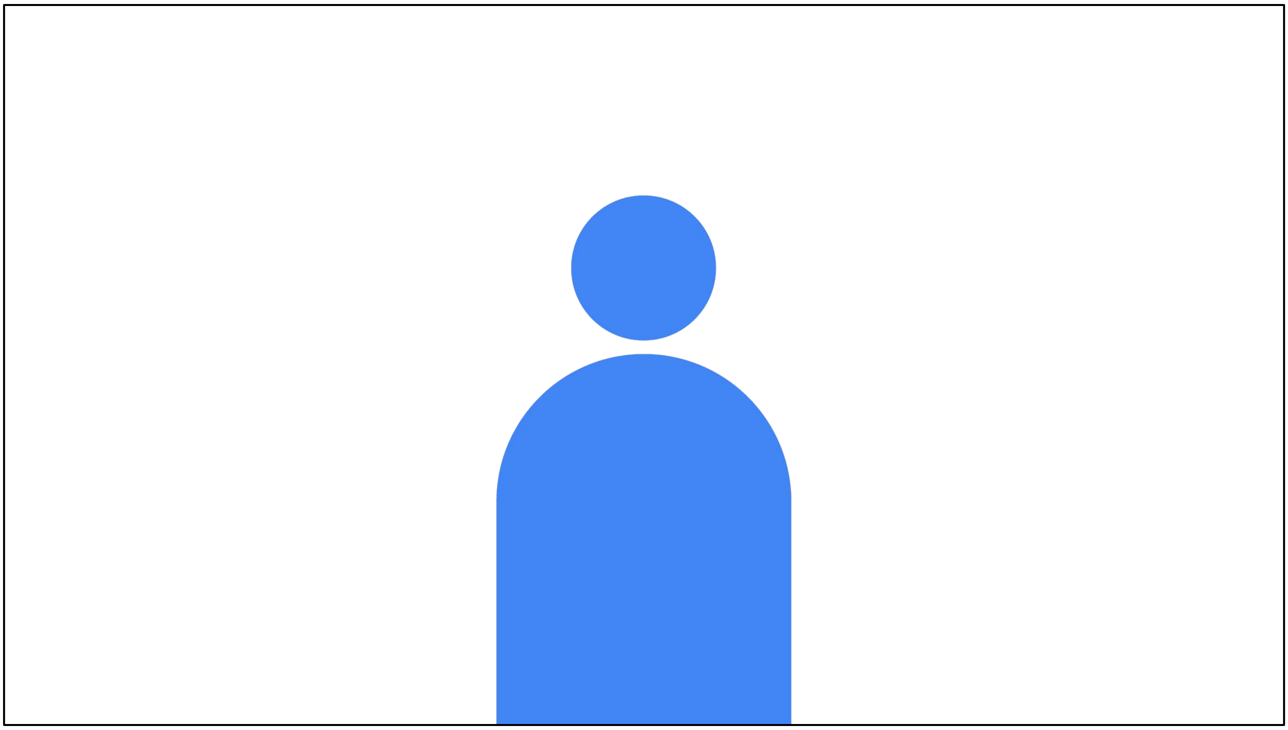
Introduction to the Looker Platform

Understanding your users’ experience

LookML project hierarchy

The Looker development environment

Last, we explored the Looker development environment and reviewed its key features for your workflow as a LookML developer.



In the next module, we will expand on these concepts to begin writing LookML code to model dimensions and measures, the key data components used by business users to analyze and visualize data.

See you there!