Groundzero

PRODUCT FEATURE GUIDE

Data

Connections, files, catalog, lakehouse, lineage and Ontology Studio

Documented October 2026

The names and values in the screenshots are examples from a sample company, Acme Retail.

Contents

1 About this guide 4
- 1.1 Start with what you want to do 4
- 1.2 Where to find things in Groundzero 4
- 1.3 How the pieces fit together 4
- 1.4 How to read the chapters 5
- 1.5 Index of features 5
- 1.6 Related guides 6

2 Data source connections 7
- 2.1 Overview 7
- 2.2 Create a connection 8
- 2.3 Test a connection 13
- 2.4 Find a connection 14
- 2.5 View connection details 17
- 2.6 Edit a connection 19
- 2.7 Delete a connection 19
- 2.8 Connector reference 20
- 2.9 Special connection types 25
- 2.10 Secrets 27
- 2.11 Limits 28
- 2.12 Tips 28
- 2.13 Troubleshooting 28
- 2.14 Related features 29

3 Files and workspaces 30
- 3.1 Organise dashboards in a workspace 30
- 3.2 Manage files in your buckets 34
- 3.3 Troubleshooting 39
- 3.4 Related 39

4 Data catalog 40
- 4.1 Find the data you need 41
- 4.2 Understand a resource 46
- 4.3 Describe a resource 51
- 4.4 Tag data 54
- 4.5 Copy the call for a dataset or job 59
- 4.6 When the catalog shows a message 61
- 4.7 Troubleshooting 63
- 4.8 Related 63

5 Lakehouse and warehouses 64
- 5.1 Permissions in short 64
- 5.2 Create a warehouse 65
- 5.3 Open a warehouse and start a session 70
- 5.4 Run a query 75
- 5.5 Read and export a result 82
- 5.6 Save and reuse queries 85
- 5.7 Look at a table's details 86
- 5.8 Protect a table with a data policy 90
- 5.9 Allow or deny a table to users and groups 91

5.10 Review the audit log 92
5.11 Back up and restore a database 94
5.12 Tips 97
5.13 Troubleshooting 97
5.14 Related 98

6 Data lineage 99
6.1 How lineage gets recorded 100
6.2 How sure is a path 100
6.3 Permissions 100
6.4 Open the map for a table 100
6.5 Read the map 103
6.6 See what breaks if a table changes 106
6.7 Trace a value back to its source 107
6.8 Follow a column 109
6.9 Check one run 110
6.10 Use the details panel 111
6.11 See every hop: complete movement 117
6.12 Start from a job or dataset 119
6.13 Export lineage 120
6.14 Legend, refresh and the "catching up" banner 122
6.15 When the page shows a message 124
6.16 Troubleshooting 126
6.17 Limits 126
6.18 Related 126

7 Ontology Studio 127
7.1 The words used here 128
7.2 Permissions 128
7.3 Find your way around 129
7.4 Create a model 130
7.5 See how things connect: the map 133
7.6 Browse and find definitions 136
7.7 Review definitions 138
7.8 Draft with AI 147
7.9 Publish a release 151
7.10 Look at past releases 155
7.11 Keep up with new columns: lineage drift 157
7.12 Export a model 157
7.13 If someone else edits at the same time 158
7.14 When the page shows a message 158
7.15 Tips 160
7.16 Troubleshooting 160
7.17 Limits 161
7.18 Related 161

A Appendix 162
A.1 Naming rules 162
A.2 Limits at a glance 162
A.3 Glossary 162

1

About this guide

This guide helps you work with the **Data** area of Groundzero: bring data in, find it, query it, trace where it came from and agree what it means. Each chapter starts with what you can get done, then walks through it step by step with screenshots, then lists every field and message you may meet.

The names and values in the screenshots are examples from a sample company, "Acme Retail".

1.1 Start with what you want to do

I want to…	Read
Connect a database, bucket, SaaS app, AI provider or lakehouse	Data source connections
Upload or download files, or keep dashboards in a folder	Files and workspaces
Find a table or column, add tags or a description	Data catalog
Create a warehouse, run SQL, export a result, protect a table, back up a database	Lakehouse and warehouses
See what feeds a table, or what breaks if it changes	Data lineage
Agree and publish what business terms mean for AI Chat and agents	Ontology Studio

For SQL itself, see the **Lakehouse runtime guide**.

1.2 Where to find things in Groundzero

The top navigation bar has six items. Four belong to this guide.

Navigation item	Chapter
Workspaces	Files and workspaces
Connections	Data source connections
Lakehouse	Lakehouse and warehouses
Catalog	Data catalog, Data lineage and Ontology Studio (use the menu on the left of the Catalog page)
Designer	Build documentation: datasets, visualizations and dashboards
DE/ML	Build documentation: notebooks, ETL jobs, runtimes and MLOps

The Catalog menu also has **Consumption Metrics**, which is in the Analyze and Admin documentation.

1.3 How the pieces fit together

1. A **connection** tells Groundzero how to reach a source or a target.
2. Datasets and ETL jobs read and write through connections. Tables written to a **lakehouse warehouse** can be queried with SQL.

3. Each time a job or dataset runs, Groundzero records **lineage**: which tables it read and wrote, and which columns map to which.
4. The **catalog** lists everything and lets you tag and describe it.
5. **Ontology Studio** uses lineage to propose business definitions that you review and publish for AI Chat and agents.

1.4 How to read the chapters

Most chapters follow the same pattern: the task and its outcome, numbered steps, a reference of fields and options, what happens next, tips, and troubleshooting as symptom, cause and fix. Messages in quotation marks are the exact text you see on screen.

Convention	Meaning
Bold	A label you see on screen (button, tab, field)
`Code`	Something you type, or a permission name
Tip box	A shortcut or good practice
Note box	Something worth knowing
Warning box	A limit, a surprise or a risk

NOTIFICATIONS

Success and error messages appear in the bottom-right corner. They stay for 10 seconds. Identical messages are shown once, and all are cleared when you go to another page.

1.5 Index of features

Feature	Chapter	Where to find it
Create, test, edit, delete a connection	Data source connections	Connections
Connector reference	Data source connections	Connector reference
Dashboards in workspaces	Files and workspaces	Workspaces
Browse, upload, download, delete files	Files and workspaces	Workspaces > My Files
Find resources by name, column or tag	Data catalog	Catalog
Descriptions (with AI drafts) and tags	Data catalog	Catalog > Overview
Warehouse, table and column tags	Data catalog	Catalog > Schema
Call a dataset or job from outside	Data catalog	Catalog > Contract
Create and delete warehouses	Lakehouse and warehouses	Lakehouse
Sessions, schema tree, running queries	Lakehouse and warehouses	Lakehouse > a warehouse
Page, download and export results	Lakehouse and warehouses	Query editor
Saved queries	Lakehouse and warehouses	Lakehouse > Saved Queries

Feature	Chapter	Where to find it
Table properties and maintenance	Lakehouse and warehouses	Tree > Properties
Data policies and Table Access Manager	Lakehouse and warehouses	Tree > table menu
Audit log	Lakehouse and warehouses	Tree > audit > logs
Backups, schedules, restore	Lakehouse and warehouses	Tree, Schedules, Backups
Open a table's map, depth, columns	Data lineage	Catalog > Data Lineage
Impact of a change	Data lineage	Details panel > Impact
Runs, owners, insights	Data lineage	Details panel
Complete movement, export	Data lineage	Map, Export
Create a model, review definitions	Ontology Studio	Catalog > Ontology Studio
Draft with AI, fix missing sources	Ontology Studio	Toolbar, Review
Concepts and join suggestions	Ontology Studio	Review
Publish, releases, drift, export	Ontology Studio	Status bar, Releases

1.6 Related guides

- **Lakehouse runtime guide**: SQL, tables, maintenance and examples.
- **Build documentation**: notebooks, ETL jobs, datasets, runtimes and MLOps.
- **Analyze, AI and Admin documentation**: visualizations, dashboards, chat and agents, settings, billing, Consumption Metrics and permission administration.

2

Data source connections

A **connection** stores the details Groundzero needs to reach an external system: a database, a data warehouse, object storage, a SaaS application, a streaming service, an AI model provider, an MCP server or a Groundzero lakehouse. Datasets, ETL jobs, the lakehouse and AI features refer to connections by name, so you set a connection up once and reuse it everywhere.

Open **Connections** in the top navigation bar.

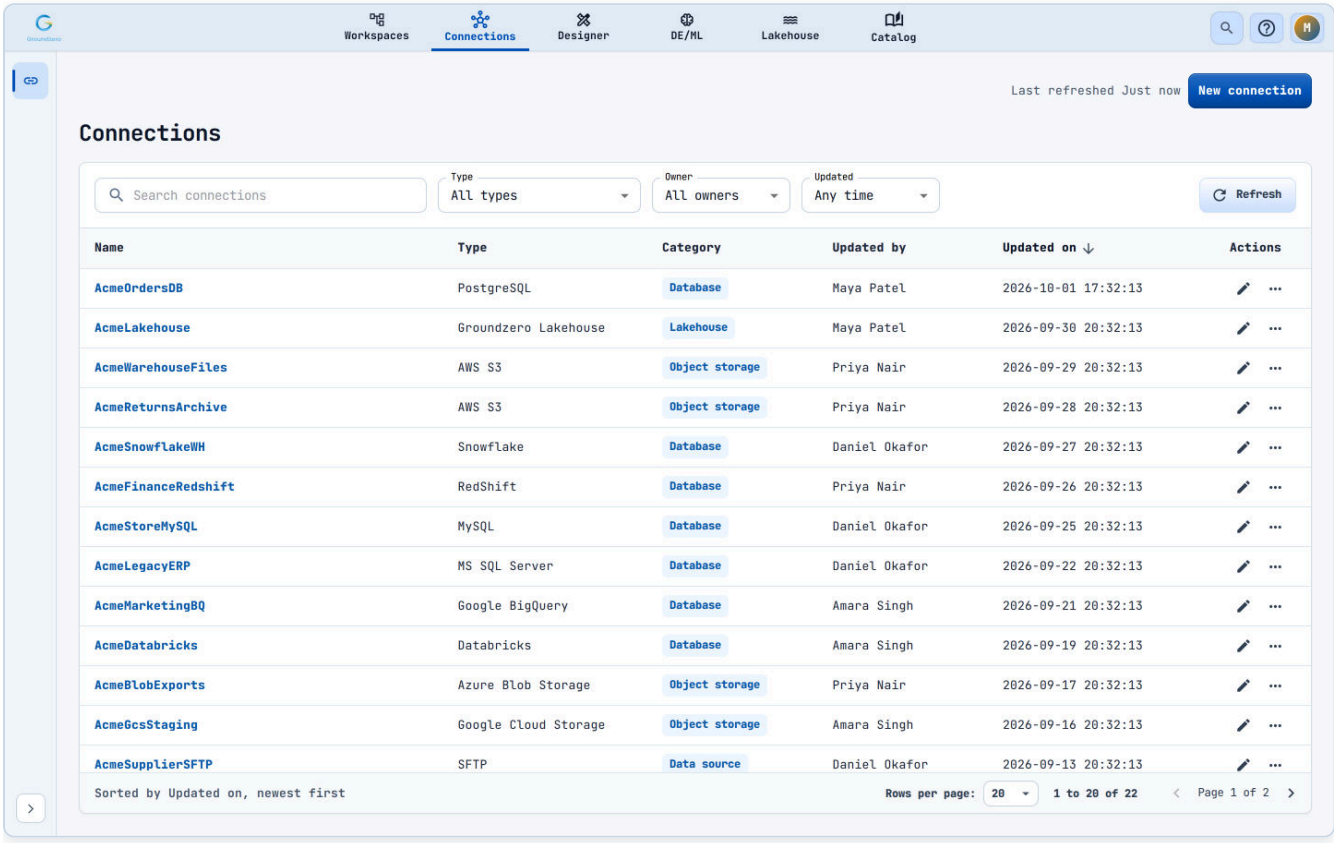


Figure 1. The Connections page lists every connection you are allowed to see, with search, three filters and a sortable table.

2.1 Overview

2.1.1 What you can do

Task	Where
Create a connection (66 connector types)	**New connection**
Check that a connection works	**Test Connection** inside the dialog
See how a connection is set up (secrets stay hidden)	Click the connection name
Change a connection	Pencil icon, or **Edit** in the details dialog
Delete a connection	Row menu, **Delete**
Find a connection	Search box, three filters and sortable columns

2.1.2 When to use connections

- You want a dataset (Designer) or an ETL job (DE/ML) to read from or write to an external system.
- You want Groundzero AI Chat or Agent Builder to use a model provider, an MCP server or another Groundzero agent service.
- You want the Lineage page to explain where data came from; lineage identifies assets by the connection they live on.
- You want to browse an object-storage bucket under **My Files** (see the Files and workspaces chapter).

2.1.3 Who can do what

Every signed-in user can open the page, but what you see and can do depends on your **permissions on the resource type "Connection"**. An administrator grants them in the admin console (Access Manager).

Action	Permission needed (either one)	Message when denied
List connections	`All` or `List`	You do not have "All" or "List" permissions on Connection.
View details	`All` or `Describe`	You do not have "All" or "Describe" permissions on Connection.
Create	`All` or `Add`	You do not have "All" or "Add" permissions on Connection.
Edit	`All` or `Edit`	You do not have "All" or "Edit" permissions on Connection.
Delete	`All` or `Delete`	You do not have "All" or "Delete" permissions on Connection.

An administrator can grant permissions to a user or a group, either for all connections or for one connection. An explicit deny wins. Whoever creates a connection automatically receives `All` on it. If you see a "You do not have ... permissions" message, ask an administrator for the permission it names.

2.2 Create a connection

Before you start, have the host name or address, the account or user name and the password or key for the system you want to reach. Ask the owner of that system if you do not know them.

1. Click **New connection**. The **Add Connection** dialog opens with a single field, **Connection Type**. **Save** and **Test Connection** are disabled.

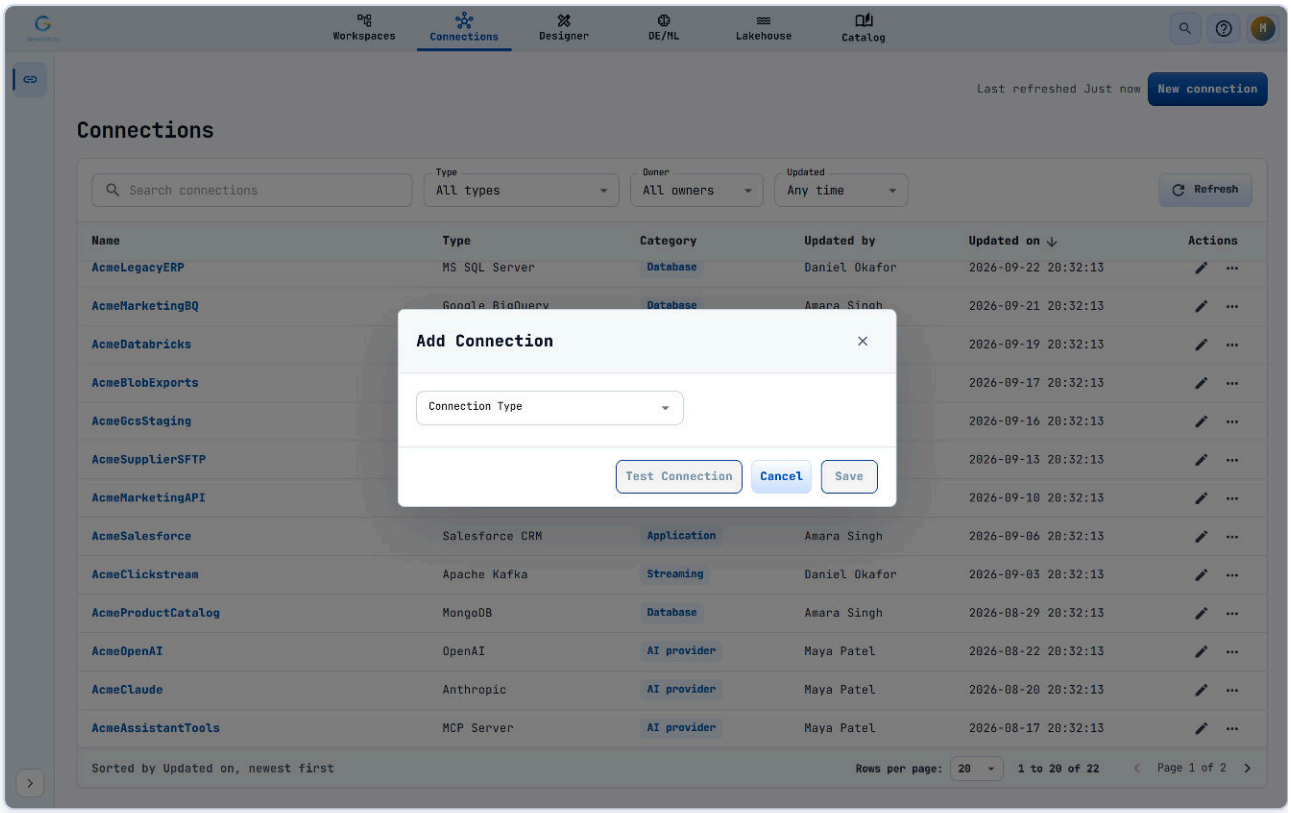


Figure 2. The Add Connection dialog before a type is chosen.

2. Click **Connection Type** and type to filter the alphabetical list, or scroll. Each option shows the connector logo.

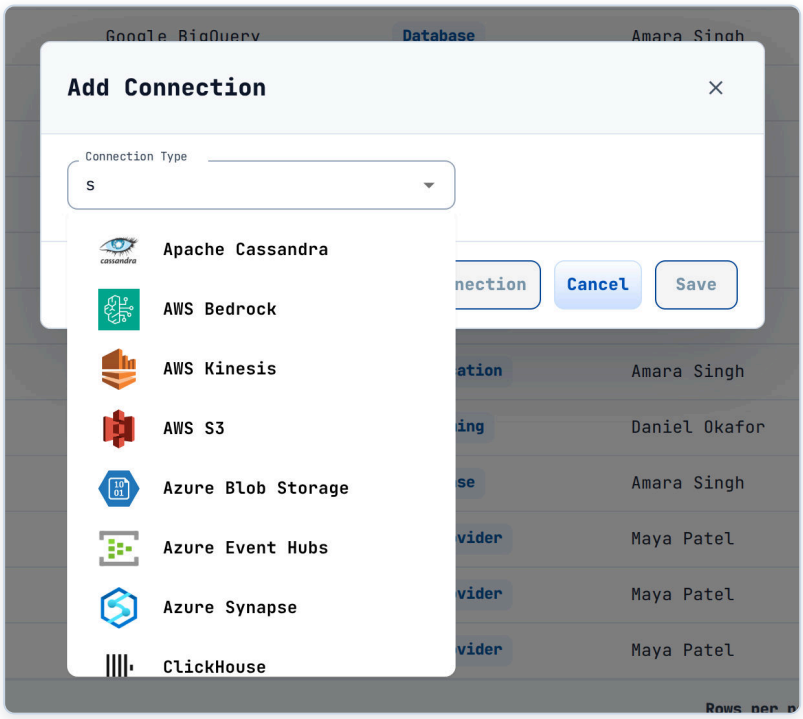


Figure 3. Typing in Connection Type narrows the list of connectors.

3. Pick a type. The form for that connector appears. The first field is always **Connection Name**. Fill in the remaining fields (see the connector reference below).

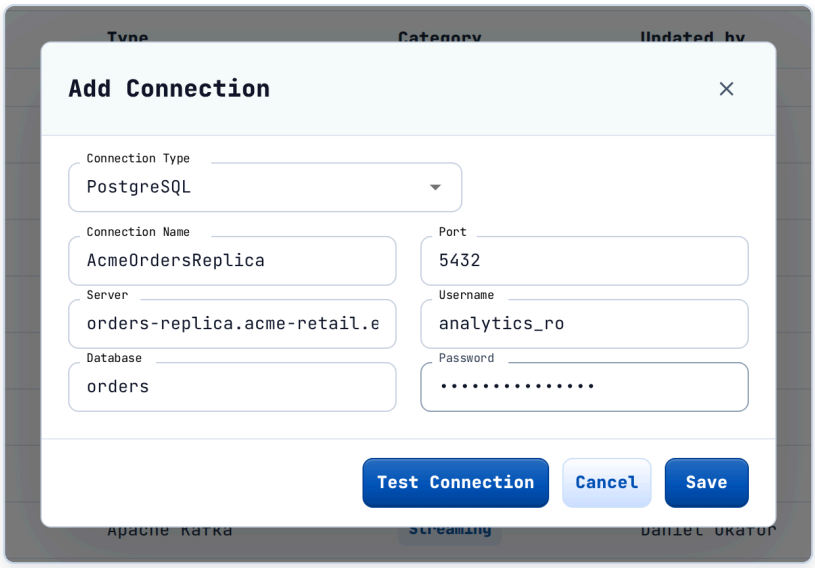


Figure 4. A PostgreSQL connection with its fields filled in; the default port 5432 is pre-filled.

4. Click **Test Connection** (optional but recommended). A success or failure message appears as a notification.

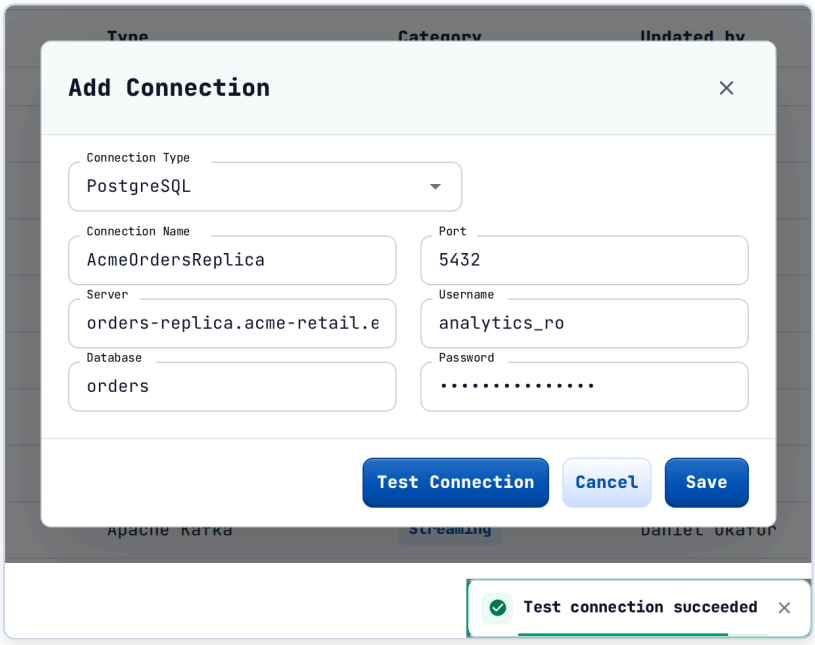


Figure 5. A successful test shows the message returned by the connector.

5. Click **Save**. The dialog closes, the list reloads and the notification "Connection created successfully" appears. Your new connection is now in the list, and datasets and jobs can use it by name.

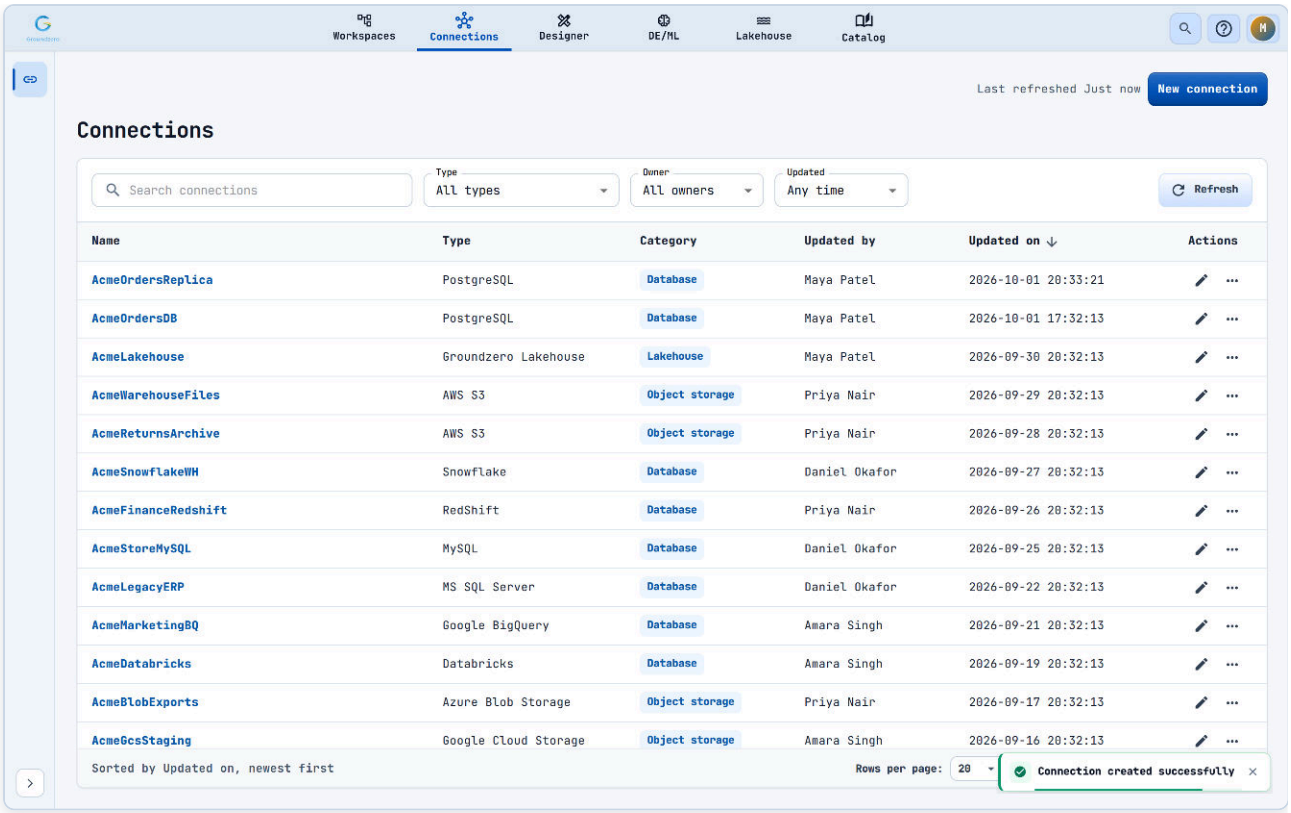


Figure 6. After saving, the new connection appears in the list.

2.2.1 Example: connect PostgreSQL in five steps

1. Click **New connection** and choose **PostgreSQL**.
2. Enter **Connection Name** `acme-orders-db`.
3. Enter **Server** `orders.acme-retail.example.com`, **Database** `orders`, keep **Port** `5432`, and enter **Username** `analytics_reader` and the **Password**.
4. Click **Test Connection**. You should see "Test connection succeeded".
5. Click **Save**. `acme-orders-db` appears in the list.

2.2.2 Connection name rules

A name must start with a letter and may contain letters, digits and single hyphens in the middle: no spaces, no underscores, no hyphen at the end and no two hyphens in a row. Names are unique per site.

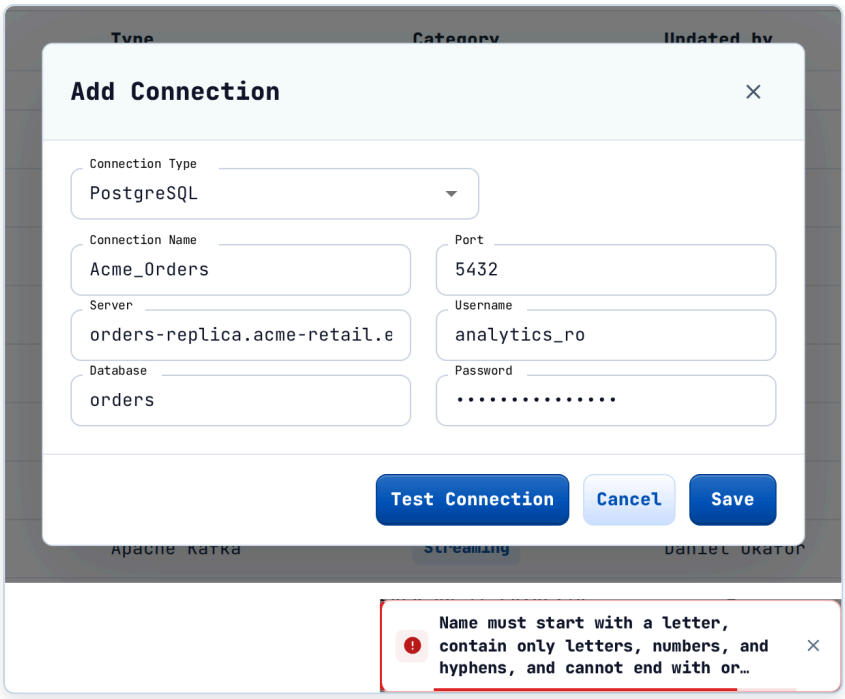


Figure 7. An underscore in the name is rejected with a plain-language message.

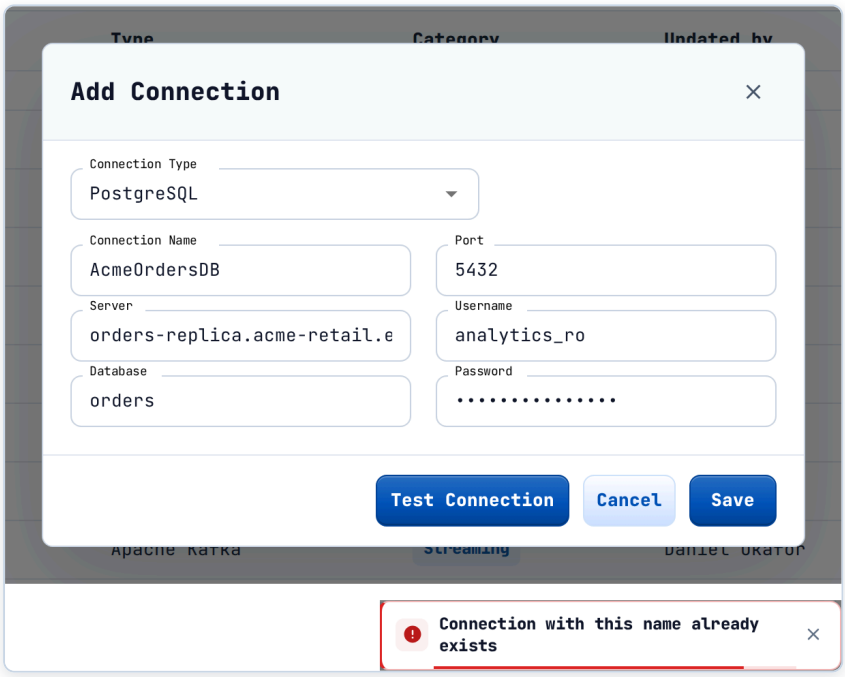


Figure 8. Creating a second connection with an existing name is rejected.

2.2.3 When Save and Test Connection are enabled

Both buttons stay disabled until all of these are true:

- you picked a connection type;
- **Connection Name** is not blank;
- every field is filled in, except fields labelled "(optional)". Numbers (including 0) and True or False choices count as filled.

If **Save** stays disabled, look for a field that is not labelled "(optional)" and is still empty. This includes fields that look unimportant, such as a pre-filled default you cleared.

2.2.4 Messages you may see while creating

Message	Cause	What to do
Fill all required connection details before saving	The form is incomplete.	Fill every field that is not marked "(optional)".
Name must start with a letter, contain only letters, numbers, and hyphens, and cannot end with or contain consecutive hyphens.	The name breaks the name rule.	Rename it, for example `acme-orders-db`.
Connection with this name already exists (also shown as "Connection name already exists")	Duplicate name.	Choose another name.
AI Chat Storage is already configured	Only one AI Chat Storage connection may exist.	Edit the existing one.
AICHATSTORAGE must use the reserved name GZ-CHAT-STORAGE, and that name cannot be used by other connection types.	Reserved name.	Use the name the form asks for.
You do not have "All" or "Add" permissions on Connection.	Missing permission.	Ask an administrator.
Internal server error occurred.	A failure on the server.	Try again; contact your administrator if it persists.
An error occurred while creating connection	No more detail was returned.	Try again.

2.3 Test a connection

Test Connection uses the values currently in the form and never saves anything. While it runs, the button reads "Testing...". Test before you save, and again after you edit.

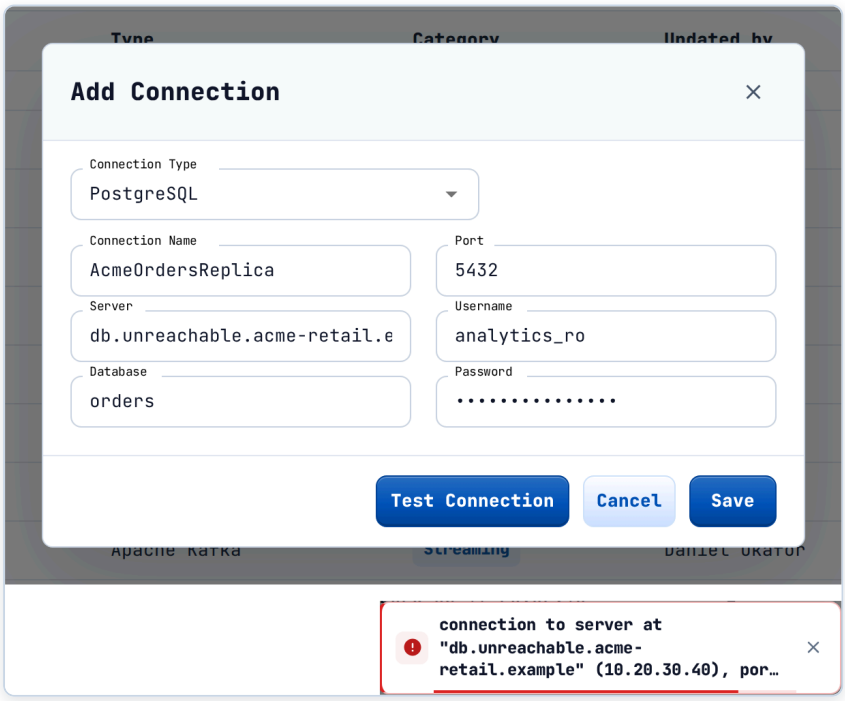


Figure 9. A failed test shows the driver's own error text.

What a successful test says, by family:

Family	Success message
Databases and warehouses	"Test connection succeeded" (some connectors say "Connection successful" or, for Cassandra, "Connected to Cassandra version <v>")
Object storage	AWS S3 and AI Chat Storage check the bucket; Azure Blob checks the container; Google Cloud Storage says "Connection tested successfully"
Streaming	Kafka and Kinesis say "Successfully created a connection."; Event Hubs says "Connection tested successfully"
AI model providers	"Connection successful". **Custom LLM** only checks your saved fields and says "Configuration valid..."
MCP Server	"Test connection succeeded. Found N tool(s)."
Groundzero Agents	"Connection successful"
Groundzero Lakehouse	"Connection successfully tested". A failure shows the message in quotes, for example `"Invalid parameters"`

When a test fails, read the message first. Typical ones are "Bucket does not exist", "Access denied to bucket" and "Container does not exist" for storage. For a database, the message is the database's own error (wrong password, host not reachable). If the message is only "Error occurred while testing connection", check the host, port and credentials.

When you edit an MCP Server connection and its token or header values are still masked, the test is blocked with "Re-enter or replace saved MCP credentials before testing this connection."

2.4 Find a connection

The page loads every connection you may list. The header shows **Last refreshed** (Just now, N min ago, and so on) and the **New connection** button.

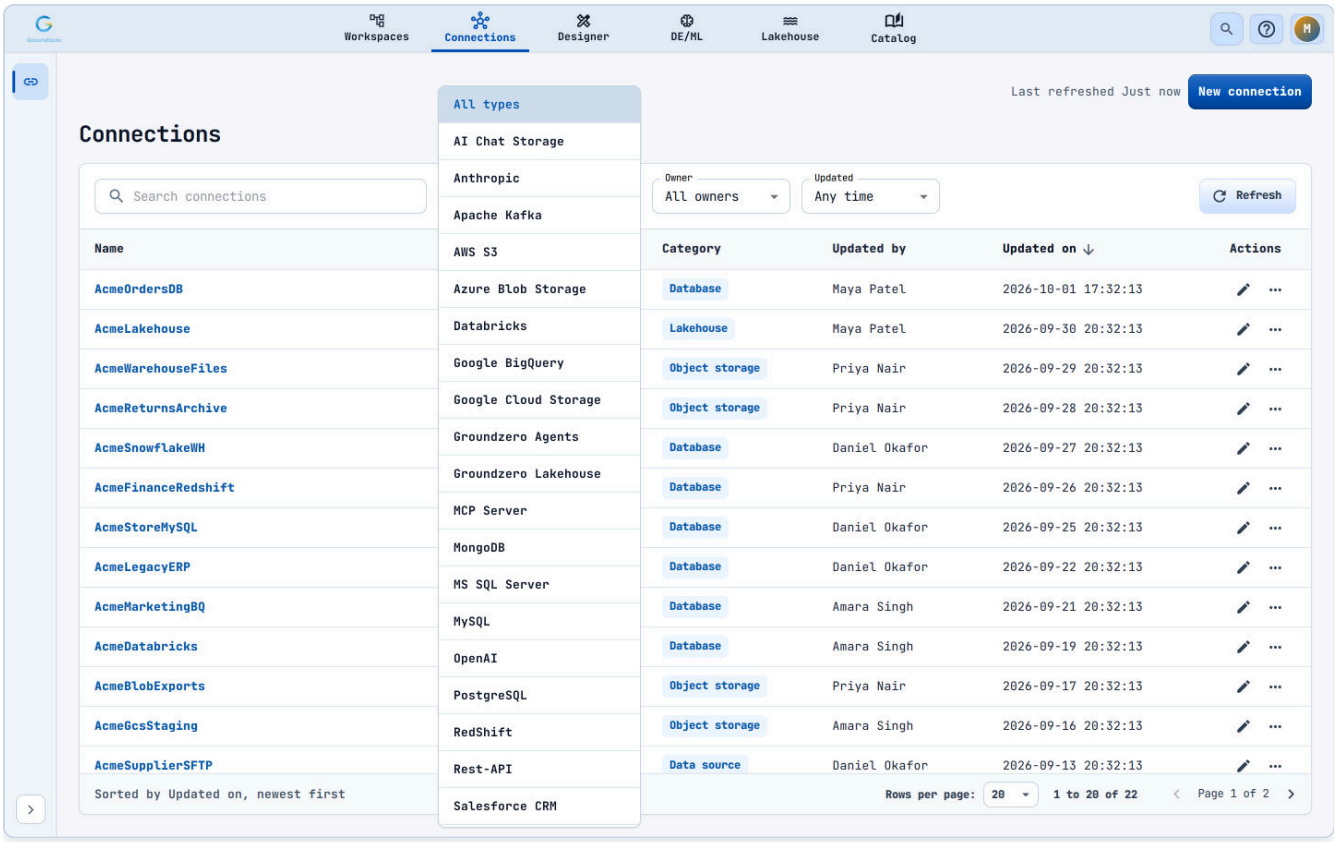


Figure 10. The Type filter lists every connector type that appears in your loaded rows.

2.4.1 The toolbar

Control	What it does
Search connections	Case-insensitive text search across name, type, category, "Updated by", workspace and description. Press Esc or use **Clear search** to empty it.
Type	"All types" plus each type present in the loaded rows.
Owner	"All owners" plus each distinct "Updated by" person. It filters on the last person who modified the connection.
Updated	Any time, Today, Last 7 days, Last 30 days (rolling, based on the "Updated on" date).
Refresh	Reloads the list (disabled while loading).

Active filters appear as removable chips (for example `Type: PostgreSQL`) with a **Clear all** button. Changing a filter returns you to page 1.

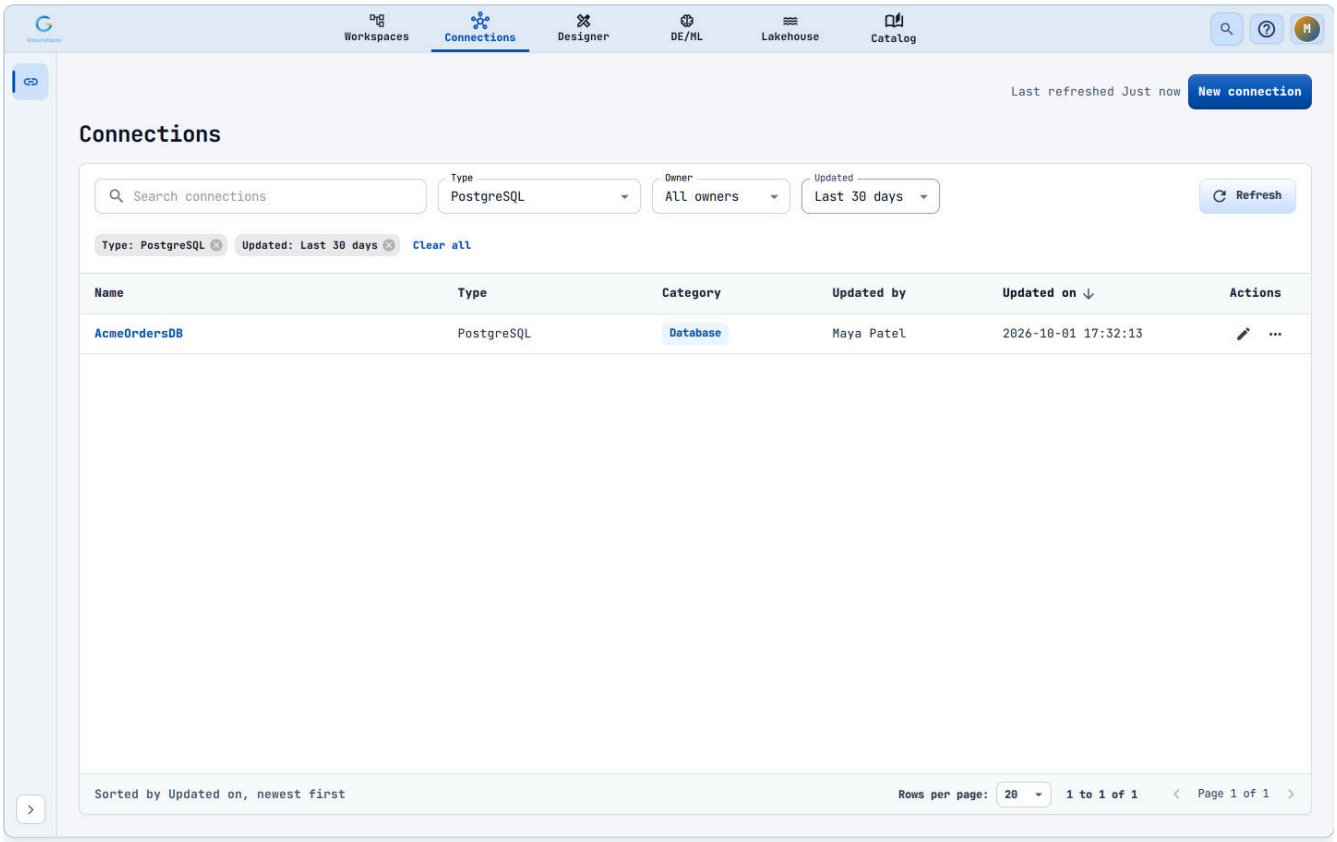


Figure 11. With a Type and an Updated filter active, matching rows and removable chips are shown.

2.4.2 The table

Column	Content	Sortable
Name	Link-style button that opens the details dialog.	Yes
Type	Display name of the connector, for example PostgreSQL.	Yes
Category	Chip: AI provider, Lakehouse, Object storage, Database, Application, Streaming or Data source.	Yes
Updated by	The last person who changed the connection ("priya.nair" is shown as "Priya Nair").	Yes
Updated on	`YYYY-MM-DD HH:mm:ss` in your local time. This is the default sort, newest first.	Yes
Actions	Pencil (**Edit**) and "..." menu (**View details**, **Delete**).	No

Click a column header to sort ascending, click again to reverse. The footer states the sort (for example "Sorted by Name, ascending"). Page size is 20, 50 or 100 rows. Rows are keyboard focusable: press Enter on a focused row to open its details.

How the Category chip is chosen. AI provider: OpenAI, Anthropic, Bedrock, Cohere, Custom LLM, Fireworks, Gemini, Hugging Face, MCP, Groundzero Agents, Together, xAI. Lakehouse: lakehouse and Iceberg connections. Object storage: S3, Google Cloud Storage, Azure Blob and AI Chat Storage. Database: PostgreSQL, SQL Server, MySQL, Oracle, Redshift, Snowflake, BigQuery, Databricks, MongoDB and similar. Application: HubSpot, Salesforce, ServiceNow, Zoho, Shopify. Streaming: Event Hubs, Kafka, Kinesis. Everything else, for example Athena or Rest-API, is shown as "Data source".

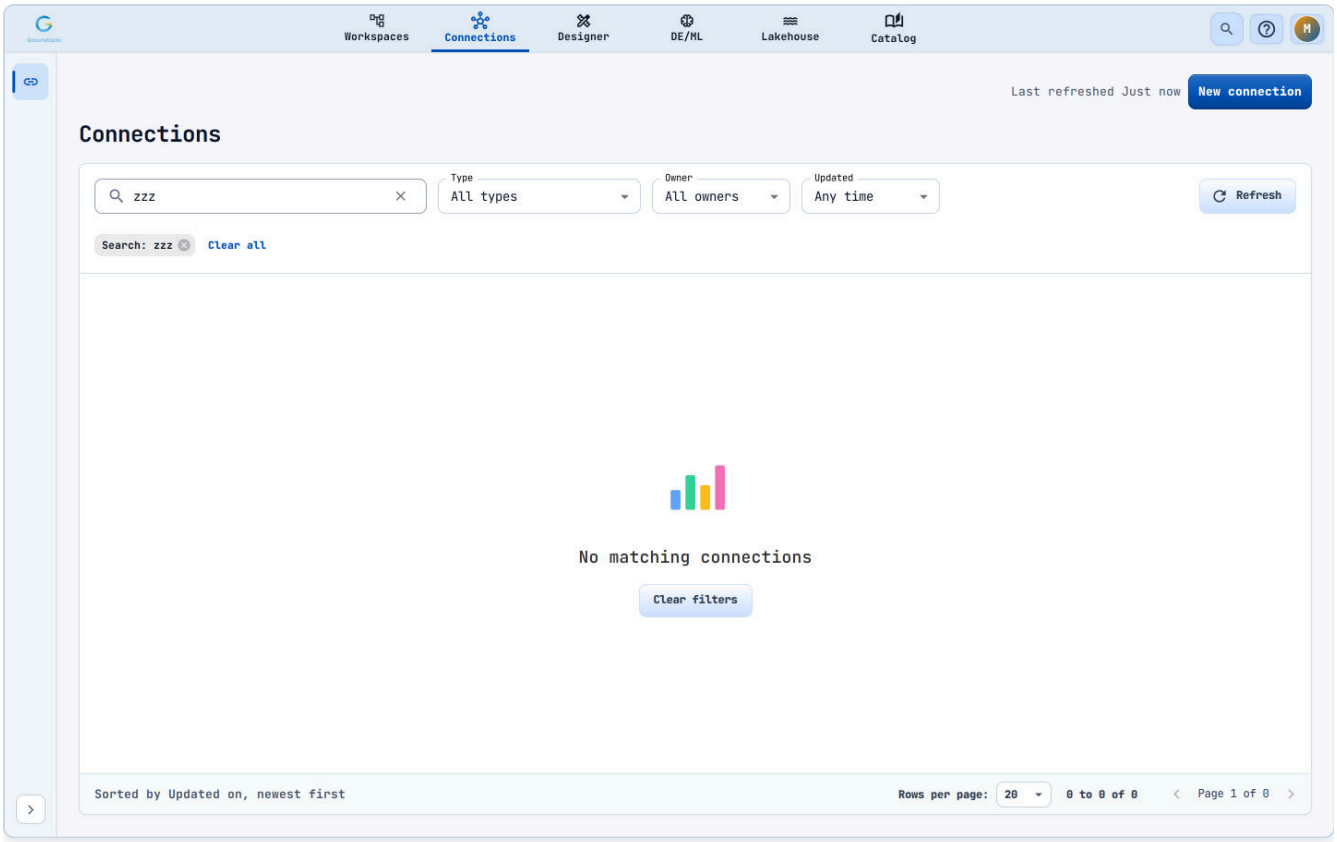


Figure 12. When nothing matches, an empty state offers Clear filters.

SYSTEM CONNECTION

The page also lists the system connection `GZ-CHAT-STORAGE` (type AI Chat Storage) when it exists. Other pages hide it.

2.5 View connection details

Click a connection name, choose **View details** from the "..." menu, or press Enter on a focused row.

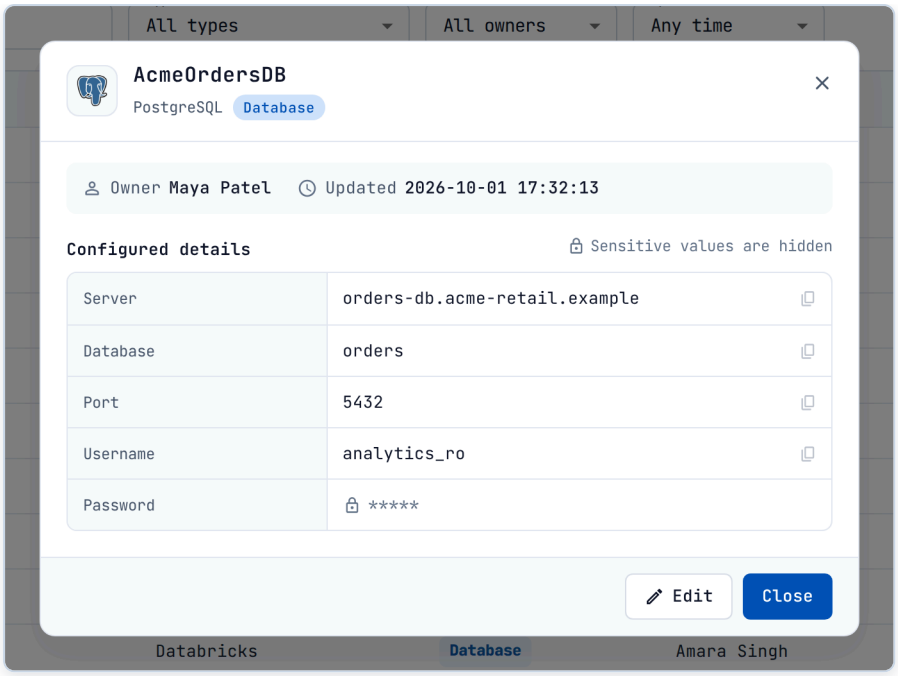


Figure 13. The details dialog shows the configuration. Secret values are replaced by a lock and five asterisks.

The dialog shows the connector logo, name, type and category, a summary (Owner, Updated; Workspace only if present), and **Configured details**: one row per setting, with a readable label such as Bucket Name. Values that look secret (password, secret, token, credential, private key, access key, API key, account key, client secret, security token) show a lock and `*****`, and nobody, including an administrator, can display them again. Other values have a copy button. The footer buttons are **Edit** and **Close**.

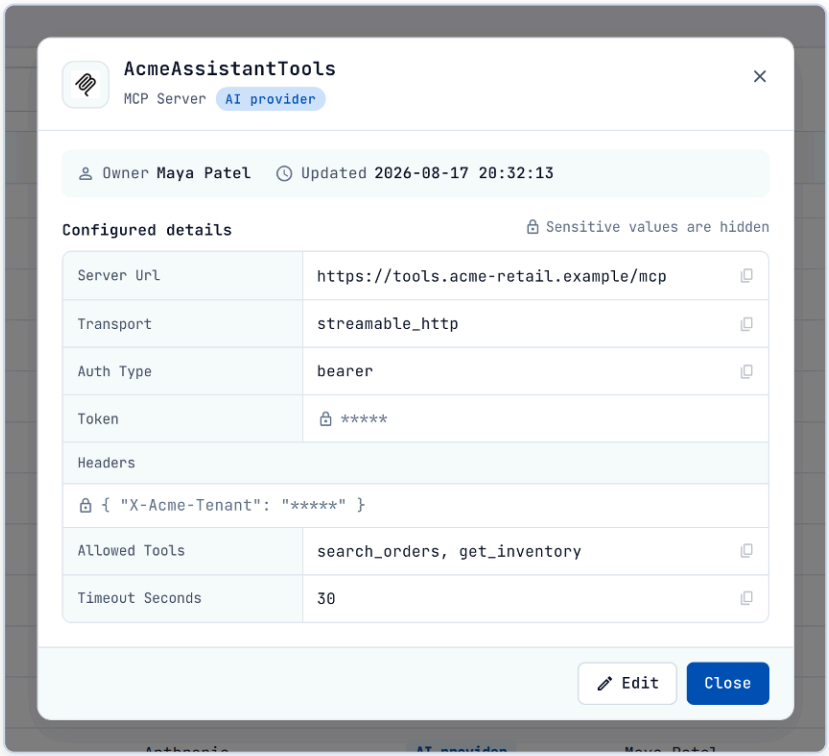


Figure 14. An MCP Server connection masks its token and every header value.

2.6 Edit a connection

Click the pencil icon (**Edit <name>**) or **Edit** in the details dialog.

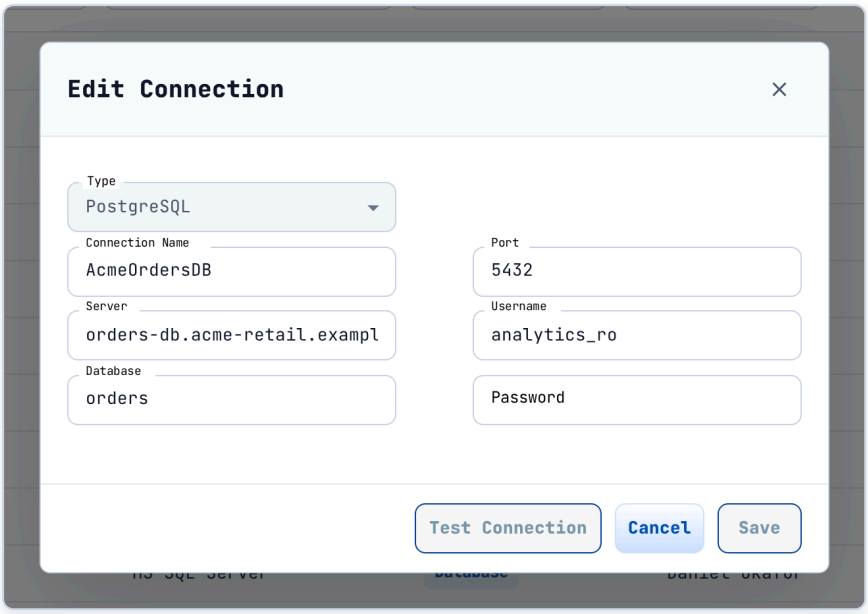


Figure 15. The Edit Connection dialog: the Type is locked and the password must be typed again.

1. Change the fields you need. The **Type** cannot be changed; **Connection Name** can.
2. Type every secret again. Non-secret values are pre-filled, but **secrets arrive empty**, so **Save** and **Test Connection** stay disabled until the form is complete. AI model provider forms and MCP Server show existing secrets as `••••••••••••` and keep them if you leave them unchanged.
3. Click **Test Connection**, then **Save**. **Save** replaces the whole configuration. On success the notification reads "<Name> edited successfully".

Datasets and jobs refer to a connection by name, so renaming a connection can break them. Edit errors: the message returned, or "Unable to edit connections"; "Fill all required connection details before saving" when the form is incomplete. For MCP Server, if the stored secret is missing you see "Saved MCP credentials could not be preserved. Re-enter or replace them and try again."

2.7 Delete a connection

1. Open the row menu (**Open actions for <name>**) and choose **Delete**.
2. Confirm in **Delete Connection**: "Are you sure you want to delete this connection <name>?".

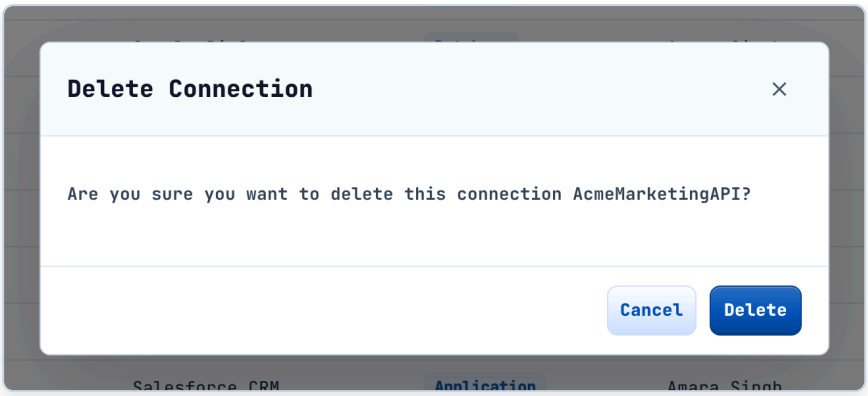


Figure 16. The delete confirmation names the connection.

On success the notification reads "Connection deleted successfully" and the list reloads. Failures show the message returned or "Unable to delete connections".

> **DELETING DOES NOT CHECK REFERENCES**
>
> Deleting removes the connection and its permissions without checking whether datasets, jobs or agents still use it. Check usage (for example with Data Lineage) before deleting.

2.8 Connector reference

The **Connection Type** list offers 66 connector types in families. For each family this section explains the fields they share once, then lists the fields each connector shows. Field names are the labels in the form. " (masked)" means the input hides what you type. "(optional)" means you may leave it empty. Defaults are what the form pre-fills.

Tips that apply to every connector:

- Create a dedicated user or key for Groundzero with only the access it needs. A read-only database user is enough for analytics.
- Copy values exactly. A trailing space or a wrong region is the most common reason a test fails.
- Use **Test Connection** before you save.

2.8.1 Relational databases

Use these for operational databases and classic SQL servers.

Shared fields: **Server** (host name or address), **Database**, **Port** (the form pre-fills the usual port), **Username** and **Password** (masked). Ask your database administrator for the host and for a login that can read the tables you need.

Connector	Fields	Notes
PostgreSQL	Server, Database, Port (default 5432), Username, Password	
MySQL	Server, Database, Port (default 3306), Username, Password	
MS SQL Server	Server, Database, Port (default 1433), Username, Password	
Oracle	Server, Database, Port (default 1521), Username, Password	

Connector	Fields	Notes
RedShift	Server, Database, Port (default 5439), Username, Password	
Azure Synapse	Server, Database, Username, Password, Port (default 1433)	
CockroachDB	Server Hostname, Database Name, Username, Password, Port Number	
Timescale	Host, Port, Username, Password, Database	
SingleStore	Server, Port (default 3306), Username, Password, Database	**Database** is masked although it is not a secret.
SkySQL	Server, Port, Username, Password, Database, SSL Disabled	**Database** is masked. **SSL Disabled** is a True or False choice (default False).
DB2	Host Name, Port, Username, Password	
SAP HANA	User, password, Database, Server, Port	
Teradata	Database, Password, Server, Username	
Exasol	DSN, User, Password, Schema	
ClickHouse	ClickHouse Host, Username, Password, Port (optional), Boolean for HTTPS (optional), Database Name (optional)	

Example: connect PostgreSQL. See "Example: connect PostgreSQL in five steps" above. The other relational connectors work the same way with their own port.

Common mistakes. Using the wrong port; pointing to a host that is only reachable inside your company network; leaving **Password** empty when editing, because secrets must be typed again.

2.8.2 Cloud data warehouses and lakehouse engines

Use these for warehouses and analytics engines in the cloud.

Connector	Fields	Notes
Snowflake	Account, Database, Username, Password, Schema	**Account** is your Snowflake account identifier, the first part of your Snowflake sign-in address.
Google BigQuery	Credentials File, Parent Project	Paste the service-account JSON into **Credentials File** (a single-line text box, not a file picker).
Databricks	Databricks Server Hostname, HTTP Path, Access Token, Catalog (optional), Schema (optional)	Copy the hostname and HTTP path from your Databricks compute settings. **Access Token** shows what you type.
Athena	Data Source, Database, Port (default 443), S3 Staging Directory, Access Key, Secret Access Key	**S3 Staging Directory** is the S3 location where Athena writes query results.
Firebolt	Account Name, Client ID, Client Secret, Database, Engine Name	
MotherDuck	Token, Database	
Apache Hudi	Storage Type, Base Path, S3 Access Key, S3 Secret Key, S3 Endpoint	**Storage Type** offers one choice, "S3".

Connector	Fields	Notes
Groundzero Lakehouse	Lakehouse URL, Warehouse Name, Database Name, Site Name, Service Account Username, Service Account Password	See "Groundzero Lakehouse" below.

Example: connect Snowflake. Choose **Snowflake**, name it `acme-snowflake`, enter **Account** `acme-retail01`, **Database** `ANALYTICS`, **Schema** `PUBLIC`, **Username** `GZ_READER` and the **Password**, then click **Test Connection** and **Save**.

Common mistakes. Pasting the full web address into **Account**; using a user without access to the chosen database or schema.

2.8.3 Other databases and search or vector stores

Use these for document, graph, key-value, search, time-series, log and vector stores.

Connector	Fields	Notes
MongoDB	Username, Server, Password, Cluster	
Apache Cassandra	Contact Points, Port, Username, Password, Keyspace	
ScyllaDB	Contact Points/Hosts, Port, Username, Password, Keyspace Name, Local Datacenter (optional)	
DynamoDB	Access Key Id, Secret Access Key, Region	
CosmosDB	Account Endpoint, Account Key, Database, Container	
Couchbase	Connection String, Username, Password, Bucket Name, Profile (optional)	
ElasticSearch	Host, Port, API Key	**API Key** shows what you type.
InfluxDB	Host, Token, Organization, Database	
Neo4j	URL, Username, Password	
Pinecone	API Key, Index Name	
Weaviate Cloud	Cluster URL, API Key, Headers Dictionary (optional)	**API Key** shows what you type.
Splunk	Splunk Host, Port (default 8089), Username, Password, Scheme (default https)	

Common mistakes. Using a key for a different region or project; leaving the default port when your server uses another.

2.8.4 Object storage and file sources

Use these for buckets, drives and file servers. Only AWS S3 buckets can be browsed under **My Files**.

Connector	Fields	Notes
AWS S3	Bucket Name, Region, Access Key, Secret Access Key	Region is the AWS region of the bucket, for example `us-east-1`.

Connector	Fields	Notes
AI Chat Storage	Bucket Name, Region, Access Key, Secret Access Key	Uses the AWS S3 form. See "AI Chat Storage" below.
Azure Blob Storage	Account Name, Account Key, Container Name	**Account Key** shows what you type.
Google Cloud Storage	Bucket Name, Credentials File	Paste the service-account JSON.
Google Drive	Credentials File, Folder ID (optional)	Paste the service-account JSON.
OneDrive	Client ID, Client Secret, Tenant ID, Drive ID, Folder Path (optional)	
Sharepoint	Tenant ID, Client ID, Client Secret, SharePoint Site URL, Refresh Token (optional)	
SFTP	Username, Server, Password, Port	

Example: connect AWS S3. Choose **AWS S3**, name it `acme-landing-bucket`, enter **Bucket Name** `acme-retail-landing`, **Region** `us-east-1`, then the **Access Key** and **Secret Access Key** of a user that can read the bucket. **Test Connection** should report success. After you save, the bucket appears under **My Files**.

Common mistakes. A wrong bucket name or region; a key without access to the bucket ("Access denied to bucket"); JSON containing an empty value in **Credentials File**, which keeps **Save** disabled because every value must be filled.

2.8.5 Streaming

Use these for event streams.

Connector	Fields	Notes
Apache Kafka	Bootstrap Servers, Security Protocol, API Key, API Secret, SASL Mechanism, Starting Offsets	
AWS Kinesis	Stream Name, Region, Access Key, Secret Key	
Azure Event Hubs	Event Hub Namespace, Shared Access Key Name, Access Key, Event Hub Name	

Example: connect Kafka. Choose **Apache Kafka**, name it `acme-orders-stream`, enter your **Bootstrap Servers** (host:port, comma separated if several), the **Security Protocol** and **SASL Mechanism** your cluster requires, then the **API Key** and **API Secret**, and **Starting Offsets**. Click **Test Connection**.

Common mistakes. Mismatched **Security Protocol** and **SASL Mechanism**; a key that has no access to the topic.

2.8.6 Applications and APIs

Use these for SaaS applications and generic web APIs.

Connector	Fields	Notes
Rest-API	Base URL, Endpoint, Endpoint Params (optional), AuthType	**AuthType** is free text.
Salesforce CRM	Username, Password, Security Token, Domain	

Connector	Fields	Notes
ServiceNow	URL, Username, Password	
Hubspot	Access Token	**Access Token** shows what you type.
Shopify	Merchant/Store Name, API Version, Access Token	**Access Token** shows what you type.
Zoho CRM	Client Id, Client Secret Id, Refresh Token, URL (default https://www.zohoapis.in/)	**Refresh Token** shows what you type.
Microsoft Dataverse	Tenant ID, Client ID, Client Secret, Dataverse Environment URL	

OAuth-style connectors (OneDrive, Sharepoint, Microsoft Dataverse, Zoho CRM) take the client ID, secret, tenant and refresh-token values pasted into text boxes. Copy them from the app registration in the source system.

Example: connect a REST API. Choose **Rest-API**, name it `acme-pricing-api`, enter **Base URL** `https://api.acme-retail.example.com`, **Endpoint** `/v1/prices` and **AuthType**. Click **Test Connection**.

Common mistakes. Putting the endpoint path in **Base URL**; an expired token.

2.8.7 AI model providers

Use these for the models behind AI Chat and Agent Builder. Model-provider connections feed the model picker in AI Chat.

Shared fields: **API Key** (masked), a base URL that is pre-filled with the provider's standard address, and an optional **Model**.

Connector	Fields	Notes
OpenAI	API Key, Base URL (default https://api.openai.com/v1), Organization ID (optional), Project ID (optional), Model (optional)	
Anthropic	API Key, Base URL (default https://api.anthropic.com), API Version (default 2023-06-01), Workspace ID (optional), Model (optional)	
Azure OpenAI	API Key, Resource Endpoint, Deployment Name (optional), API Version (default v1), Management Token (masked, optional), Subscription ID (optional), Resource Group (optional), Account Name (optional), Management Base URL (default https://management.azure.com, optional), Management API Version (default 2025-06-01, optional)	
AWS Bedrock	Access Key, Secret Access Key, Region (default us-east-1), Assume Role ARN (optional), Model Job Role ARN (optional), Session Token (masked, optional), Model (optional)	
Cohere	API Key, Base URL (default https://api.cohere.com), Model (optional)	
Fireworks AI	API Key, Base URL (default https://api.fireworks.ai/inference/v1), Control Plane URL (default https://api.fireworks.ai, optional), Account ID (optional), Model (optional)	
Google Gemini	API Key, Base URL (default https://generativelanguage.googleapis.com/v1), Model (optional)	
Hugging Face	API Key, Hub Base URL (default https://huggingface.co), Inference Base URL (default https://router.huggingface.co/v1), Model (optional)	

Connector	Fields	Notes
Together AI	API Key, Base URL (default https://api.together.ai/v1), Model (optional)	
xAI	API Key, Base URL (default https://api.x.ai/v1), Model (optional)	
Custom LLM	Invocation endpoint, Model ID, Bearer API key, Context window (optional)	See "Custom LLM" below.

Example: connect OpenAI. Choose **OpenAI**, name it `acme-openai`, paste the **API Key**, keep the default **Base URL** and click **Test Connection**. Success reads "Connection successful". The model then shows up in the AI Chat model picker.

Common mistakes. A key from a different provider or project; changing **Base URL** by accident. An edit form shows saved keys as `............` and keeps them if you do not change them.

2.8.8 MCP Server and Groundzero Agents

Connector	Fields	Notes
MCP Server	Server URL, Transport, Auth Type, Bearer Token, Custom Headers JSON, Allowed Tools, Timeout Seconds	See "MCP Server" below.
Groundzero Agents	Agent API URL (pre-filled with your site's agent API URL), Service account username, Service account password (masked)	MCP and Groundzero Agents connections feed AI Chat tools and Agent Builder (see the Analyze/AI documentation).

2.9 Special connection types

2.9.1 MCP Server

The MCP form checks what you type. **Server URL** must be a public HTTPS URL without credentials or a fragment; otherwise the helper text "Use a public HTTPS URL without credentials or a fragment." appears. **Transport** is "Streamable HTTP" (default) or "Server-Sent Events". **Auth Type** is "No auth" (default), "Bearer token" (shows **Bearer Token**, masked) or "Custom headers" (**Custom Headers JSON** becomes required). **Allowed Tools** is a comma- or newline-separated list of tool names you permit. **Timeout Seconds** defaults to 30 and must be 1 to 300.

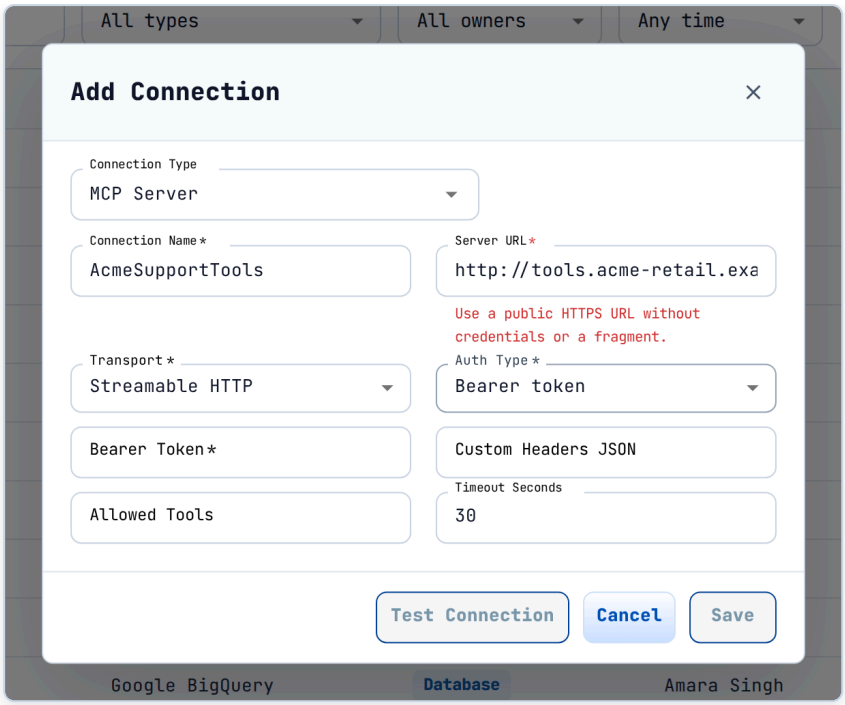


Figure 17. The MCP form flags a non-HTTPS URL and reveals the Bearer Token field when Auth Type is "Bearer token".

Test it: success reads "Test connection succeeded. Found N tool(s)." If it fails, check that the URL is public HTTPS and reachable.

2.9.2 Custom LLM

Use this for a model behind your own HTTPS endpoint.

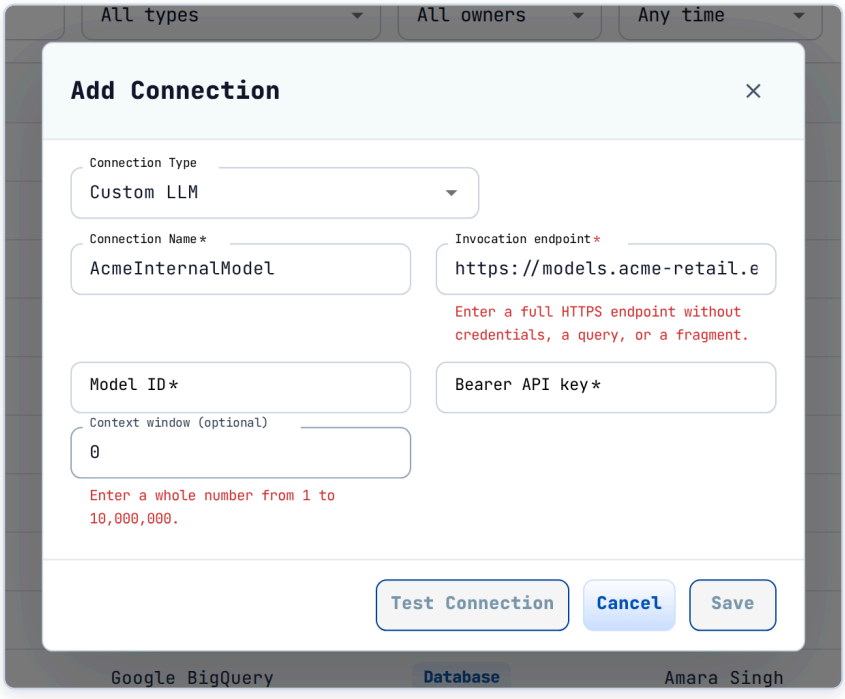


Figure 18. The Custom LLM form checks endpoint and context window as you type.

Field	Rule	Message
Invocation endpoint	HTTPS URL, up to 2,048 characters, no spaces or backslash, no credentials, query or fragment	"Enter a valid HTTPS endpoint without spaces." / "Enter a full HTTPS endpoint without credentials, a query, or a fragment." / "Enter a valid HTTPS endpoint."
Model ID	Up to 512 characters, no leading, trailing or control characters	"Enter the exact model ID without leading, trailing, or control characters."
Bearer API key	Visible ASCII, up to 8,192 characters	"Use visible ASCII characters without spaces, up to 8,192 characters."
Context window (optional)	Whole number 1 to 10,000,000	"Enter a whole number from 1 to 10,000,000."

2.9.3 Groundzero Lakehouse

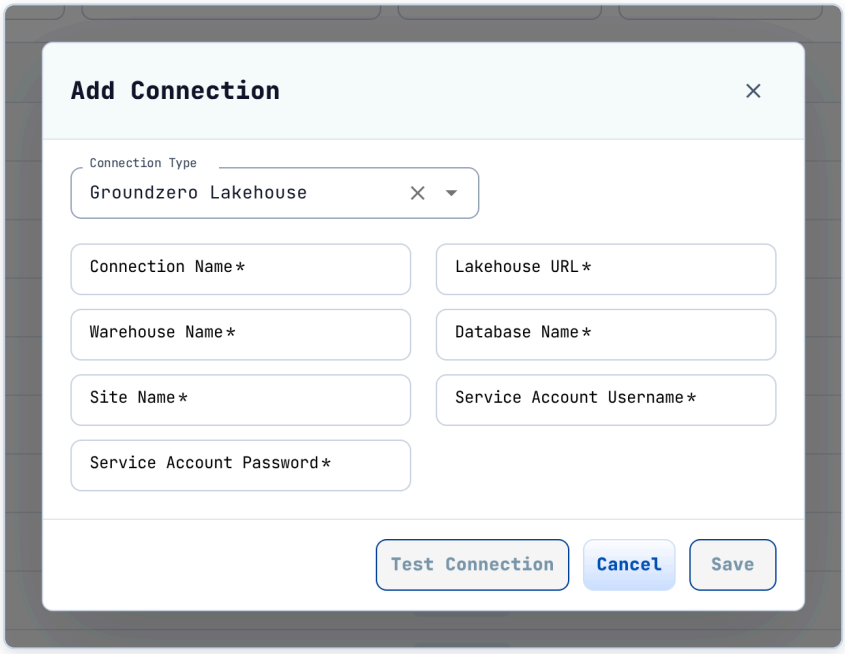


Figure 19. The Groundzero Lakehouse form asks for the lakehouse host, warehouse, database, site and a service account.

Lakehouse URL is a host name only (without `https://`). Warehouse and database names are free text. Use a service account with access to that warehouse. See the Lakehouse chapter for how warehouses are created and how to copy the Lakehouse URL.

2.9.4 AI Chat Storage

Uses the AWS S3 form. Its name is fixed to `GZ-CHAT-STORAGE`, only one may exist, and once it exists the type disappears from the type list.

2.10 Secrets

- After you save, secret values show only as `*****`.
- Some fields show what you type: Azure Blob **Account Key**, Databricks **Access Token**, ElasticSearch **API Key**, HubSpot **Access Token**, Shopify **Access Token**, Weaviate **API Key**, Zoho **Refresh Token**, and the

Credentials File fields of Google Cloud Storage, Google BigQuery and Google Drive. Avoid screen sharing while you fill them.

- When you edit a connection you retype its secrets, except for MCP Server and the AI model provider forms.

2.11 Limits

Item	Limit
Connection name	Letters, digits, single interior hyphens; starts with a letter; unique per site; `GZ-CHAT-STORAGE` is reserved
Page size	20, 50 or 100 rows; no cap on the number of connections
Custom LLM	Endpoint 2,048 characters; model ID 512; key 8,192; context window 1 to 10,000,000
MCP	Timeout 1 to 300 seconds; HTTPS URL required

2.12 Tips

> **TIP**
>
> Run **Test Connection** before **Save**. A test never stores anything and uses exactly what is in the form, so you can correct a typo and retest freely.

> **TIP**
>
> Use descriptive, hyphen-free names such as `AcmeOrdersDB`. Datasets and jobs refer to connections by name, so renaming a connection later can break references.

> **TIP**
>
> Give the connection a read-only database user if the use is analytics: the connection decides what every dataset and job built on it may do.

2.13 Troubleshooting

Symptom	Cause	Fix
Test Connection and **Save** are greyed out	Some field is empty	Fill every field without "(optional)"; when editing, retype the secrets
Name rejected	Underscore, space, or a hyphen at the end	Start with a letter and use letters, digits and single hyphens
"Connection with this name already exists"	Names are unique per site	Choose another name or edit the existing connection
"AI Chat Storage is already configured"	Only one `GZ-CHAT-STORAGE` is allowed	Edit the existing one

Symptom	Cause	Fix
"You do not have … permissions on Connection."	Missing permission	Ask an administrator for the permission named in the message
Saved but the bucket is missing under My Files	Only AWS S3 connections are listed there	See the Files and workspaces chapter
"Bucket does not exist" or "Access denied to bucket"	Wrong bucket name or region, or a key without access	Check the bucket name and region and the key's permissions
MCP test fails	The URL is not public HTTPS or not reachable	Use a public HTTPS URL; re-enter masked tokens or headers ("Re-enter or replace saved MCP credentials…")
Lakehouse test shows a quoted message	The message comes from the lakehouse, for example `"Invalid parameters"`	Fill username, password, warehouse and database

2.14 Related features

- **Files and workspaces**: browse S3 buckets that connections point to.
- **Lakehouse**: warehouses referenced by Groundzero Lakehouse connections.
- **Data catalog and Data lineage**: lineage identifies assets by the connection they belong to.
- **Designer datasets and DE/ML jobs** (Build documentation): read connections by name.

3

Files and workspaces

Groundzero has two places for files and folders, both under **Workspaces** in the top navigation bar:

- **Workspaces** hold dashboards you have imported, organised into named folders.
- **My Files** lets you browse, upload, download and delete files in your AWS S3 buckets without leaving Groundzero.

I want to…	Go to
Keep related dashboards together in a folder	Workspaces, then **New Workspace**
Put a published dashboard into a workspace	Open the workspace, then **Import Dashboard**
Upload a CSV file for a dataset or job	My Files, open a bucket, then **New** > **Upload File**
Download or clean up files in a bucket	My Files, open the bucket

You need two things in place before you start. For Workspaces, an administrator must have given you permission on workspaces (see "Workspace permissions" below). For My Files, you need an **AWS S3** connection that you are allowed to see (see the Data source connections chapter).

3.1 Organise dashboards in a workspace

A **workspace** is a named folder of imported dashboards. It is not general file storage; use My Files for that. Open **Workspaces** in the left rail.

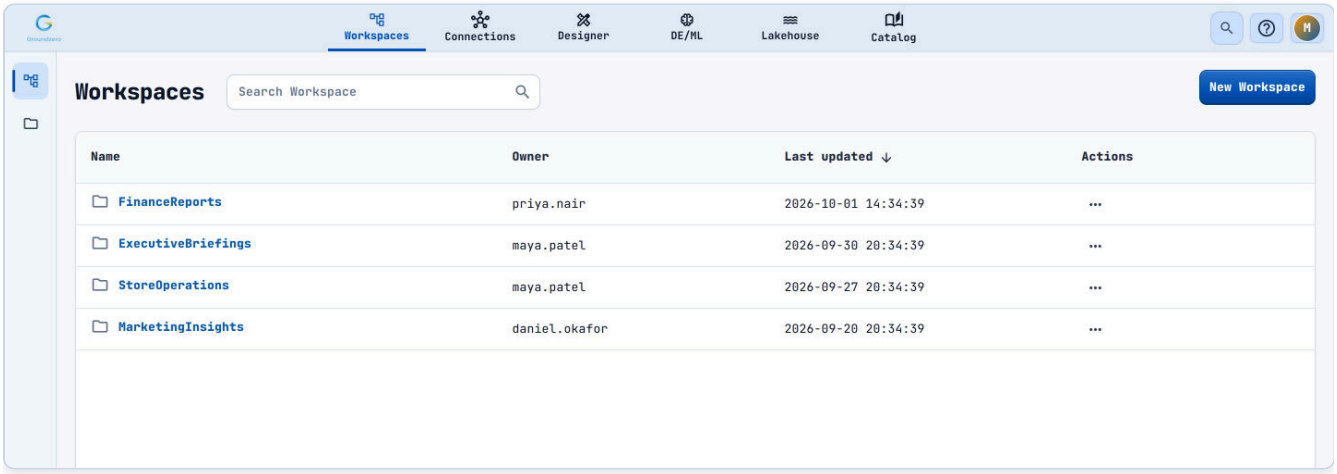


Figure 20. The Workspaces list shows each workspace with its owner and last-updated time.

The list shows the workspaces you may see, newest change first, with the **Owner**, the **Last updated** time and an **Actions** menu ("..."). Type in **Search Workspace** to filter by name. You can show 20, 50 or 100 rows per page.

3.1.1 Create a workspace

1. Click **New Workspace**.

2. Type a name. It must start with a letter and may contain letters, digits and single hyphens, for example `Finance-Reports`. No spaces, underscores, trailing hyphen or double hyphen.

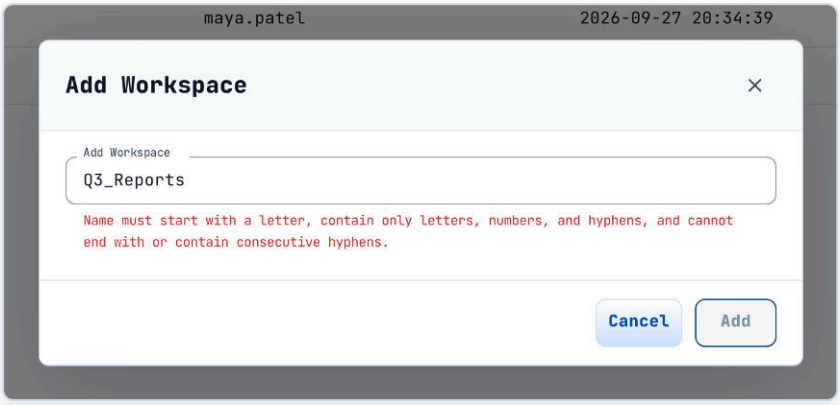


Figure 21. An underscore is rejected before you can save; the Add button stays disabled.

3. When the red helper text disappears, **Add** turns on. Click it.

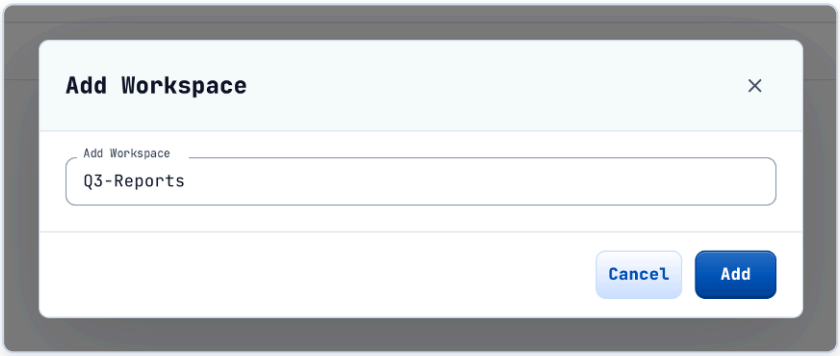


Figure 22. A valid name enables Add.

You see "<name> added successfully" and the workspace appears in the list. You are its owner.

If you see "Workspace with this name already exists", choose another name.

3.1.2 Import a dashboard into a workspace

Only **published** dashboards can be imported. The imported copy opens read-only.

1. Click the workspace name. A breadcrumb (**Workspaces** > name) appears, the owner column is now called **Created by**, and the main button is **Import Dashboard**.

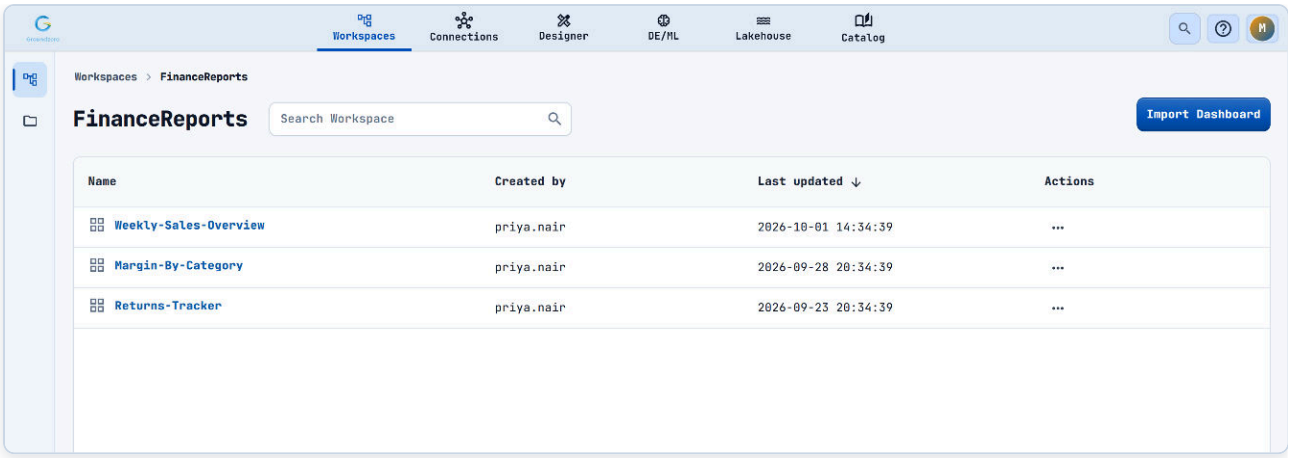


Figure 23. Inside a workspace the primary button is Import Dashboard and the owner column becomes Created by.

2. Click **Import Dashboard** and choose a dashboard from **Select a dashboard**. The list is searchable.

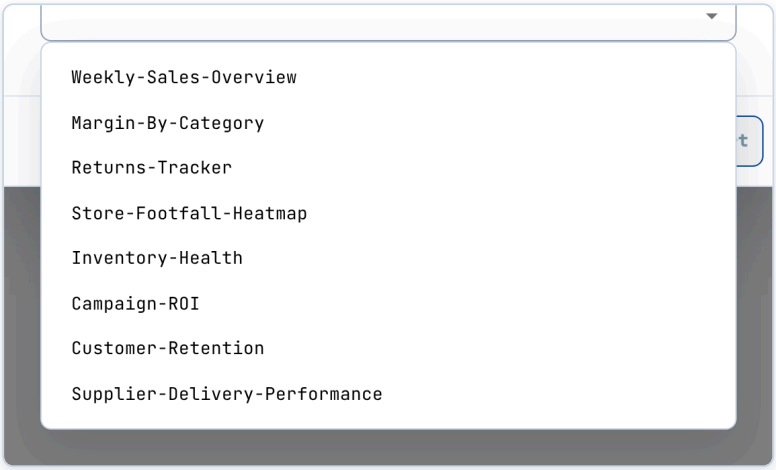


Figure 24. The dropdown lists published dashboards.

3. Click **Import**. You see "Import dashboard successful".
4. Click the dashboard name to open it.

If a dashboard of the same name is already in the workspace, the **Dashboard Name Conflict** dialog asks what to do.



Figure 25. The conflict dialog offers Replace, Keep Both or Cancel.

Button	Result
Replace	Overwrites the existing dashboard.

Button	Result
Keep Both	Imports the new one as `<name>-<date and time>` and keeps the old one.
Cancel	Does nothing.

> **TIP**
>
> If you are not sure, choose **Keep Both**. You can delete the older copy later.

3.1.3 Rename, delete or hand over a workspace

Open the "..." menu on a row (**Open actions for <name>**).

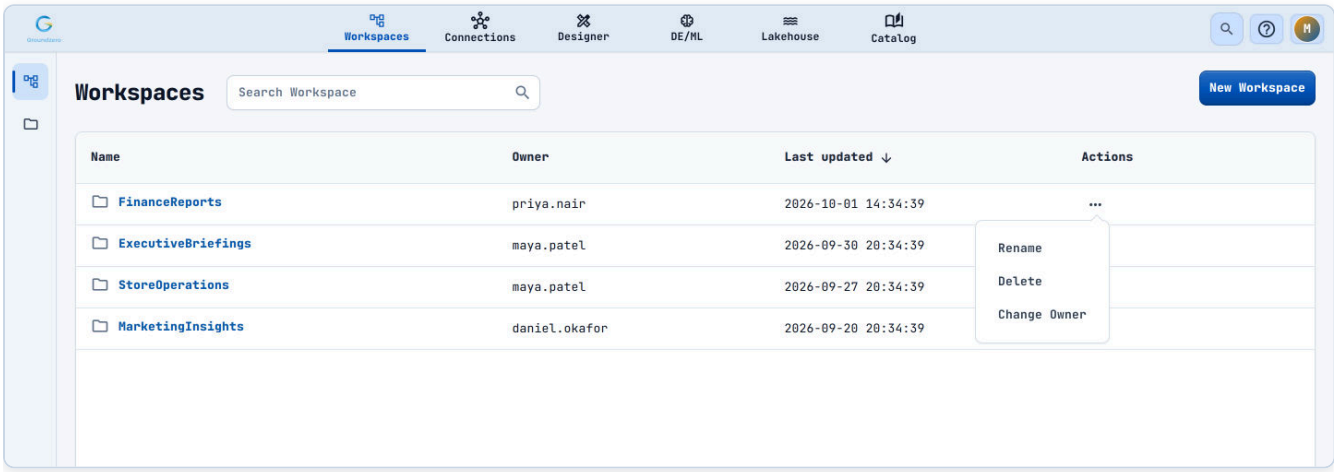


Figure 26. The row menu offers Rename, Delete and (for site administrators) Change Owner.

- **Rename** opens a dialog with the current name. The same naming rule applies.

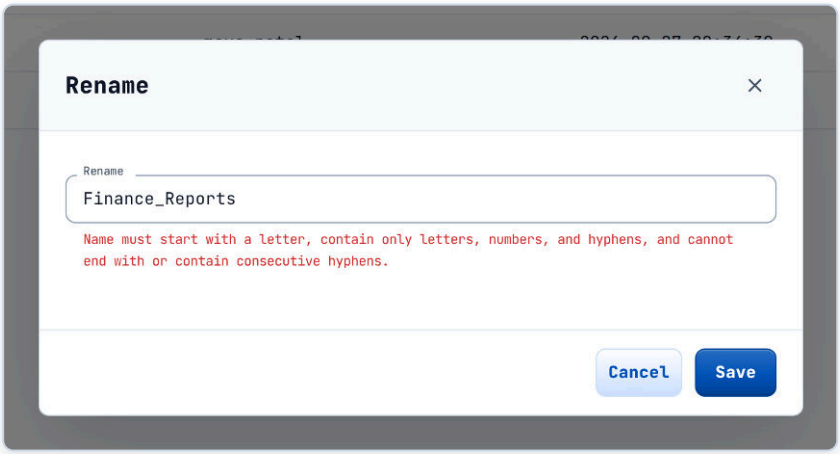


Figure 27. The Rename dialog shows the same name rule.

- **Delete** asks you to confirm, then shows "<name> deleted successfully".
- **Change Owner** is shown to site administrators only. Choose the new owner in **New Owner** and click **Submit**.



Figure 28. Change Owner lists the site's users.

WARNING

Deleting a workspace removes the dashboards imported into it. The originals stay where they were published.

3.1.4 Workspace permissions

An administrator grants these in Access Manager. If one is missing you see `You do not have "All" or "<Permission>" permission(s) on Workspace.`; ask an administrator for the permission it names.

Permission	Lets you
`All`	Do everything below
`Add`	Create a workspace
`List`	See a workspace and its contents
`Edit`	Rename a workspace
`Delete`	Delete a workspace
`AddDashboard`	Import a dashboard into it
`ViewDashboard`	Open a dashboard
`DeleteDashboard`	Delete a dashboard from it

3.2 Manage files in your buckets

My Files shows the AWS S3 buckets behind your connections. Use it to upload CSV, YAML or image files, download files, or tidy up folders. Open **My Files** in the left rail.

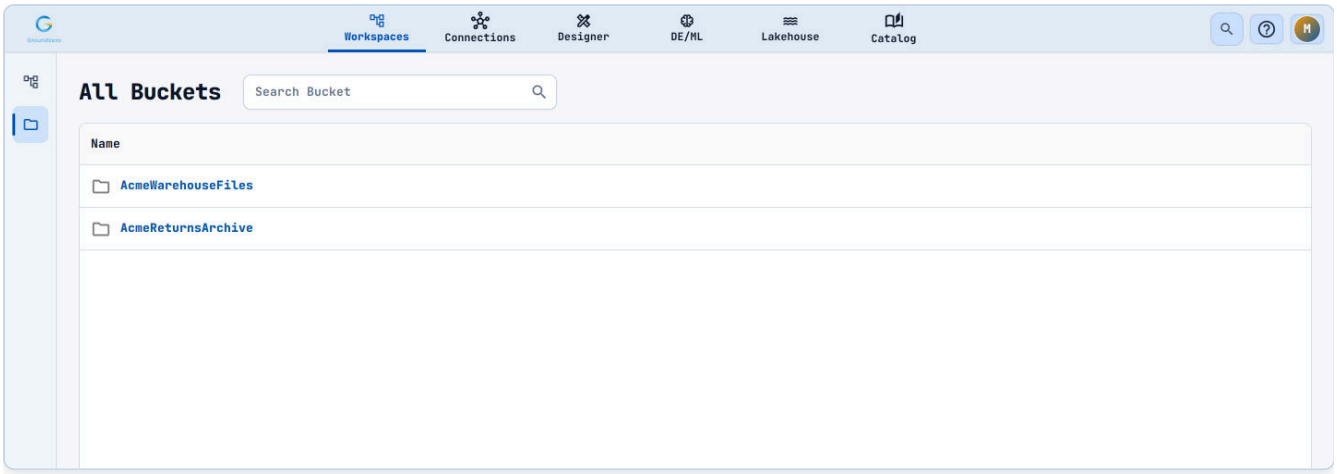


Figure 29. All Buckets lists the S3 storage connections you may see.

All Buckets lists the AWS S3 connections you may see. Use **Search Bucket** to filter. If a bucket is missing, see "Troubleshooting" below.

> **ONLY AWS S3 APPEARS**
>
> Azure Blob Storage, Google Cloud Storage and AI Chat Storage connections are not listed here.

3.2.1 Browse a bucket

1. Click a bucket. A breadcrumb (**All Buckets** > connection > folders) shows where you are.
2. Use the **Folders** tab for the sub-folders of the current folder and the **Files** tab for its files. The search box switches between **Search Folder** and **Search File**.

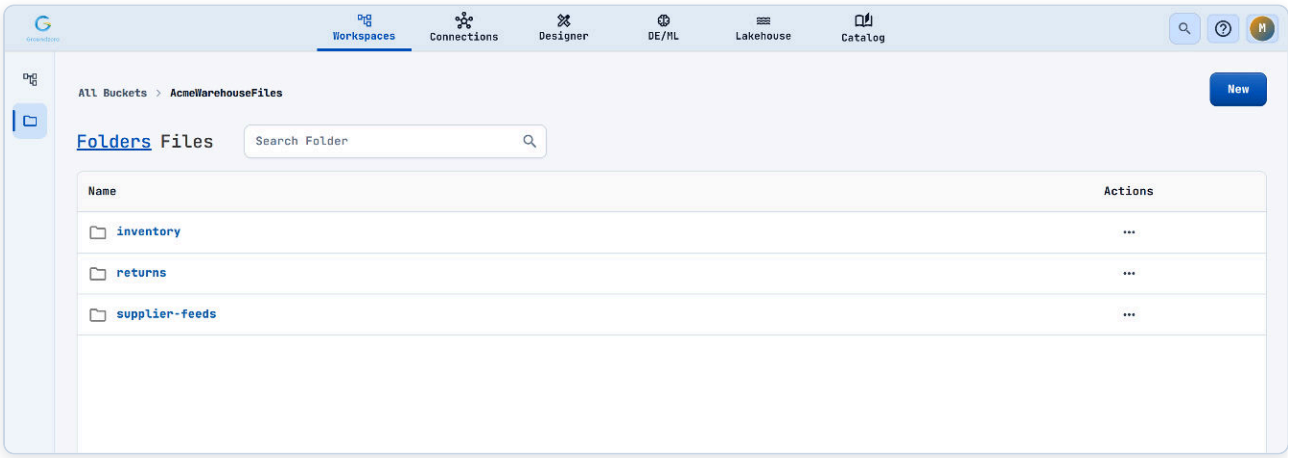


Figure 30. The Folders tab lists the immediate sub-folders of the current path.

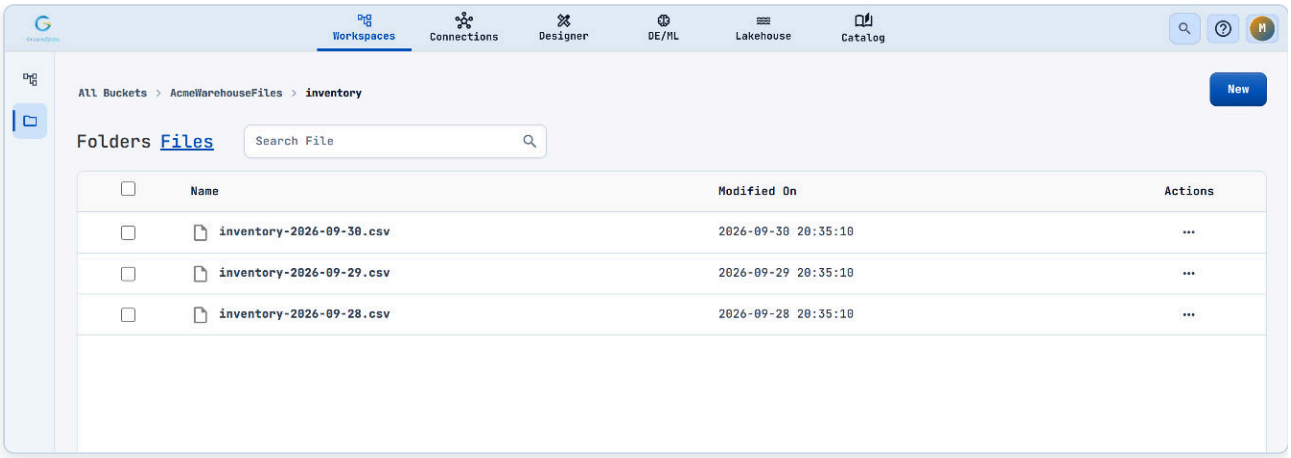


Figure 31. The Files tab lists the files in the current folder.

3. Click a folder to open it; use the breadcrumb to go back.

Each folder lists its first 1,000 files. Tables show 20, 50 or 100 rows per page.

3.2.2 Create a folder

1. Click **New**, then **Create Folder**.

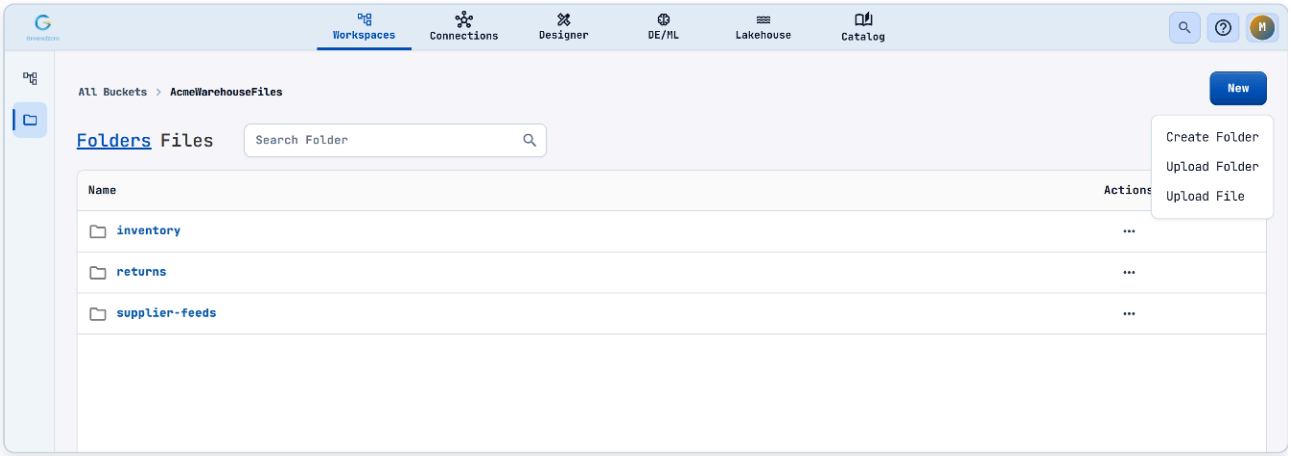


Figure 32. The New menu offers Create Folder, Upload Folder and Upload File.

2. Type a name in **Create Folder** and click **Create**. Typing a slash creates a folder inside a folder.

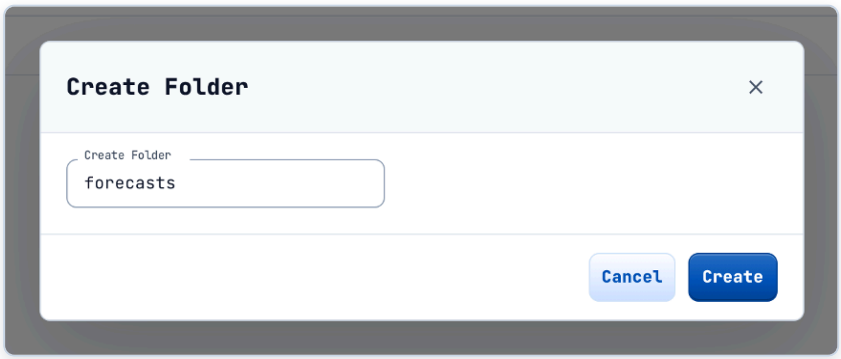


Figure 33. The Create Folder dialog.

3. You see "Folder created successfully" and the folder appears in the list.

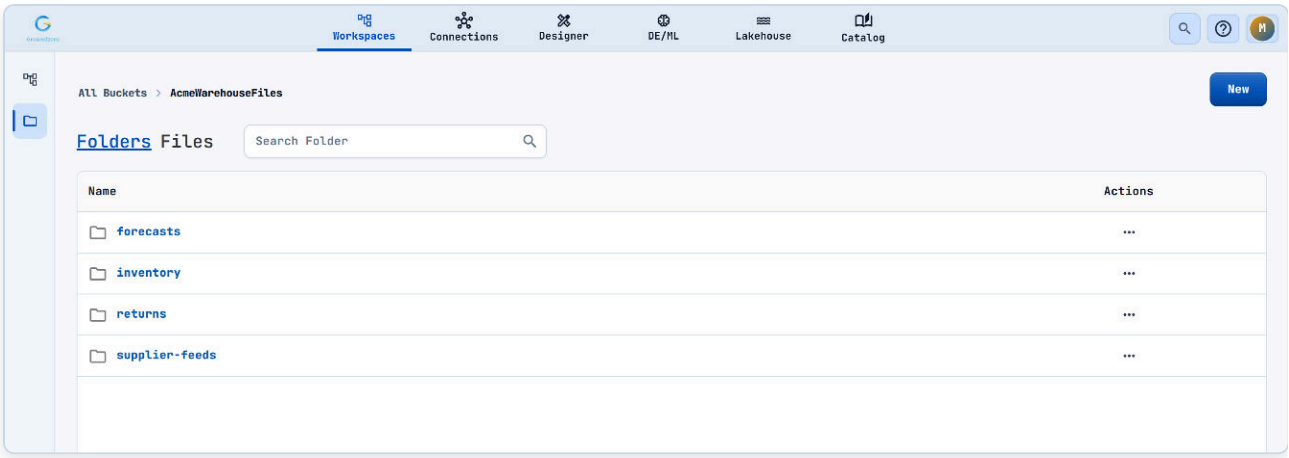


Figure 34. The new folder appears in the list.

An empty folder holds no data. Upload a file into it to keep it.

3.2.3 Upload files or a folder

1. Click **New**, then **Upload File** (or **Upload Folder**).
2. Click **Choose File** (or **Choose Folder**). CSV, YAML and image files are suggested, but you can pick other types. You can choose several files; their names are listed in the dialog.

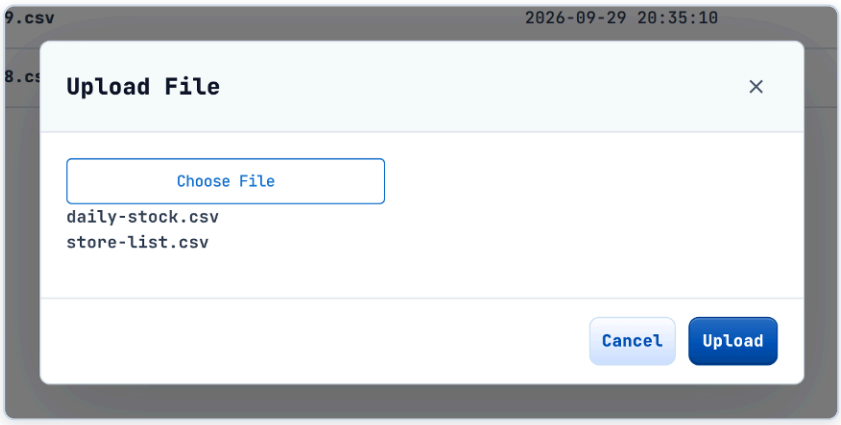


Figure 35. The chosen file names are listed before you upload.

3. Click **Upload**. You see "<n> files uploaded successfully," and the files appear in the **Files** tab.

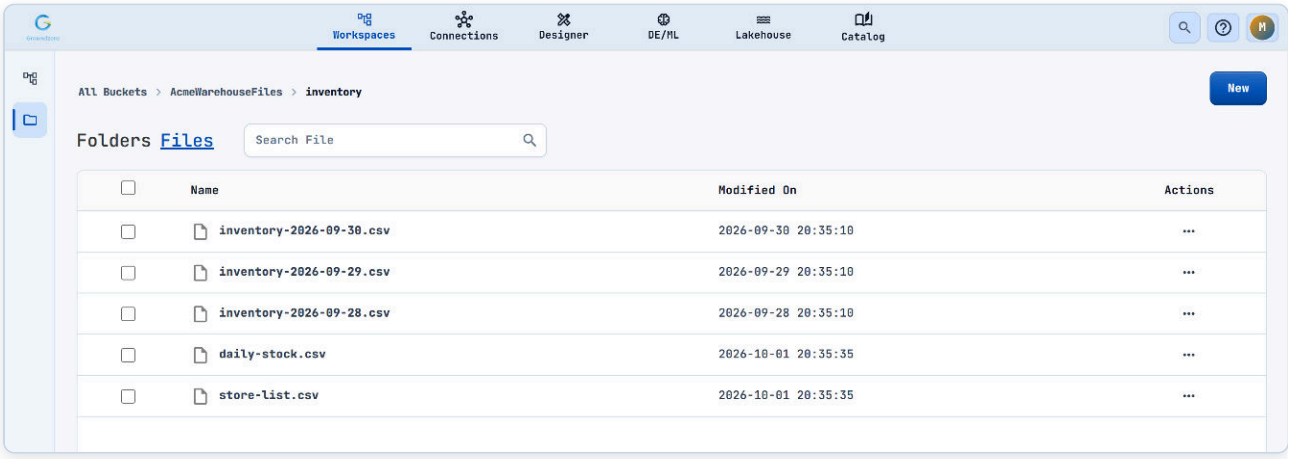


Figure 36. After uploading, the files are listed in the folder.

UPLOADING OVERWRITES

A file with the same name in the same folder is replaced without a prompt.

Always check the **Files** tab afterwards. If the upload fails you see "Error in uploading files" or "Error in uploading folder".

3.2.4 Download files

- **One file:** open the "..." menu on its row and choose **Download**. You see "File downloaded successfully".
- **Several files:** tick the rows (or **Select all rows**). Download and delete icons appear at the top right of the table. Download gives you one `files.zip`.

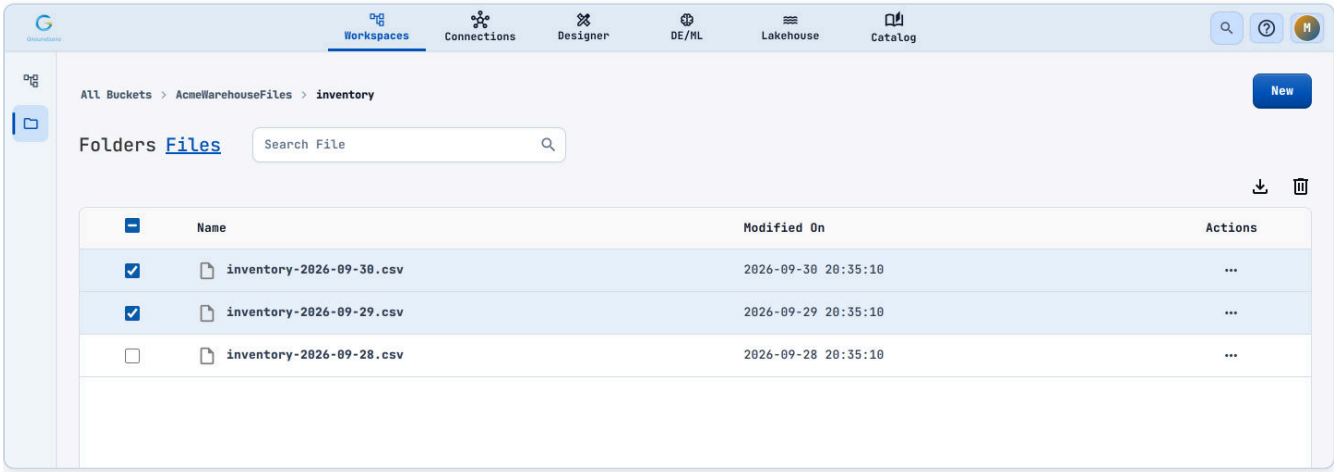


Figure 37. Selecting rows reveals bulk download and delete icons.

3.2.5 Delete files or folders

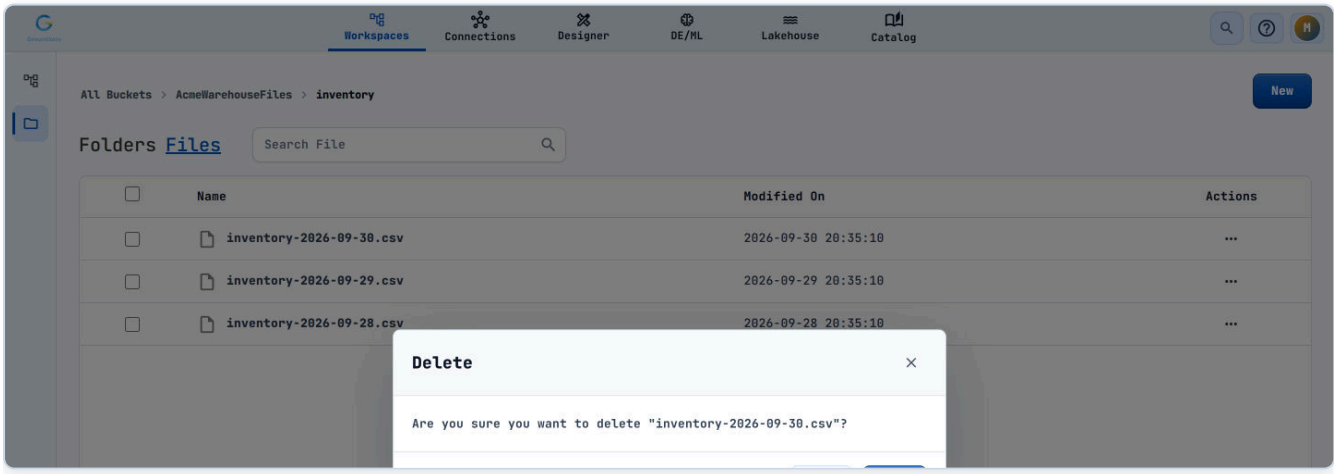


Figure 38. Deleting a file asks for confirmation and names the file.

- **A file:** "..." menu > **Delete**, or tick rows and use the delete icon. Confirm the question, then you see "<name> deleted successfully".
- **A folder:** on the **Folders** tab, "..." menu > **Delete**.

DELETING A FOLDER DELETES EVERYTHING INSIDE IT

All files and nested folders go with it, and Groundzero does not ask a second time.

3.2.6 Who can see and change a bucket

Access follows the connection. Anyone who can see a bucket in **All Buckets** can also upload, download and delete in it. Share the connection (the `Describe` permission) only with people who should have that access.

> **TIP**
> Use a dedicated bucket, with a dedicated S3 connection that has narrow credentials, for files you want your team to manage in Groundzero.

3.3 Troubleshooting

Symptom	Likely cause	Fix
A bucket is missing from **All Buckets**	The connection is not an AWS S3 connection, or you do not have `Describe` on it	Check the connection type; ask an administrator for access
"Access denied. Check your permissions." or "The access key ID is invalid."	The keys in the connection are wrong or expired	Edit the connection and type the secrets again
"The bucket does not exist."	The bucket name in the connection is wrong	Correct the bucket name in the connection
"File not found."	Someone removed the file after you opened the list	Reload the folder
An upload shows success but the file is not listed	The list was not refreshed, or the upload failed silently	Open the **Files** tab again and retry
A workspace will not rename or create	The name breaks the naming rule	Start with a letter; use only letters, digits and single hyphens
Change Owner is not in the menu	Only site administrators can change owners	Ask an administrator
Change Owner is rejected with "Owner name must start with a letter..."	The owner name format is not accepted	Ask an administrator to change the owner

3.4 Related

- **Data source connections**: create the AWS S3 connection that My Files browses.
- **Designer documentation**: publishing dashboards and the dashboard viewer.
- **Lakehouse**: governed tables, an alternative to raw files.

4

Data catalog

The **Catalog** is where you find out what data your team has and what it means. It lists three kinds of resource:

- **Datasets** built in Designer;
- **ETL jobs** built in DE/ML (a job moves or transforms data);
- **Lakehouse warehouses**, with their databases, tables and columns.

For each resource you can read a description, see who owns it and when it changed, add tags, browse a warehouse's tables and columns, and copy the call that runs a dataset or job from another tool.

A **tag** is a short label written as `key: value`, for example `tier: gold` or `classification: pii`. Tags make data easy to find and govern.

Open **Catalog** in the top navigation bar. A collapsible menu on the left switches between **Catalog**, **Consumption Metrics**, **Data Lineage** and **Ontology Studio**. The last two have their own chapters; Consumption Metrics is in the Analyze and Admin documentation.

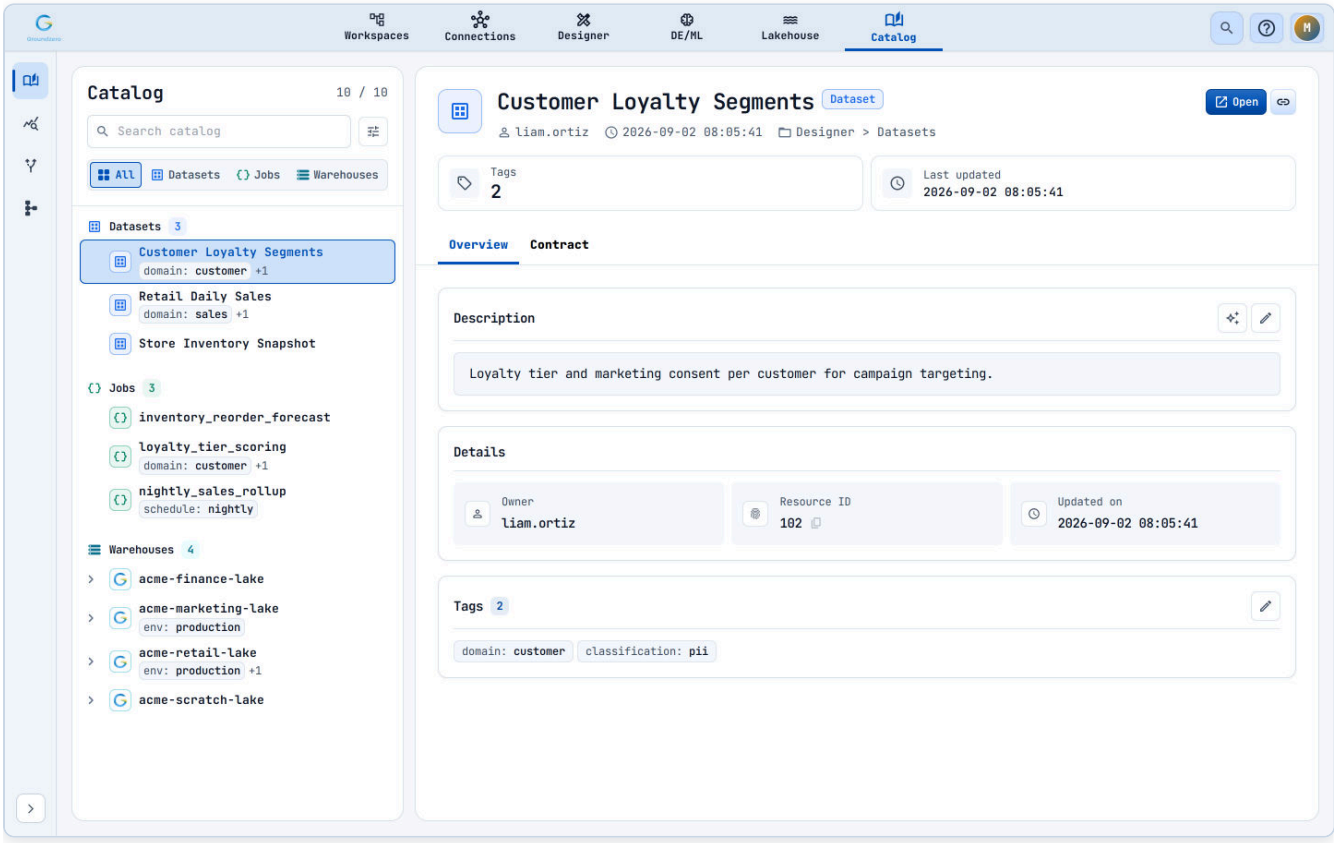


Figure 39. The Catalog page: a navigator on the left and the inspector for the selected resource on the right.

The page has two halves. The **navigator** on the left lists resources. The **inspector** on the right shows the one you selected. The first resource is selected for you.

I want to…	Go to
Find a table or column	"Find the data you need"

I want to…	Go to
Understand what a resource is	"Understand a resource"
Label data so others can find it	"Tag data"
Write or draft a description	"Describe a resource"
Call a dataset or job from another tool	"Copy the call for a dataset or job"

Who can do what:

Action	Who can do it
Open the catalog	Any signed-in user of the site
See a warehouse and its tables	Users who may list that warehouse
Change tags on a warehouse, database, table or column	Administrators; other users can look but not save
Edit a warehouse description	Users with `All` permission on that warehouse
Edit dataset or job tags and descriptions	As your permissions on that dataset or job allow
Draft a description with AI	Anyone who can use Groundzero chat

4.1 Find the data you need

4.1.1 Search

Type in **Search catalog**. Matching ignores capital letters, and `_`, `-` and `/` count as spaces, so `order_items`, `order-items` and `order items` all find the same thing.

The search looks at names, locations, owners, projects and tags (as `key: value`). For warehouses it also looks at database, table and column names.

Type at least **two characters** to search column names in every warehouse. Groundzero then loads the warehouse tables in the background, which can take a moment.

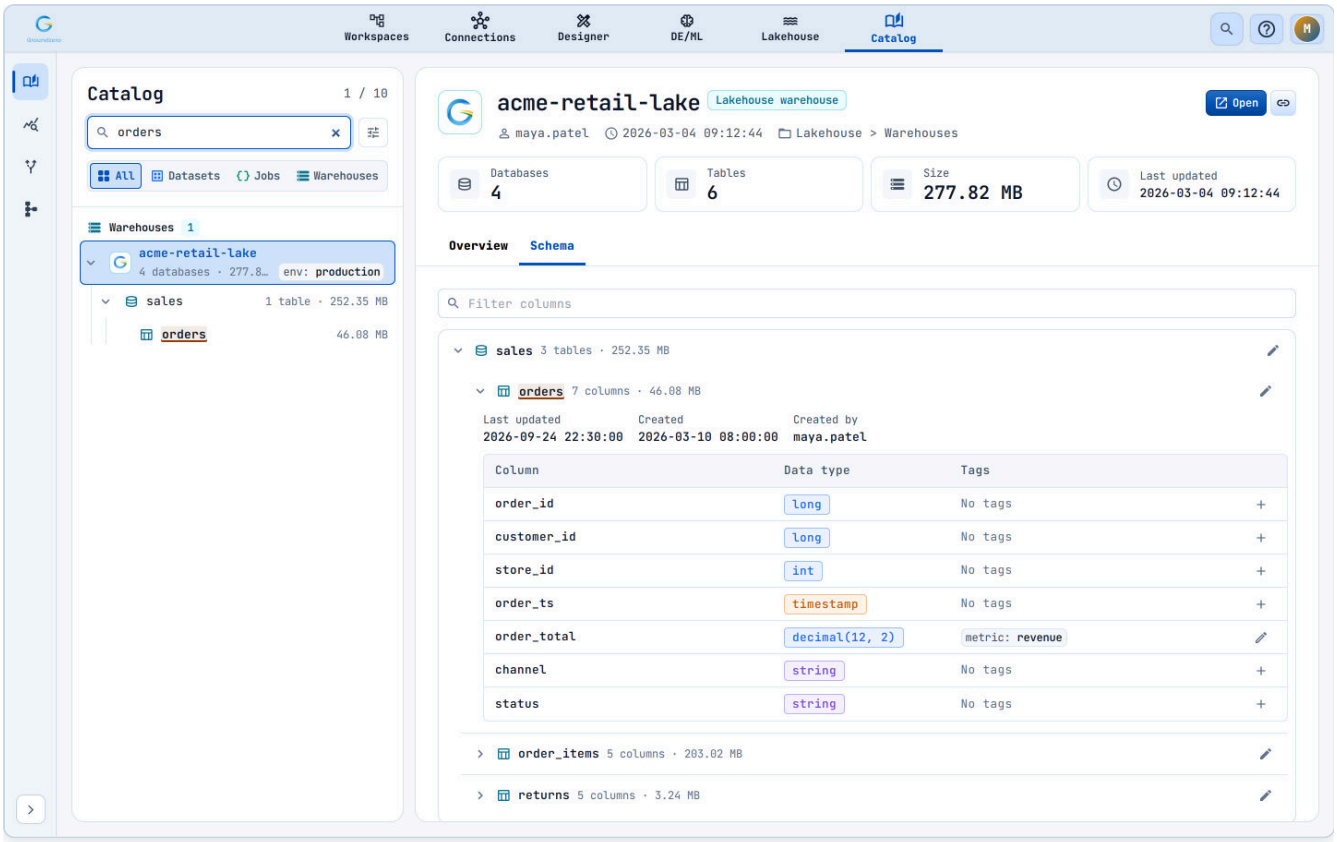


Figure 40. Searching for "orders" selects the matching warehouse, jumps to its Schema tab and highlights the table.

What happens as you type:

- Matching text is highlighted. Under a matching table the tree shows the matching column names.
- If the search matches something inside the selected warehouse, the inspector opens its **Schema** tab at the best match.
- If the selected resource no longer matches, the first result is selected.

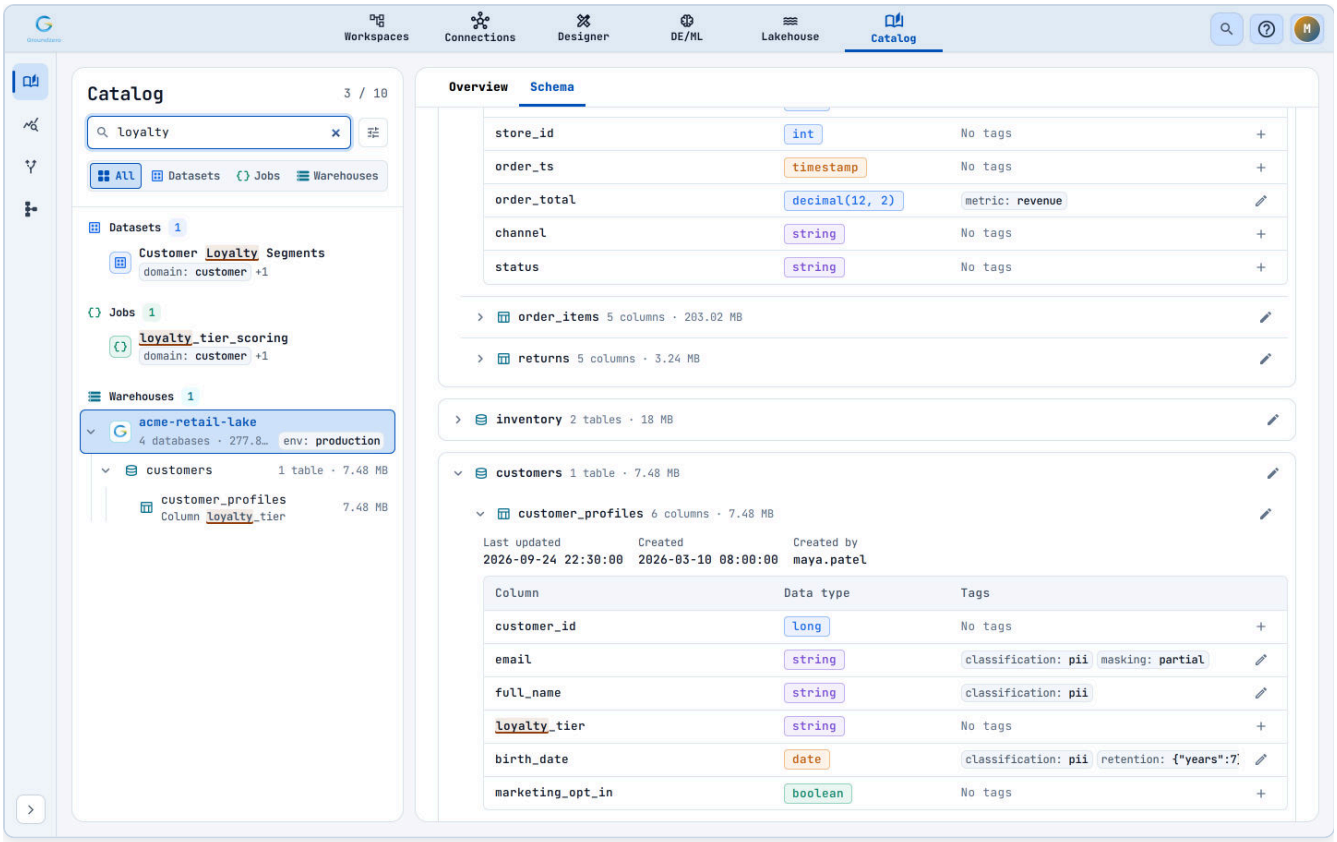


Figure 41. Searching "loyalty" matches datasets and jobs by name and a warehouse by column name.

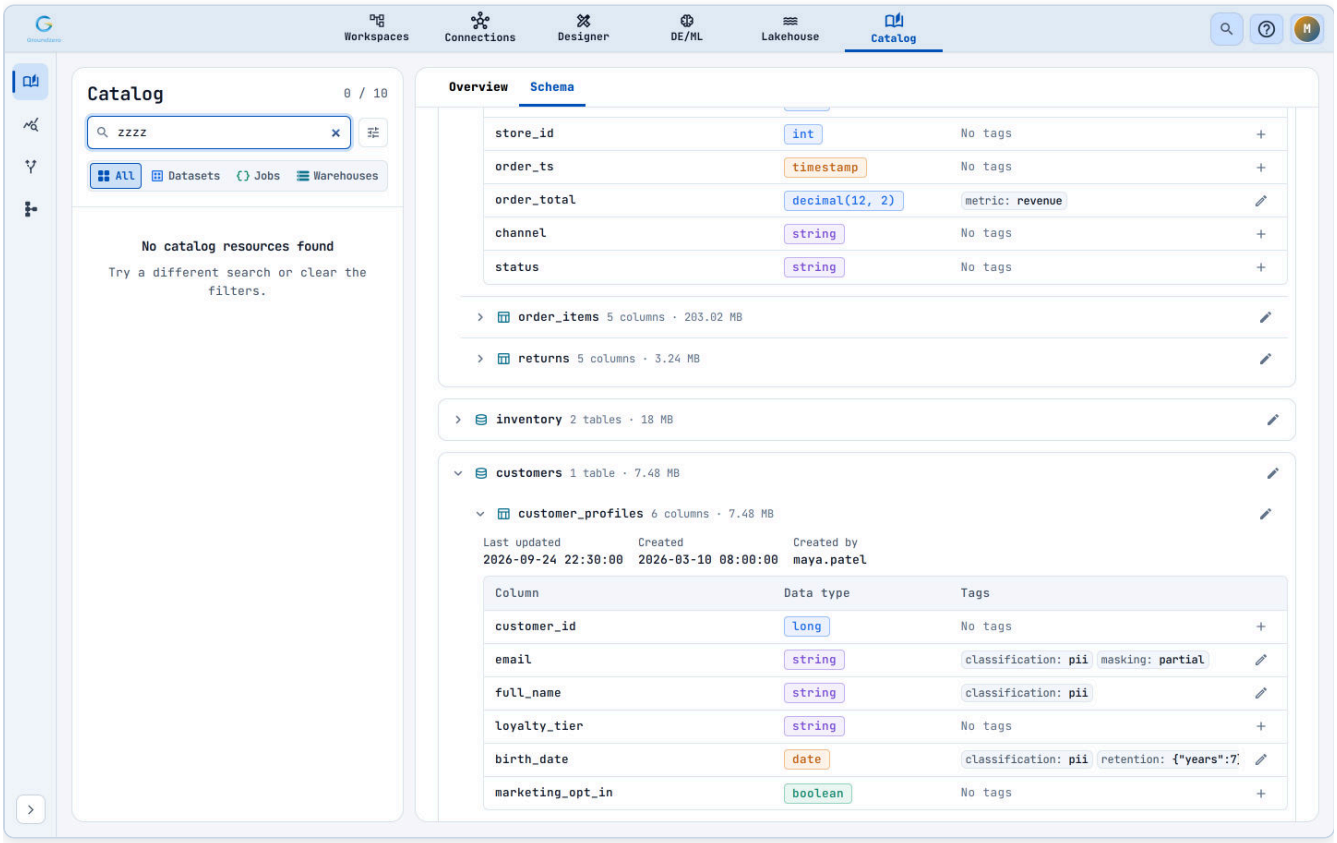


Figure 42. When nothing matches, the navigator says so; the inspector keeps the previously selected resource.

4.1.2 Narrow by kind

Use the buttons **All**, **Datasets**, **Jobs** and **Warehouses** under the search box. The counter next to the title shows how many resources match out of the total.

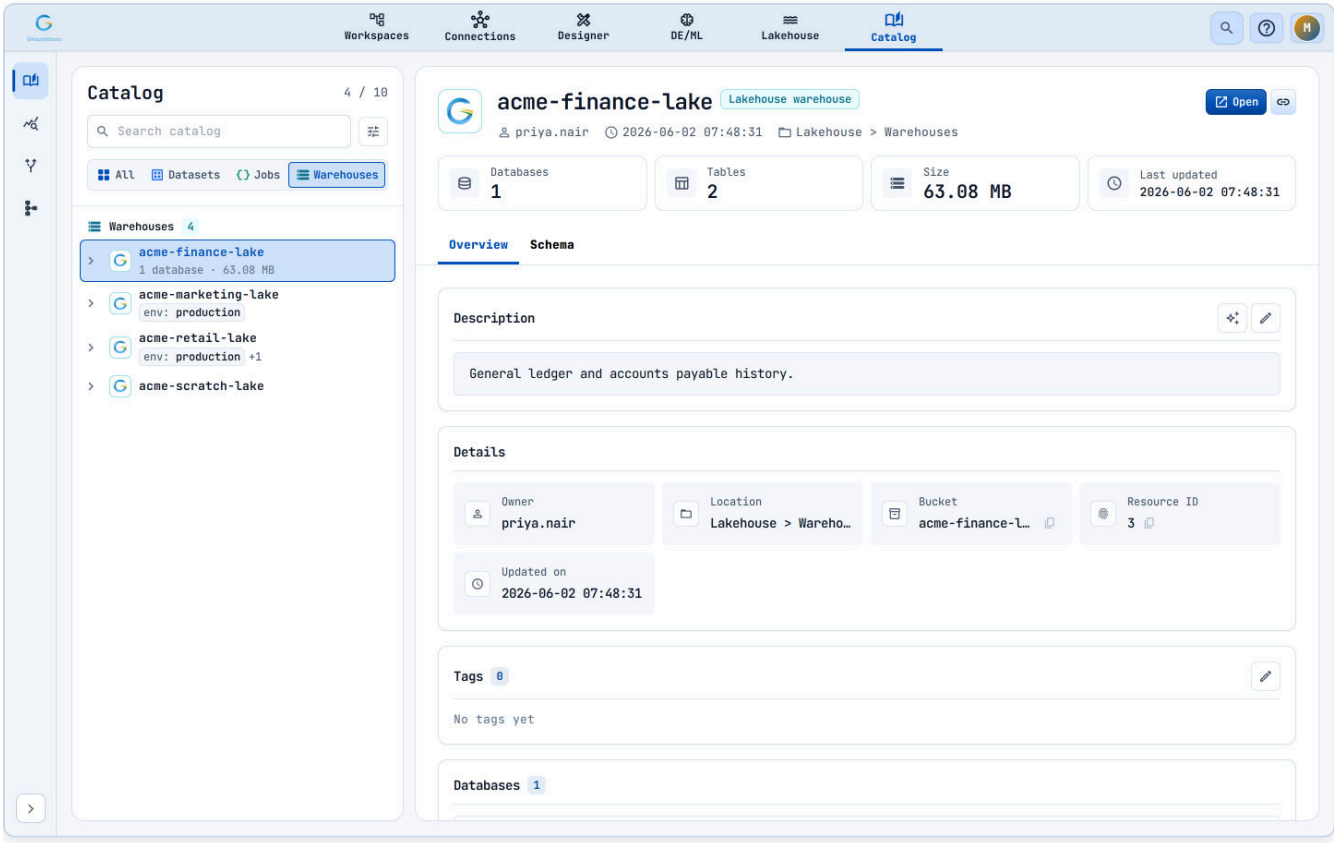


Figure 43. The Warehouses kind button limits the list to warehouses.

4.1.3 Filter by tag or date

Click **Filters**. The button shows how many filters are active. All filters combine with the search and the kind buttons.

Filter	What it does
Tag	Keeps resources that have exactly one chosen `key: value` tag.
Tags	**Tagged** keeps resources with at least one tag; **Missing tags** keeps resources with none.
From / **To**	Keeps resources updated on or after / on or before a date. Resources with no update date drop out once you set a date.

Clear filters resets these four filters. It does not reset the search or the kind buttons.

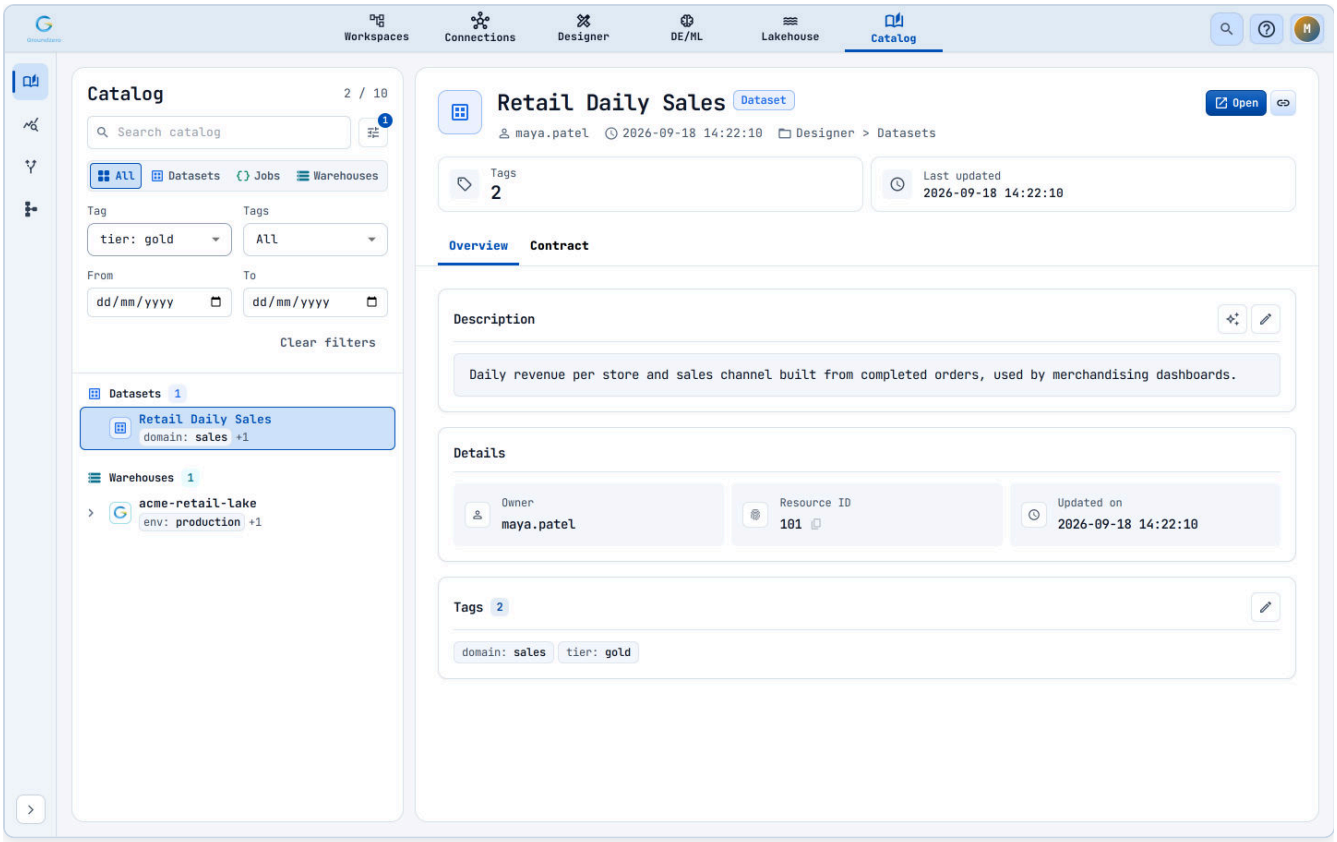


Figure 44. Choosing "tier: gold" in the Tag filter leaves the matching resources.

4.1.4 Walk through a warehouse

Warehouses have an arrow. Open it to see its databases, and open a database to see its tables with their sizes. Click a table to jump to the warehouse's **Schema** tab at that table.

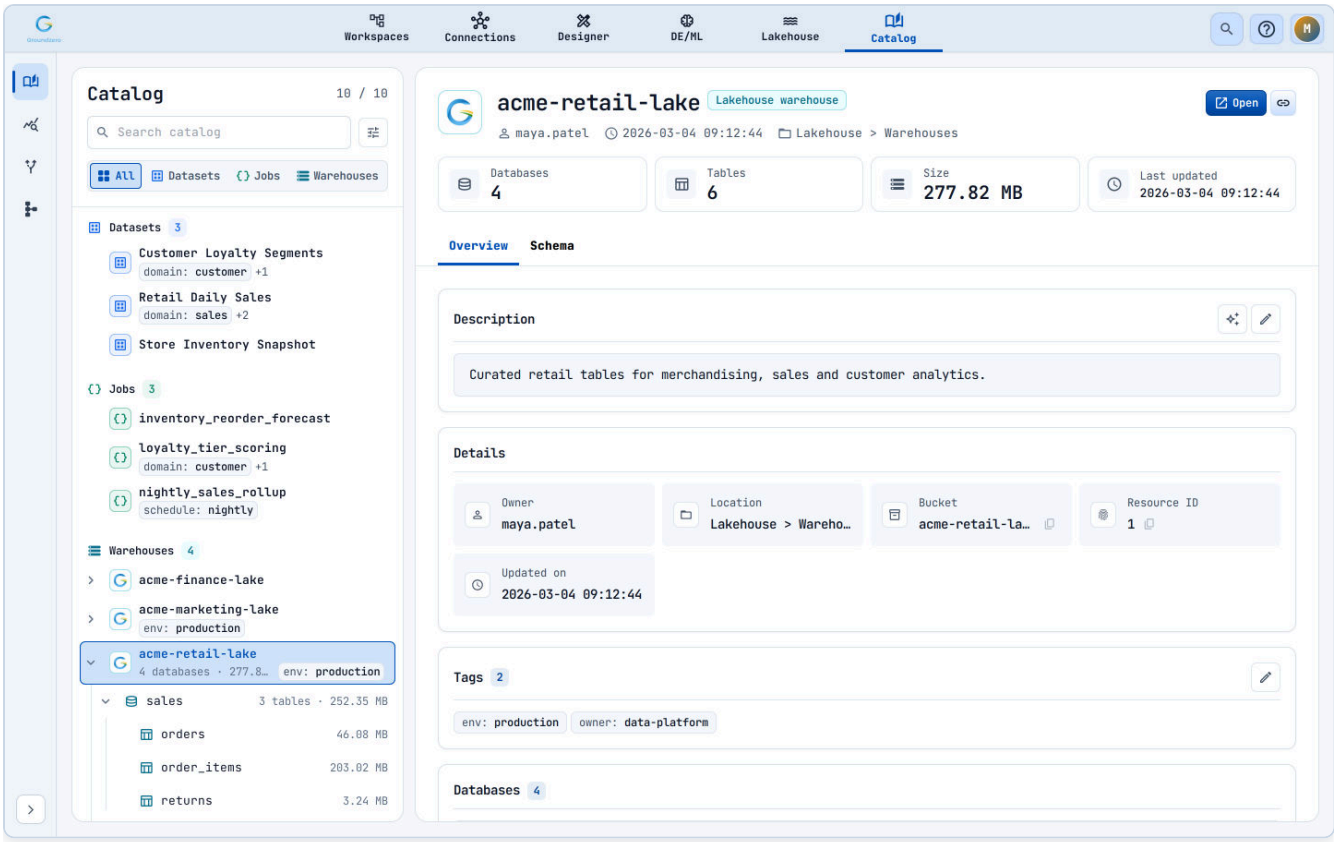


Figure 45. Expanding a warehouse and a database in the tree reveals tables with their sizes.

You can also use the keyboard: Tab into the tree, Arrow Up and Down to move, Home and End to jump, Enter or Space to open.

4.2 Understand a resource

The inspector header shows the name, the kind (**Dataset**, **ETL job** or **Lakehouse warehouse**), the owner, the update time and the location. Two buttons help you move on:

- **Open** goes to the resource's own page: the dataset in Designer, the job in DE/ML, the warehouse in the Lakehouse editor.
- **Copy link** copies the address of that page. You see "Copied link".

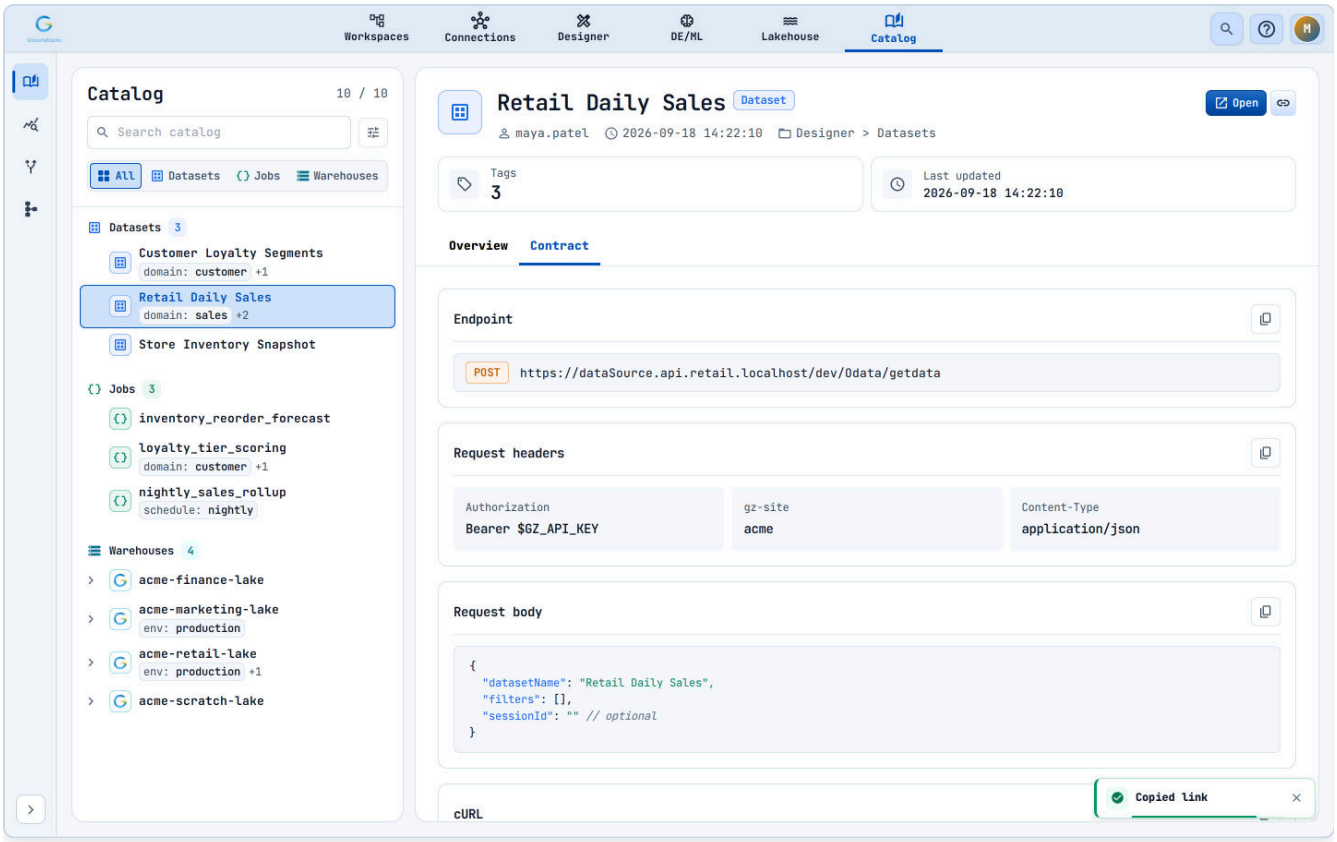


Figure 46. Copy link confirms with a notification.

Cards under the header show counts: tags and last update for datasets and jobs; databases, tables, size and last update for warehouses.

Datasets and jobs have the tabs **Overview** and **Contract**. Warehouses have **Overview** and **Schema**. The tab returns to Overview whenever you select another resource.

The **Details** card on Overview lists the owner, location, project (jobs), bucket (warehouses), resource ID and update time, with copy icons where useful. Empty values read "Not available".

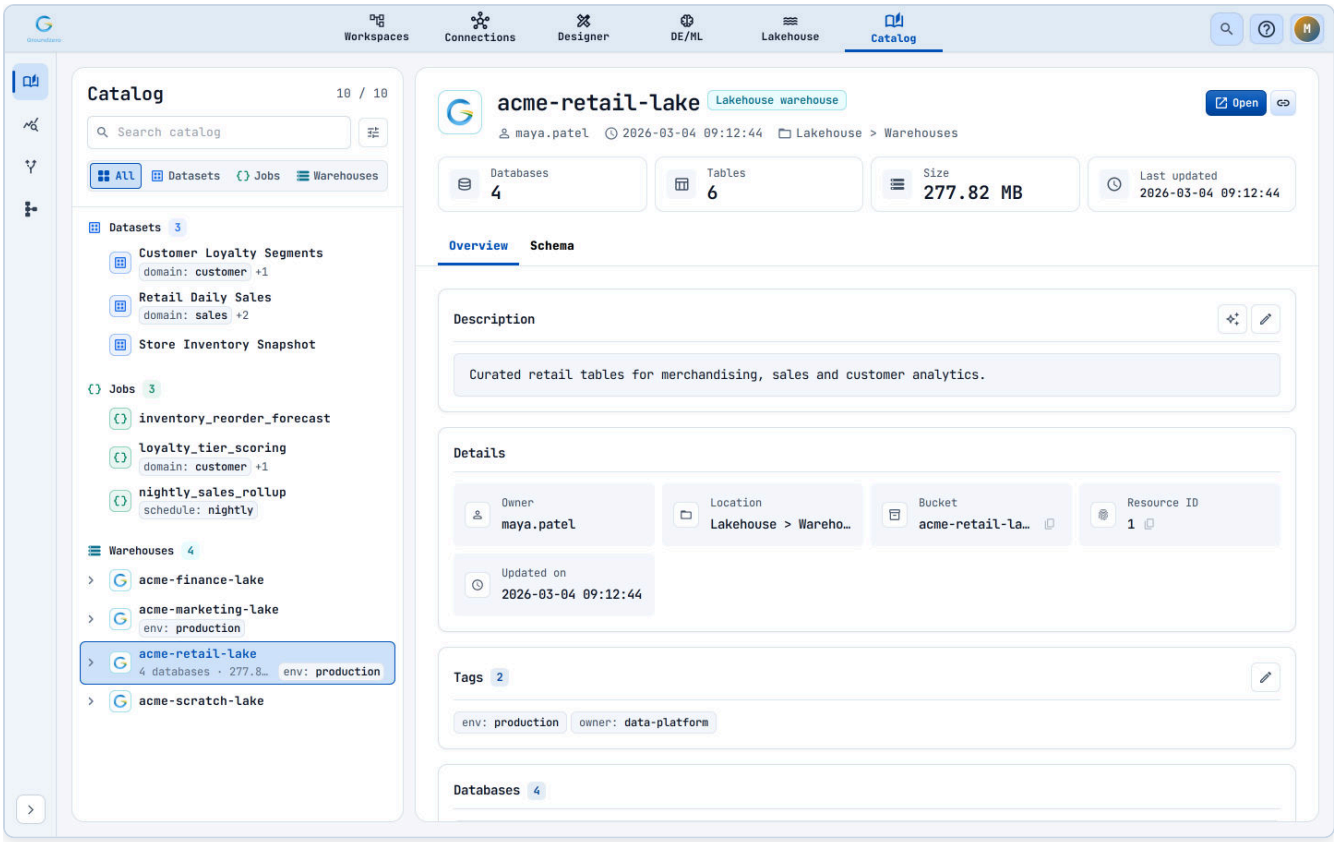


Figure 47. A warehouse overview: stat cards, description, details, tags and a Databases card.

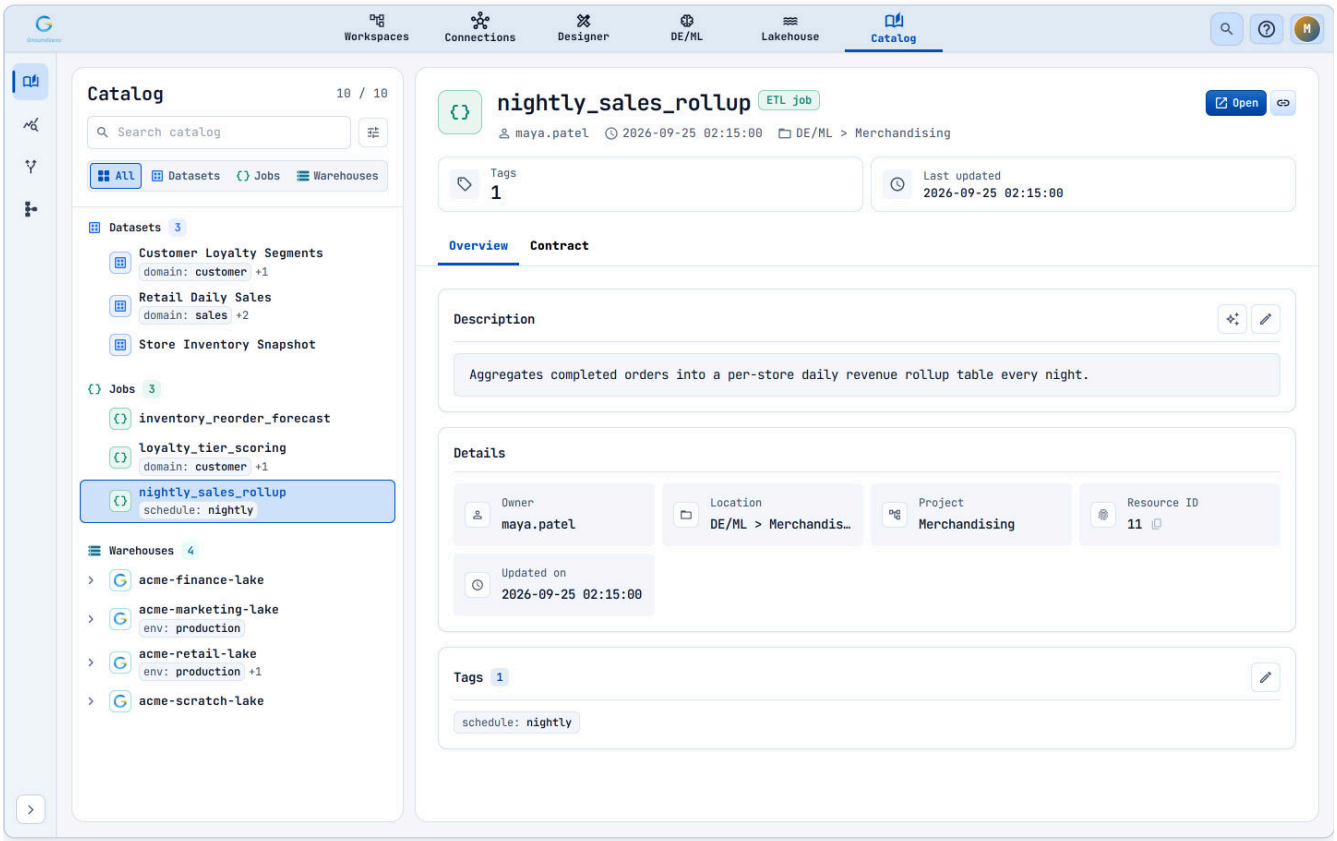


Figure 48. A job overview shows its project and location.

For a warehouse, the **Databases** card lists each database with its table count and size. Click one to open the Schema tab there.

4.2.1 See the tables and columns of a warehouse

Open a warehouse and choose the **Schema** tab. Databases are collapsible sections; open a table to see its columns, their data types and their tags.

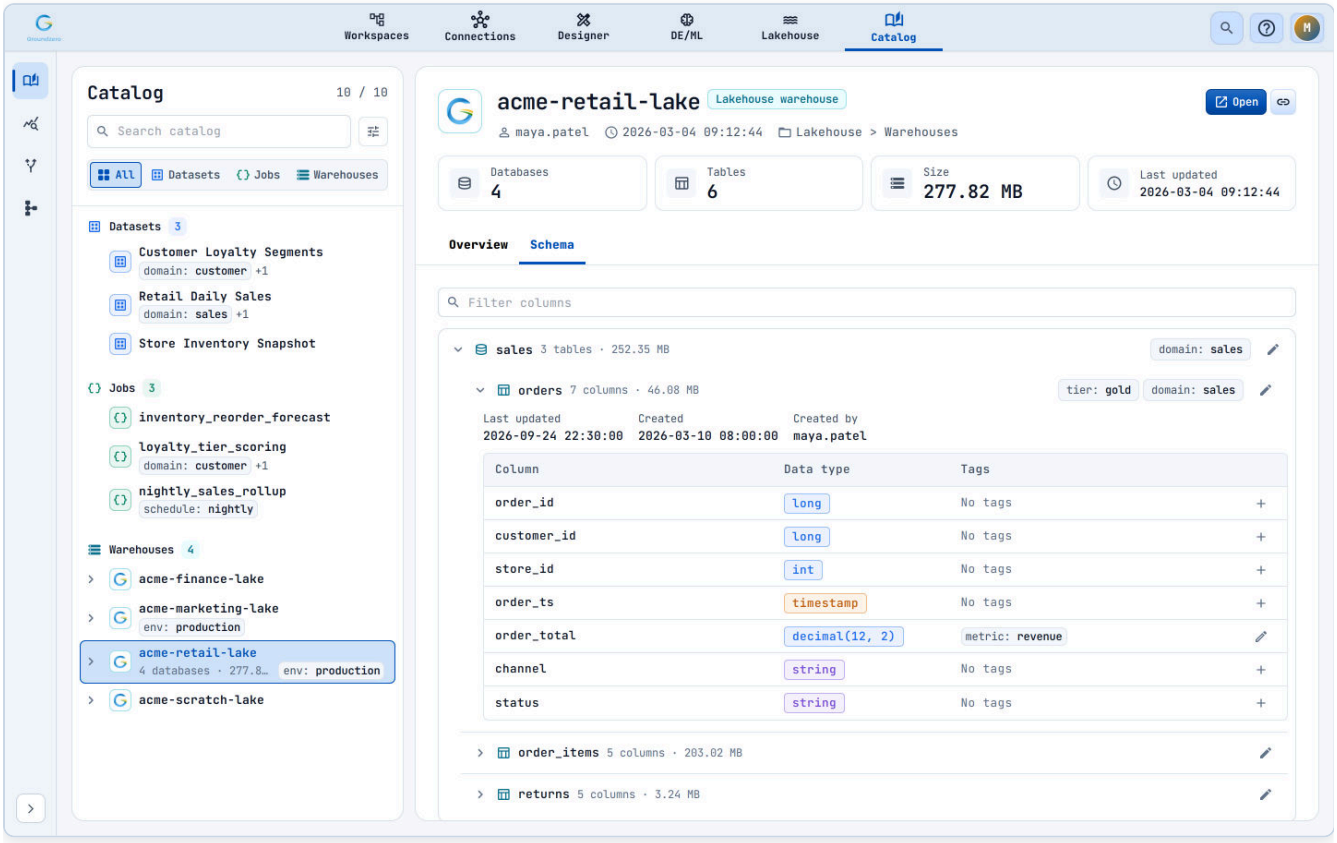


Figure 49. Databases are collapsible sections; opening a table shows its columns, types and column tags.

1. Type in **Filter columns** to keep only columns whose name or data type matches. If nothing matches you see `No columns match "<text>"`.

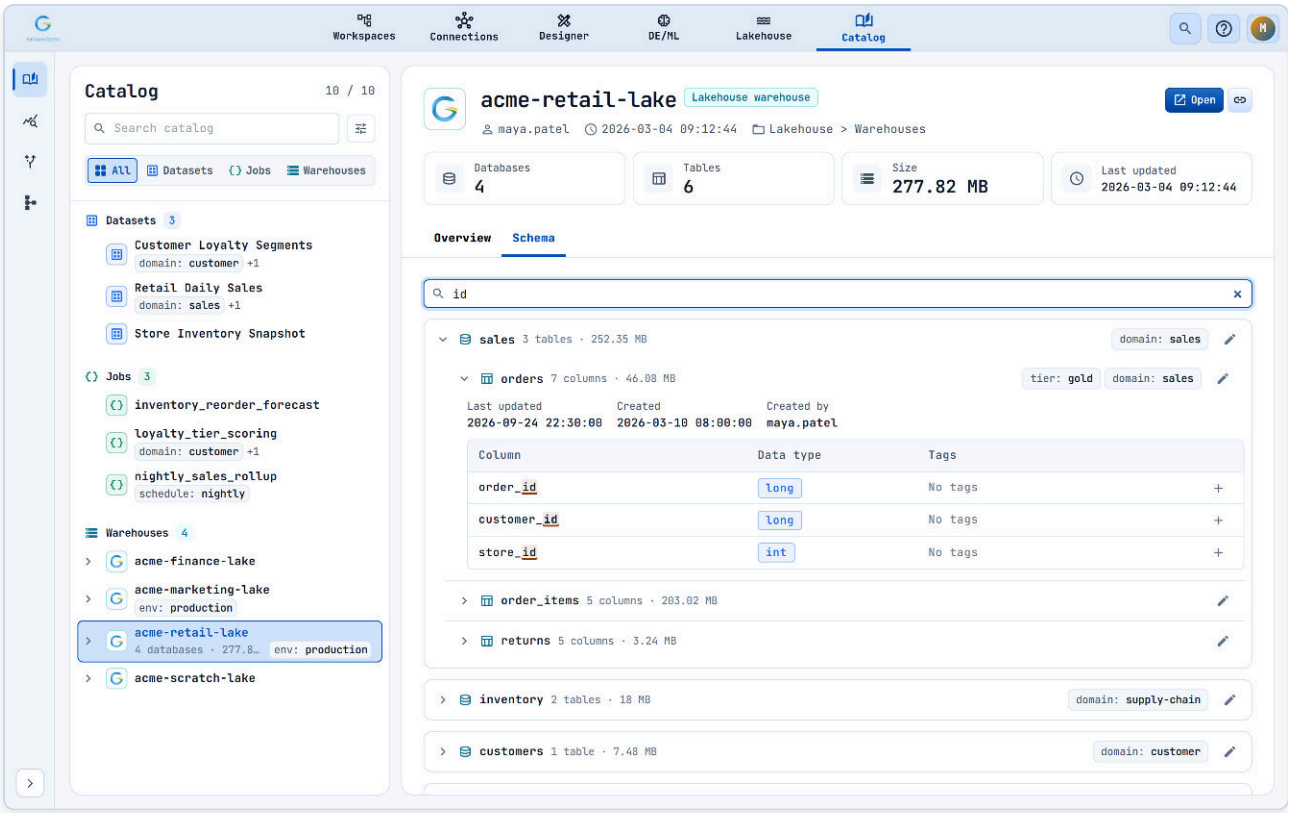


Figure 50. Filtering columns by "id" keeps only matching columns and highlights the match.

2. Open a table to see when it was created and last updated.

If a warehouse has no tables you see "No schema available for this warehouse yet." You see the same message if you may not list the warehouse.

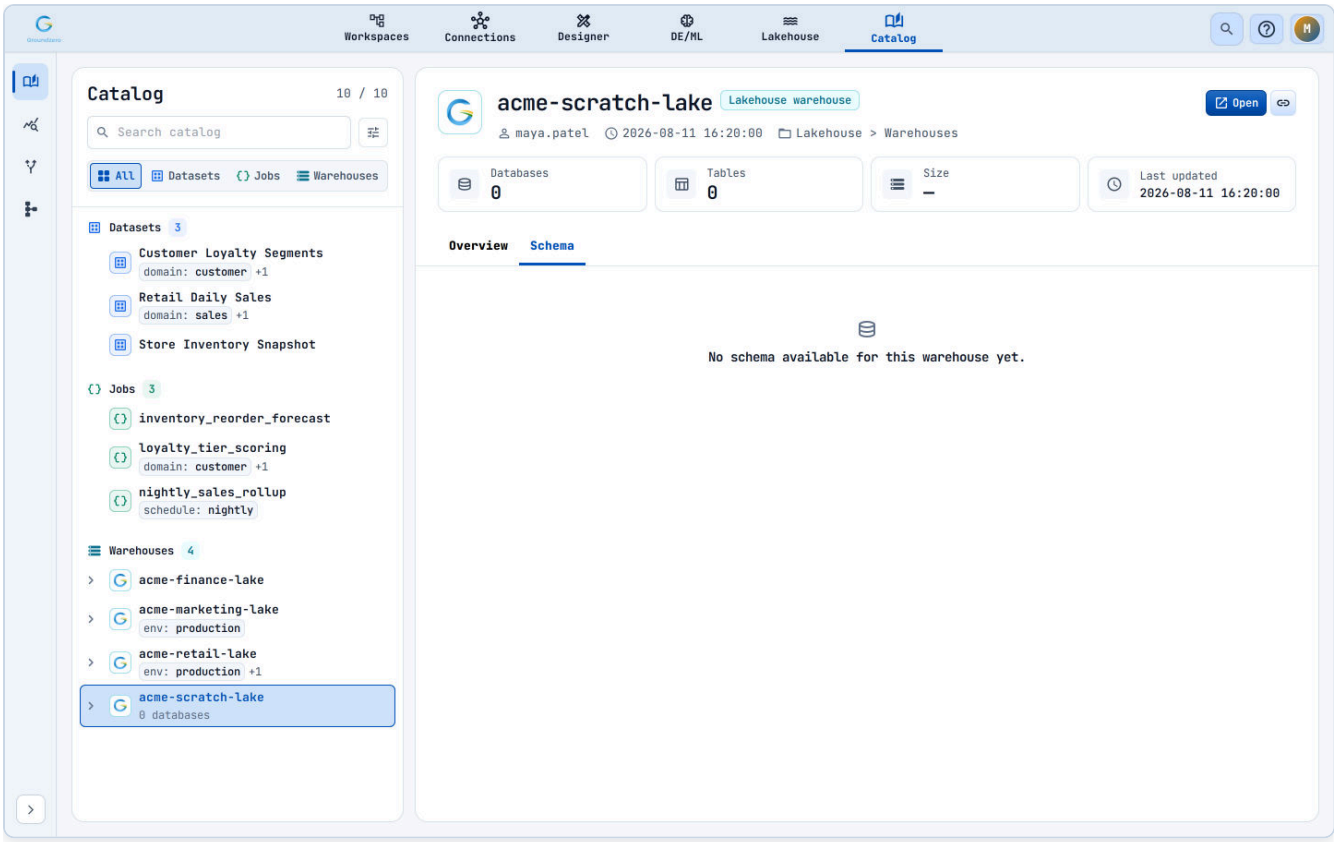


Figure 51. An empty warehouse has no schema.

4.3 Describe a resource

A good description says what the resource provides and when to use it. The **Description** card on Overview shows it. When there is none, it reads "No description yet. Write one or draft it with AI."

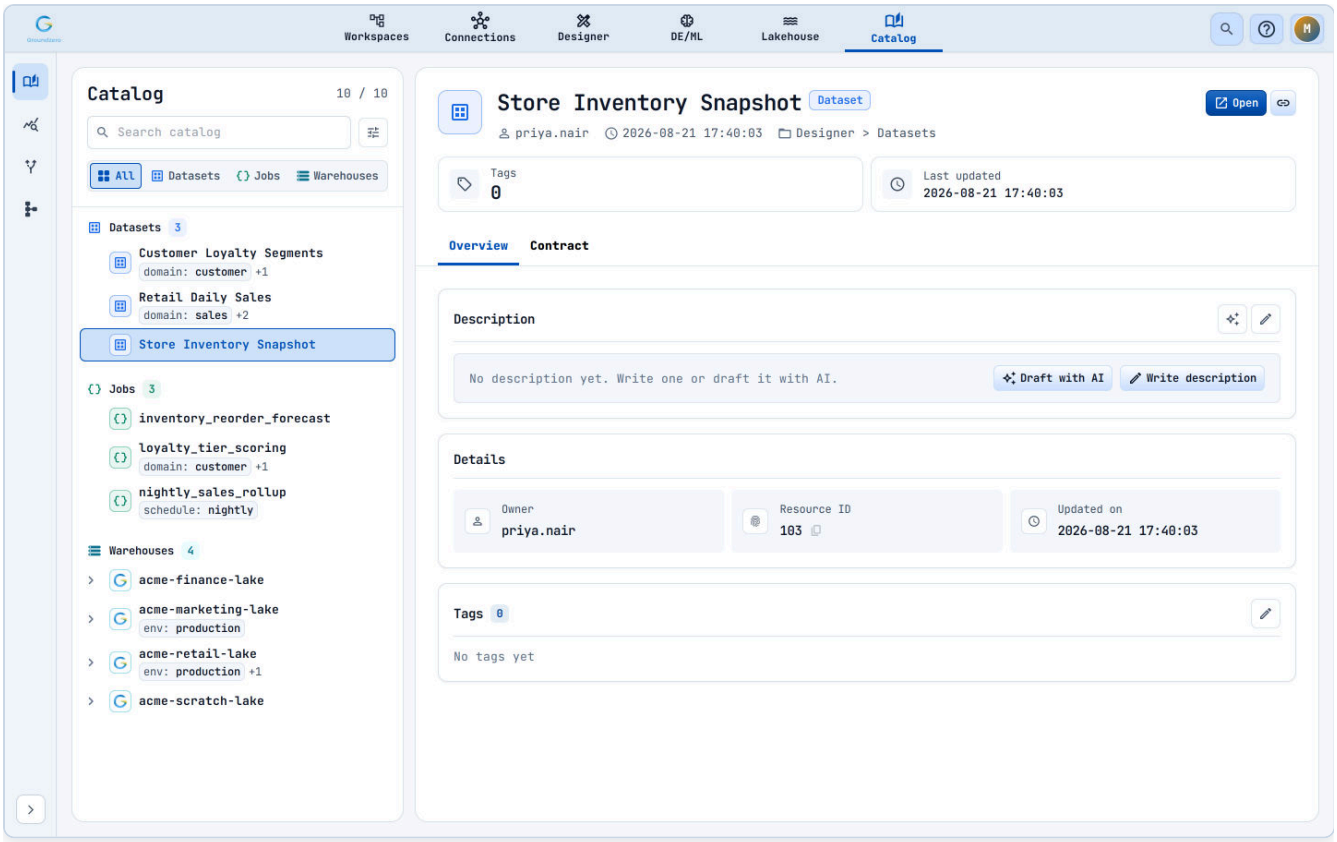


Figure 52. A dataset without a description offers Draft with AI and Write description.

4.3.1 Write a description

1. Click **Write description** or **Edit description** (pencil).
2. Type the text. Enter adds a new line; Ctrl or Cmd plus Enter saves.
3. Click **Save**. You see "Saved". **Cancel** throws the draft away.

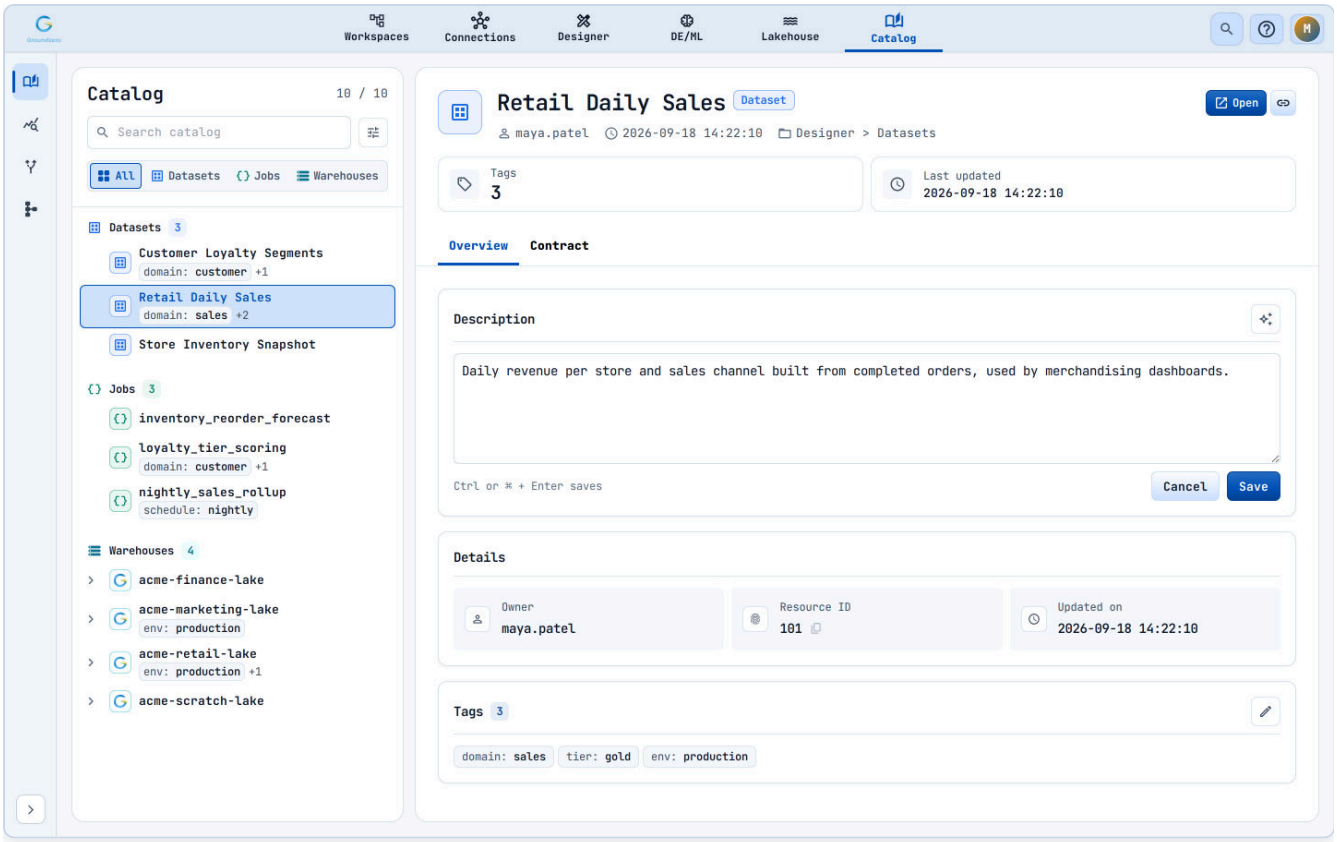


Figure 53. The description editor with Cancel and Save.

Saving an empty text clears the description. A warehouse description can be up to 1,000 characters; longer text is rejected with "description must not exceed 1000 characters".

4.3.2 Draft a description with AI

1. Click **Draft with AI** (it becomes **Regenerate description with AI** once there is text).
2. Wait a moment. The editor opens with the draft and you see "Draft ready to review."
3. Edit the draft, then click **Save**.

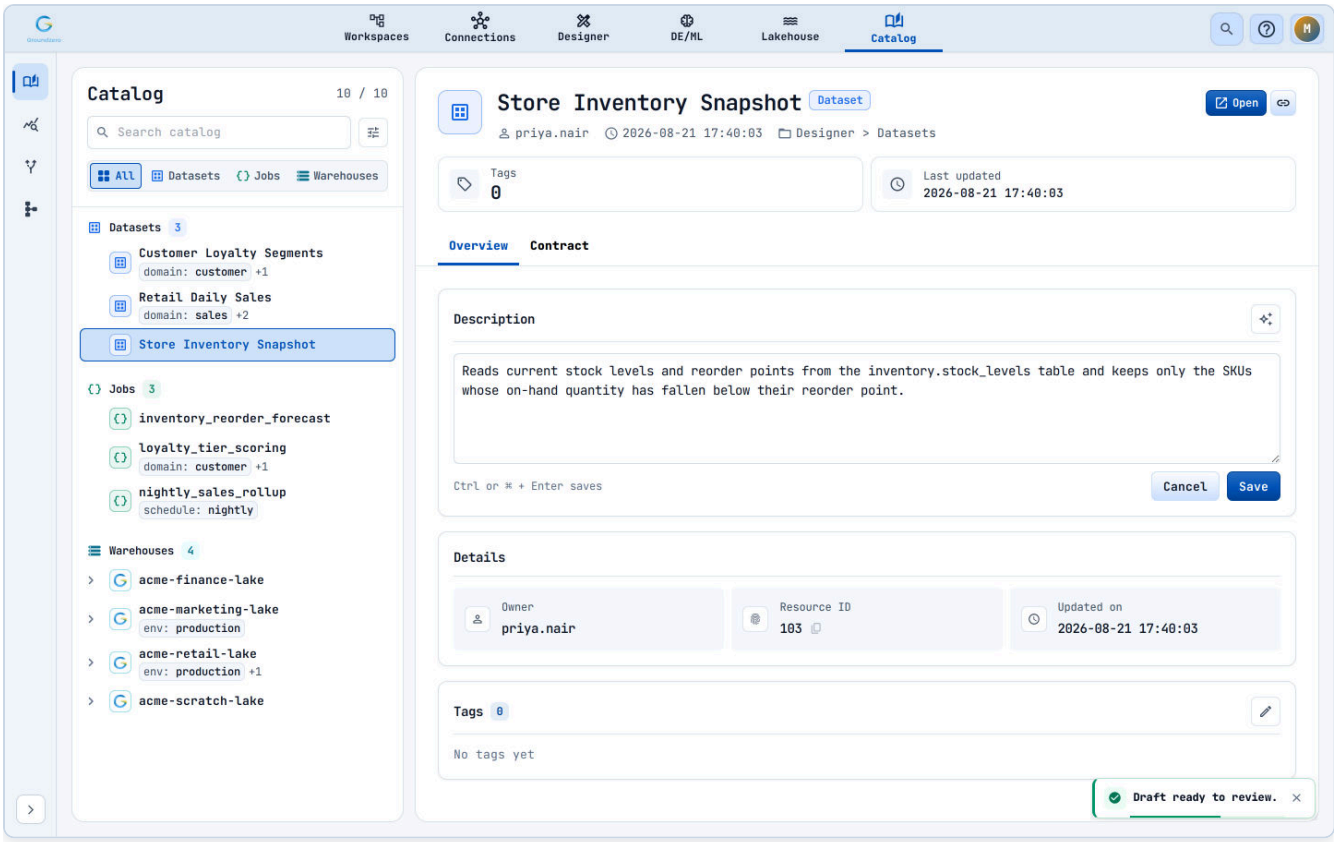


Figure 54. The AI draft opens in the editor for review.

Nothing is saved until you click Save. The AI reads the resource's configuration (for a dataset, its queries; for a job, its code; for a warehouse, its database and table names and tags). Secret-looking values such as passwords and keys are hidden before anything is sent. The draft is two to four short sentences. It counts against your AI usage budget.

If you see "Unable to generate description" or "AI returned an empty description.", try again. If you select another resource while a draft is being written, the draft is discarded.

4.4 Tag data

4.4.1 Tag a dataset, job or warehouse

1. On the **Tags** card of Overview, click the pencil (**Edit tags**). It is disabled until the resource has finished loading.
2. Type a **New tag key** and **New tag value**, then click **Add** (or press Enter). Adding a key that already exists replaces its value.
3. To remove a tag, click the x on its chip.
4. Click **Save tags**. You see "Tags updated". **Cancel** throws your changes away.

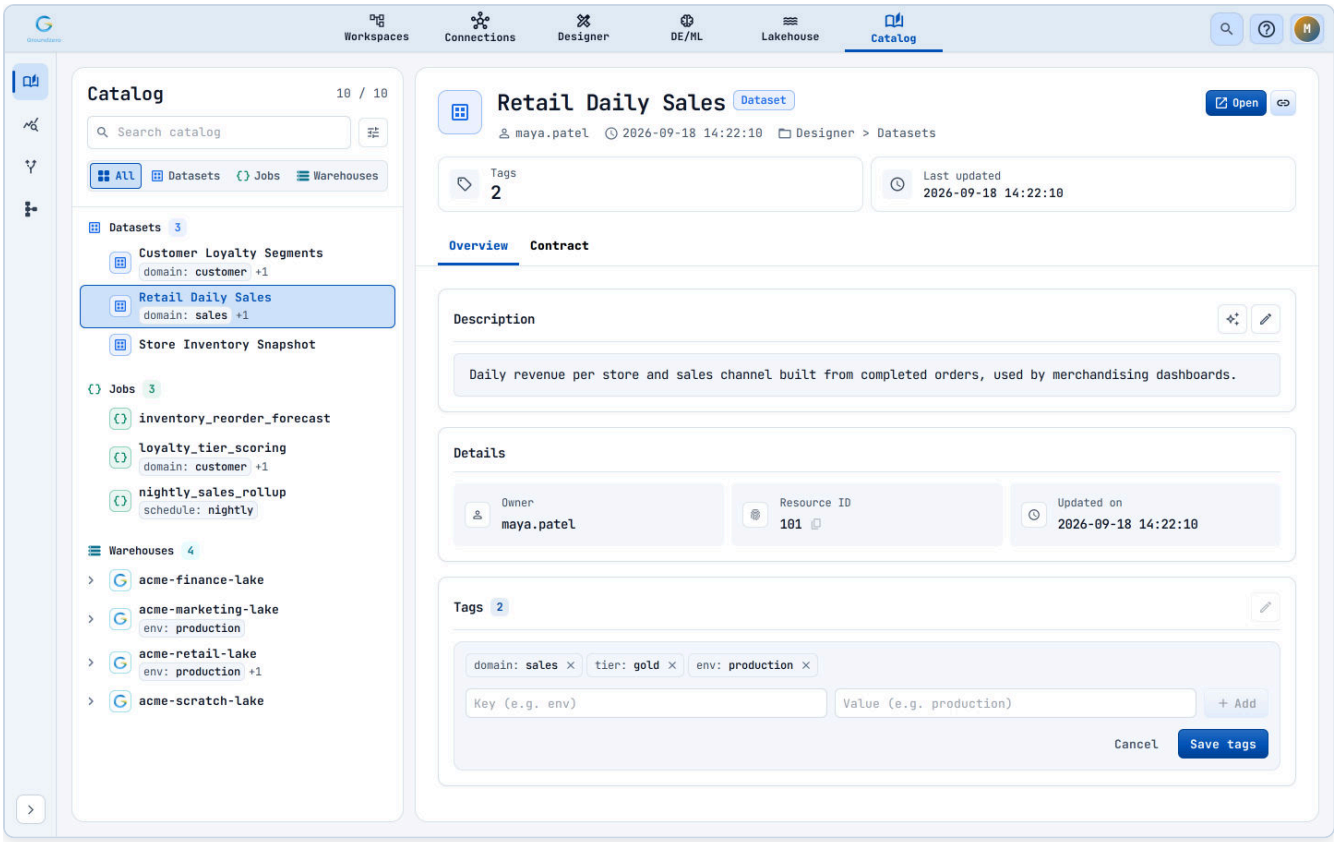


Figure 55. The tag editor with a new tag added.

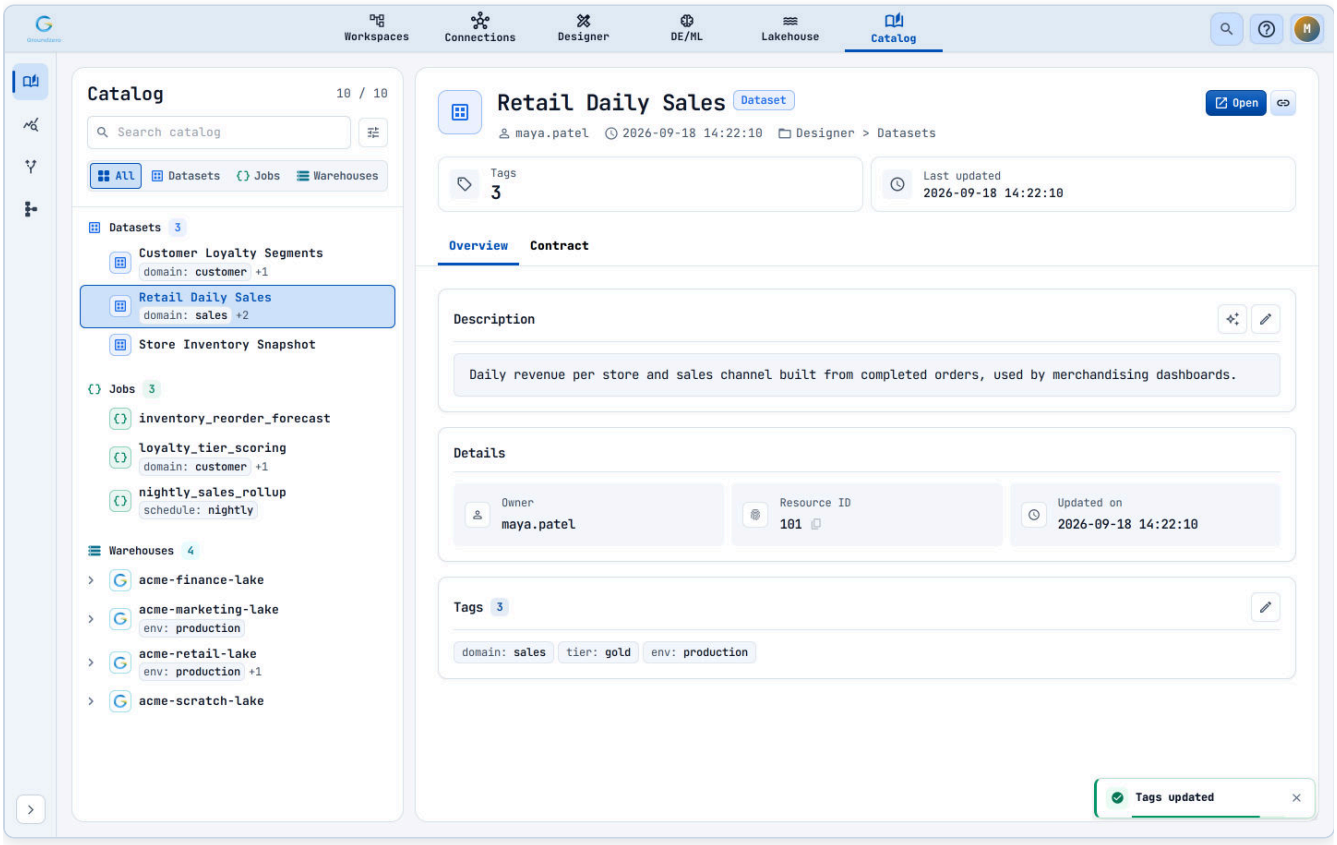


Figure 56. After saving, the new tag is shown and counted.

The Tags card shows the first 8 tags, then "+N more" and "Show less".

ONLY ADMINISTRATORS CAN SAVE WAREHOUSE TAGS

On a warehouse, other users can open the editor but see "Unable to update warehouse tags" when they save.

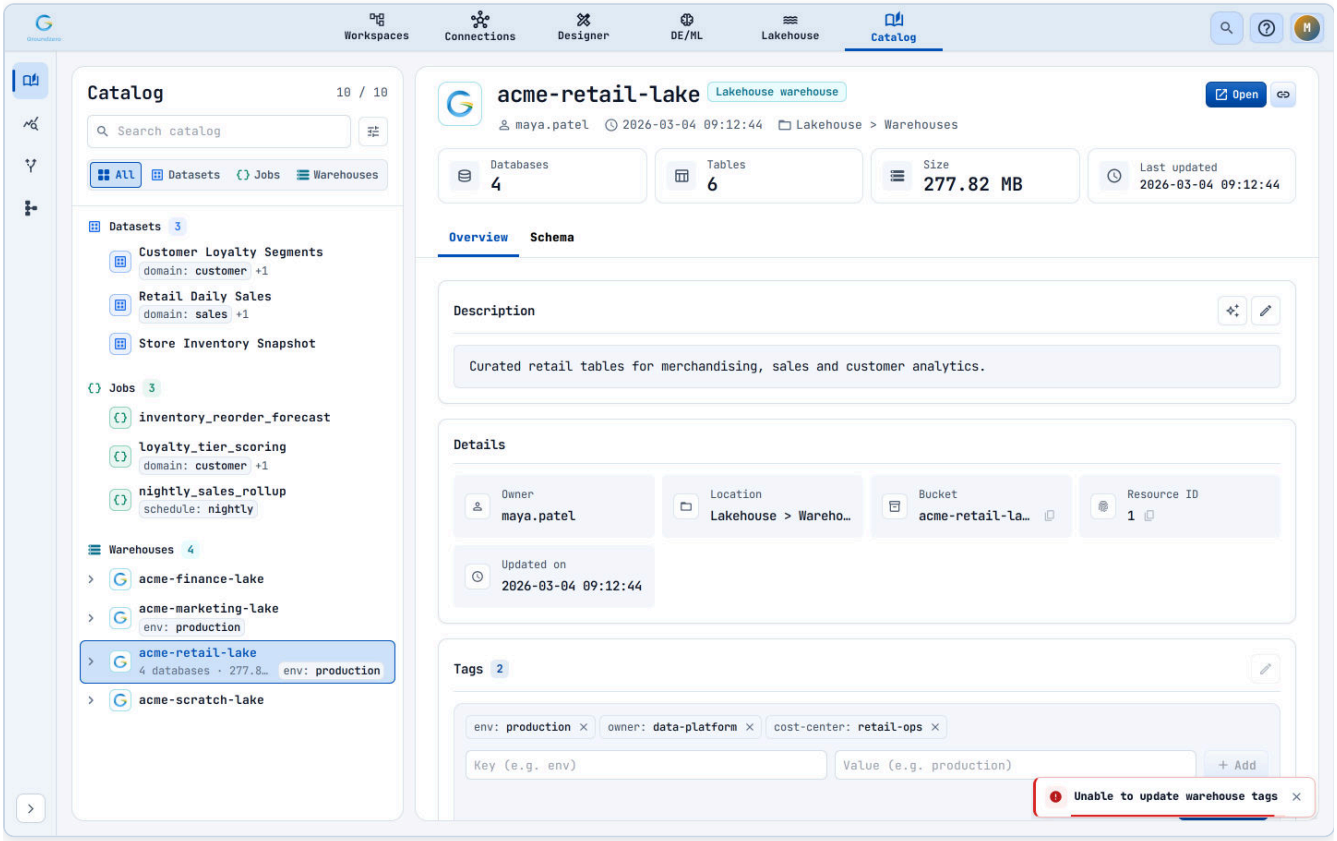


Figure 57. A forbidden tag save shows a generic notification.

4.4.2 Tag a database or table

1. On the **Schema** tab, click the pencil beside the database or table.
2. Change an existing tag or add one with **New tag key**, **New tag value** and **Add**.
3. Click **Save tags**. The editor closes; there is no success message. **Cancel** closes it without saving.

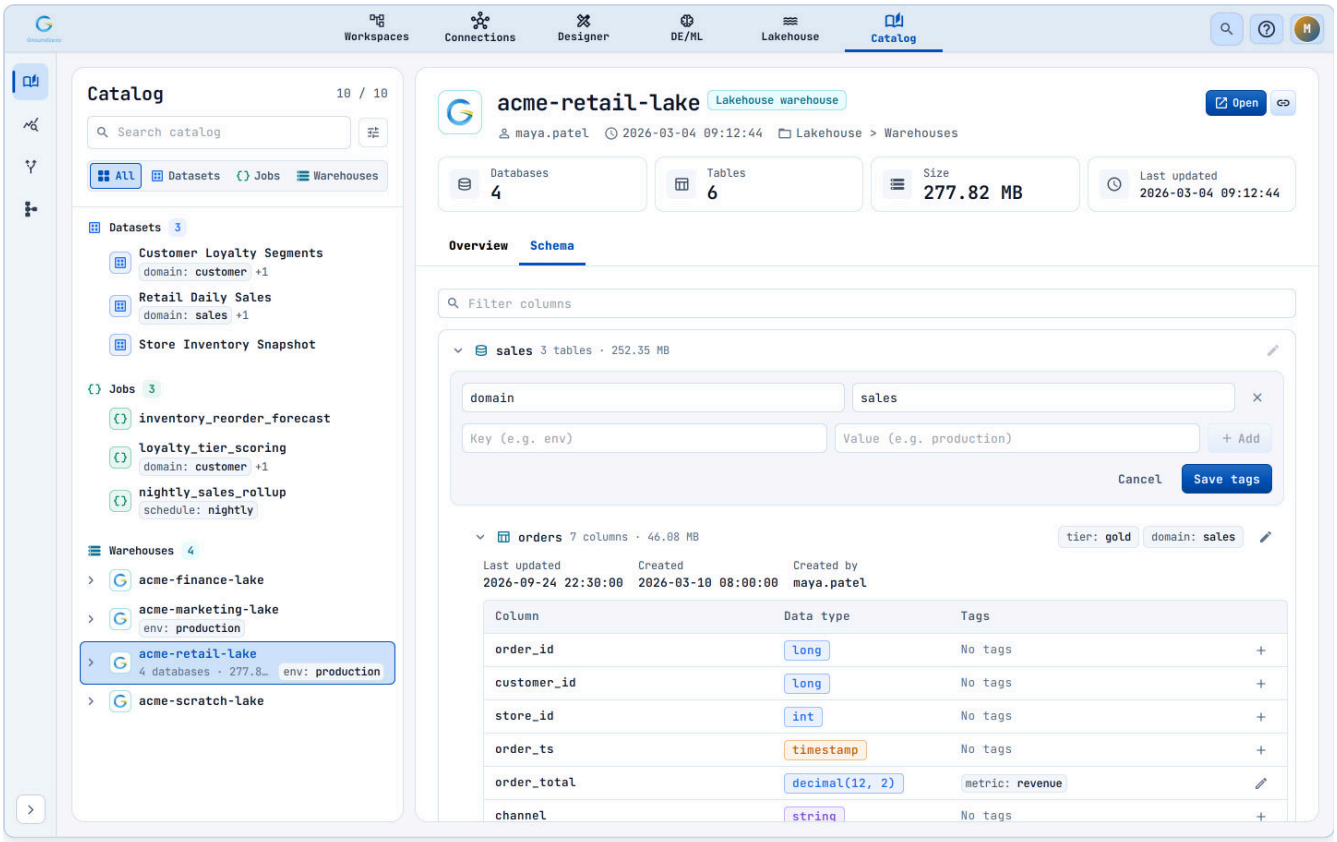


Figure 58. The database tag editor shows editable rows.

REMOVING A ROW DELETES THE TAG IMMEDIATELY

The x on a row deletes that tag at once, before you click Save. If the delete fails, the row comes back with an error.

4.4.3 Tag a column

Each column has a tag button: **Add tags**, **Edit tags** or **View tags** (administrators can edit; others can only view). The button is disabled if the column cannot be tagged. The tag cell shows "Loading tags...", "Tags unavailable", "No tags", or up to two tags plus "+N".

The **Tags for <column>** dialog edits all tags of the column as one JSON object, for example `{ "classification": "restricted" }`.

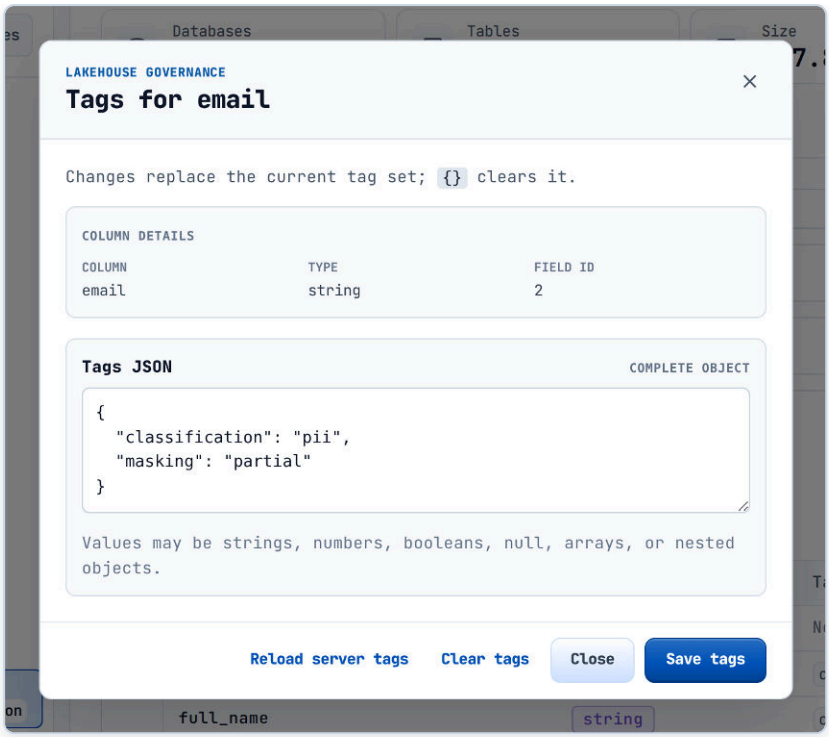


Figure 59. The column tag dialog edits one JSON object per column.

Button	What it does
Reload server tags	Fetches the latest saved tags
Clear tags	Empties the box; the tags are removed when you save
Save tags	Replaces the whole tag set with what is in the box
Close	Closes the dialog

If someone else changed the tags first you see "Tags changed on the server. Your draft is kept. Reload server tags, review the changes, then save again." Your text is kept; reload, review and save again.



Figure 60. A conflicting save keeps your draft and explains what to do.

Other messages: "Fix the JSON before saving.", "Use a JSON object, such as { "classification": "restricted" }.", "Unable to load column tags. Check your connection and try again."

> **TIP**
> Use one vocabulary everywhere, for example `domain: sales`, `tier: gold`, `classification: pii`. Then the **Tag** filter finds everything with one tag across datasets, jobs and warehouses. Use **Missing tags** to find resources nobody has classified yet.

4.5 Copy the call for a dataset or job

The **Contract** tab of a dataset or job shows how to run it from outside Groundzero, for example from a script. Each card has a copy icon.

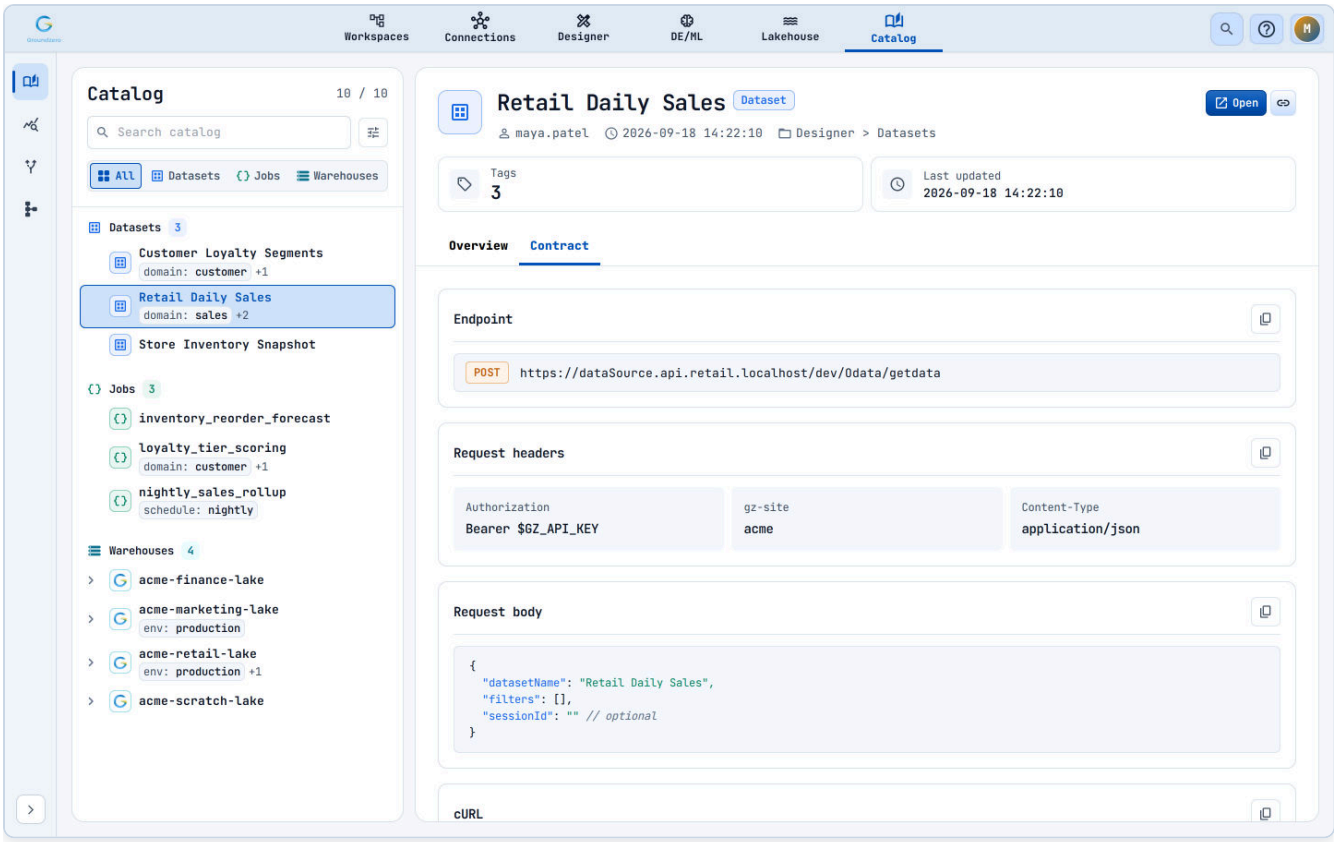


Figure 61. The dataset contract: endpoint, headers, body and cURL.

Card	What it gives you
Endpoint	The URL to call. It reads "Resolving endpoint..." for a moment first.
Request headers	The headers to send.
Request body	The values to send; lines marked `// optional` can be left out.
cURL	A ready command that combines the three

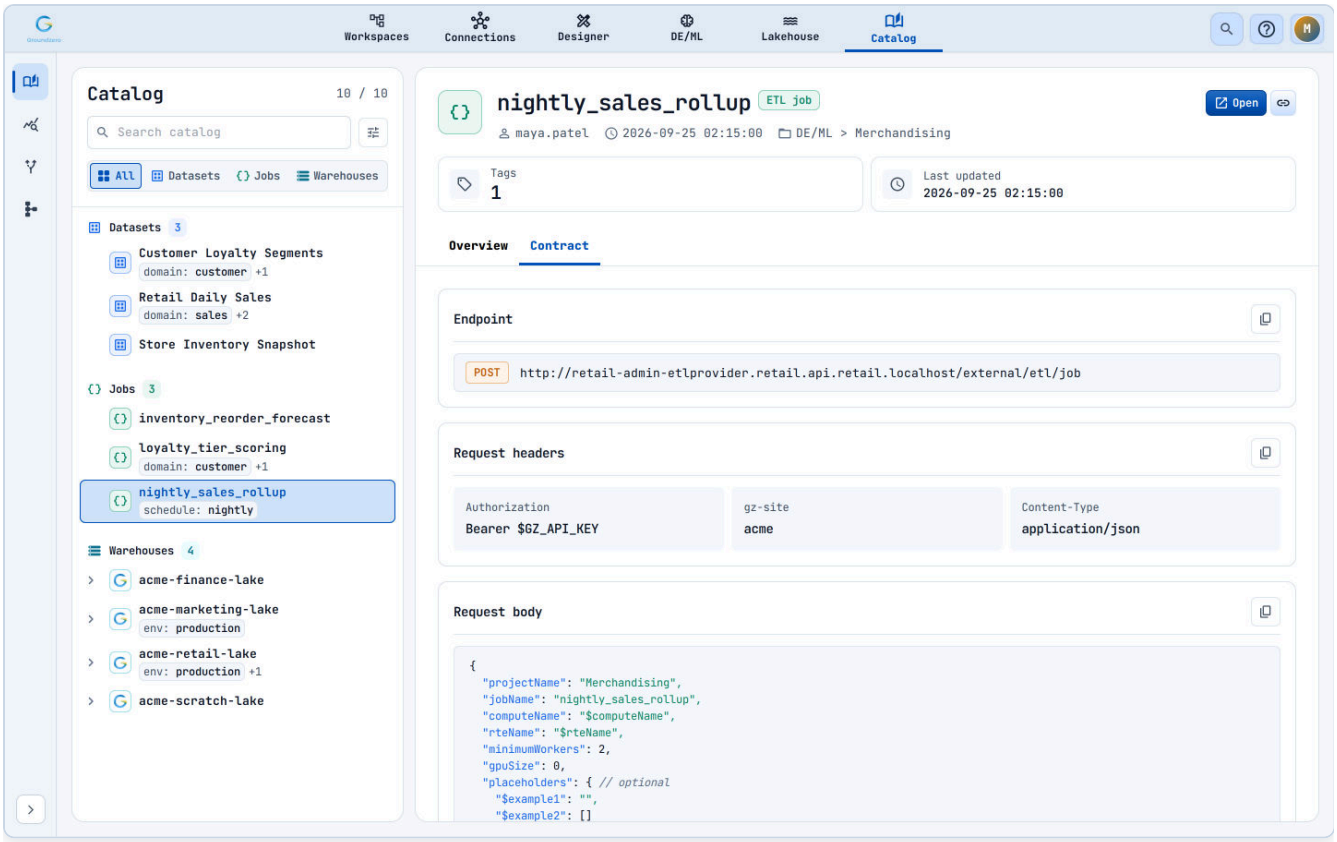


Figure 62. The job contract shows the job endpoint and a job request body.

Values that start with `$`, such as `$GZ_API_KEY`, are placeholders you fill in yourself. The catalog does not create API keys. Who may run the dataset or job still depends on their permissions.

4.6 When the catalog shows a message

What you see	What it means
Shimmering bars, "Loading catalog..."	Resources are loading
"No catalog resources yet"	No dataset, job or warehouse exists yet
"Datasets failed to load", "ETL jobs failed to load" or "Lakehouse warehouses failed to load"	One source failed; the other two still load
"No catalog resources found. Try a different search or clear the filters."	Your search or filters match nothing
"Unable to load <name> config"	The details of that resource did not load; select it again

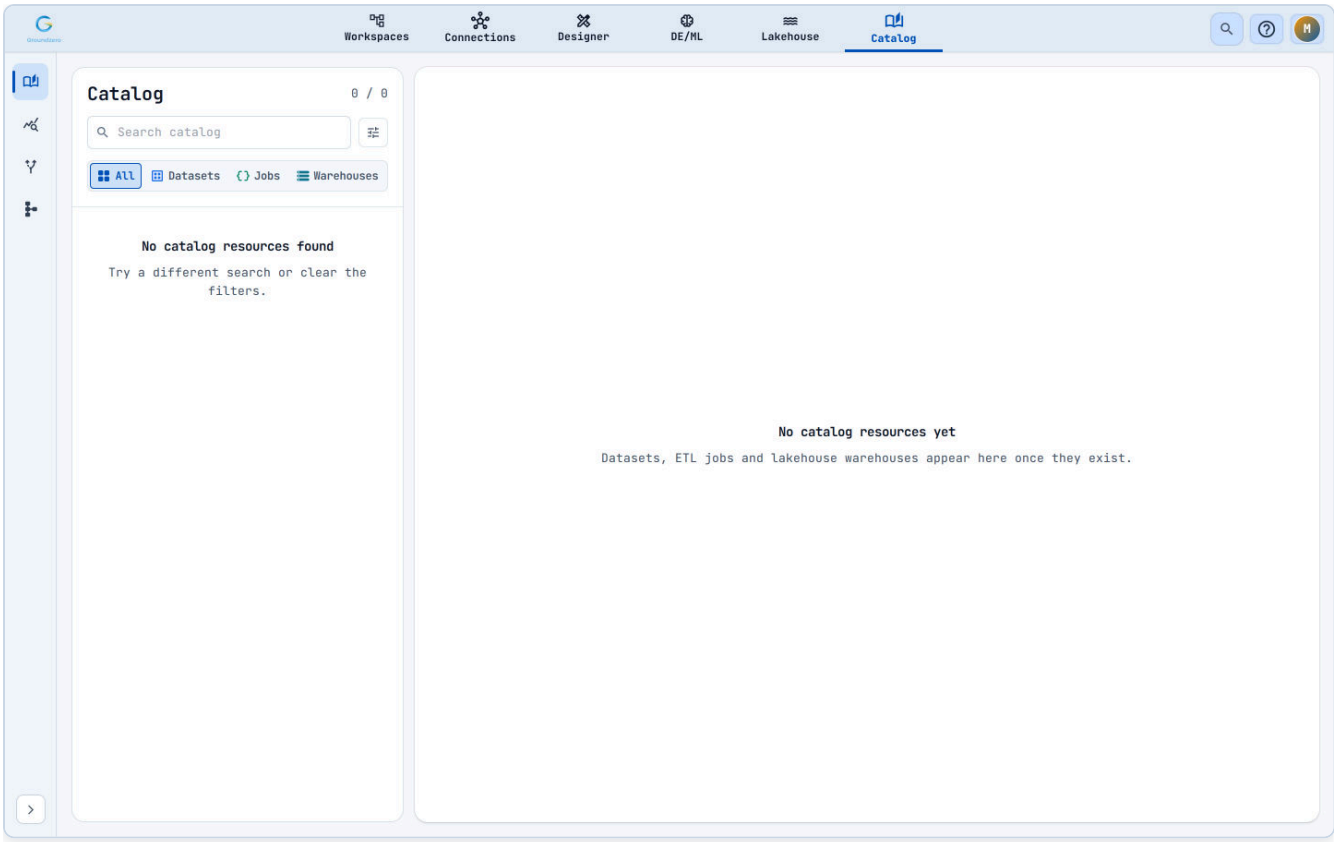


Figure 63. With no resources the catalog explains what will appear.

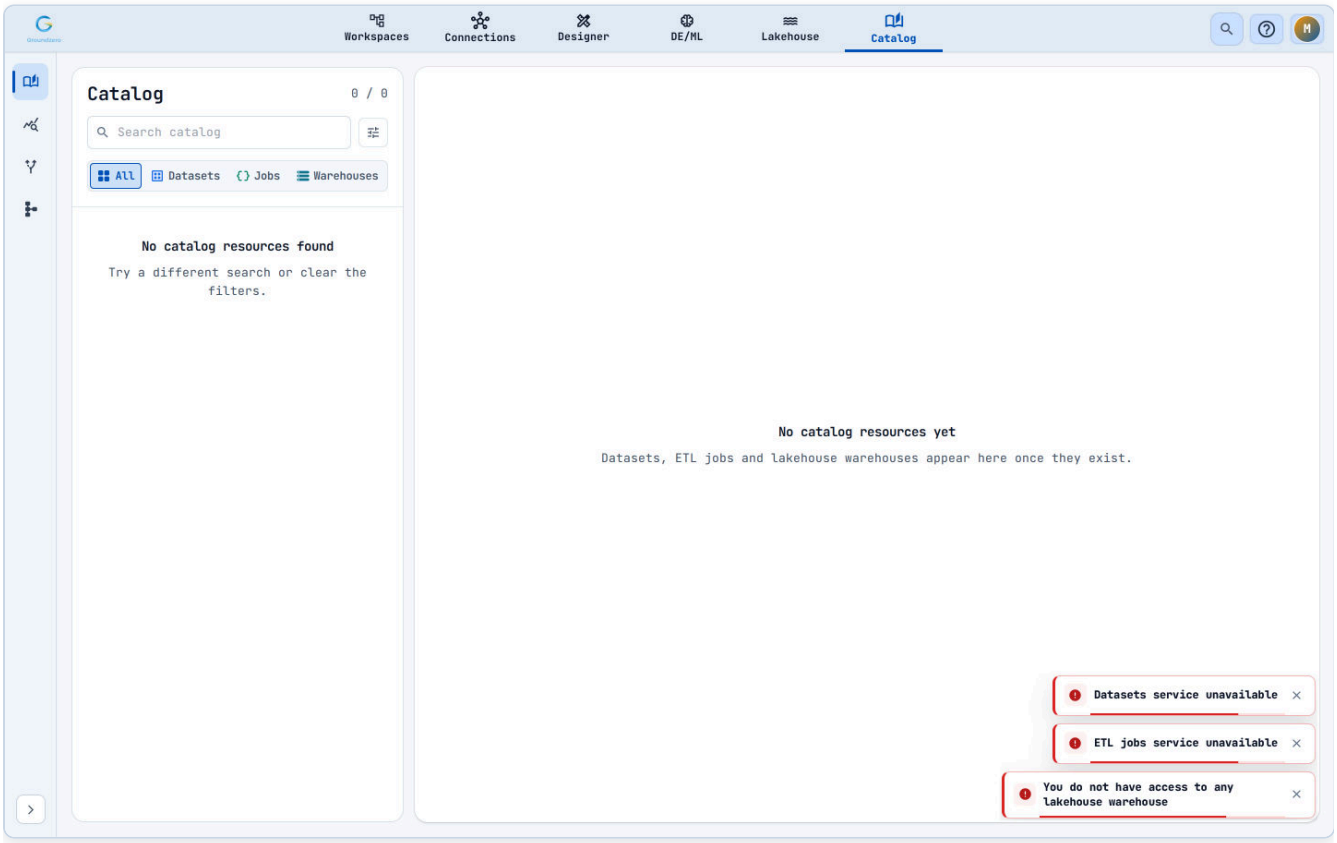


Figure 64. When a source fails, the others still load and an error notification appears for each failure.

4.7 Troubleshooting

Symptom	Cause	Fix
The catalog is empty but you expect items	One source failed, or you may not list the warehouse	Read the error notifications; ask an administrator for list access
"No schema available for this warehouse yet."	The warehouse has no tables, or you may not list it	Create a table, or ask for access
Search finds nothing inside a warehouse	Warehouses are still loading, or you typed one character	Type at least two characters and wait a moment
"Unable to update warehouse tags"	You are not an administrator	Ask an administrator to save the tags
The column dialog says the tags changed	Someone saved first	Click **Reload server tags**, review, save again
The column tag button is disabled	That column cannot be tagged	None needed
A change made elsewhere is not shown	The catalog keeps details for your session	Reload the page
Copy buttons do nothing	The browser blocked clipboard access	Allow clipboard access for the site

4.8 Related

- **Data lineage** and **Ontology Studio**: the other two entries in the Catalog menu.
- **Lakehouse**: the warehouses, tables and policies that the Schema tab shows.
- **Designer** and **DE/ML** (Build documentation): where datasets and jobs are created.

5

Lakehouse and warehouses

The **Lakehouse** is Groundzero's managed store for analytic tables. You register an S3 bucket as a **warehouse**, then create databases and tables in it, run SQL, export results, protect sensitive data and back databases up, all from the Lakehouse page.

This chapter is about using the Lakehouse page. For the SQL itself (statements, data types, creating and changing tables, time travel, maintenance and worked examples) see the **Lakehouse runtime guide**.

Open **Lakehouse** in the top navigation bar. The left rail has **Warehouses** and **Saved Queries**. The rail starts collapsed to icons; use the toggle at the bottom to expand it.

Term	Meaning
Warehouse	An S3 bucket you have registered. Use its name as the catalog name in SQL.
Database	A group of tables inside a warehouse.
Table	A table stored in the open Iceberg format. You refer to it as `database.table`.
Session	A query engine started just for you, for one warehouse. SQL does not run until you start one.
Data policy	A row filter and column masks that apply to every read of a table.
Table Access Manager	Allow and deny rules for one table, per user or group.

I want to…	Go to
Register a bucket	"Create a warehouse"
Run my first query	"Start a session", then "Run a query"
Get all rows of a large result	"Export a result"
Reuse a query	"Save and reuse queries"
Hide personal data from some readers	"Protect a table with a data policy"
See who ran what	"Review the audit log"
Keep a copy of a database	"Back up and restore a database"

5.1 Permissions in short

An administrator grants permissions in Access Manager. The creator of a warehouse automatically gets `All` on it.

To do this	You need
See a warehouse in the list	`List` or `All` on the warehouse
Create or delete a warehouse	`Create` or `Delete` (or `All`) on warehouses
Create or drop a database	`CreateDatabase` or `DropDatabase` (or `All`) on the warehouse

To do this	You need
Run SQL	The matching action on the warehouse, database and table (see "Why a statement is refused")
Open a table's data policy	`Select` on the database; `All` to save it
Open Table Access Manager	`All` on the database

5.2 Create a warehouse

Outcome: an S3 bucket that Groundzero can store tables in. Open **Warehouses**.

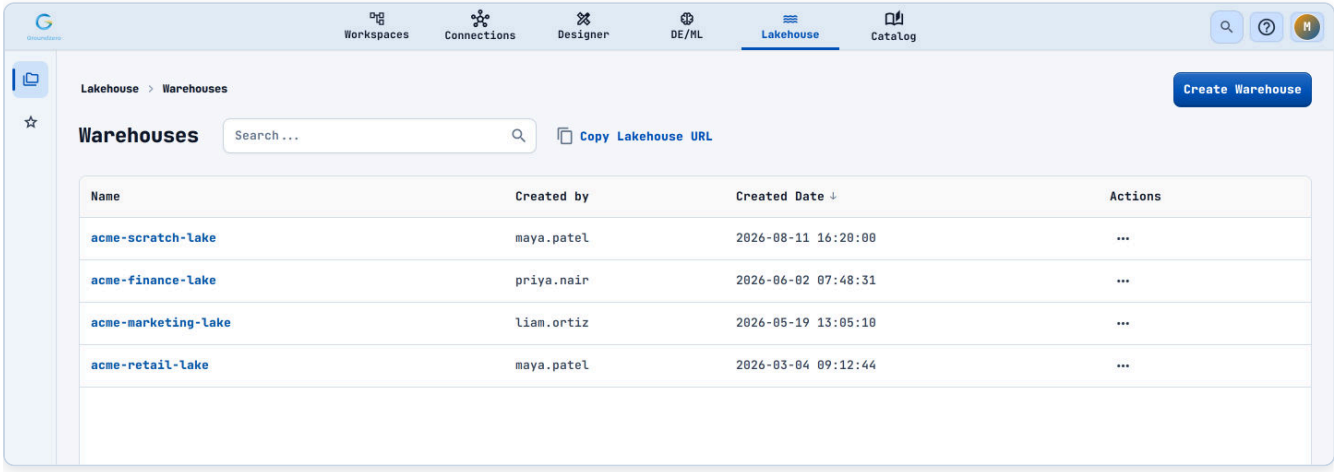


Figure 65. The Warehouses list shows name, creator and creation date.

The list has a **Search...** box, **Copy Lakehouse URL**, **Create Warehouse** and a table with **Name**, **Created by**, **Created Date** and an **Actions** menu. If nothing exists you see "No warehouses to show".

1. Create the S3 bucket in AWS first. Its name must be `gz-lakehouse-<name>`.
2. Click **Create Warehouse**. The dialog says "Register an S3 bucket as a lakehouse warehouse."

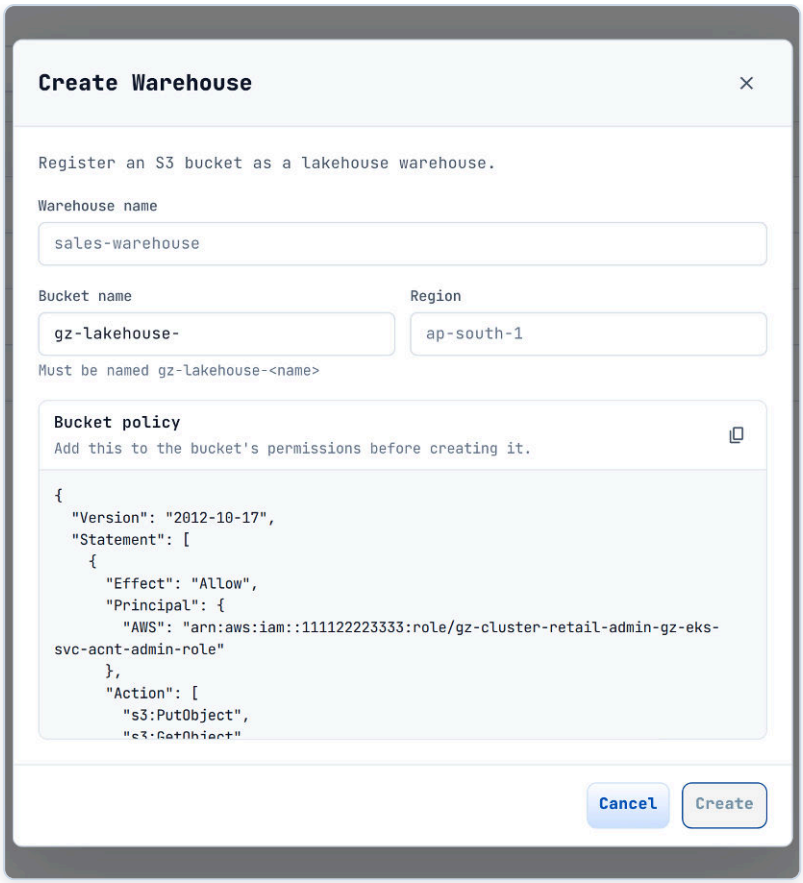


Figure 66. The Create Warehouse dialog with the generated bucket policy.

3. Fill in the three fields.

Field	Example	Rule
Warehouse name	`sales-warehouse`	Letters, digits and single hyphens; starts with a letter; no underscore
Bucket name	`gz-lakehouse-sales`	Starts with `gz-lakehouse-`
Region	`ap-south-1`	The AWS region of the bucket

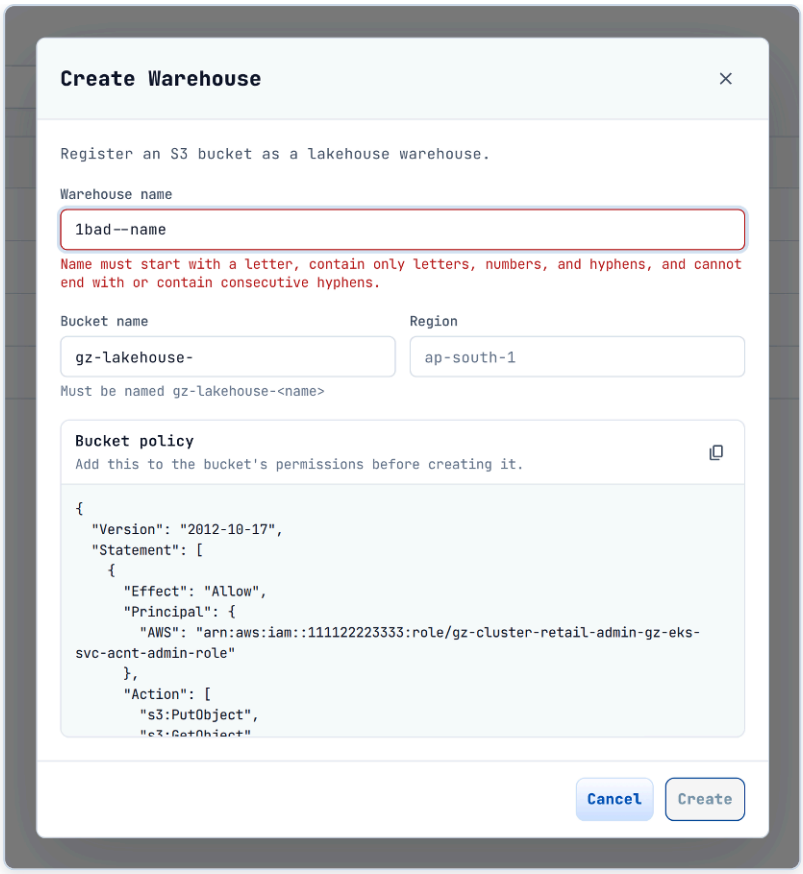


Figure 67. An invalid name is flagged immediately.

4. Click **Copy policy** next to **Bucket policy** and add that policy to the bucket's permissions in AWS. It lets Groundzero use the bucket. Do this *before* you click **Create**. You see "Policy is copied to clipboard successfully".
5. Click **Create**. It turns on once all three fields are valid.

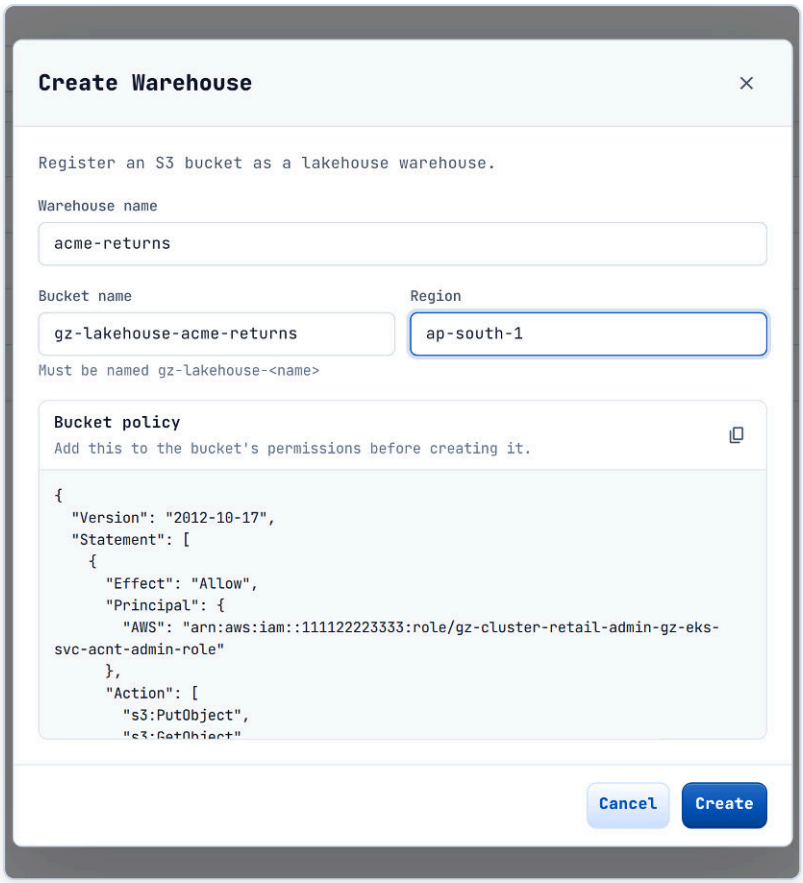


Figure 68. The completed form; the policy refers to the new bucket.

You see "Warehouse created successfully" and the warehouse appears in the list.

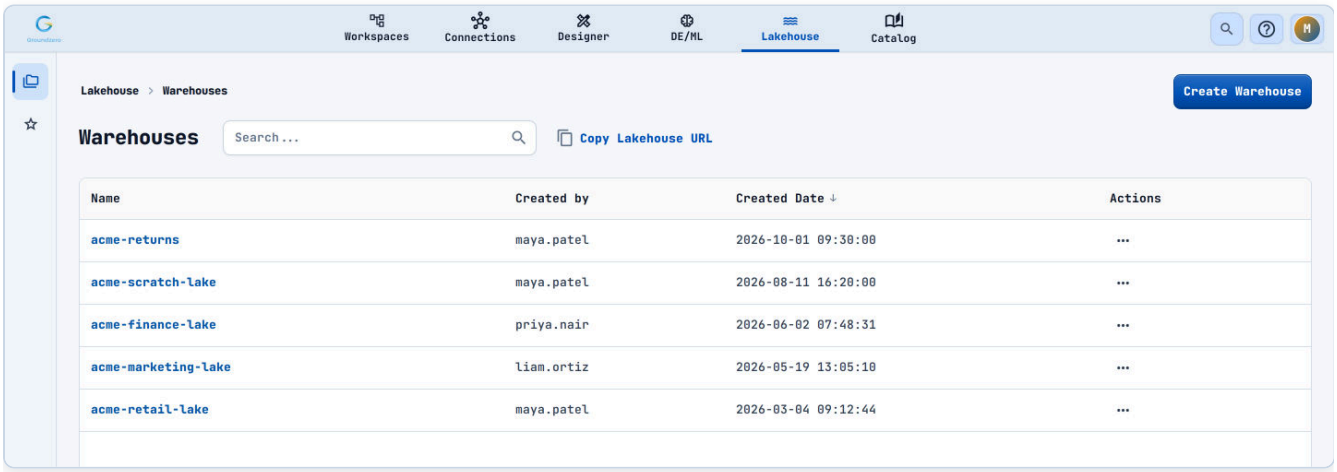


Figure 69. The new warehouse appears in the list.

If it does not appear, check the name and your `Create` permission.

5.2.1 Other warehouse actions

Open the "..." menu on a row.

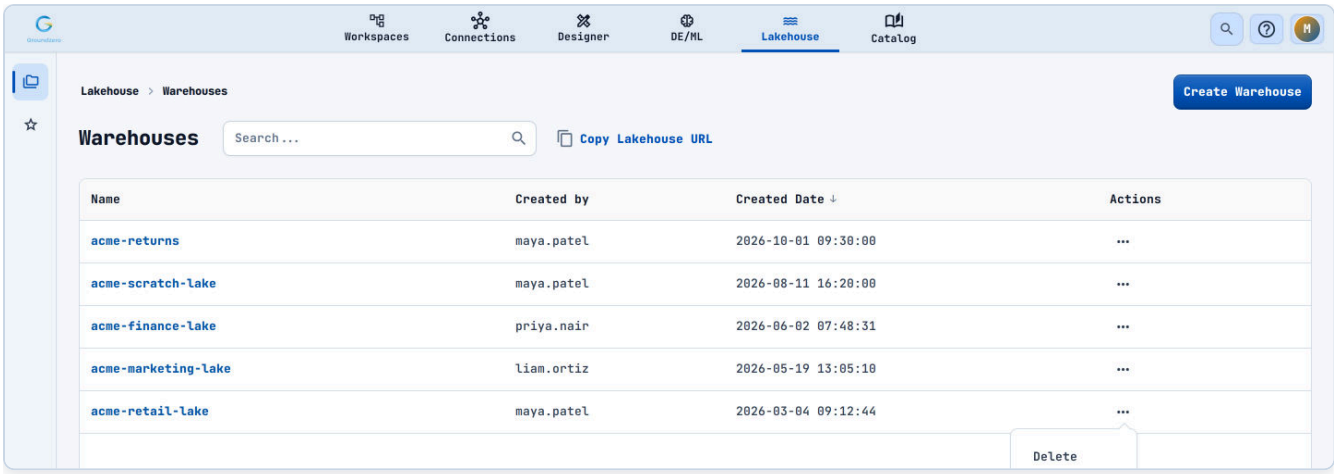


Figure 70. The row menu offers Delete and Copy Policy.

- **Copy Policy** copies the bucket policy again for that warehouse.
- **Delete** asks "Are you sure you want to delete this warehouse <name> ?". Choose **Yes**; you see "Deleted successfully.".

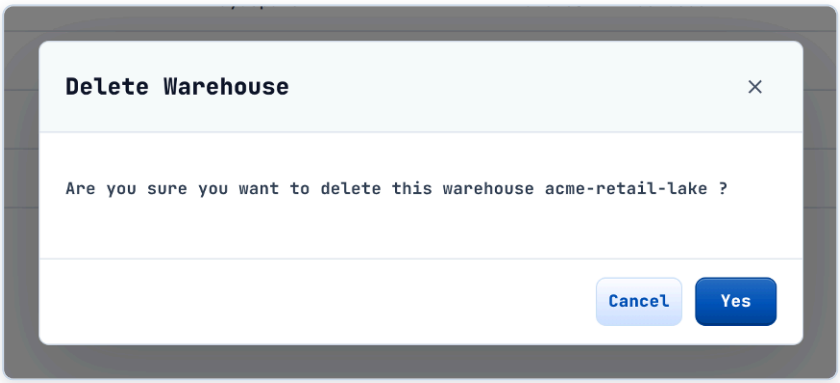


Figure 71. The confirmation names the warehouse.

A warehouse that still has databases, tables or views cannot be deleted:

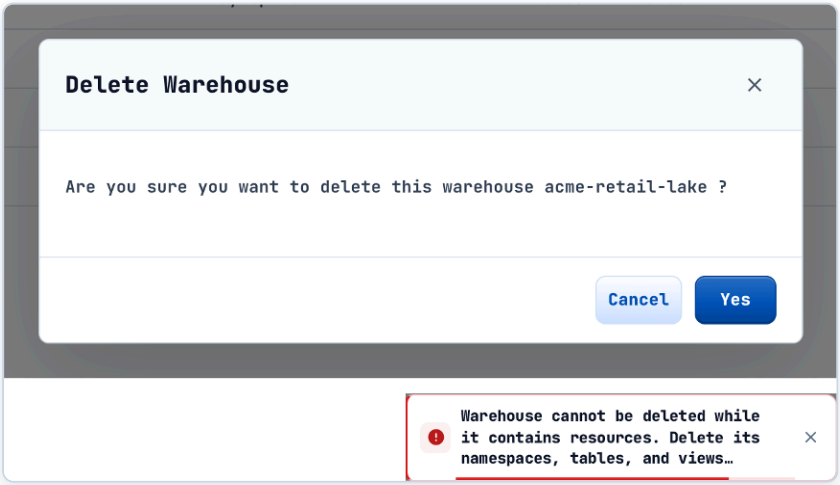


Figure 72. A non-empty warehouse is protected.

You see "Warehouse cannot be deleted while it contains resources. Delete its namespaces, tables, and views first.". The built-in `audit` database does not block deletion.

- **Copy Lakehouse URL** (on the page header) copies the lakehouse host name. Paste it into the **Lakehouse URL** field of a Groundzero Lakehouse connection (see the Data source connections chapter). If it reads "Lakehouse URL is not available yet", wait a moment.

5.3 Open a warehouse and start a session

Click a warehouse name to open its workspace.

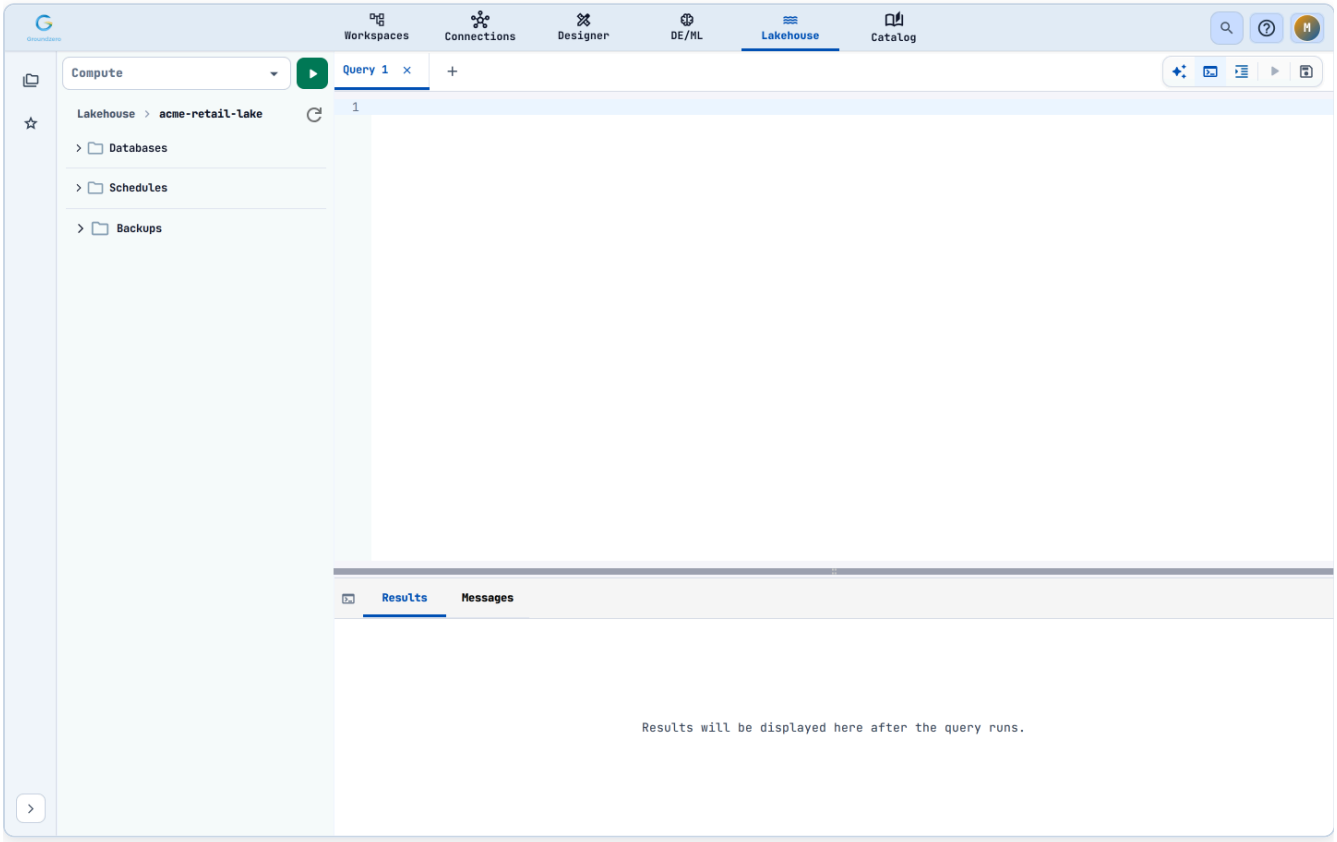


Figure 73. A workspace before a session is started: compute selector, schema tree, schedules, backups and an empty editor.

The left column has the compute selector, the session button, the database tree, **Schedules** and **Backups**. The right side has the query tabs and the editor.

5.3.1 Start a session

Outcome: a running query engine for you, so that **Run query** works.

1. Open the **Compute** list. Each option shows its memory and vCPU. Choose one. If you have no permission to use compute, the control reads "Insufficient Permission" and is disabled.
2. Set **Minimum Workers** with the minus and plus buttons. The default is 2; more workers help with large scans.

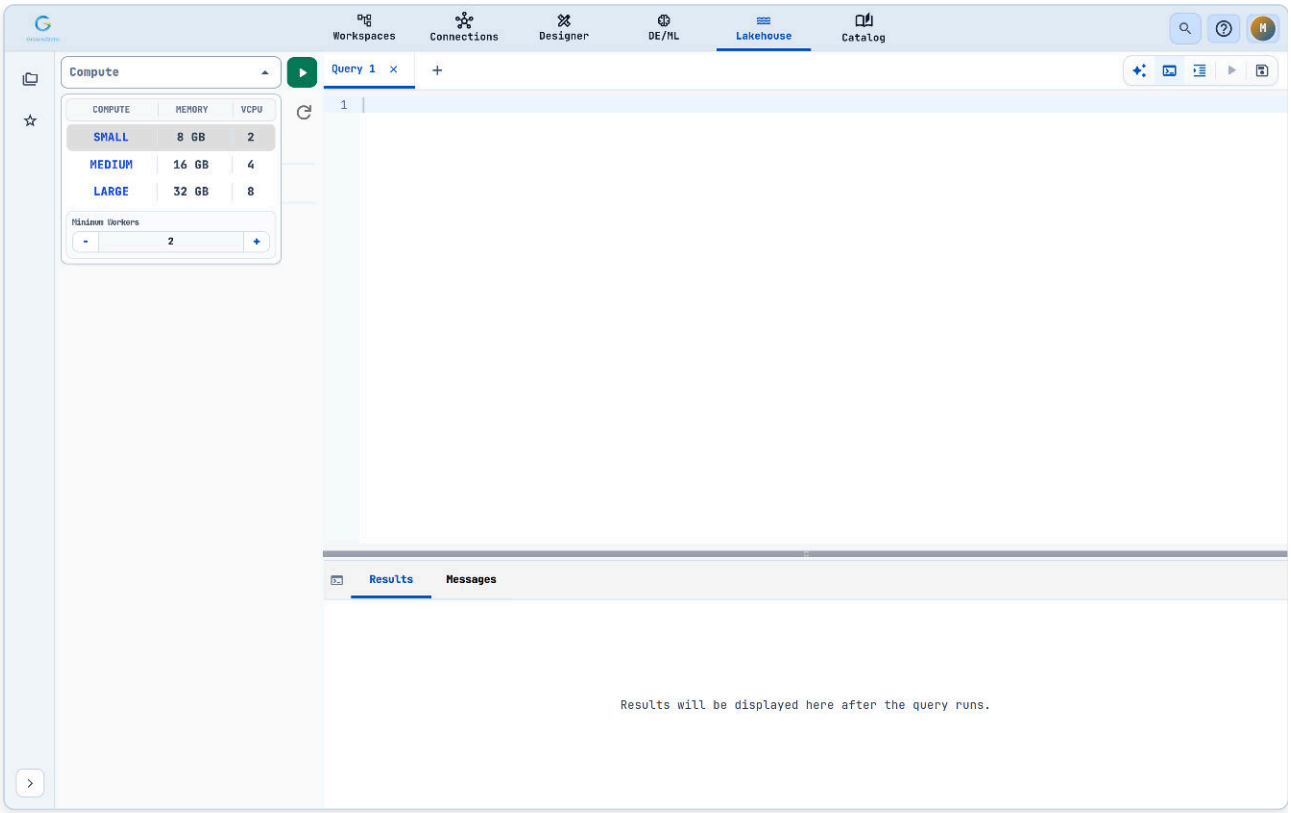


Figure 74. The Compute list and the Minimum Workers stepper.

3. Click the green **Start lakehouse session** button. If you have not chosen a compute you see "please select Compute".

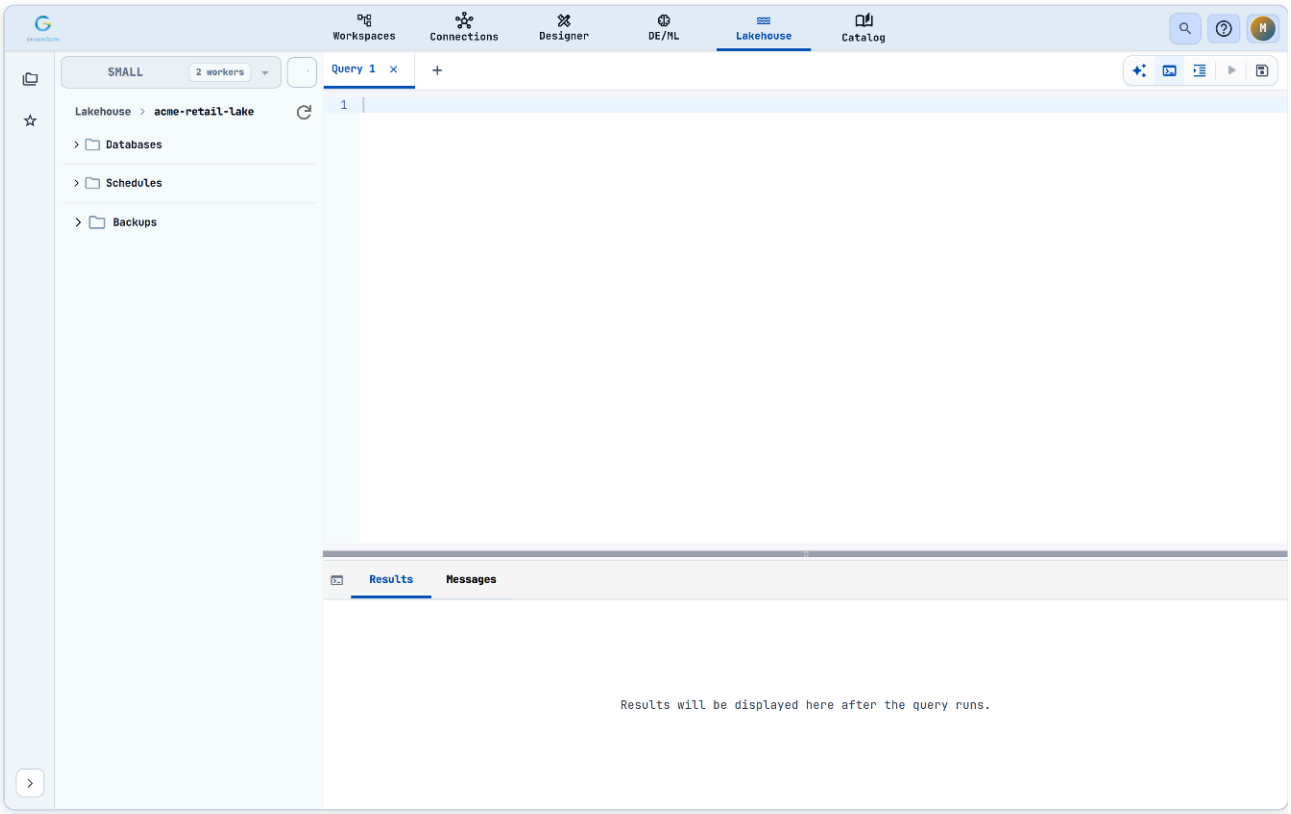


Figure 75. While the session starts, the button shows a spinner.

4. Wait for "Session started successfully". The button turns red and reads **Stop lakehouse session**, and **Run query** is on.

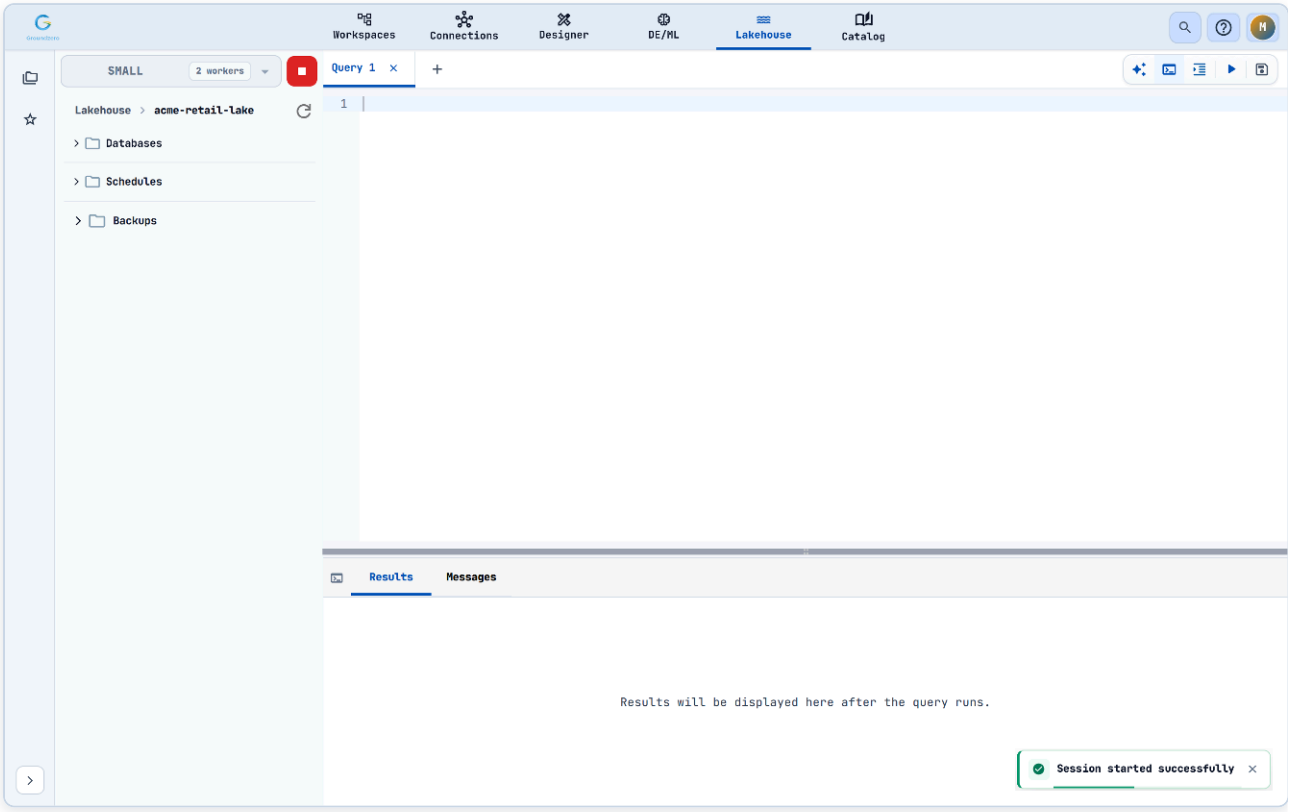


Figure 76. A running session: the compute is locked and Stop is available.

What to know about sessions:

- Starting can take up to 3 minutes. If it takes longer you see "Session start failed (200)"; try again, or choose a larger compute.
- Leaving the page and coming back within 30 minutes keeps your running session.
- A session stops by itself after 30 minutes without use. You then see "The lakehouse session is not reachable. Start the session and try again." Start it again.
- Click **Stop lakehouse session** when you are done ("Session stopped successfully"). This releases the compute.

5.3.2 Browse databases, tables and columns

Expand **Databases** in the tree to see databases (the built-in `audit` database comes first), then **Tables** and **Columns**. Columns are shown as `<column> (<type>)`. Hover a node and click **Open row actions** for its menu.

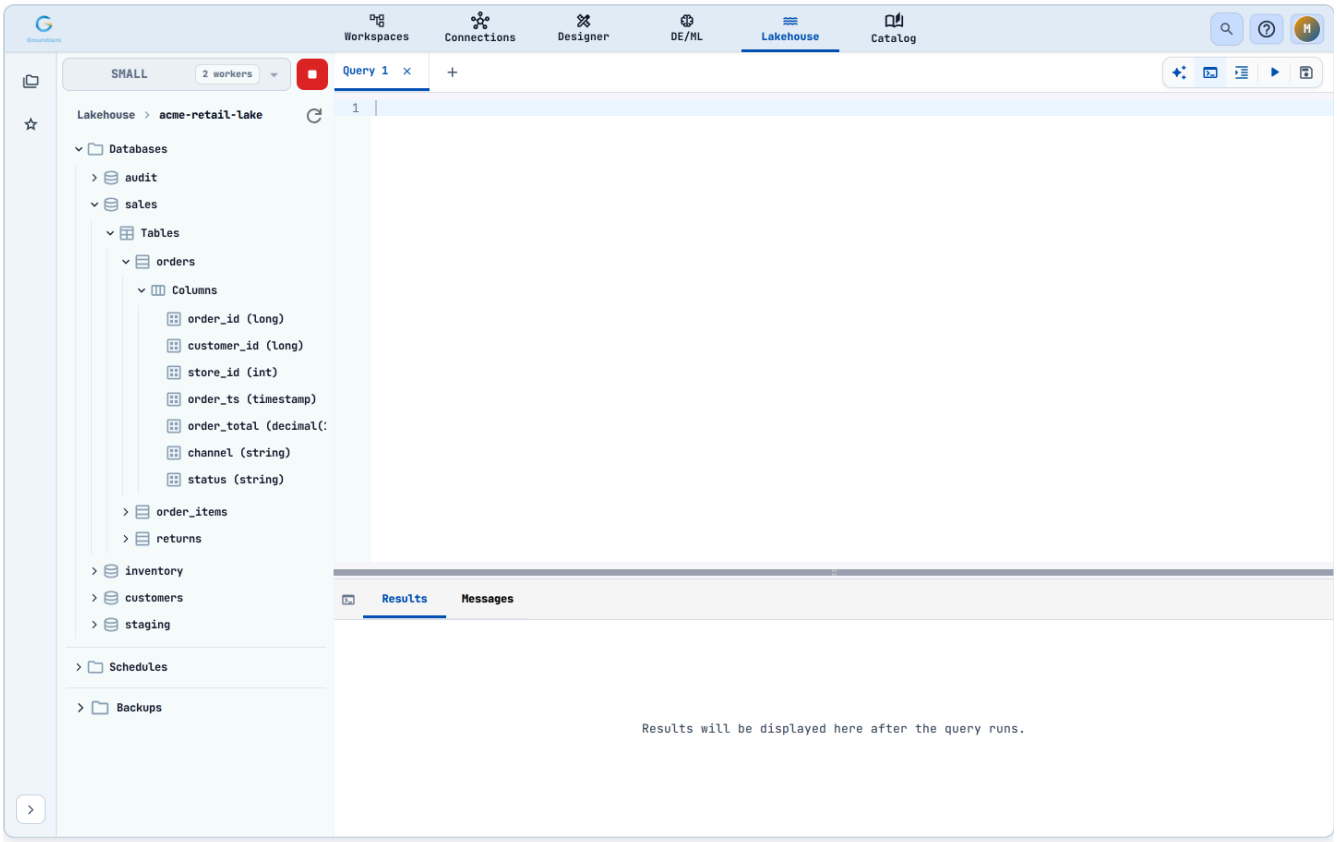


Figure 77. Tables expand to their columns.

Node	Menu items
Databases	Import Schema, Restore Database, Mark stale backups failed
A database	Properties, Export schema, Backup Database, Restore Database
A table	Add table on cursor, Properties, Data policy, Table Access Manager
`audit.logs`	Open audit logs

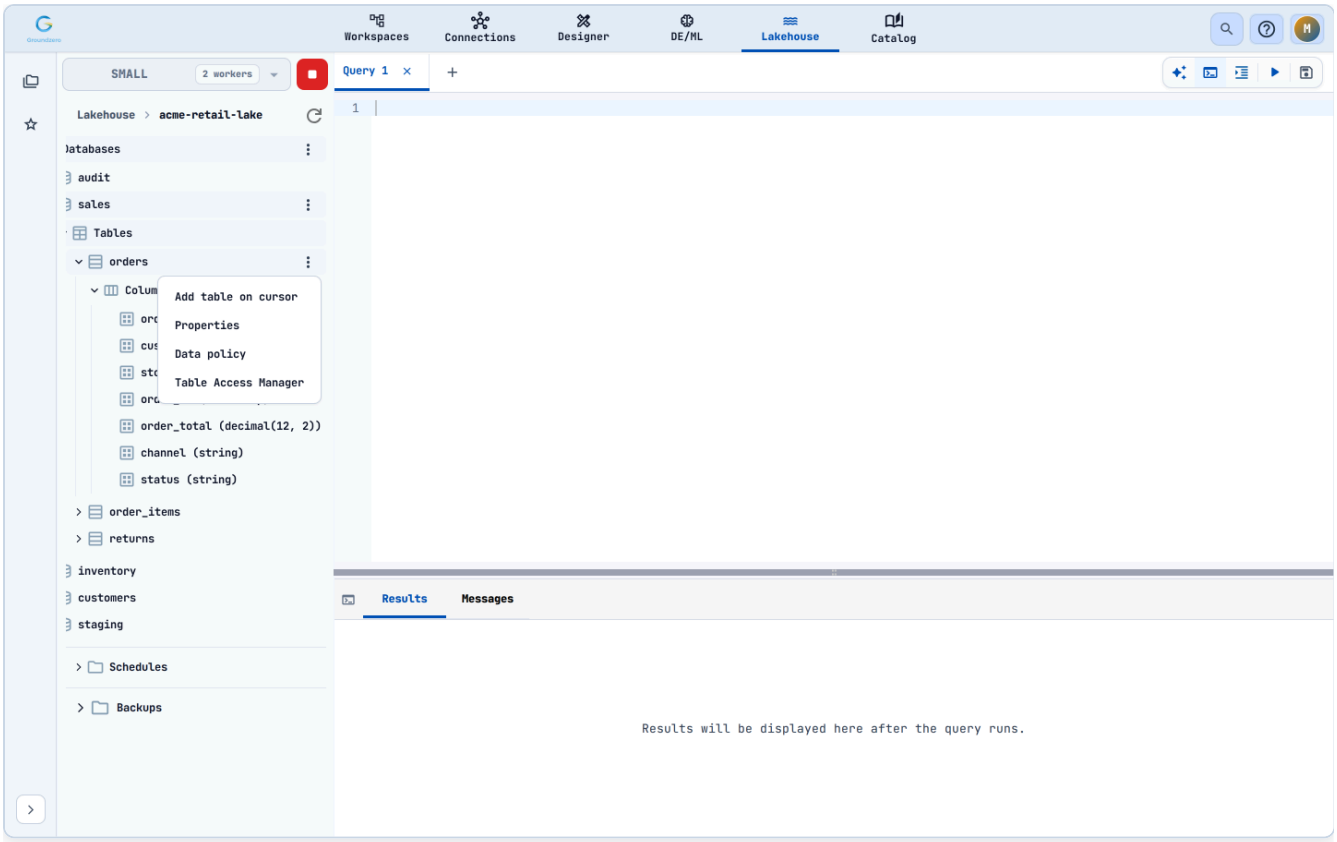


Figure 78. The table menu.

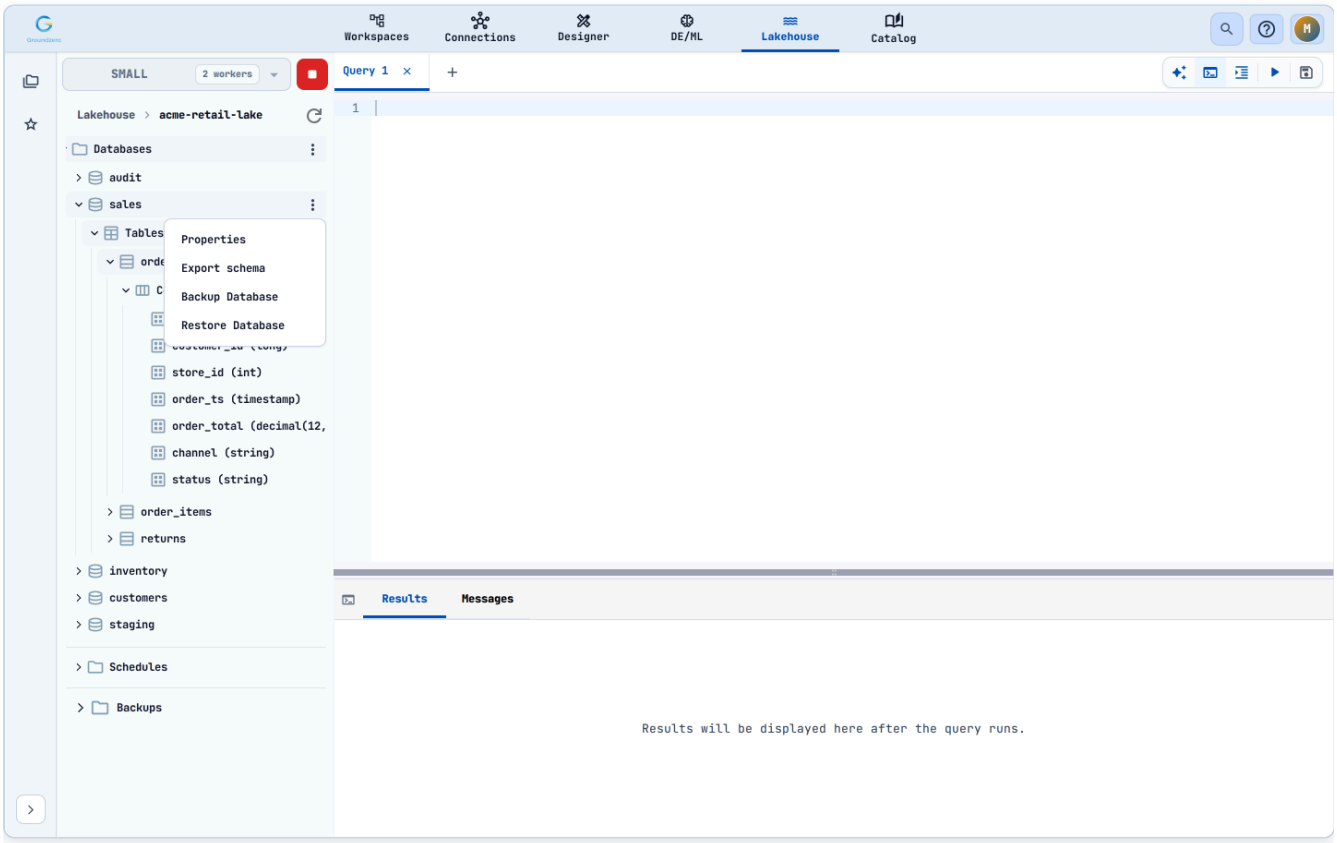


Figure 79. The database menu.

The refresh icon beside the breadcrumb reloads the tree. The editor also suggests names from the tree as you type.

Export schema downloads `schema.txt`, a list of the columns of every table in the database (no data). **Import Schema** does the reverse: choose a schema file and a **Database Name**, and Groundzero creates the database with one empty table for each entry. Existing tables are skipped.

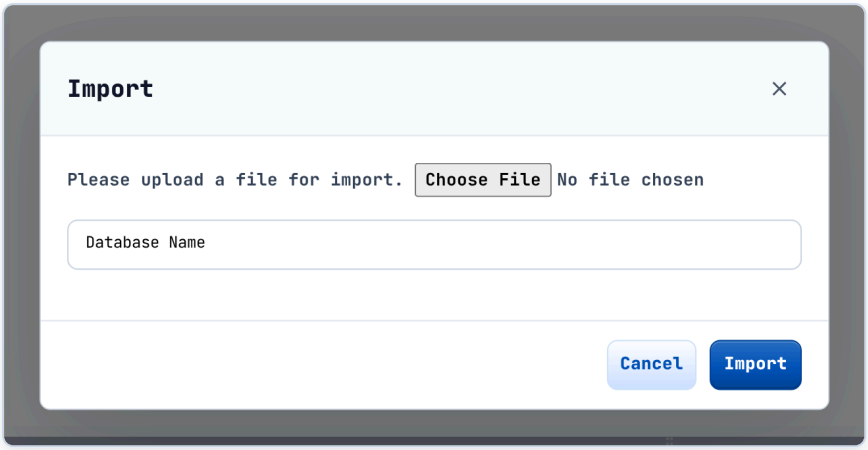


Figure 80. Import Schema asks for a file and a database name.

5.4 Run a query

Outcome: a result you can read, page through and export. You need a running session.

1. Use the first query tab, or click **+** for another ("Query 2", and so on).
2. Type your SQL. Always write tables as `database.table`, for example `sales.orders`. Autocomplete suggests warehouse, database, table and column names.
3. Click **Run query**.

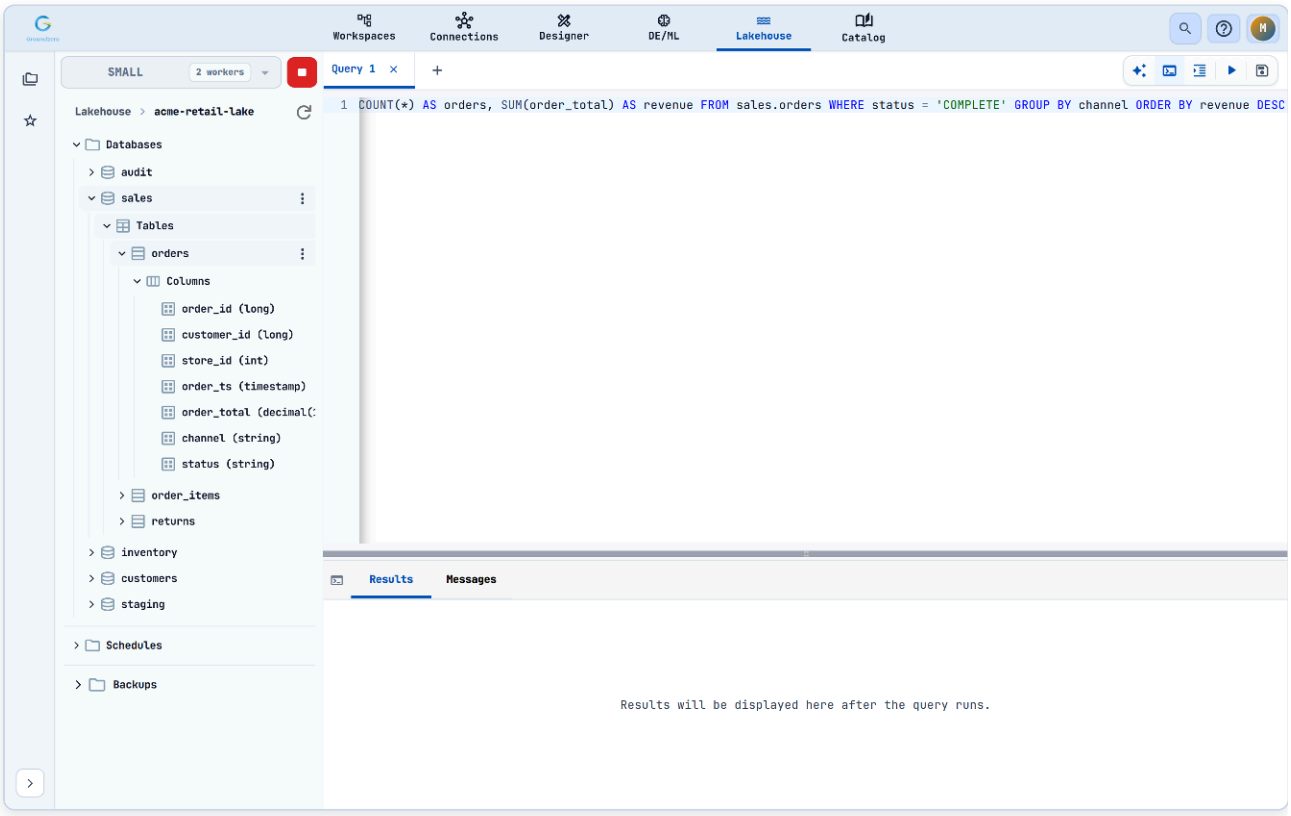


Figure 81. Typing a query in the editor tab.

4. While it runs, **Messages** shows "Executing Query..." and **Run query** becomes a red **Stop query**. A pill shows how long it has run.
5. When it finishes, the **Results** tab opens with the rows and the pill shows how long the response took.

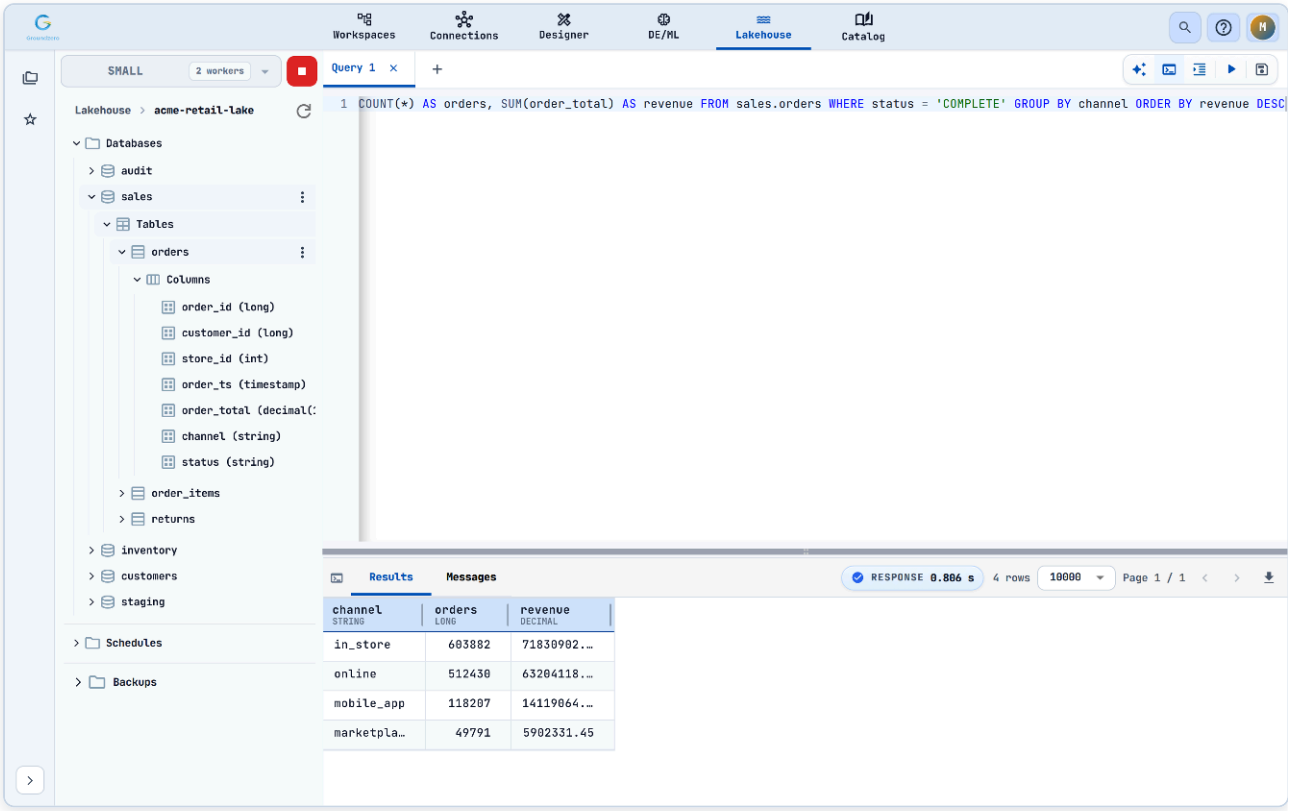


Figure 82. A result: type-labelled columns, a row count and paging controls.

6. **Messages** shows "Query executed successfully." Statements that return no rows, such as creating a table, show only this message. A query with no matching rows shows "No rows to display".

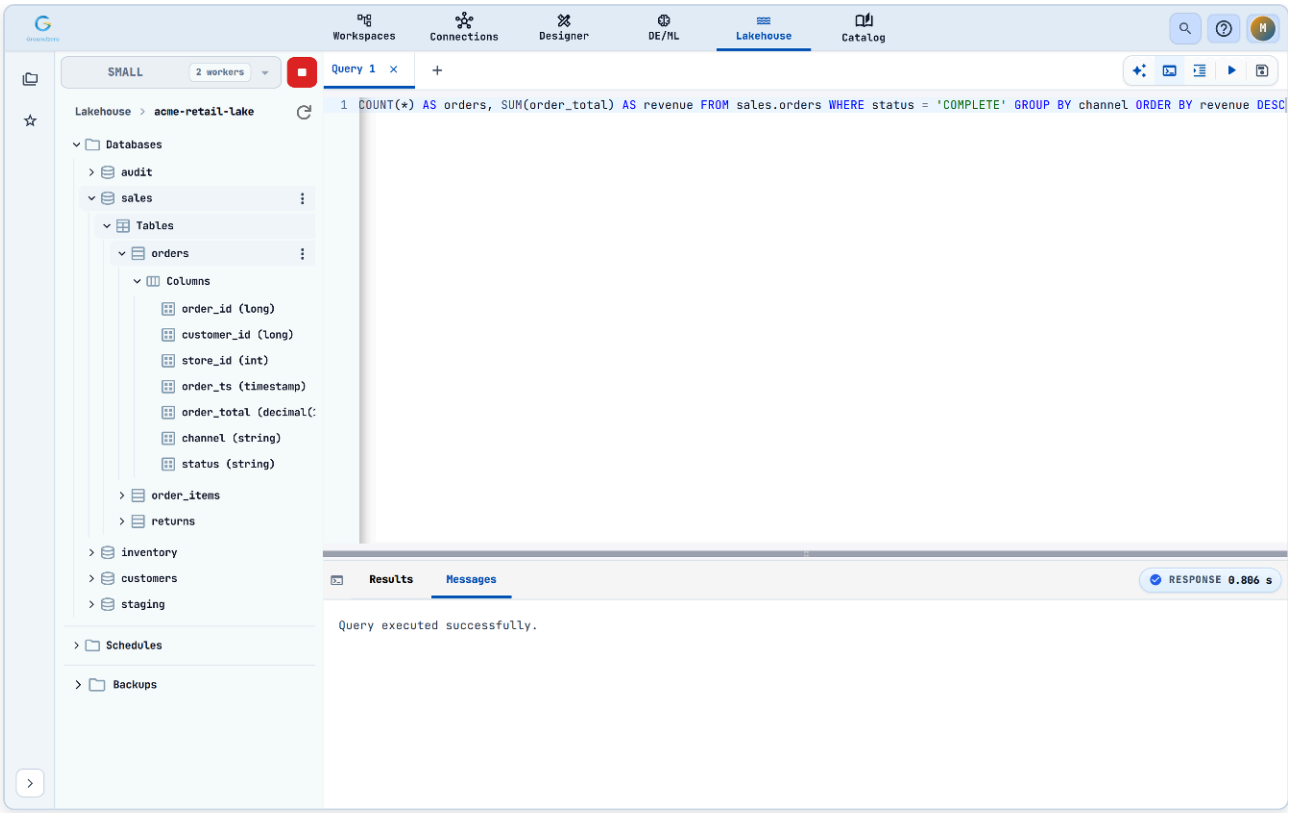


Figure 83. The Messages tab after a successful run.

The buttons at the right of the tab strip are: the Groundzero code assistant, **Toggle lakehouse output pane**, **Format SQL**, **Run query** or **Stop query**, and **Save query**.

UNSAVED QUERIES ARE LOST ON RELOAD

Query tabs live only in your browser tab. Use **Save query** to keep one.

USE BLOCK COMMENTS

Line breaks are removed before a query runs, so a `--` comment would hide everything after it. Write comments as `/* ... */`.

Format SQL tidies the query in the active tab.

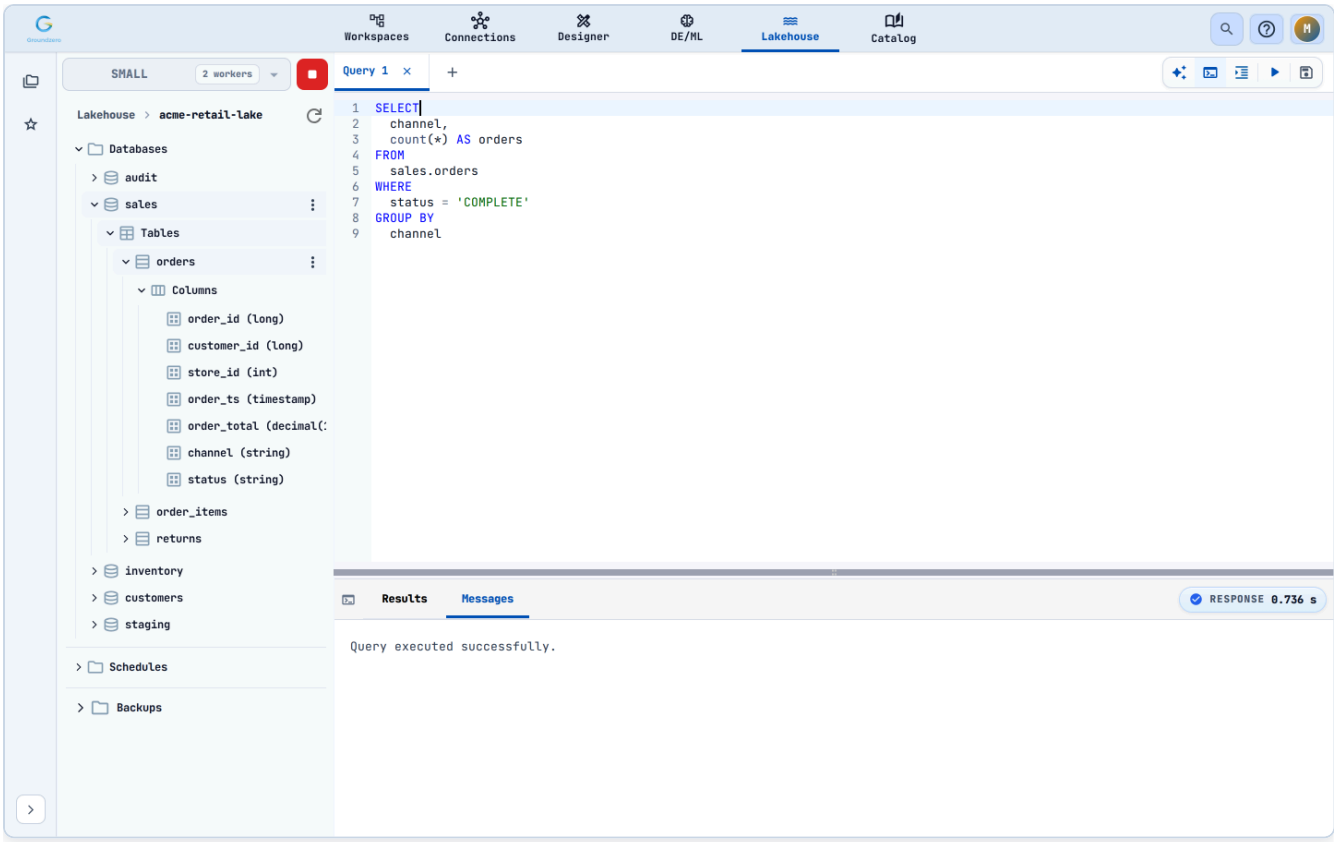


Figure 84. Format SQL lays the query out over several lines.

5.4.1 Long-running queries

- A query that takes longer than about a minute keeps running in the background. **Messages** reads "Still running. Results load here when the query finishes." and the results appear by themselves when it is done.

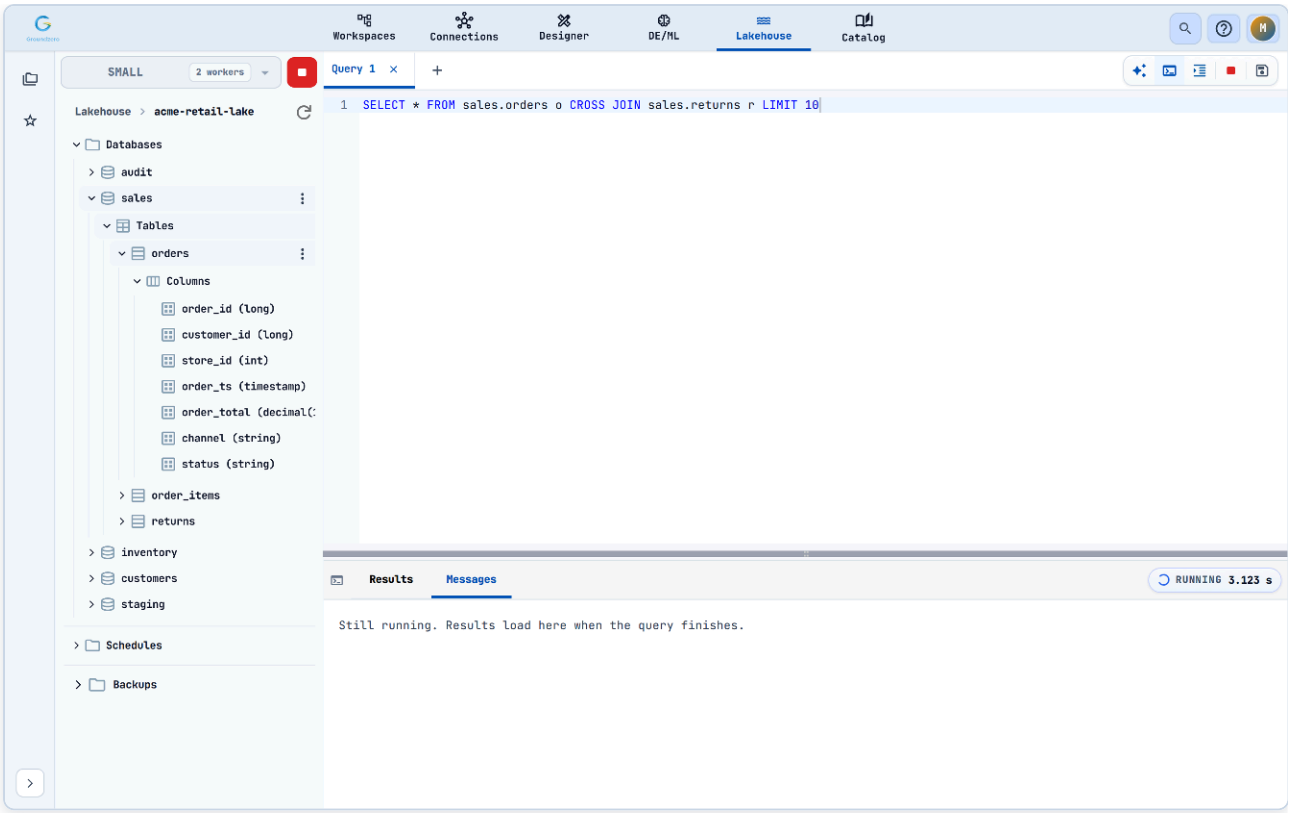


Figure 85. A detached query shows its running state.

- **Stop query** cancels it ("Query stopped.").
- A query is stopped after **30 minutes** ("Query ran longer than 30 minutes and was stopped.").
- A session runs up to **4 queries** at once. Otherwise you see "This session is already running 4 queries. Wait for one to finish or stop it, then run this query again."

5.4.2 Why a statement is refused

You see	Cause	Fix
'Use' keyword not allowed. Use fully qualified table names.	You wrote `USE <db>`	Write `database.table` instead
Use fully qualified table names.	A table was written without its database	Add the database
Following namespaces specified in the query does not exist: ...	The database does not exist	Check the name in the tree
Access denied: Unable to parse query for authorization.	The SQL could not be understood	Check the syntax
You do not have All or Select permission (or Update, Insert, and so on)	You lack that permission	Ask an administrator
Table access blocked by protected policy: <db>.<table>	The table's policy uses **Block raw reads**	Ask the table owner
Column access denied by masking policy	The column's mask is `deny`	Do not select that column
Masked columns cannot be used in WHERE clauses	A masked column cannot be filtered or sorted on	Filter on another column

You see	Cause	Fix
Access to audit.logs table is restricted to Admin group members.	Only administrators read the audit log	Ask an administrator
Audit logs can only be queried. Modifications are not permitted.	You tried to change `audit.logs`	Do not change it
The system audit database cannot be dropped.	You tried to drop `audit`	Do not drop it

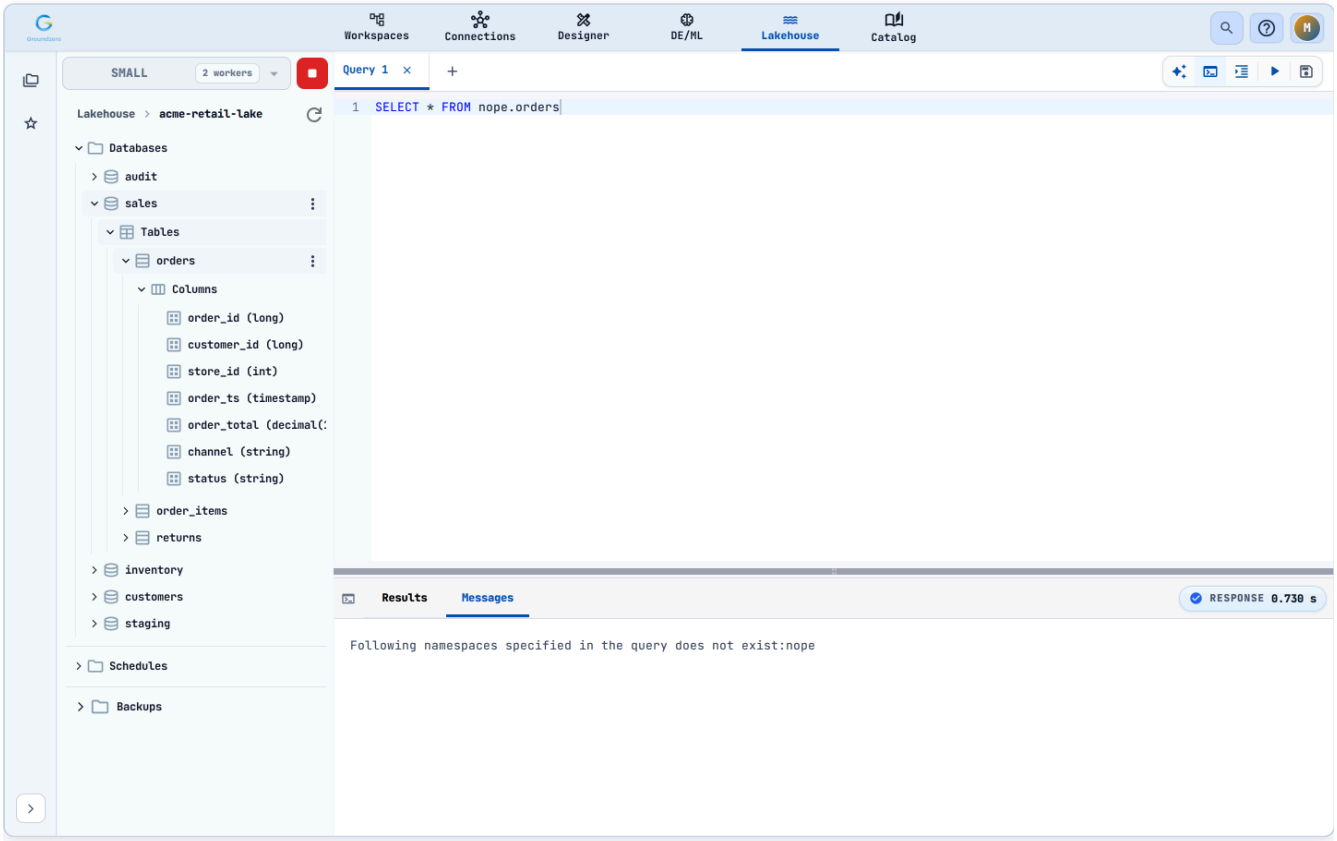


Figure 86. Referencing a database that does not exist.

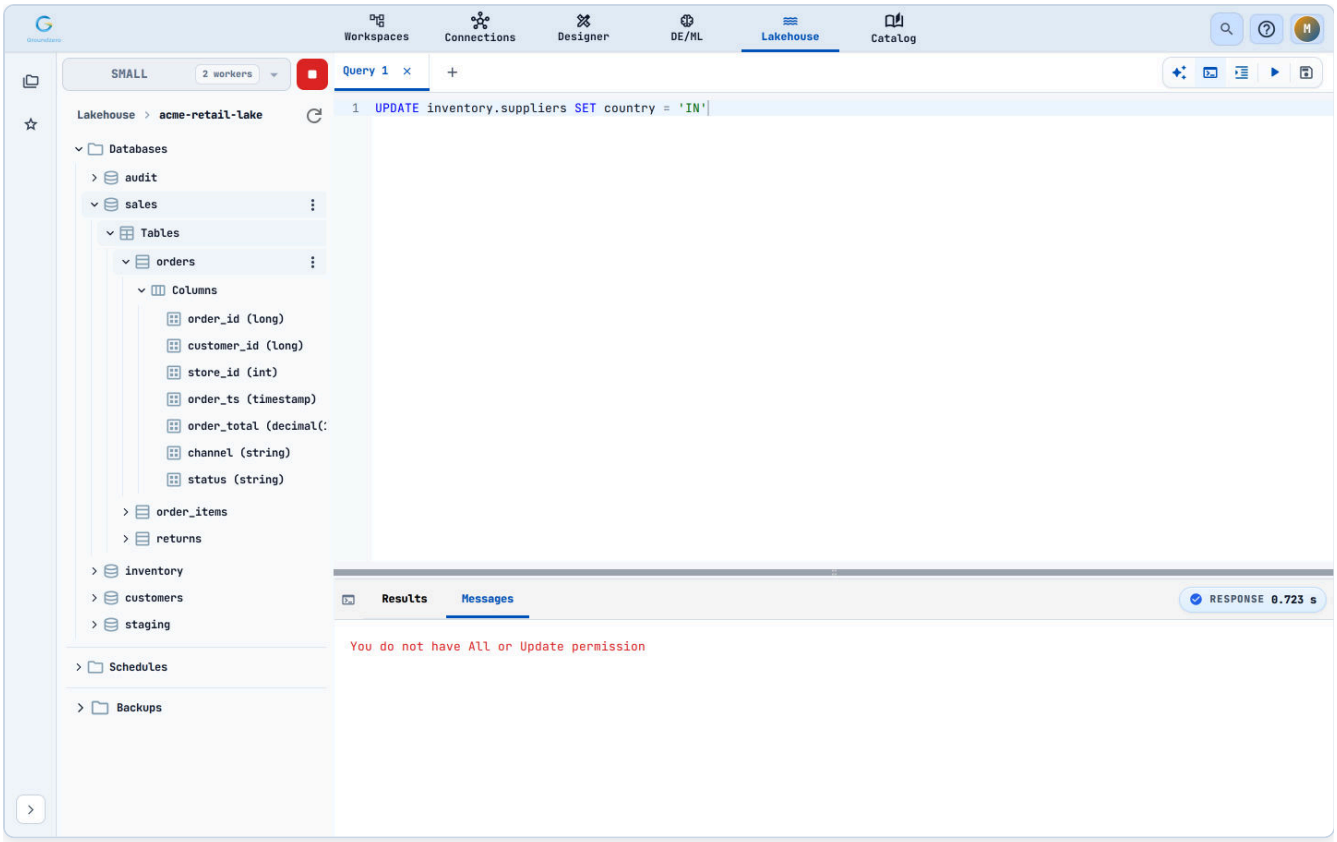


Figure 87. An UPDATE without permission is refused.

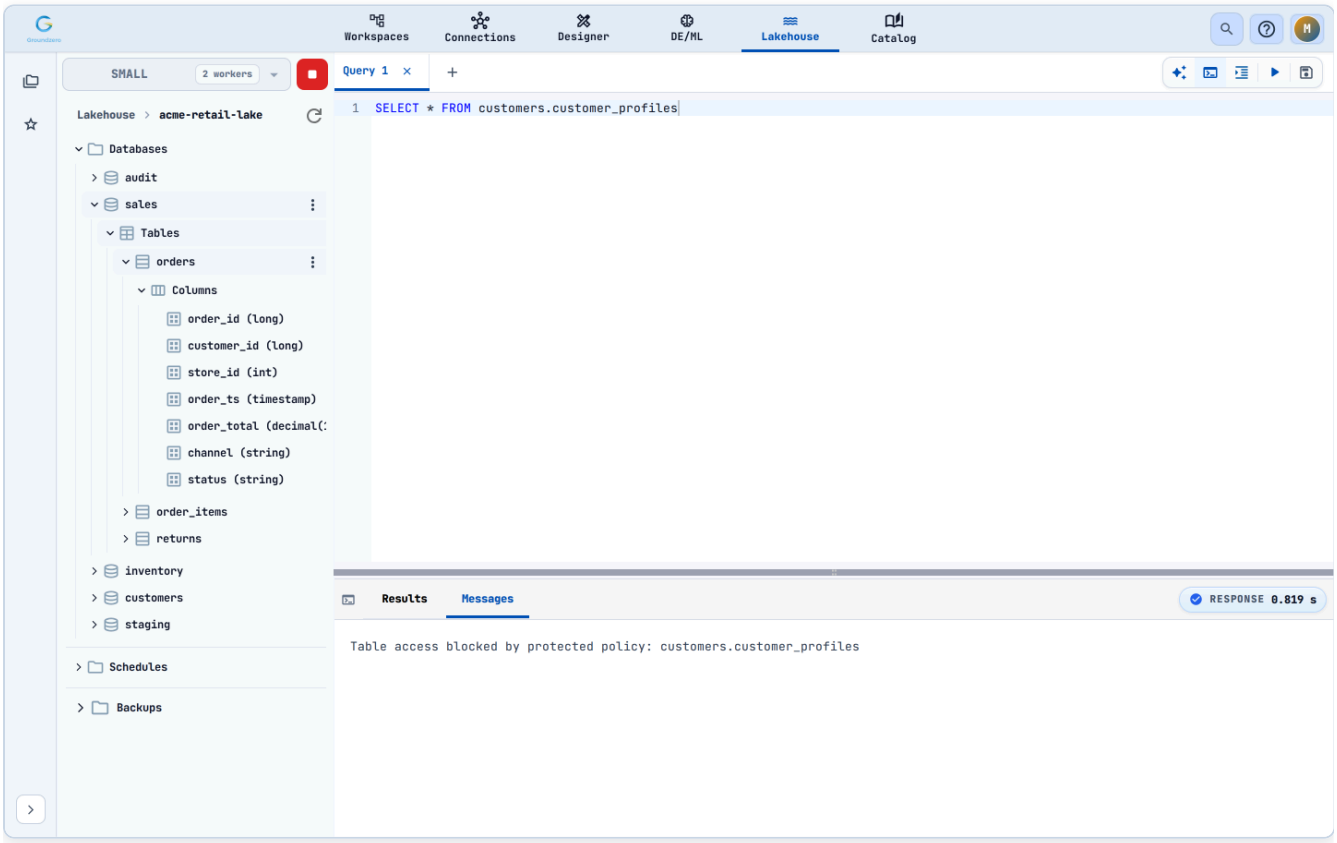


Figure 88. The error names the protected table.

Other messages in **Messages**: "An error occurred while running the query. Please try again", "Lost contact with the session. Retrying...", "The session no longer has this query. Run it again."

Permissions work at three levels: the warehouse, the database and the table. `CREATE DATABASE` needs `CreateDatabase`; `DROP DATABASE` needs `DropDatabase`; reading needs `Select`; `UPDATE`, `DELETE`, `MERGE`, `TRUNCATE` and `INSERT OVERWRITE` need `Update`; `CREATE TABLE` needs `CreateTable`. An explicit deny always wins.

5.5 Read and export a result

5.5.1 Read the grid

Each column shows its name and data type. Drag a header to reorder columns. Right-click a header for **Autosize this column**, **Autosize all columns**, **Auto-align columns** and **Reset column order**. The grid has no sort or filter; add `ORDER BY` or `WHERE` to your query. Very wide results load the first 300 columns and more as you scroll right.

5.5.2 Page through a large result

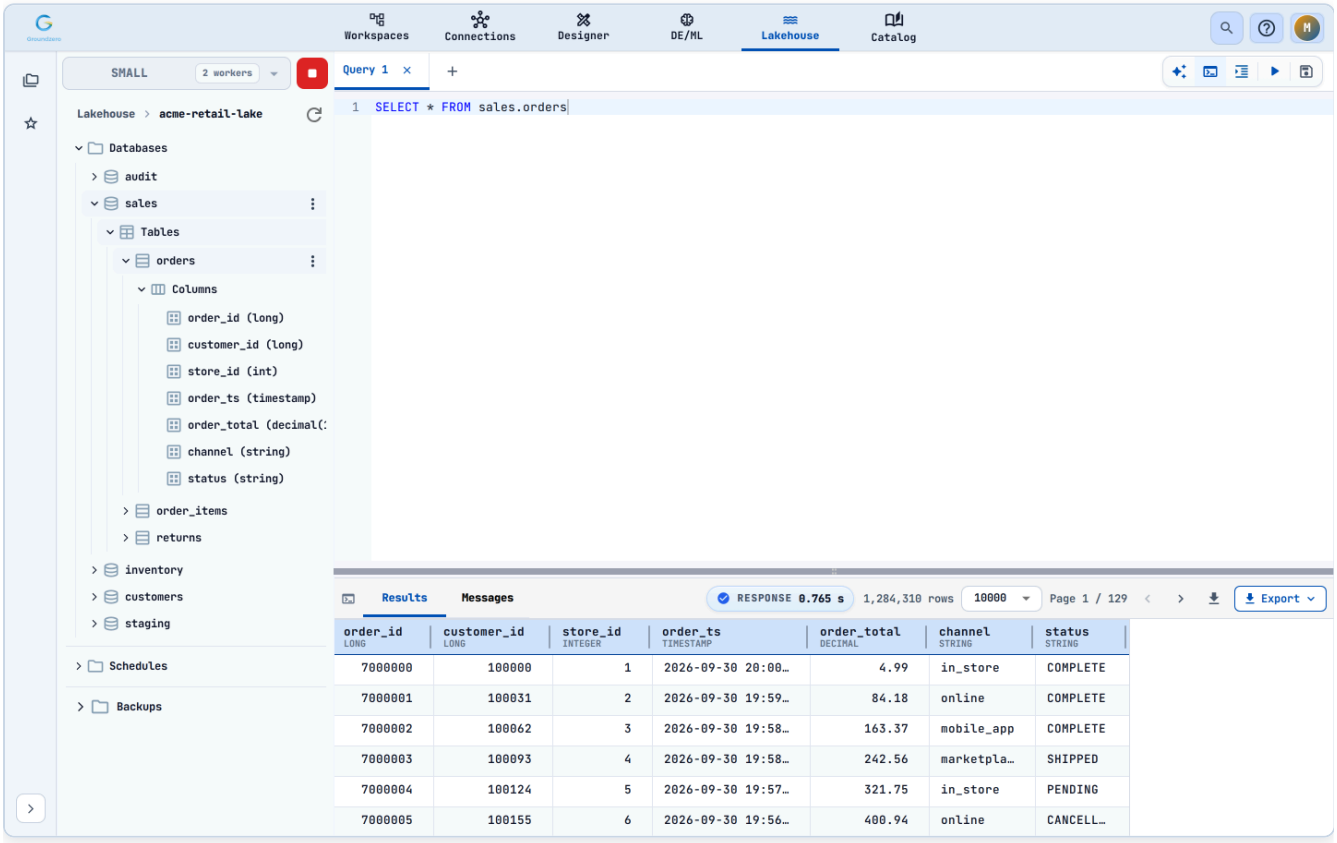


Figure 89. A large result shows the exact row count, page controls and an Export button.

Control	What it does
Row count	The total, for example "1,284,310 rows"
Rows per page	50, 100, 500, 1000 or 10000 (default 10000)
Page `n / total`, previous, next	Move between pages
Download current page as CSV	Saves the rows on this page
Export	Saves every row; appears only for results over 100,000 rows

For a very large result, paging past the first 100,000 rows first needs the full result to be prepared; you see "Preparing the full result for paging. Pages load here when it is ready."

5.5.3 Download one page as CSV

Click **Download current page as CSV**. The file is named `query_<date>.csv`. You see "CSV downloaded successfully".

5.5.4 Export a result

Outcome: one file with every row.

1. Click **Export** and choose **CSV** or **Parquet**.

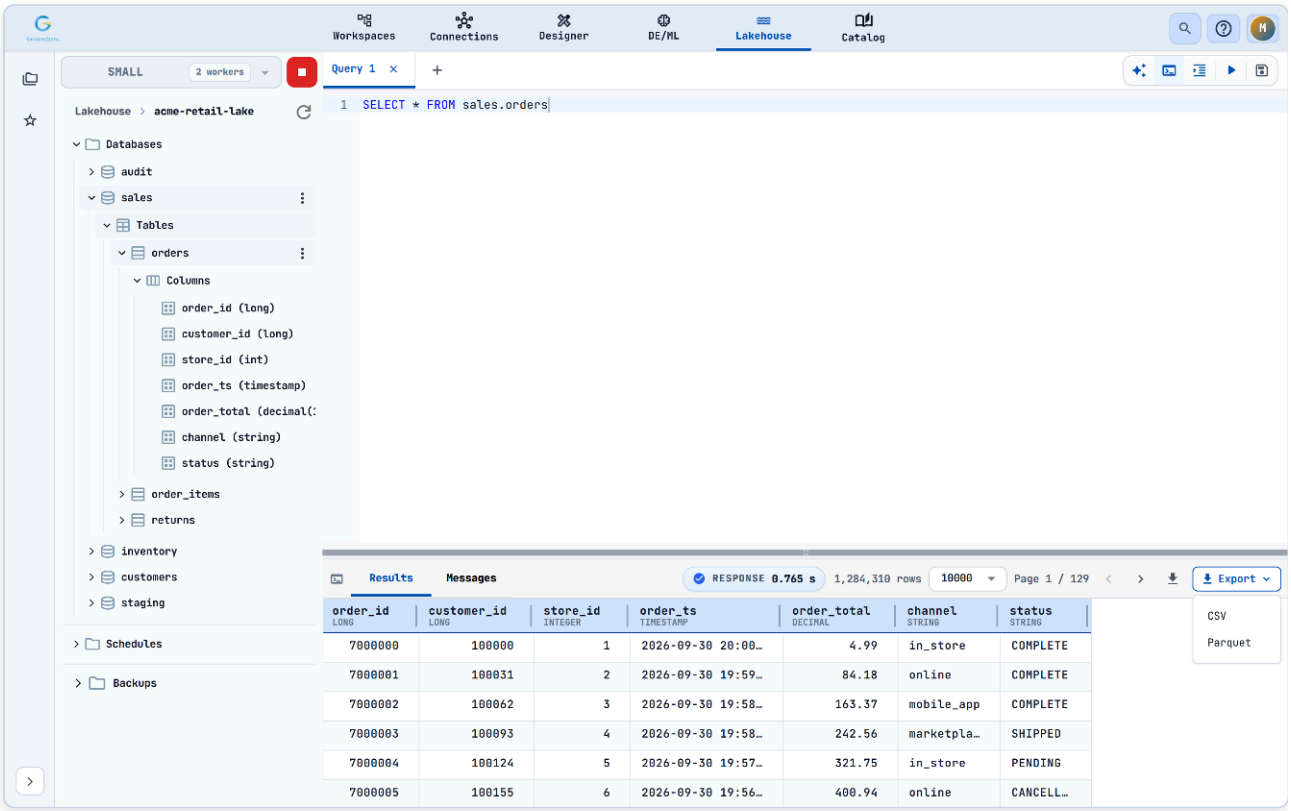


Figure 90. The Export menu offers CSV and Parquet.

2. Wait while the toolbar shows "Preparing CSV file". **Stop export** cancels.

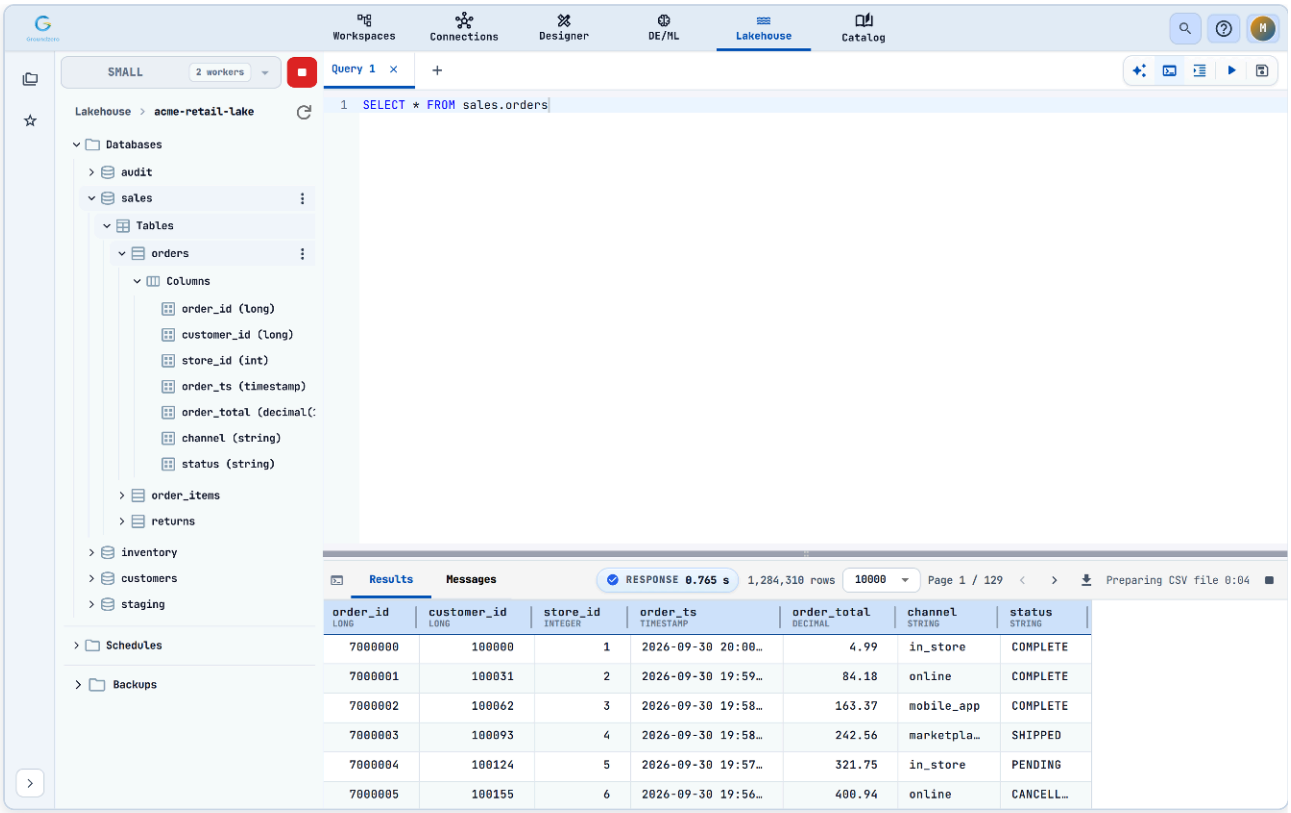


Figure 91. An export being prepared.

3. When ready, the toolbar shows a download link with the file name and size. Click it to save the file. The link works for **one hour**; export again after that.

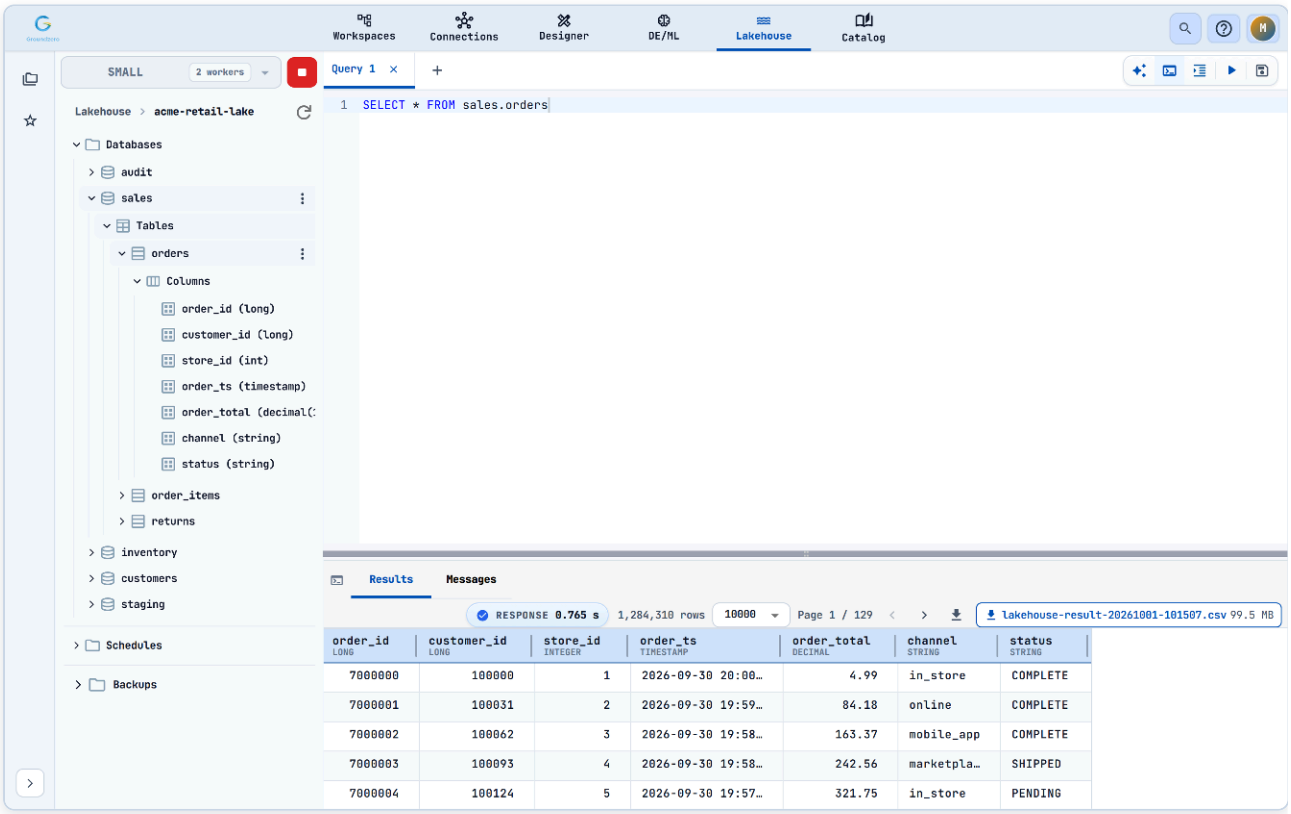


Figure 92. The finished export is a download link.

If the export fails, the reason appears in red beside **Retry export**.

You see	What to do
This export is larger than 5 GB...	Add a filter, select fewer columns, or keep the result in the lakehouse with `CREATE TABLE ... AS SELECT`
This result is too large to page through or export here (limit: 50,000,000 rows or 20 GiB)...	Add a filter, or save it as a table with `CREATE TABLE ... AS SELECT`
Preparing the full result took too long and was stopped...	Narrow the query and try again
Deep paging and export aren't available for this warehouse...	Ask an administrator to check the bucket's encryption settings
Could not get access to the warehouse bucket for this result. Try again in a moment.	Try again

Exports follow data policies: masked values and filtered rows are the same as in the grid. Every export is recorded in the audit log.

5.5.5 Limits to know

Item	Limit
Rows you can page through at once	100,000 per query (more after the full result is prepared)
Largest result you can page or export	50,000,000 rows or 20 GiB
Largest single export file	5 GB
One query	30 minutes; 4 at a time per session
Session	Starts within 3 minutes; stops after 30 idle minutes
Export link	1 hour; the exported file is deleted after 1 day

5.6 Save and reuse queries

Outcome: a named query you can open later from any session.

1. In the editor, click **Save query**.
2. In **Save As**, check the **Name** (it starts as the tab name) and save.

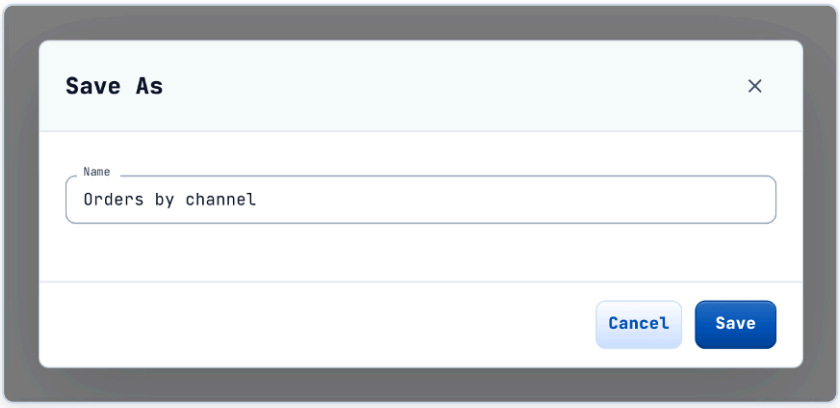


Figure 93. The Save As dialog.

You see "Query saved successfully" and the tab takes the saved name. Saving again under the same name is refused with "Key already exists."

Open **Saved Queries** in the left rail to see your queries.

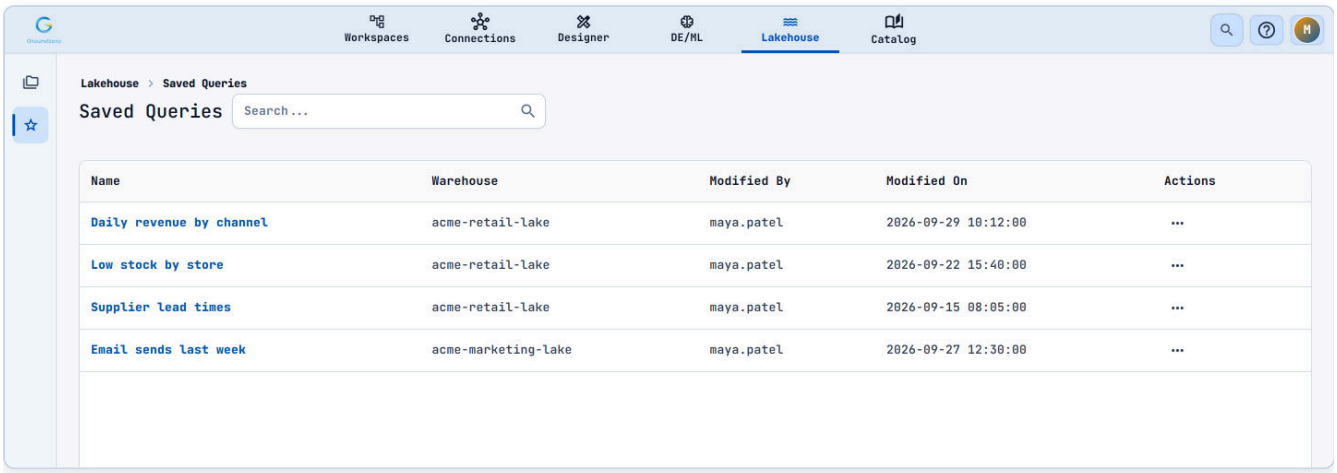


Figure 94. Saved Queries lists your queries with their warehouse.

- Click a name to open its warehouse with the SQL in a tab. Start a session before you run it.
- **Rename** asks for a **Query Name** ("Query renamed successfully").



Figure 95. Rename shows the current name.

- **Delete** asks for confirmation. The list shows only your own queries.

5.7 Look at a table's details

Table menu > **Properties** opens a **Properties** tab.

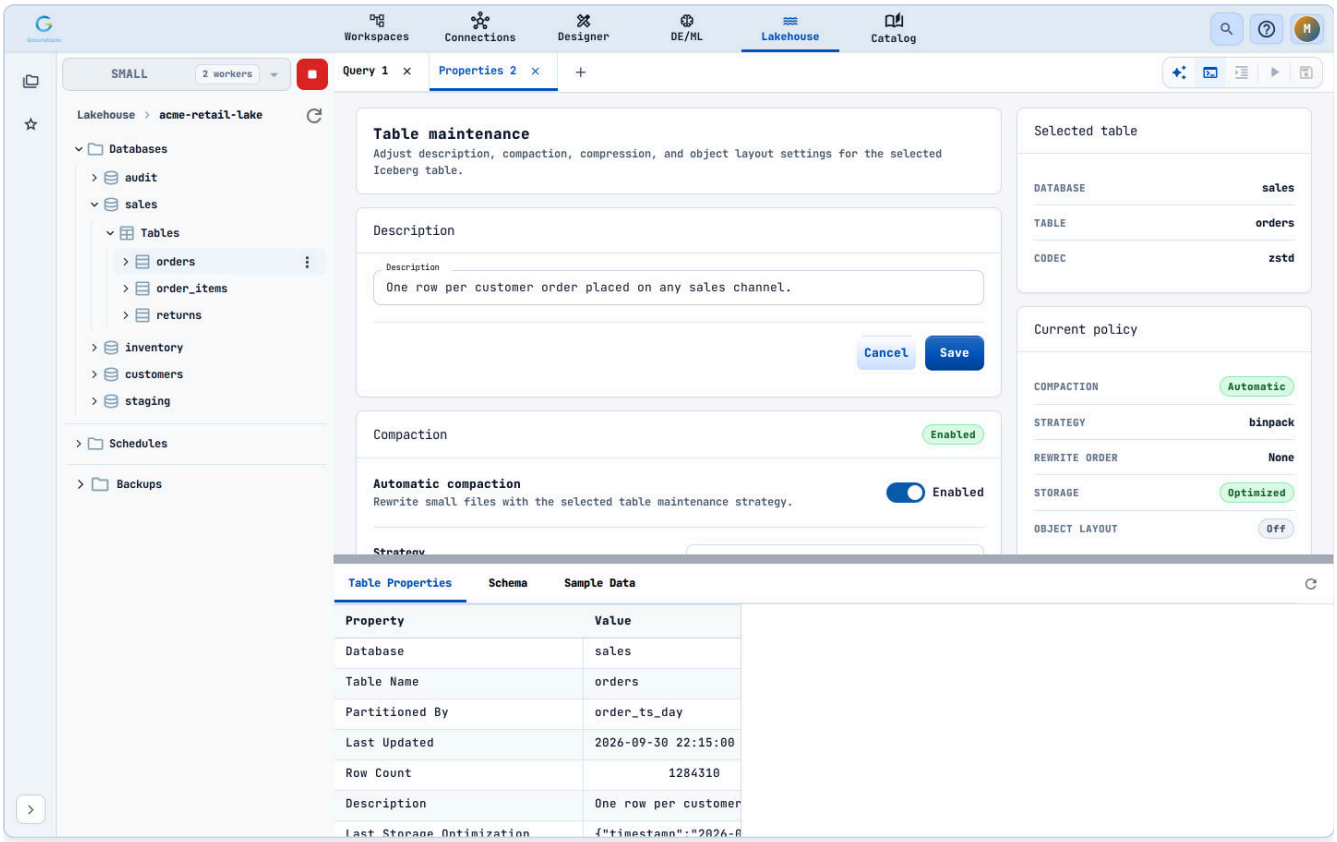


Figure 96. Table maintenance on top; Table Properties, Schema and Sample Data below.

The bottom of the tab has three parts.

- **Table Properties**: database, table, partitioning, last update, row count, size, file count, creation details, compression and more. Click the **Snapshot History** cell to read the 10 most recent snapshots. A **snapshot** is a saved version of the table at one moment.

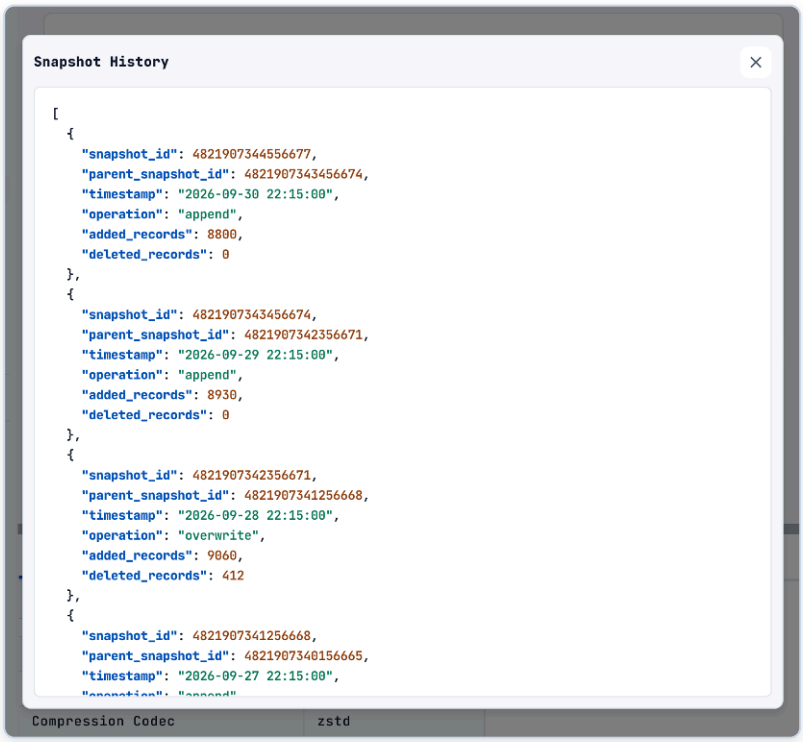


Figure 97. The Snapshot History viewer.

- **Schema**: each column with its data type and **Governance tags**. The tag dialog works as described in the Data catalog chapter.

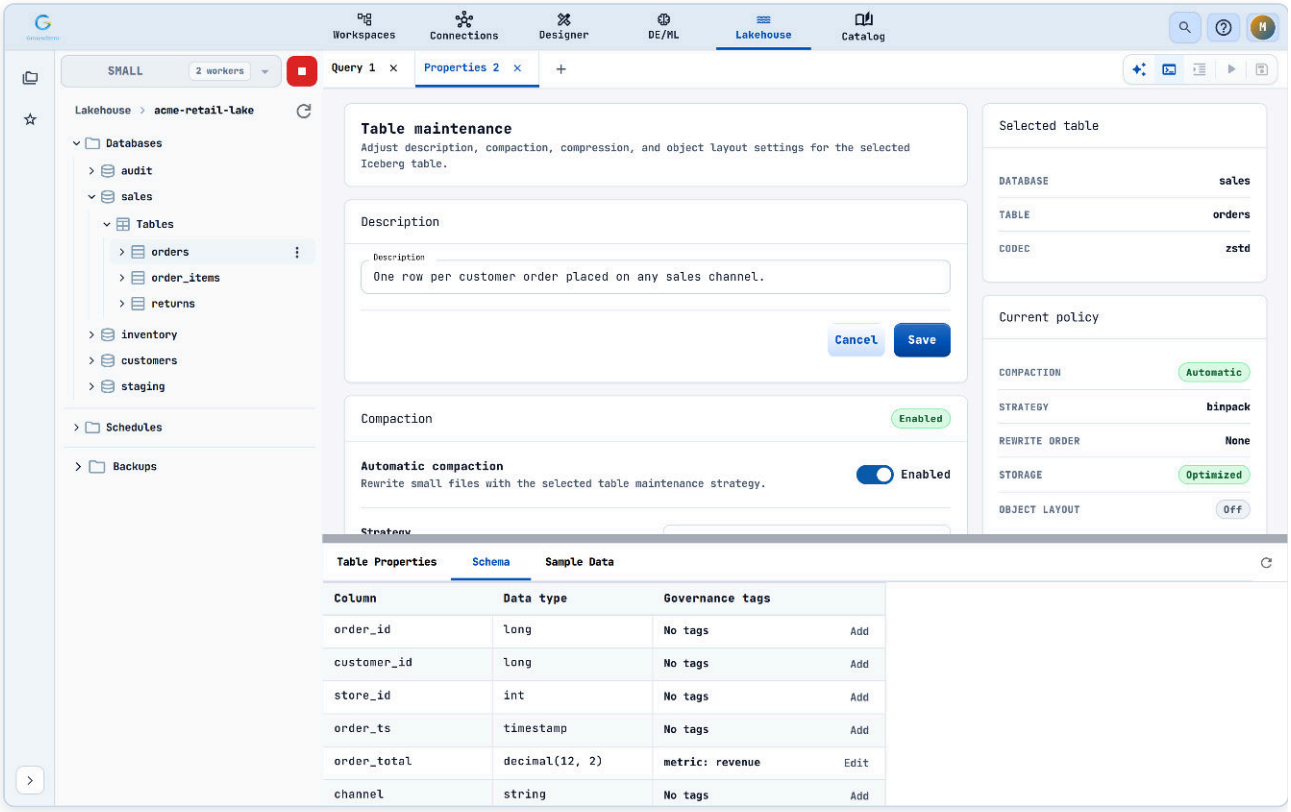


Figure 98. Governance tags per column.

- **Sample Data**: the first 10 rows. Administrators only.

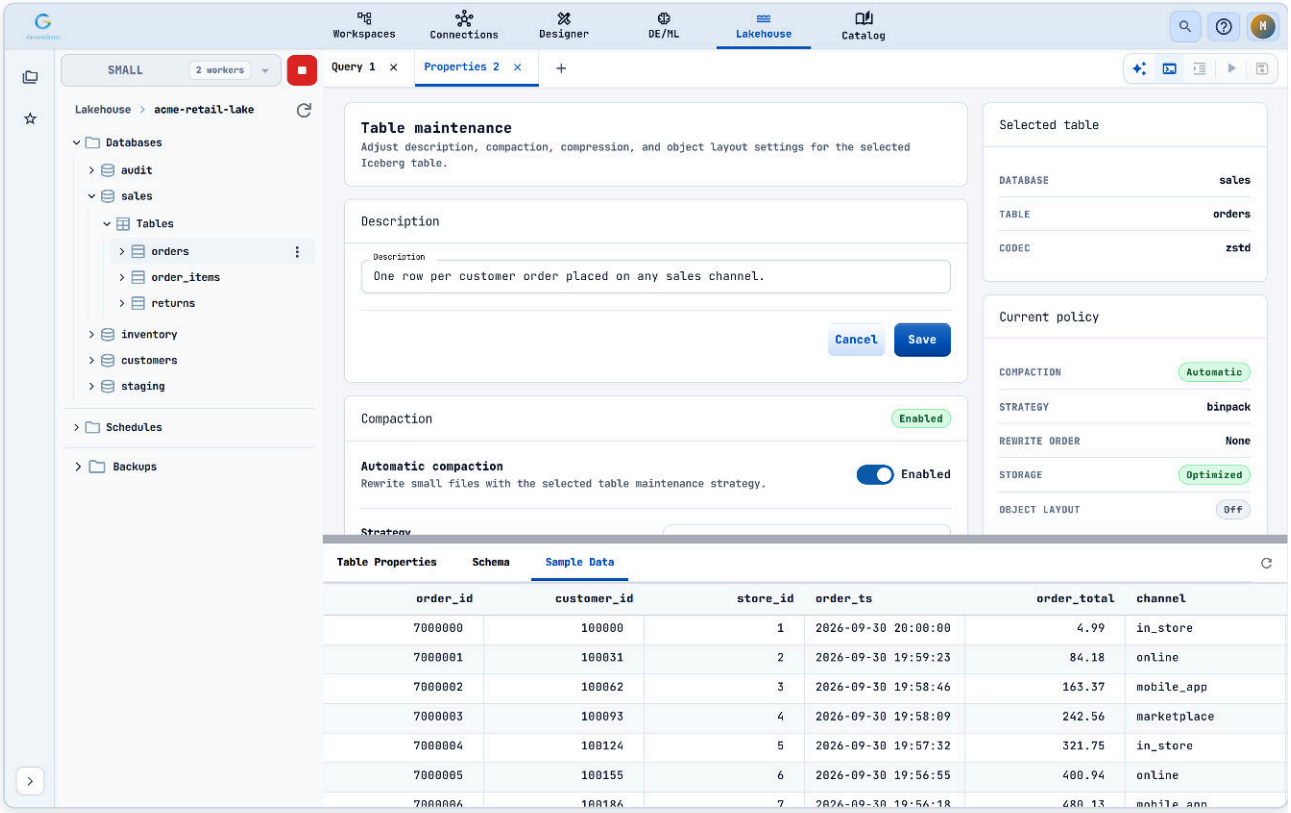


Figure 99. Sample data preview.

> **WARNING**
>
> Data policies do **not** apply to Sample Data. Only administrators can see it.

5.7.1 Keep a table fast (administrators)

The top of the **Properties** tab has maintenance panels, for administrators only.

Panel	What you can do
Description	Edit the table comment ("Description updated successfully!")
Compaction	Switch **Automatic compaction** on, or run a one-off job. Choose a **Strategy** (Binpack or Sort) and other options. You see "Compaction process initiated successfully!"
Storage	Switch **Storage optimization** on, or choose a **Compression codec** (zstd, snappy, gzip) and **Object storage layout**. You see "Storage optimization running in background..."

Compaction joins many small files into fewer large ones, which makes queries faster. The Lakehouse runtime guide explains when to use each option.

5.7.2 Look at a database

Database menu > **Properties** shows the number of tables, the total size and a **Table Sizes** list. The delete icon (**Delete database**) needs a running session, the `DropDatabase` permission and an empty database ("Delete its tables and views first."). The `audit` database cannot be deleted.

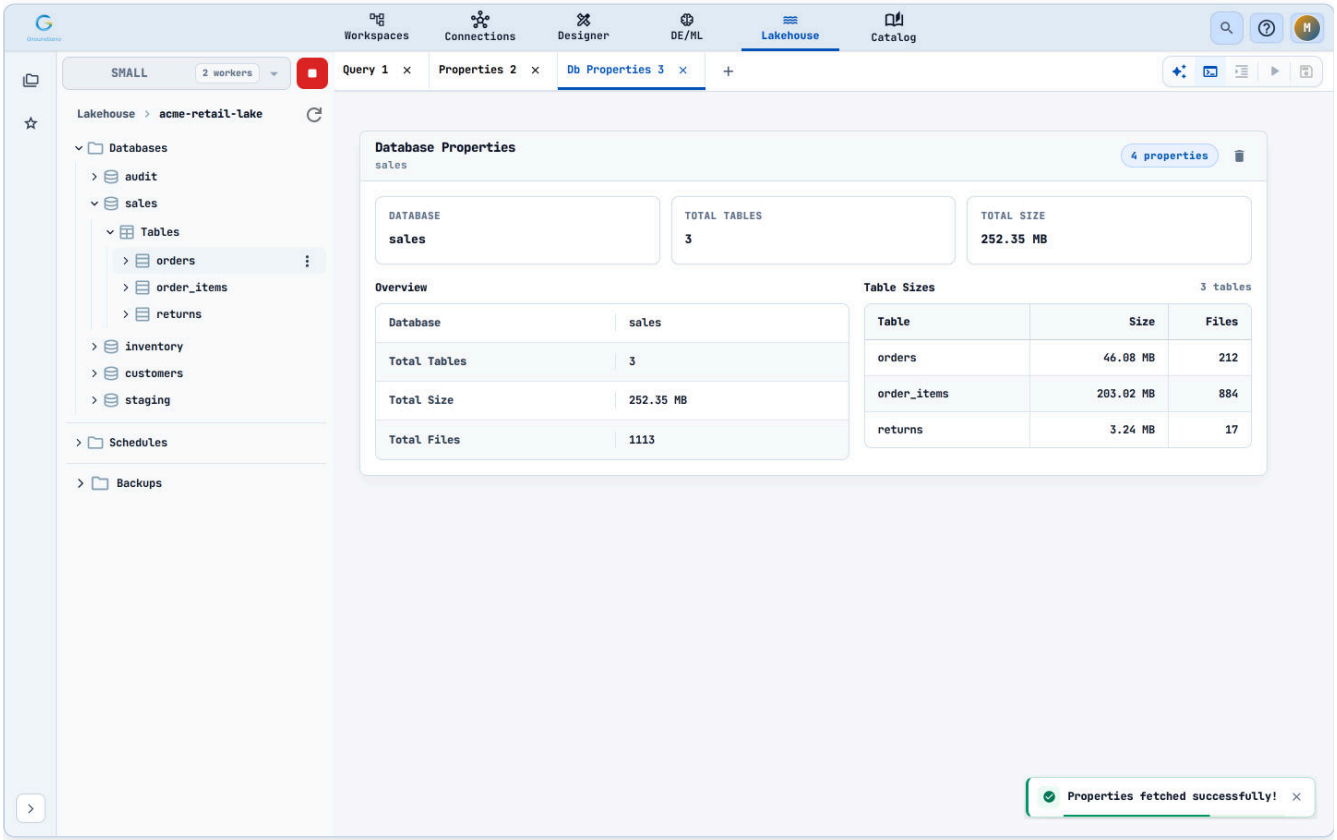


Figure 100. Database properties and the delete confirmation.

5.8 Protect a table with a data policy

Outcome: readers see only the rows and columns they should. A **data policy** is a rule set attached to one table. It filters rows and masks column values for every read: the editor, exports and saved queries.

Open the table menu > **Data policy**. You need `Select` on the database to open it and `All` to save.

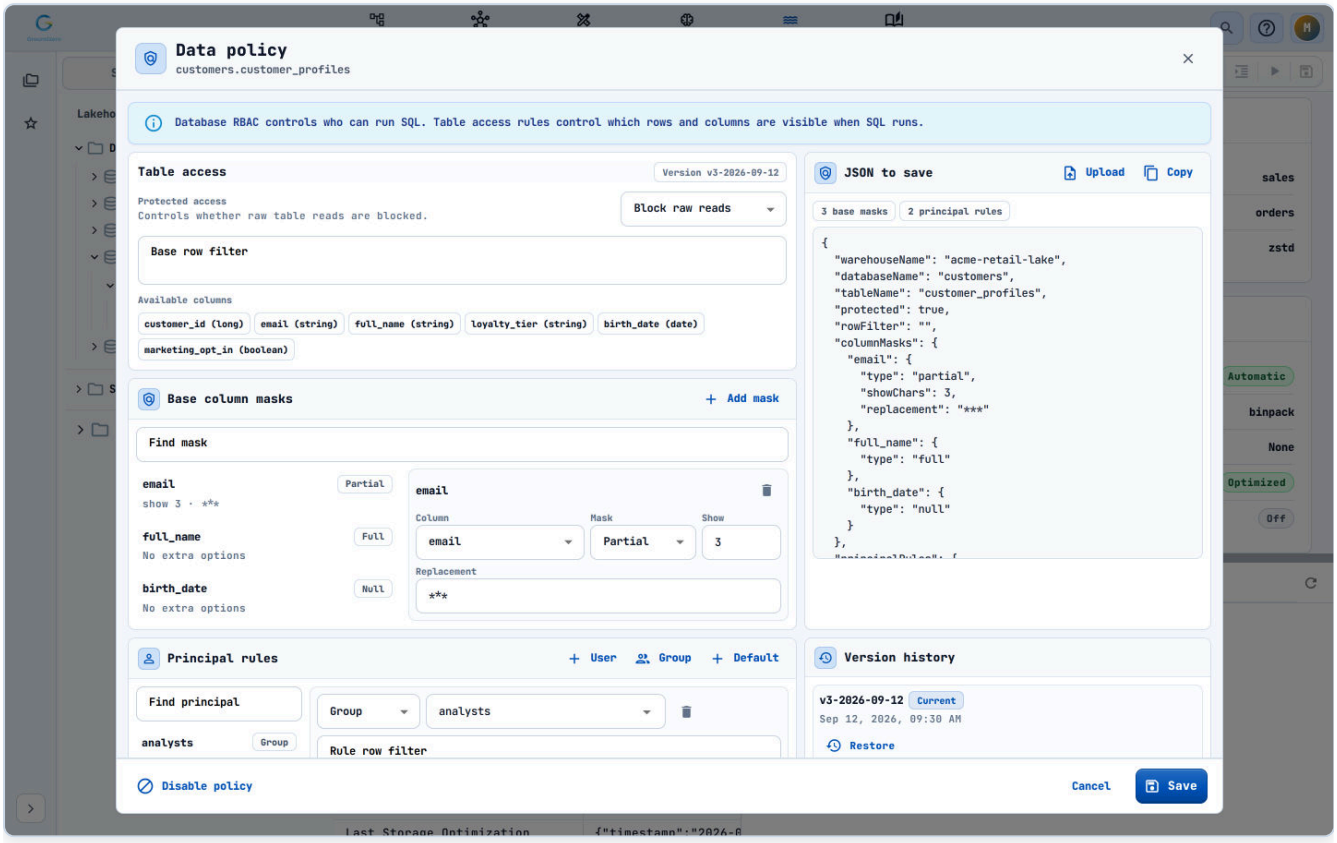


Figure 101. The Data policy dialog.

The dialog has three kinds of rule.

5.8.1 Control access to the table

- **Protected access**: **Policy only** (default) applies your filters and masks. **Block raw reads** stops any SQL read of the table; you see "Table access blocked by protected policy: <db>.<table>".
- **Base row filter**: a condition added to every read for everyone, for example `region = 'EU'`. You can use `${user}`, `${site}`, `${groups}` and `${attr.<name>}` to tailor it per reader. Click a chip under **Available columns** to add a column name. A filter that is not valid SQL is refused with "Invalid row filter SQL: <filter>".

5.8.2 Mask columns

Click **Add mask**, choose the **Column** and the **Mask** type, then set **Show** and **Replacement** if the type needs them.

Mask	What a reader sees
`full`	`****` (or your replacement) for text; empty (NULL) for other types
`partial`	The first few characters (**Show**, default 3), then the replacement (default `***`)

Mask	What a reader sees
`hash`	A one-way code in place of the value
`null`	Empty (NULL)
`deny`	The query fails with "Column access denied by masking policy"
`unmask`	Cancels a base mask for one user or group

A masked column cannot be used to filter, group, sort or join. `SELECT *` cannot be used on a table with masks when the query has several tables, joins or subqueries. If a masked column name exists in several tables you see "Ambiguous unqualified masked column '<name>'; qualify it" and need to write `table.column`.

5.8.3 Give a user or group their own rules

Click **User**, **Group** or **Default** to add a rule with its own **Rule row filter** and masks. The rule's filter is added to the base filter. Use `unmask` to let that reader see a masked column; where several masks apply, the strictest wins. A user or group rule needs a name ("User and group rules need a name before saving").

5.8.4 Save, upload and restore versions

- **Save** stores the policy ("Data policy saved"). **Disable policy** switches it off ("Data policy disabled"). **Cancel** closes the dialog.
- **JSON to save** shows the policy as text; **Upload** loads a policy from a file; **Copy** copies it.
- **Version history** lists every saved version, newest first. **Restore** brings one back ("Policy version restored").

Every save, disable and restore is a new version and is audited. Policies cover the editor, exports and saved queries, but not **Sample Data**.

> TIP
>
> Use **Block raw reads** for tables nobody should read directly. Use masks and row filters when different people should see different parts.

5.9 Allow or deny a table to users and groups

Outcome: only chosen people can run chosen actions on one table. Open the table menu > **Table Access Manager**. You need `All` on the database.

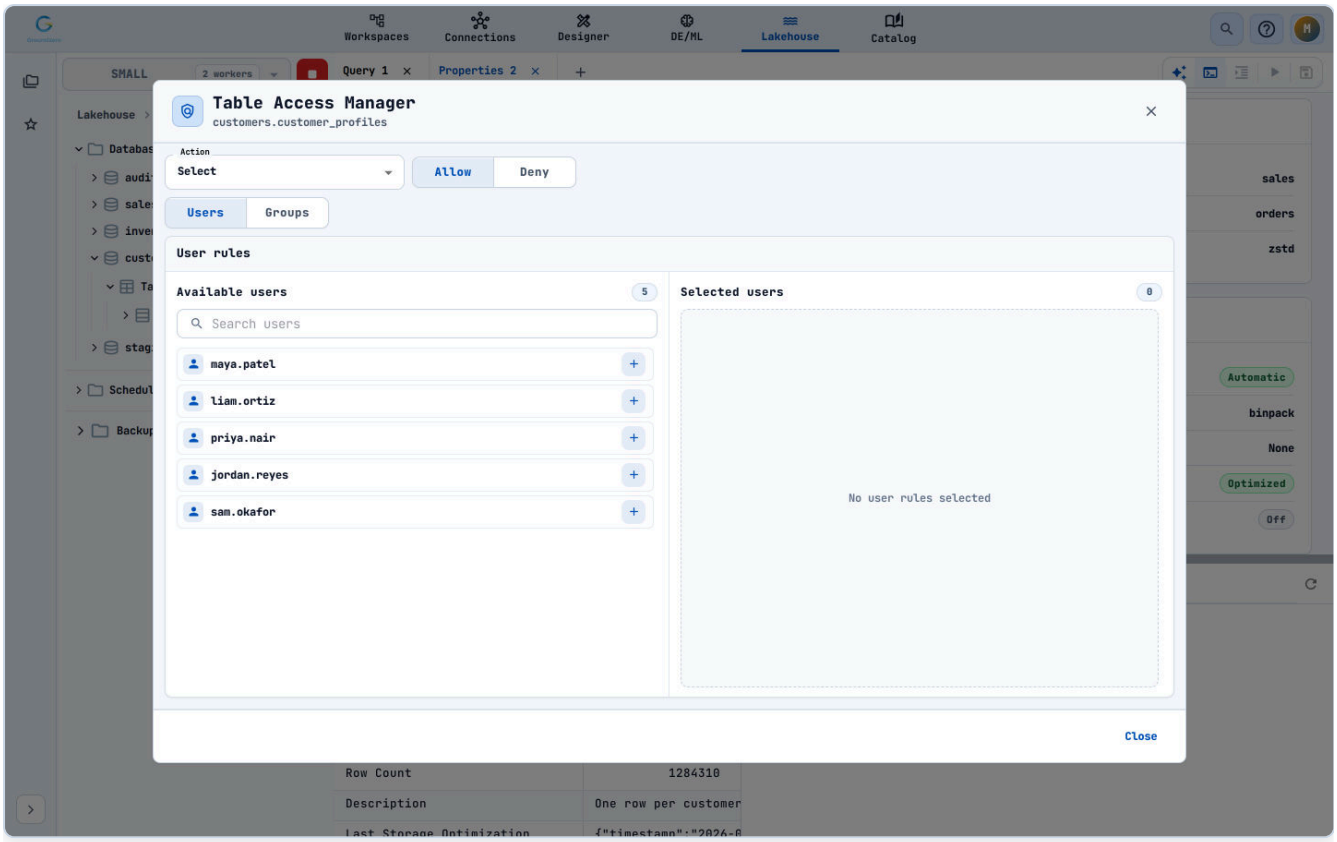


Figure 102. Table Access Manager: pick an action, an effect and users or groups.

1. Choose an **Action**: Select, Insert, Update, Alter, Drop or All.
2. Choose **Allow** or **Deny**.
3. Choose **Users** or **Groups**, then move names from **Available** to **Selected**.

Rules inherited from the database are shown, and a deny on the table can override them. **Every add or remove saves at once** ("Table access rule added", "Table access rule removed"). A deny always wins. Table rules have no version history.

5.10 Review the audit log

Outcome: see who ran which statement, when, and whether it worked. Administrators only; others see "Unauthorized access".

1. Expand `audit` > **Tables** > `logs`, then choose **Open audit logs**.

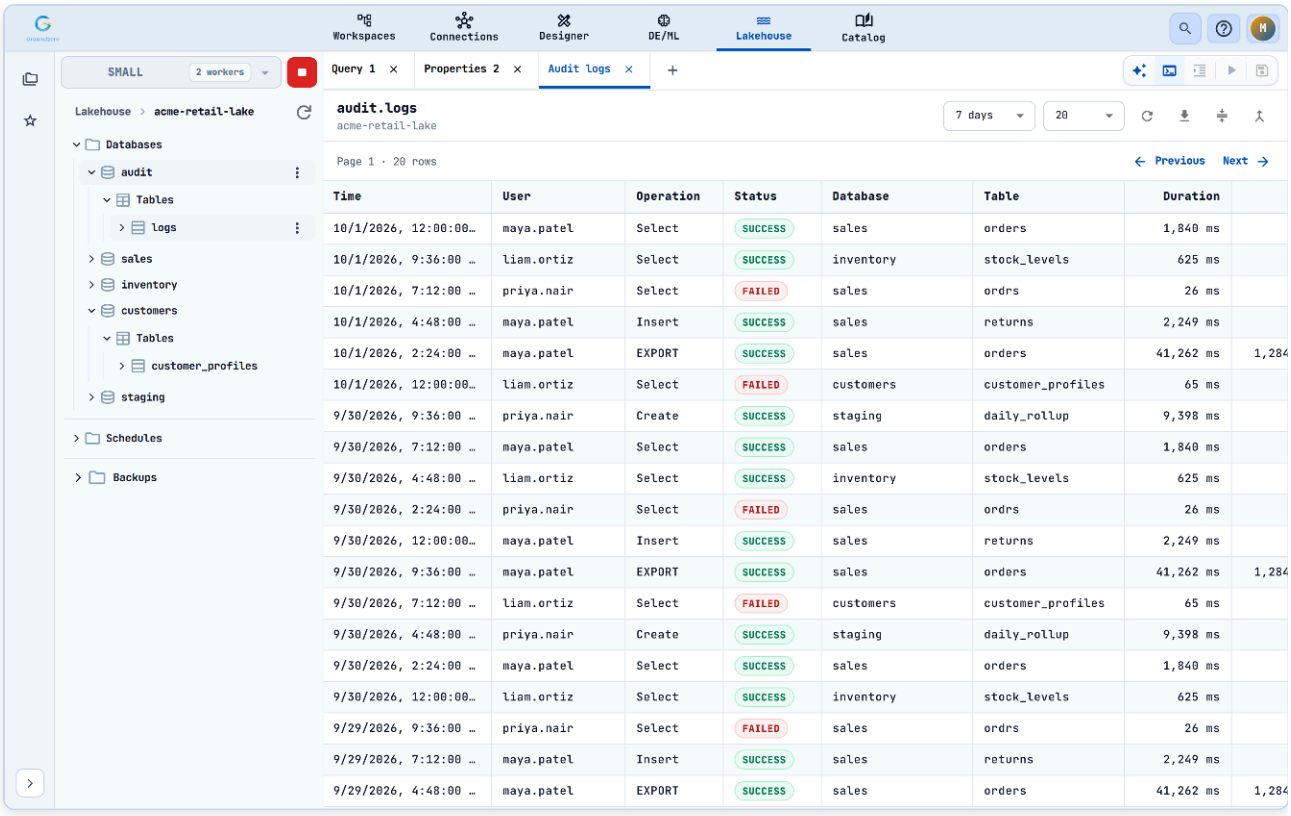


Figure 103. The audit log: time, user, operation, status, database, table and duration for every statement.

2. Choose how far back to look in **Audit log range** (7 days by default, 30, 90 or All) and how many rows per page in **Audit page size**.
3. Click any cell to open a details drawer with the full statement, the error if it failed, and timings.

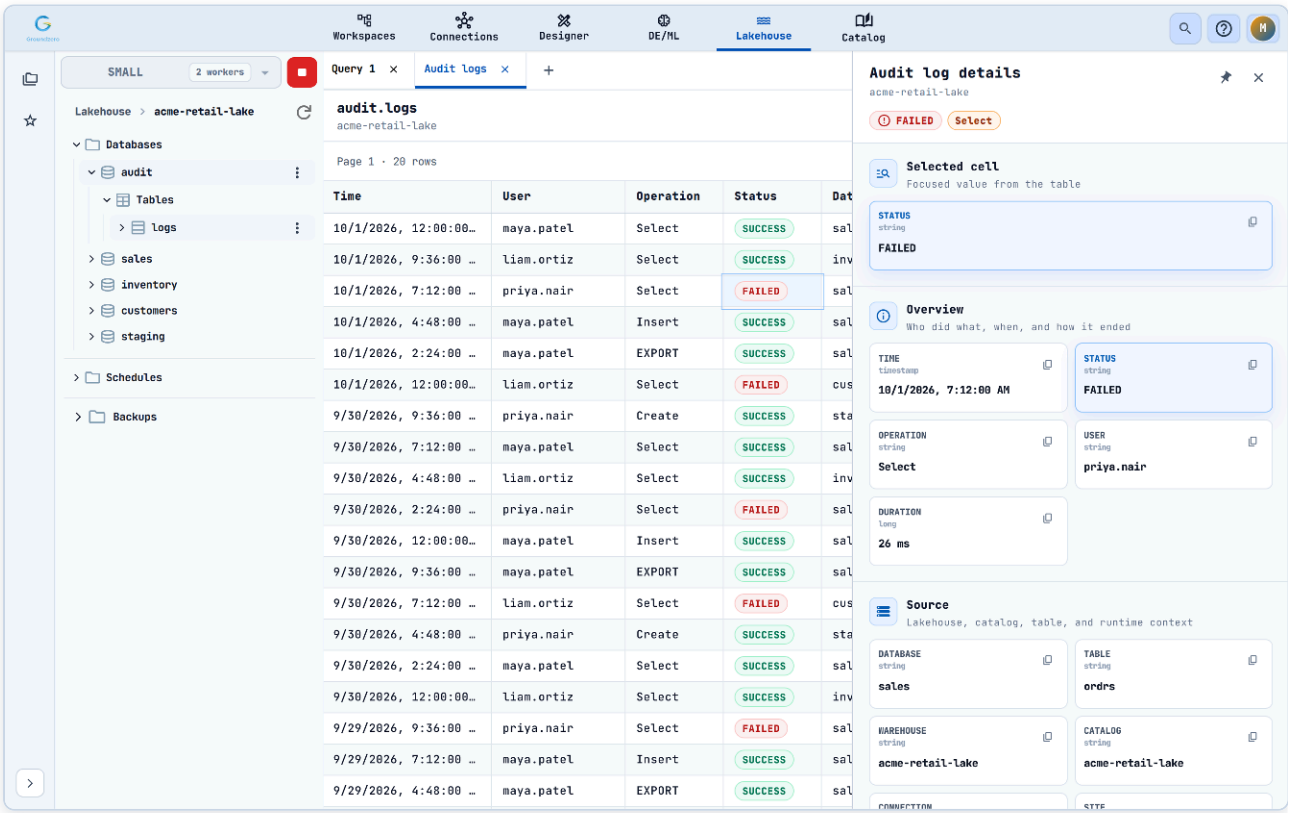


Figure 104. The details drawer for a failed statement.

Other buttons: **Refresh audit logs**, **Download current audit logs page as CSV**, **Compact audit logs** and **Rewrite audit files** (housekeeping for the log itself).

The log records every statement run in a session, failures with status FAILED and the error, and every export. New rows can take about half a minute to appear; refresh later if one is missing.

5.11 Back up and restore a database

5.11.1 Back up a database

1. Database menu > **Backup Database**.
2. Type a **Backup Name**: 3 to 50 letters, digits or underscores, starting with a letter.

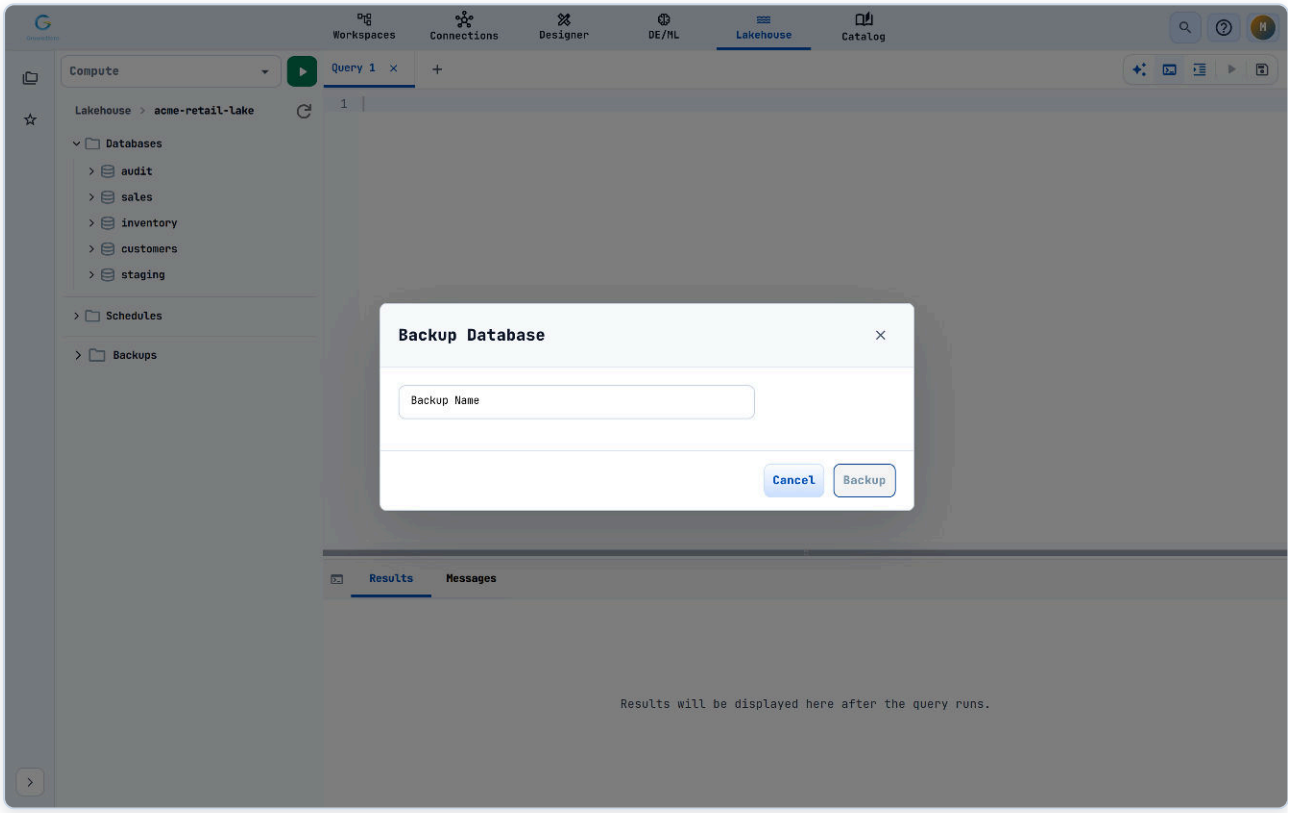


Figure 105. The Backup Database dialog.

3. Confirm. "Database backup created successfully!" means the backup has *started*. It continues in the background and shows **Running** until it finishes. If the name is rejected you see only "Database backup creation failed!"; check the name rule.

Expand **Backups** to see each database's backups with a status. A backup's menu offers **Properties** (name, database, size, files, creation time) and **Delete**.

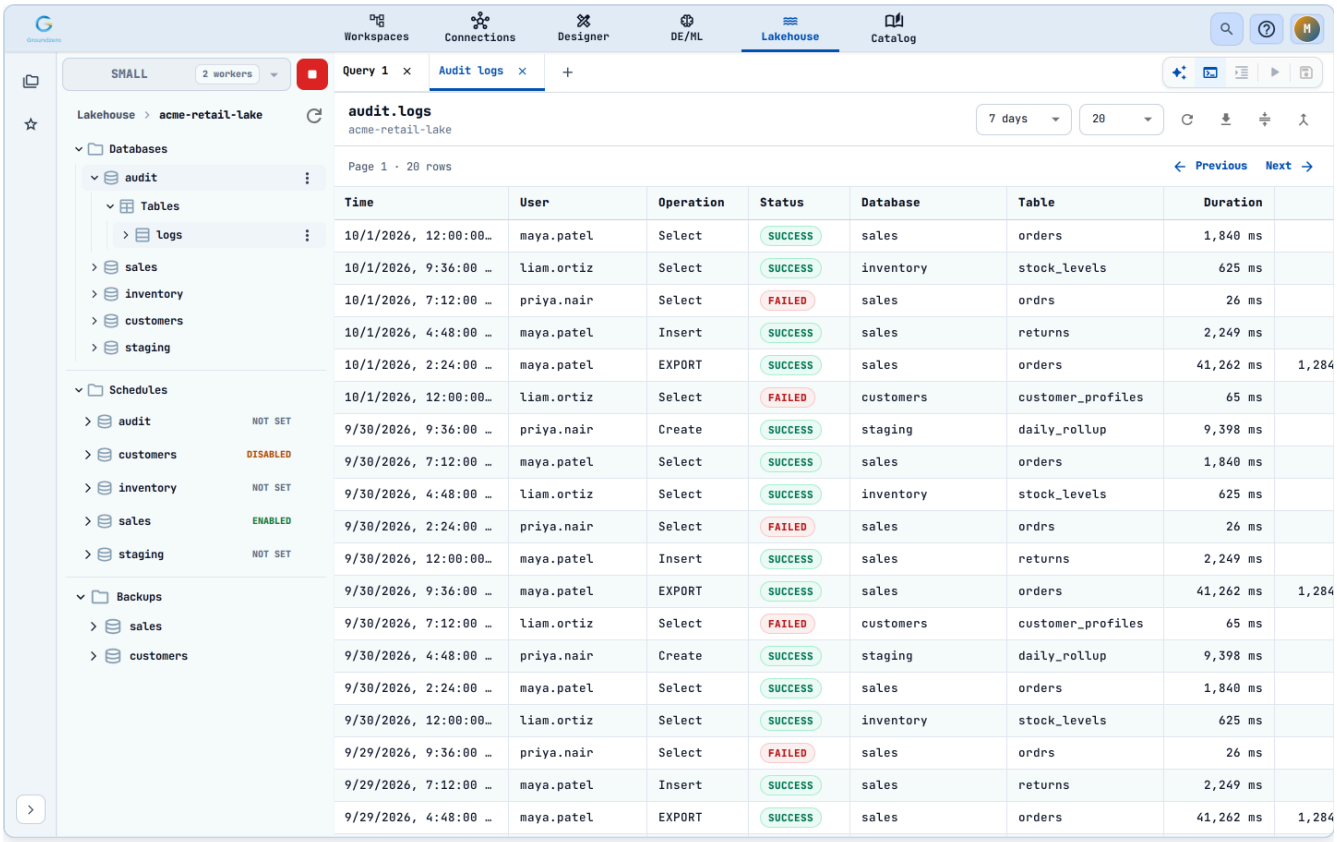


Figure 106. The Backups section lists databases and their backups.

5.11.2 Back up on a schedule

Expand **Schedules** to see each database as ENABLED, DISABLED or NOT SET. A database's menu offers **Schedule Backup** (or **Edit Schedule**), **Pause Schedule** or **Resume Schedule**, and **Delete Schedule**.

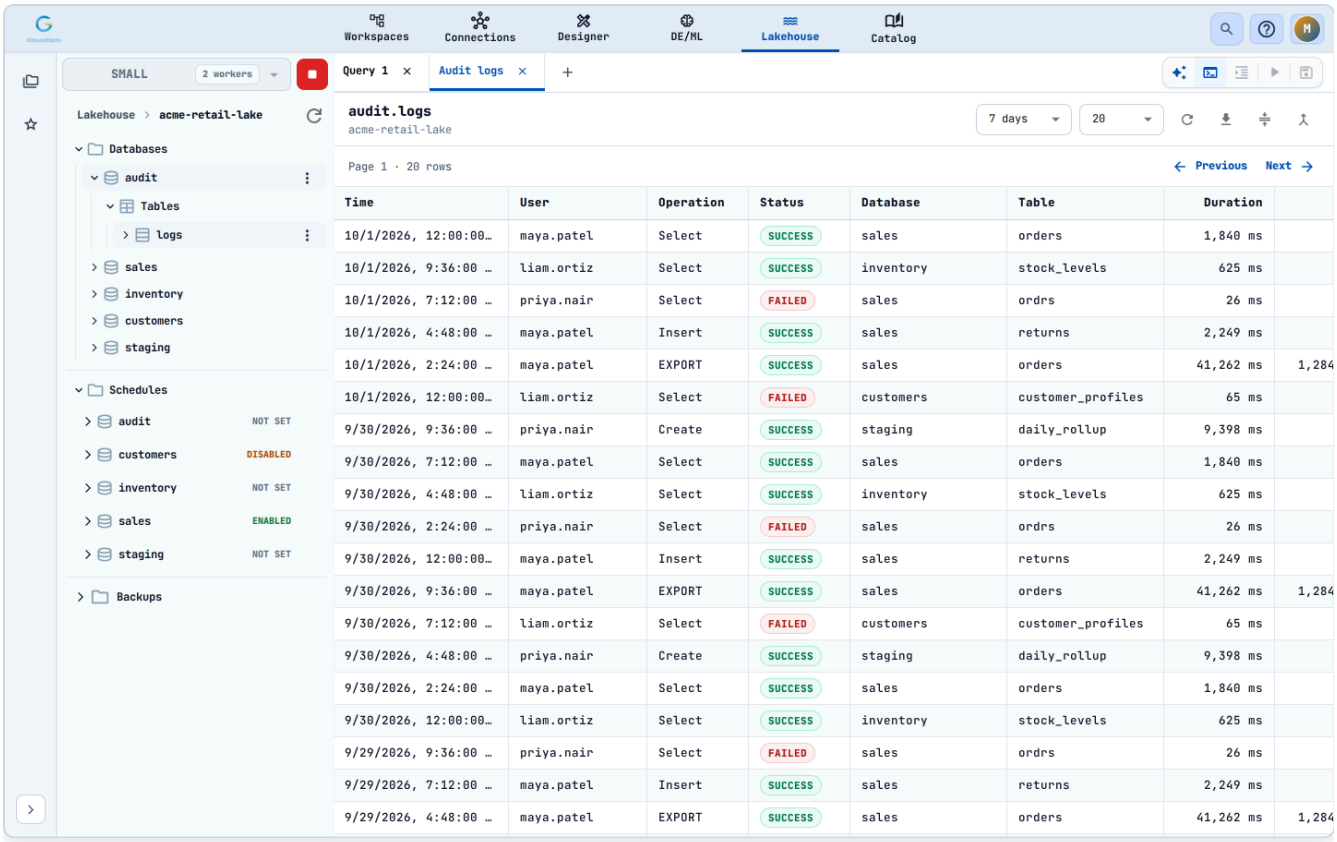


Figure 107. Schedule status per database.

The form asks for a **Backup Name**, **Retention (days)** (1 to 30), **Start time**, **Start date** and **Repeat** (Hourly, Daily, Weekly with **Days**, Monthly with **Dates**, or Yearly). Times follow your browser's time zone. Saving a schedule for a database that already has one replaces it. Deleting a schedule keeps the backups already made.

5.11.3 Restore a database

Outcome: a new database made from a backup.

1. Open the database menu (or the **Databases** menu) > **Restore Database**.
2. Choose **Select Backup**, then type a **Target Database Name** (letters, digits and underscores).

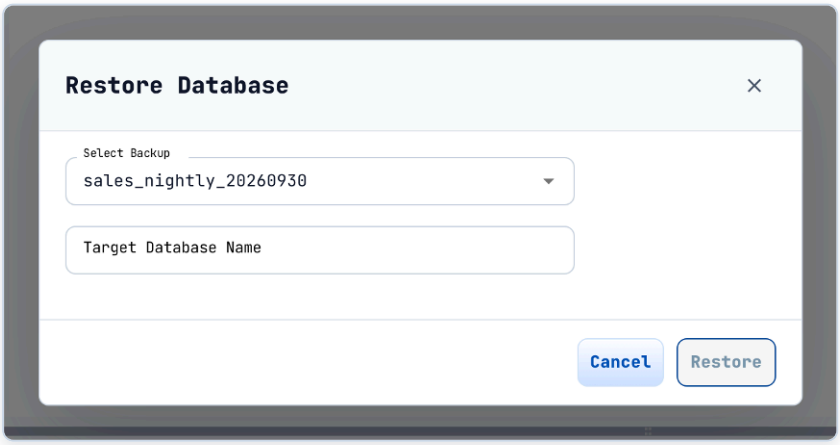


Figure 108. Restore creates a new target database from a backup.

3. Confirm. You see "Database restoration started!". Refresh the tree when it finishes to see the new database.

If a backup stays **Running** for hours, use **Mark stale backups failed** on the **Databases** menu. It marks backups that have run for over 3 hours as failed.

5.12 Tips

> **TIP**
>
> Start a session before you need it and choose enough **Minimum Workers** for big scans. Stop it when you are done.

> **TIP**
>
> For a result bigger than 100,000 rows, filter it or keep it in the lakehouse with `CREATE TABLE ... AS SELECT` instead of exporting it.

> **TIP**
>
> Always write `database.table`, and use `/* */` for comments.

5.13 Troubleshooting

Symptom	Cause	Fix
Run query is greyed out	No session is running, or no query tab is open	Choose a compute and start a session
"please select Compute"	You clicked Start without a compute	Choose one first
"The lakehouse session is not reachable..."	The session stopped (idle or stopped by you)	Start it again
"Session start failed (200)"	The session was not ready in 3 minutes	Retry, with a larger compute or more workers
Nothing happens after a `--` comment	Comments hide the rest of the query	Use `/* */`
No **Export** button	The result is under 100,000 rows	Use **Download current page as CSV**
An export link no longer works	Links last one hour	Export again
A new warehouse is missing from the list	The name is a duplicate, or you lack `Create`	Check both
"Unauthorized access" on the audit log	Administrators only	Ask an administrator
A new statement is missing from the audit log	The log updates every half minute or so	Refresh later

5.14 Related

- **Lakehouse runtime guide**: SQL reference, tables, time travel, maintenance and worked examples.
- **Data source connections**: the Groundzero Lakehouse connection used by datasets and ETL jobs.
- **Data catalog**: warehouses, databases, tables and columns with tags and descriptions.
- **Data lineage**: shows lakehouse tables as assets.

6

Data lineage

Lineage is the recorded journey of your data: which jobs and datasets write a table, and which jobs, datasets and dashboards read it. Two words you will see throughout:

- **Upstream** means before: the sources and jobs that feed a table.
- **Downstream** means after: everything that uses the table.

Pick a table and Groundzero draws a map of its upstream and downstream, down to individual columns.

Open **Catalog** in the top navigation bar, then **Data Lineage** in the module menu.

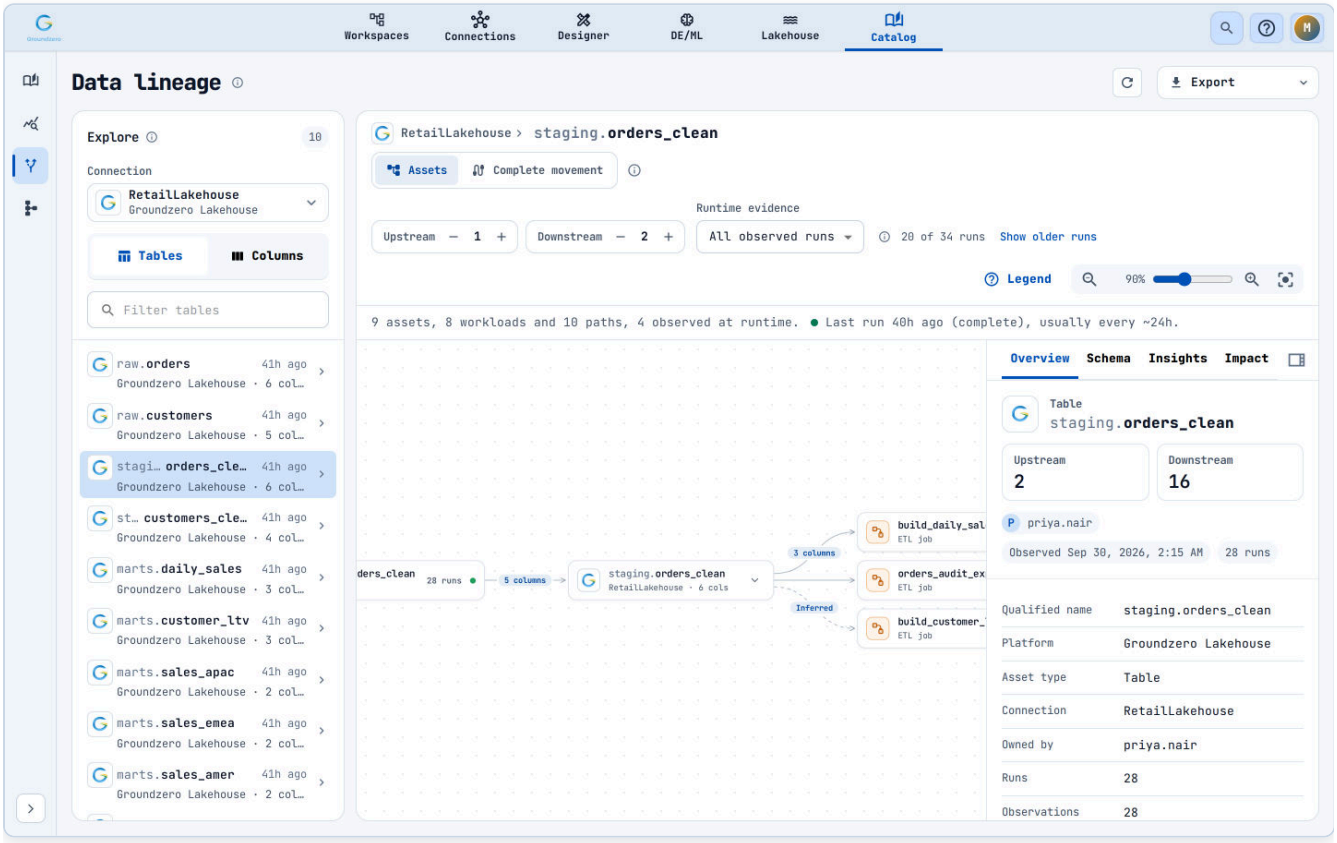


Figure 109. A map centred on staging.orders_clean with its summary line, details panel and the explorer on the left.

I want to…	Go to
Find a table and open its map	"Open the map for a table"
See what breaks if I change or drop a table	"See what breaks if a table changes"
Find where a wrong number came from	"Trace a value back to its source"
See which source columns feed a column	"Follow a column"
Check if a path was really used	"How sure is a path"
Keep a record for a change ticket	"Export lineage"

6.1 How lineage gets recorded

You do not enter lineage by hand. It appears as your jobs and datasets run.

- **From the code.** When a job or dataset runs, Groundzero reads its code (Python and SQL) to learn which tables and columns it reads and writes. If this analysis fails, the run is not affected; the lineage page shows "Lineage compilation failed".
- **From the runs.** While a run executes, Groundzero notes the tables it actually reads and writes. A finished run can take a short while to show up.
- **When you open the map.** Dashboards, owners and impact are worked out at that moment from what has been recorded.

A job or dataset that has never run has no lineage. If a map is empty, run the job once.

6.2 How sure is a path

Every line between two cards is labelled by its evidence:

- A **solid line** is **Runtime observed**: it happened in a real run.
- A **dashed line** is **Derived from code**: the code says it should happen, but no run has confirmed it yet.

Run the job to turn a dashed line solid.

6.3 Permissions

To do this	You need
Browse a connection's tables	`Describe` (or `All`) on the connection
See a table's map	Describe access to at least one of the table's connections. Anything else you may not describe appears as a grey placeholder.
See the lineage of a job or dataset	`All` or `Describe` on it
Use the Insights tab	Permission to use AI Chat
See the "catching up" banner	`All` on ETL jobs

6.4 Open the map for a table

Outcome: a map centred on the table you care about.

The left panel is titled **Explore**. A pill shows how many assets match.

1. Open the **Connection** picker ("Choose a connection") and choose the connection that holds your table. Type in **Search connections** to narrow the list. **Clear connection** removes the choice.

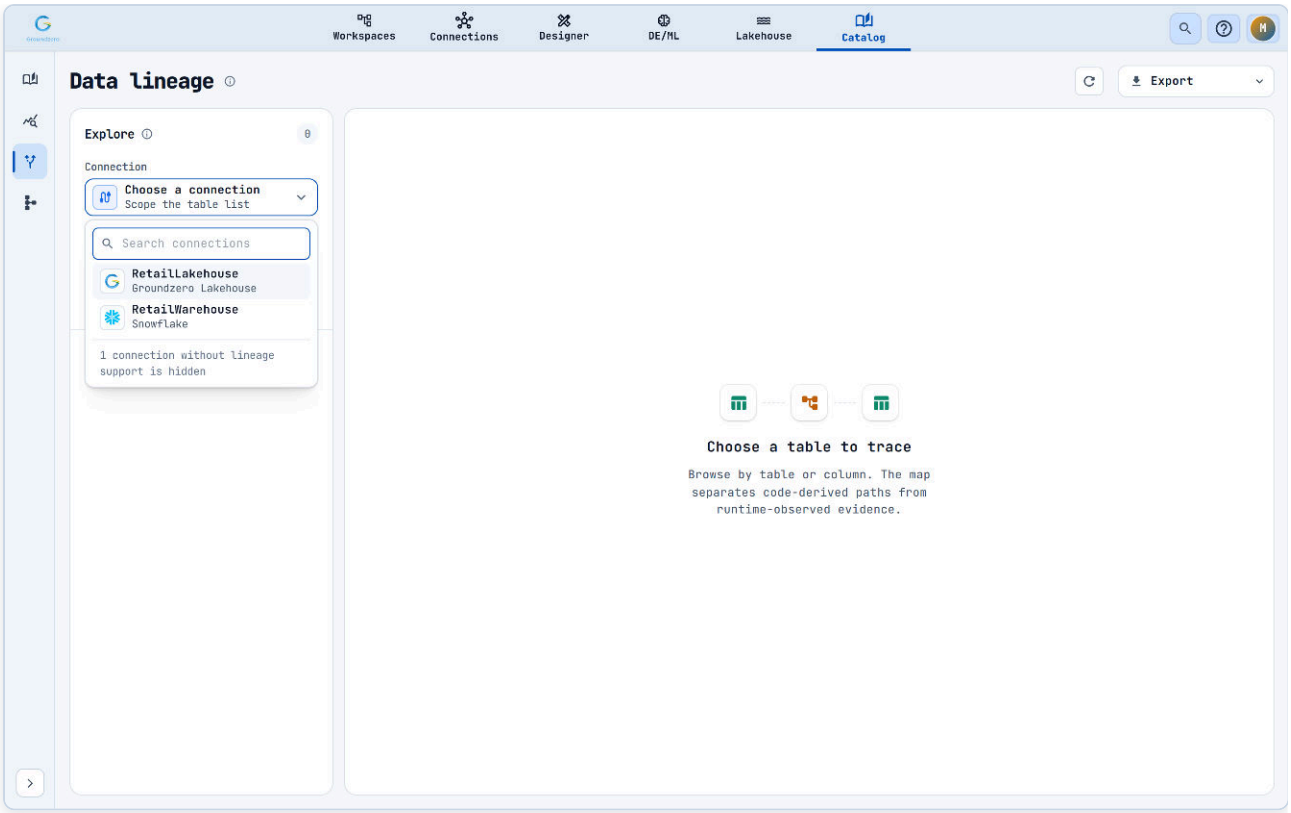


Figure 110. The connection picker lists connections that support lineage.

Only connections that can have lineage are listed (databases, warehouses, streaming and the Groundzero Lakehouse). Others, such as AI providers and REST APIs, are hidden; the footer says how many.

2. Find the table. The **Tables** tab lists tables with the time each was last observed. Type in **Filter tables** to search by name. When the connection holds several kinds of object, **Asset type** chips (for example Table and View) filter them. Click **Load more** after 50 rows.

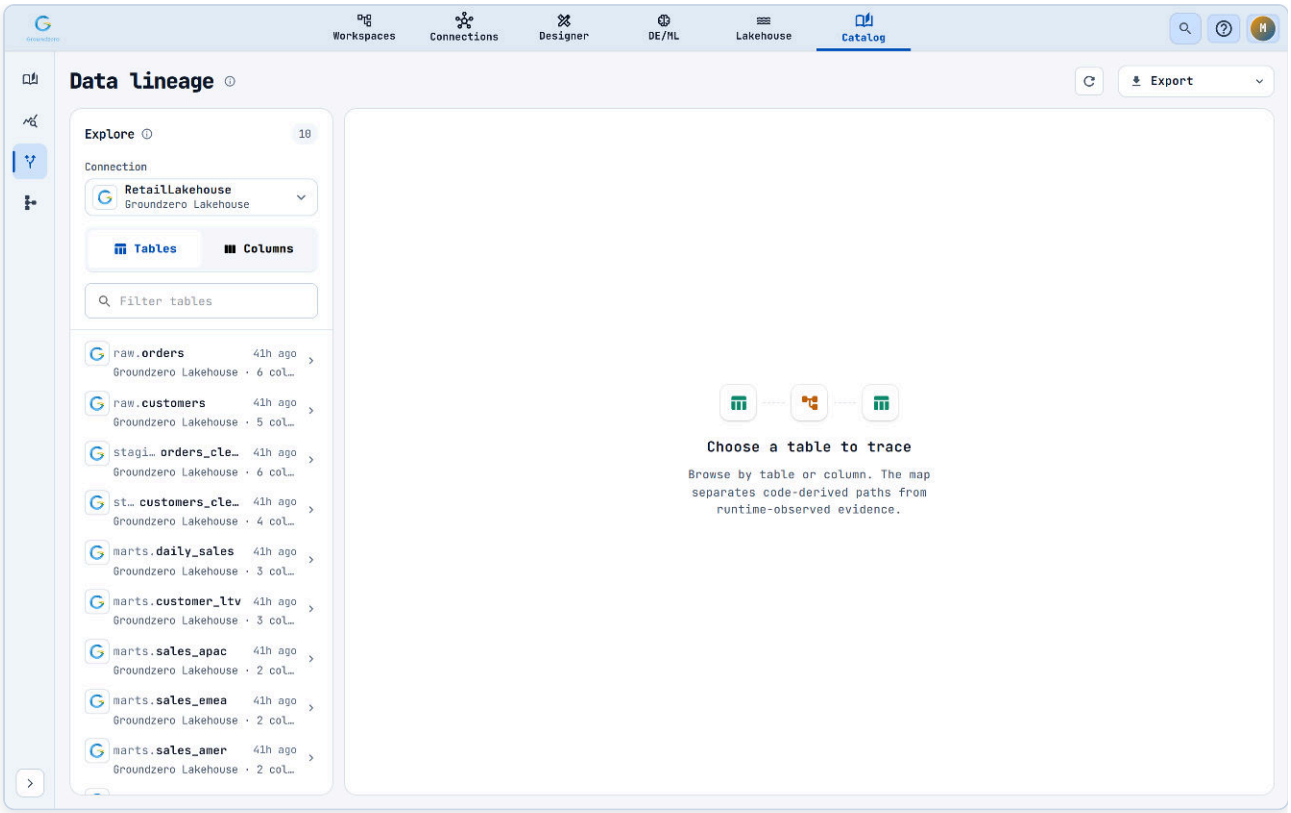


Figure 111. Choosing a connection lists its tables with the time each was last observed.

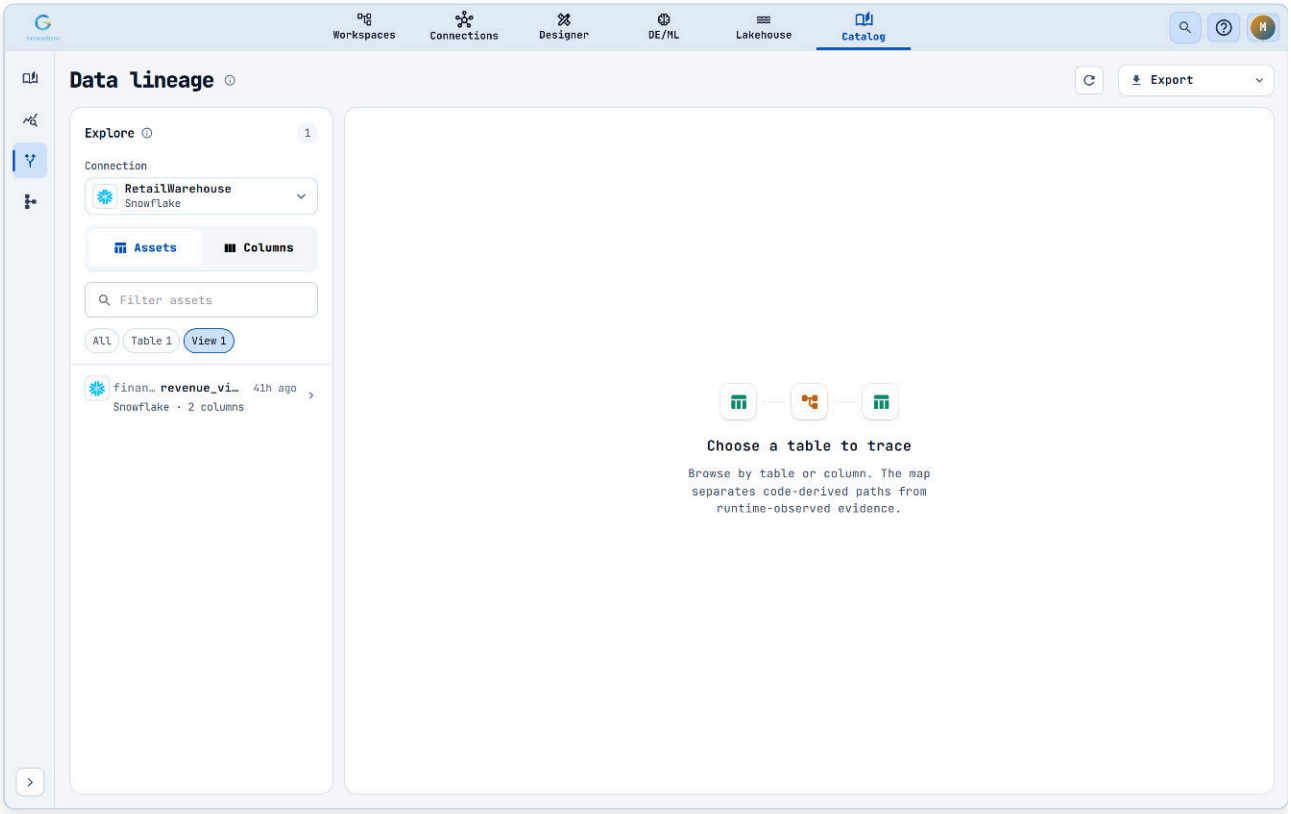


Figure 112. Asset-type chips filter a connection that holds tables and views.

3. To find a table by a column it contains, switch to the **Columns** tab and type the column name, for example `order_total`. Matching tables are listed as "Matches ...".

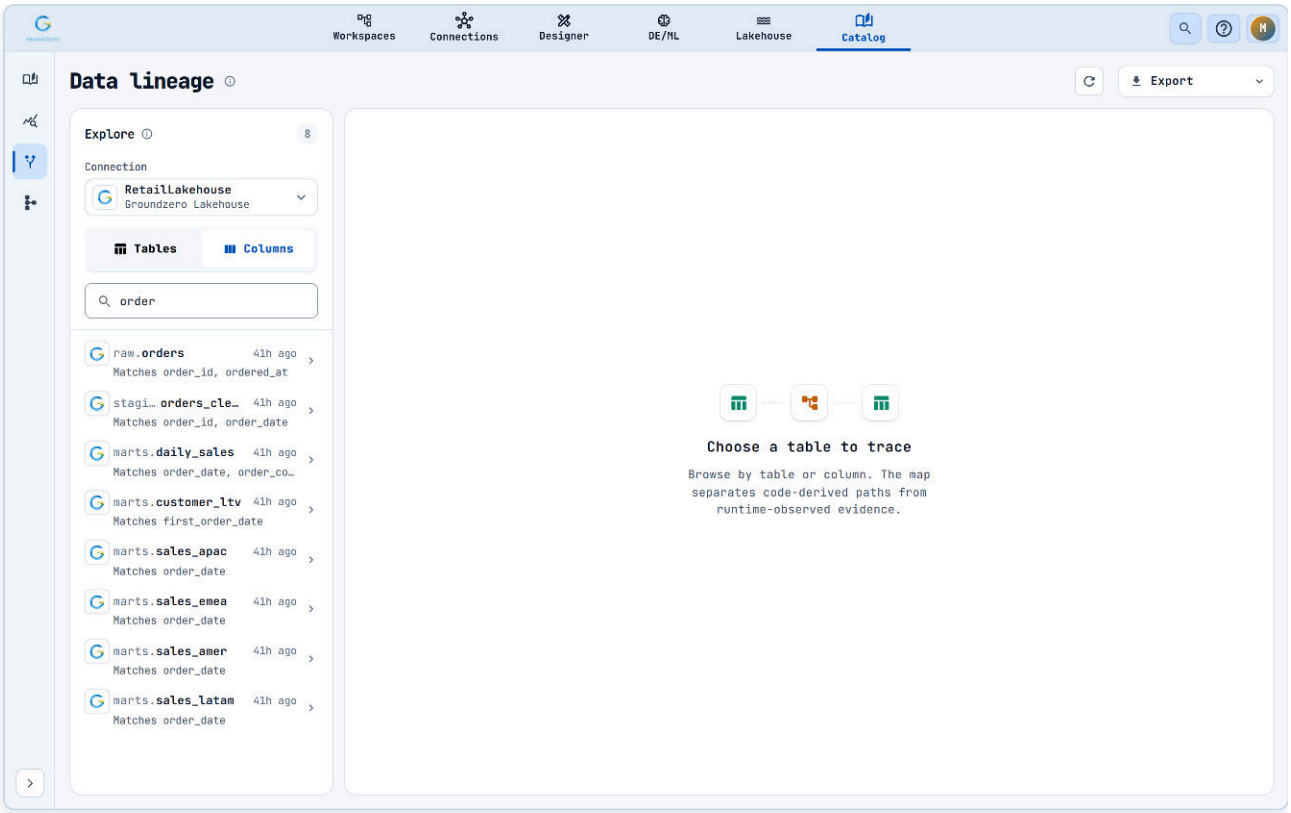


Figure 113. Searching columns finds tables whose columns match, listed as "Matches ...".

4. Click the table. Its map opens.

If a list stays empty you will see "Choose a connection to browse its lineage tables.", "Enter a column to find matching tables." or "No matching tables."

6.5 Read the map

Cards run left to right from sources to consumers.



Figure 114. The map only: sources on the left, consumers on the right; dashed lines are code-derived, solid are runtime observed.

Card	Shows
Table or asset	The connection logo, the name, the number of columns, and a **Deleted** badge if it was deleted at the source
ETL job or **Dataset**	Its name, how many times it ran (or Failed, Running, Queued, Compiling) and a status dot: green complete, red failed, amber running
Dashboard	Its name and number of widgets
Group	"N assets" or "N datasets" when more than three hang off one card; click to open it
Restricted	A placeholder for something you may not describe

A pill on a line shows "N columns", or **Inferred** when only the table-level flow is known. Hover a card to light up its whole upstream and downstream path.

How to move around:

- Drag empty space to pan. Drag a card to move it (positions last for this browser session).
- Click a card to select it; click empty space or press **Esc** to clear.
- Use **Zoom out**, the slider, **Zoom in** and **Fit lineage** to zoom. If the map looks odd, press **Fit lineage**.

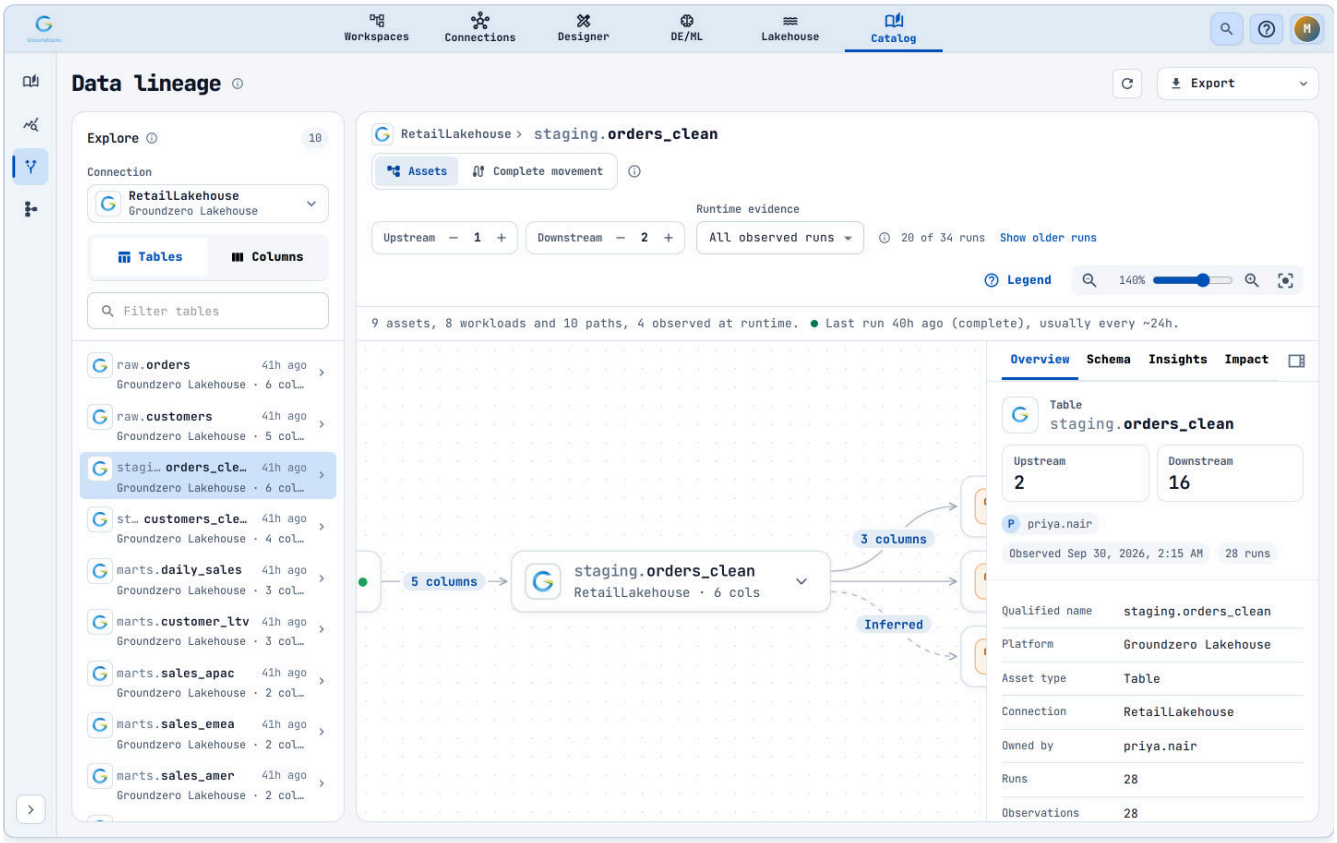


Figure 115. Zooming in with the toolbar.

6.5.1 The summary line

Under the toolbar, one line sums up the map, for example "9 assets, 8 workloads and 10 paths, 4 observed at runtime. Last run 39h ago (complete), usually every ~24h." It also tells you if the newest run is overdue, if no runs were observed, or if analysis failed (**Compile failed**).

6.5.2 Walk further upstream or downstream

The **Upstream** and **Downstream** steppers set how many steps to follow each way, from 0 to 5. The default is 1 each. At least one side must be 1 or more. The map reloads when you change them.

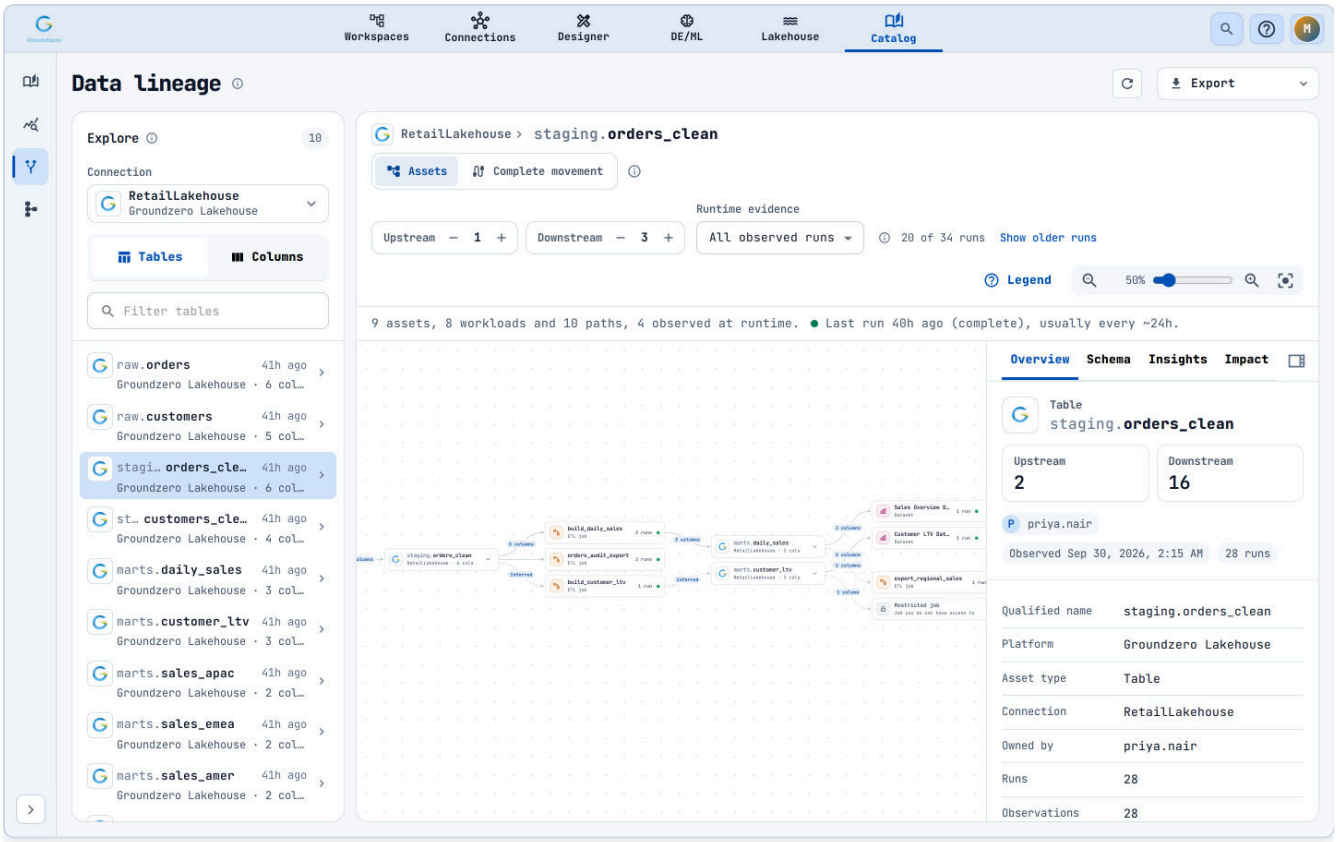


Figure 116. Downstream raised to 3 hops brings in the dashboards.

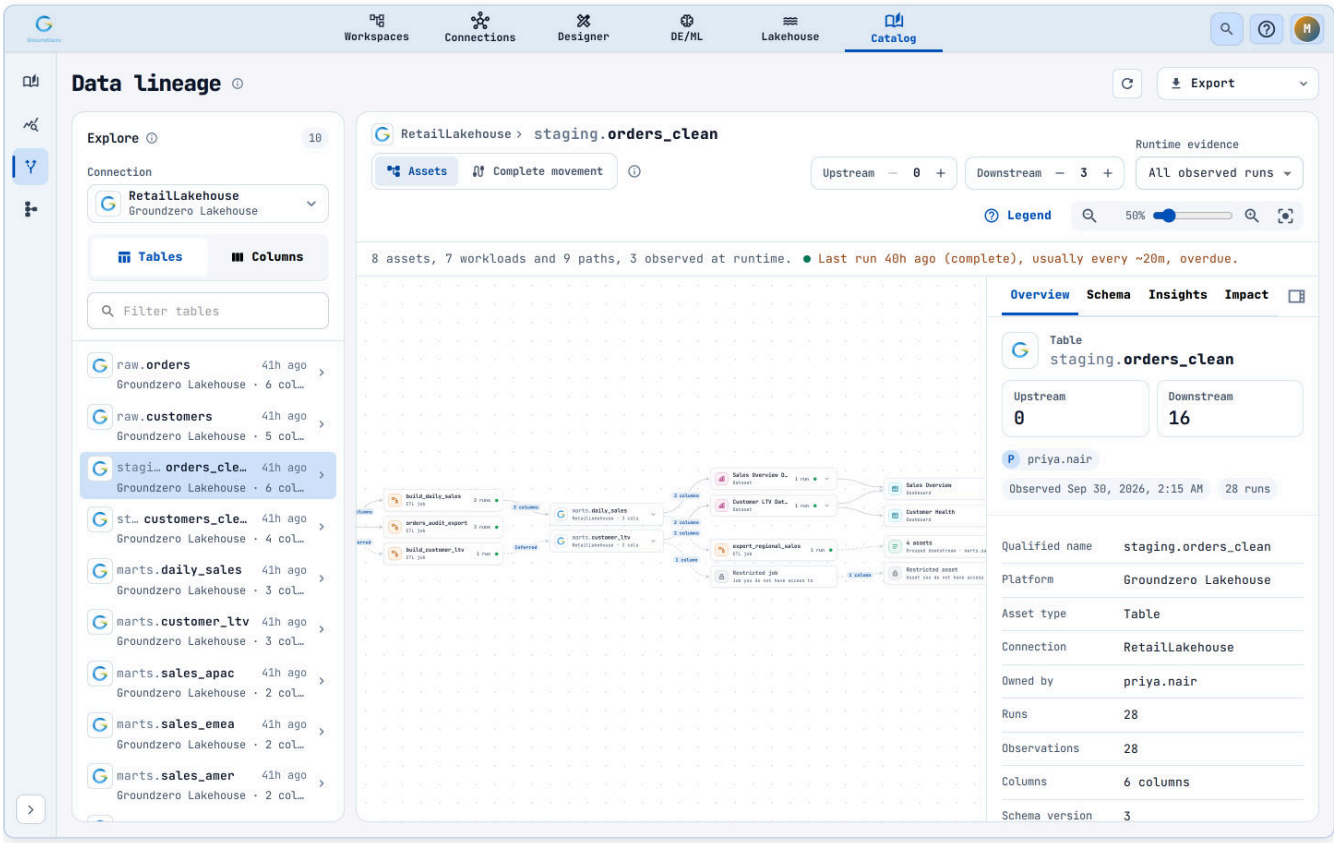


Figure 117. With Upstream at 0 only the table and what it feeds are shown.

The map draws up to 250 cards. If there are more you see "Showing {visible} of {total} nodes for interactive performance. Select a narrower asset scope to see the omitted paths." If the graph is too big, you see "This lineage scope is too large. Reduce the graph depth and try again."

> **TIP**
>
> Start with 1 step each way and add steps only where you need them.

6.6 See what breaks if a table changes

Outcome: a list of every job, dataset, dashboard and business definition that depends on the table.

1. Open the table's map and click the table card.
2. Open the **Impact** tab in the right panel.
3. Use **Depth** (1 to 5 levels, default 3) to choose how far to look.

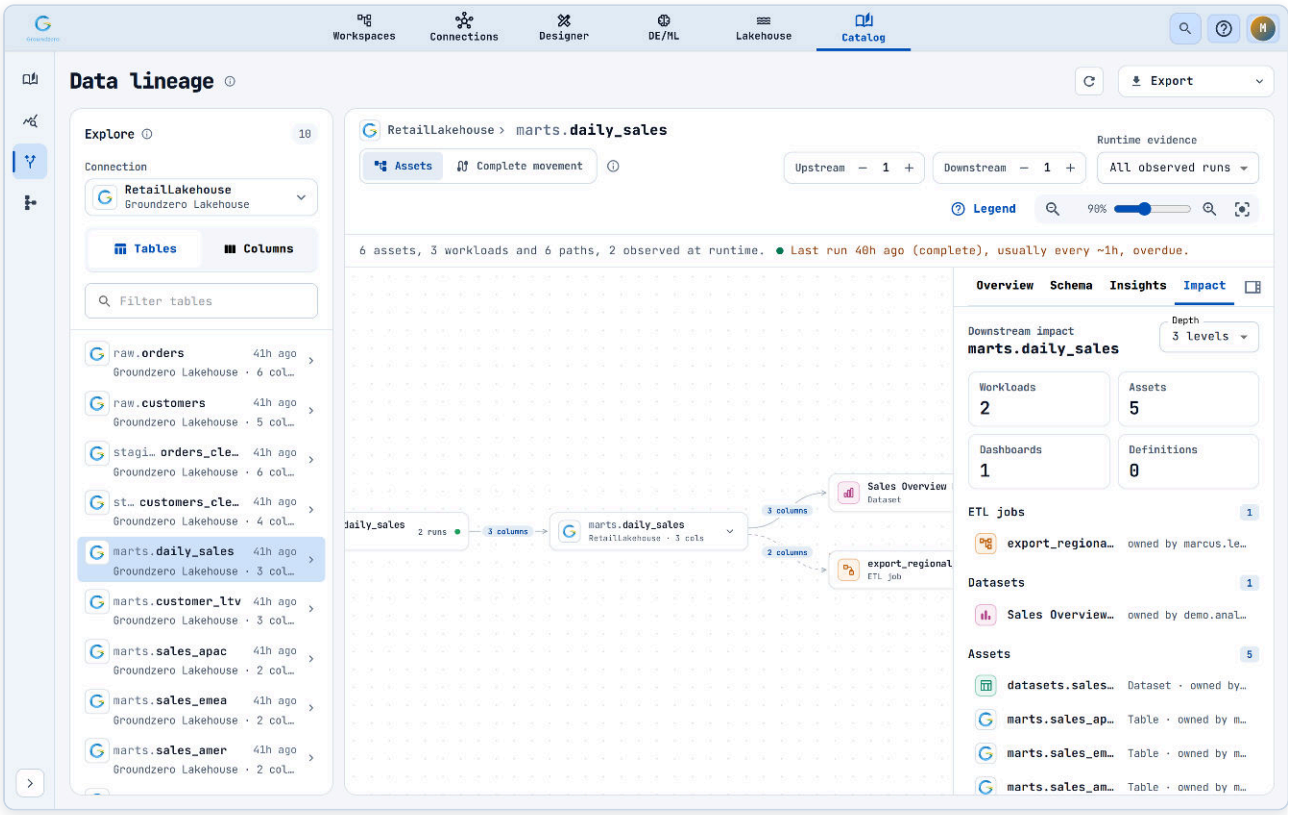


Figure 118. Impact: workloads, assets, dashboards and definitions that depend on the table.

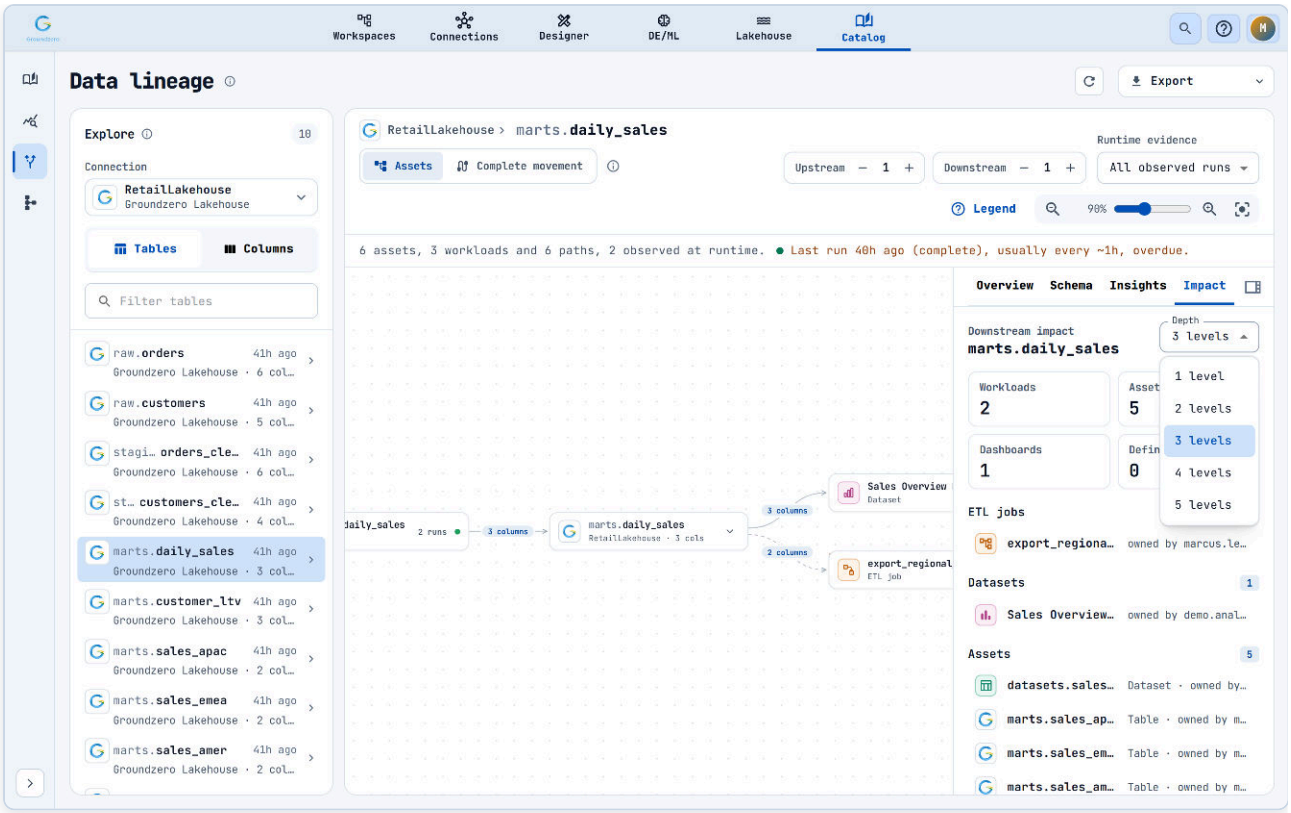


Figure 119. The depth select.

The tab shows counts for **Workloads**, **Assets**, **Dashboards** and **Definitions**, then lists them: ETL jobs, datasets, assets (with owner and last observed time), dashboards and business definitions. A business definition is a meaning recorded in Ontology Studio; you only see those you may read.

If nothing depends on the table you see "Nothing downstream reads from this asset in the recorded lineage. It is safe to change." Other notes: "Impact continues beyond the first {n} datasets shown." and "{n} restricted workload(s) not shown."

For a selected job the tab says "Impact flows from this job's outputs. Select an asset on the canvas to trace what it feeds."

> **TIP**
>
> Check Impact at depth 5 before dropping a table, and export the audit pack (see "Export lineage") for your change ticket. Lineage only knows about jobs and datasets that have run, so it can miss a user who queries the table by hand.

6.7 Trace a value back to its source

Outcome: the job, and the source table, that produced a number.

1. Open the map of the table that shows the wrong value.
2. Raise **Upstream** until you reach the source (try 3).
3. Click the job card feeding the table. The **Overview** tab shows its owner, when it last ran and its status. **Open job** or **Open dataset** goes to it.

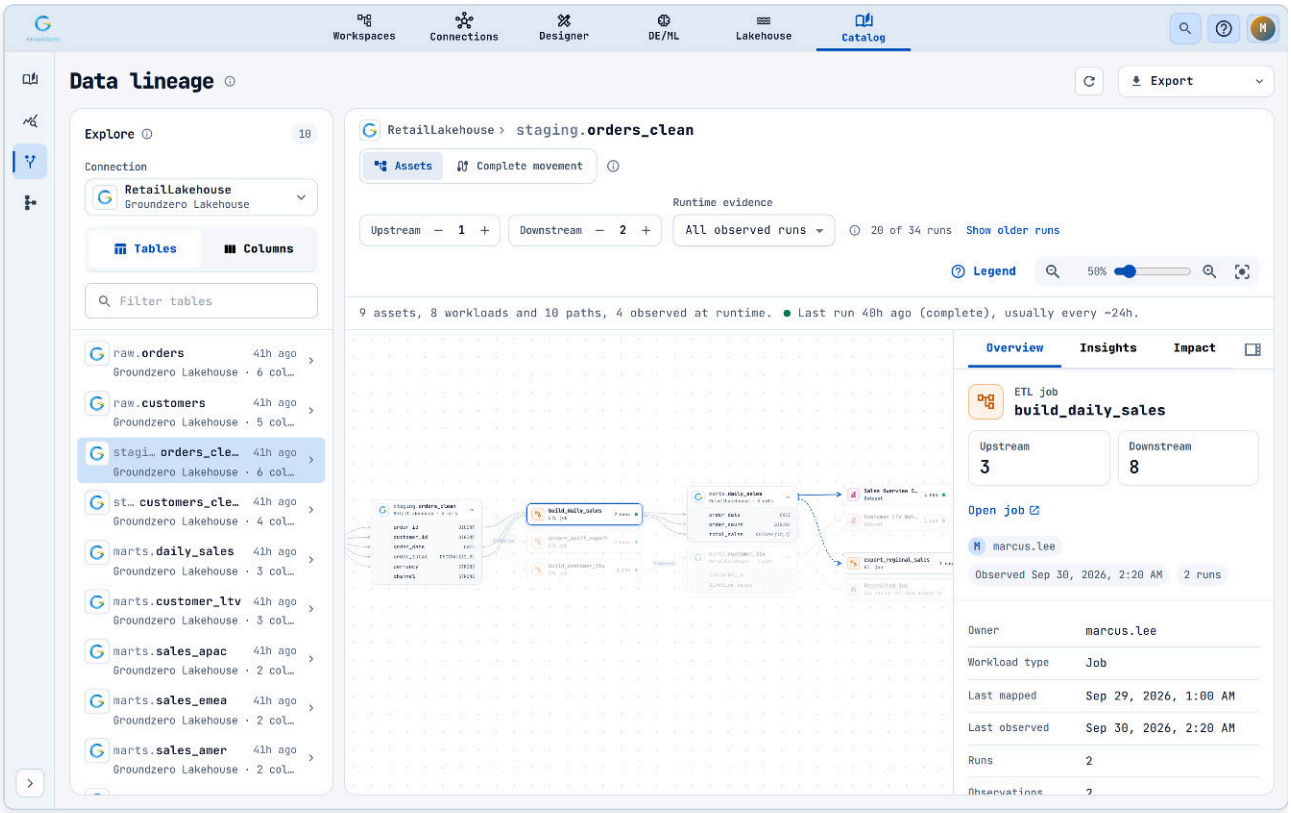


Figure 120. A job card: owner, workload type, last mapped and observed times.

4. Click a line to see how the two ends connect and whether it was observed in a run.

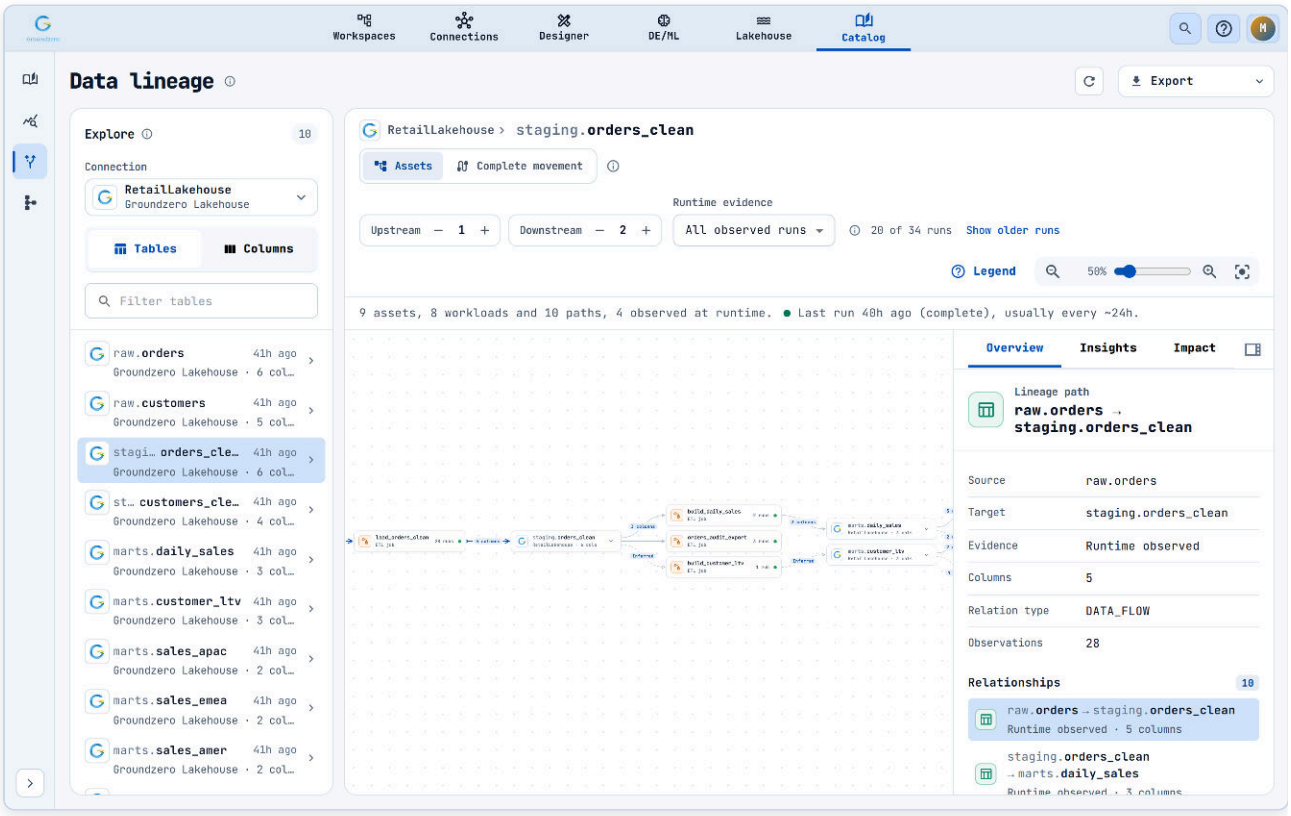


Figure 121. Clicking an edge shows the flow's evidence and relationships.

5. To see only what one run touched, choose it in **Runtime evidence** (see "Check one run").

6.8 Follow a column

Outcome: which source columns feed a column, and whether it was renamed or changed on the way.

1. Click the arrow on a table card (**Show columns for <name>**). Neighbouring tables open too. Up to 15 mapped columns are shown, then "+N more columns".
2. When two tables are open, curves join each source column to its target column through the job card. Click a column or a curve to focus on it. Click again, click empty space or press Esc to release.

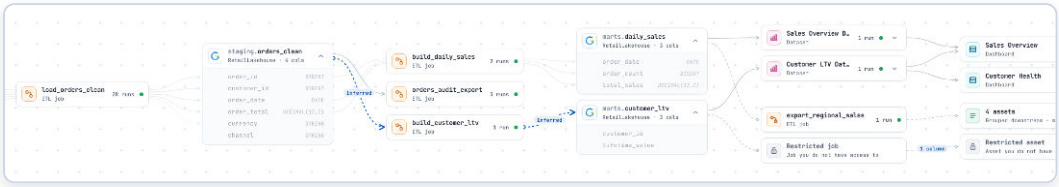


Figure 122. Columns expanded: curves connect mapped columns across the job.

Curve labels tell you what happened: **Direct mapping**, **Renamed**, **Transformed**, **Renamed + transformed**, **Transformation unknown** or **Renamed · transformation unknown**.

> **INFERRED FLOWS**
>
> When Groundzero cannot work out the columns (for example when the code uses `SELECT *`), the flow has no curves and shows **Inferred** ("Inferred, columns unknown"). The table-level flow is still correct.

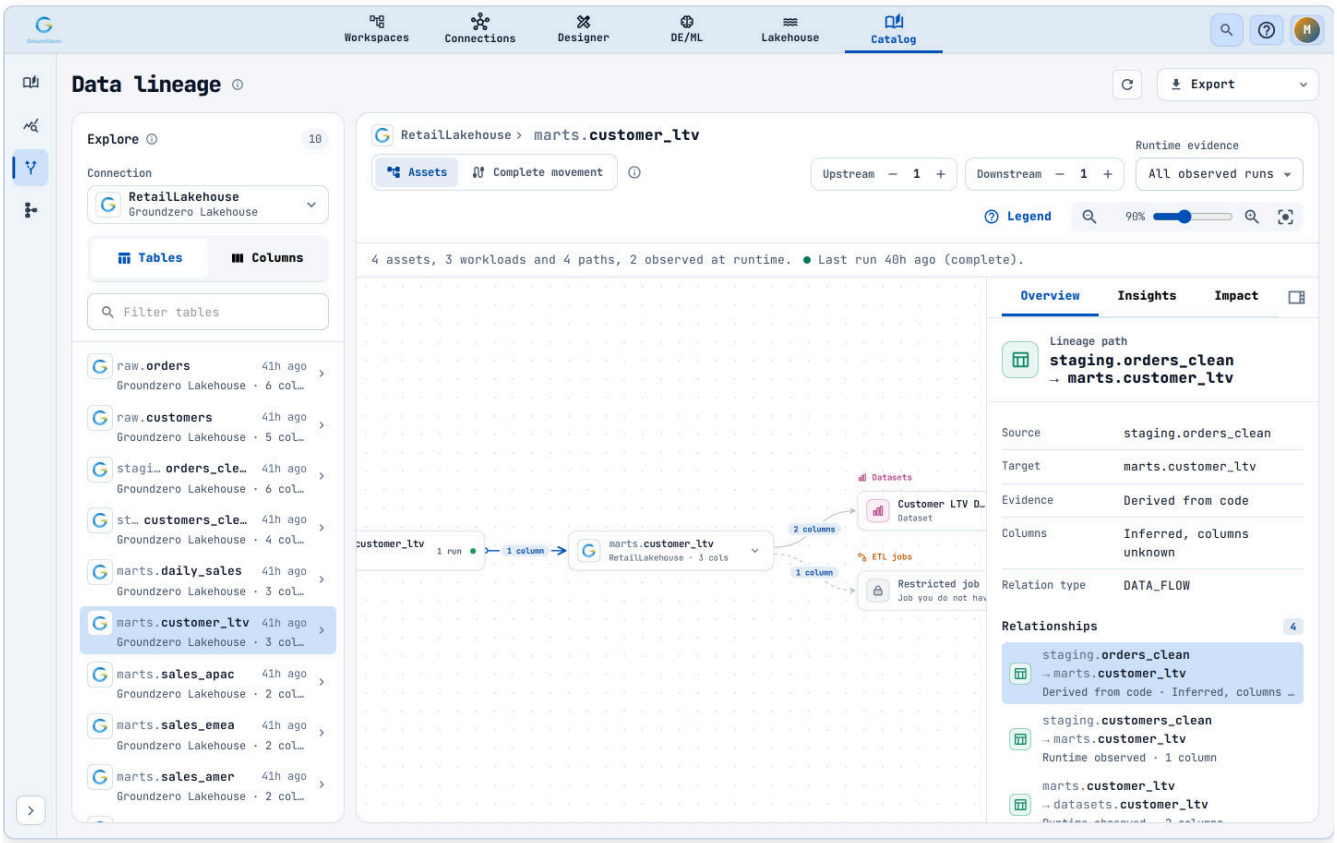


Figure 123. An inferred flow reports "Inferred, columns unknown".

6.8.1 Groups

When more than three cards hang off one card, they collapse into one group card. Click it to open the group for the rest of your visit.



Figure 124. More than three siblings collapse into one group card.

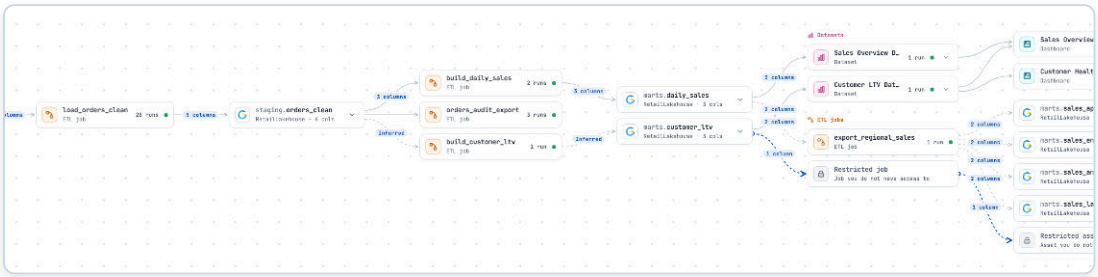


Figure 125. The expanded group shows each table.

6.9 Check one run

The **Runtime evidence** list shows "All observed runs" or one run, for example "Complete · Sep 30, 2026, 3:10 AM". Choose a run to see only what that run observed. The legend wording changes to "selected run" and the details panel gets a **Selected run** section (Run ID, Started, Stopped, Duration).

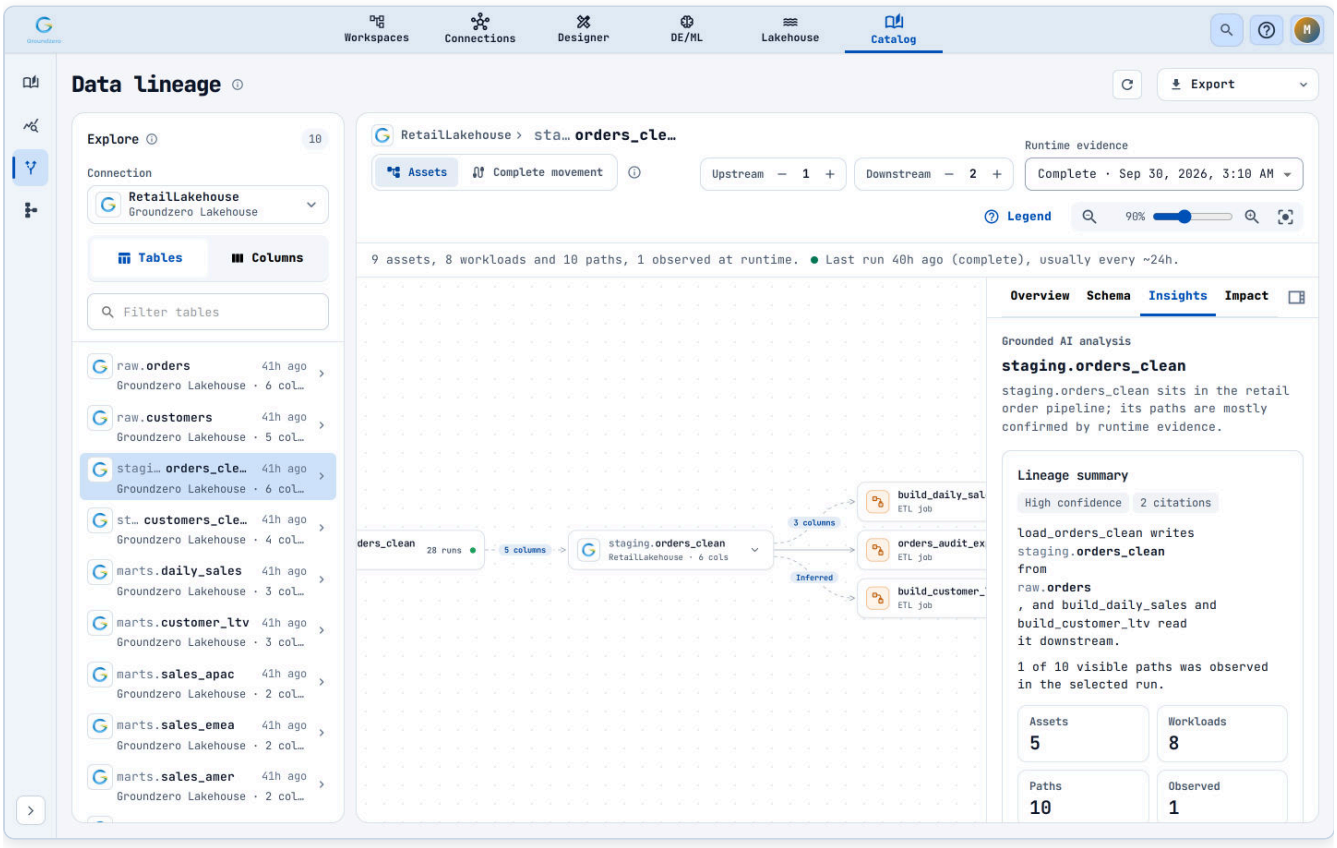


Figure 126. Choosing a run narrows the evidence; the summary now counts what that run observed.

The map loads the 20 most recent runs. The toolbar says, for example, "20 of 34 runs". Click **Show older runs** to load more (up to 100). The shape of the map is always complete; only the run history is limited. If it fails you see "Older runs could not be loaded." A map with a running or queued run refreshes itself every 10 seconds.

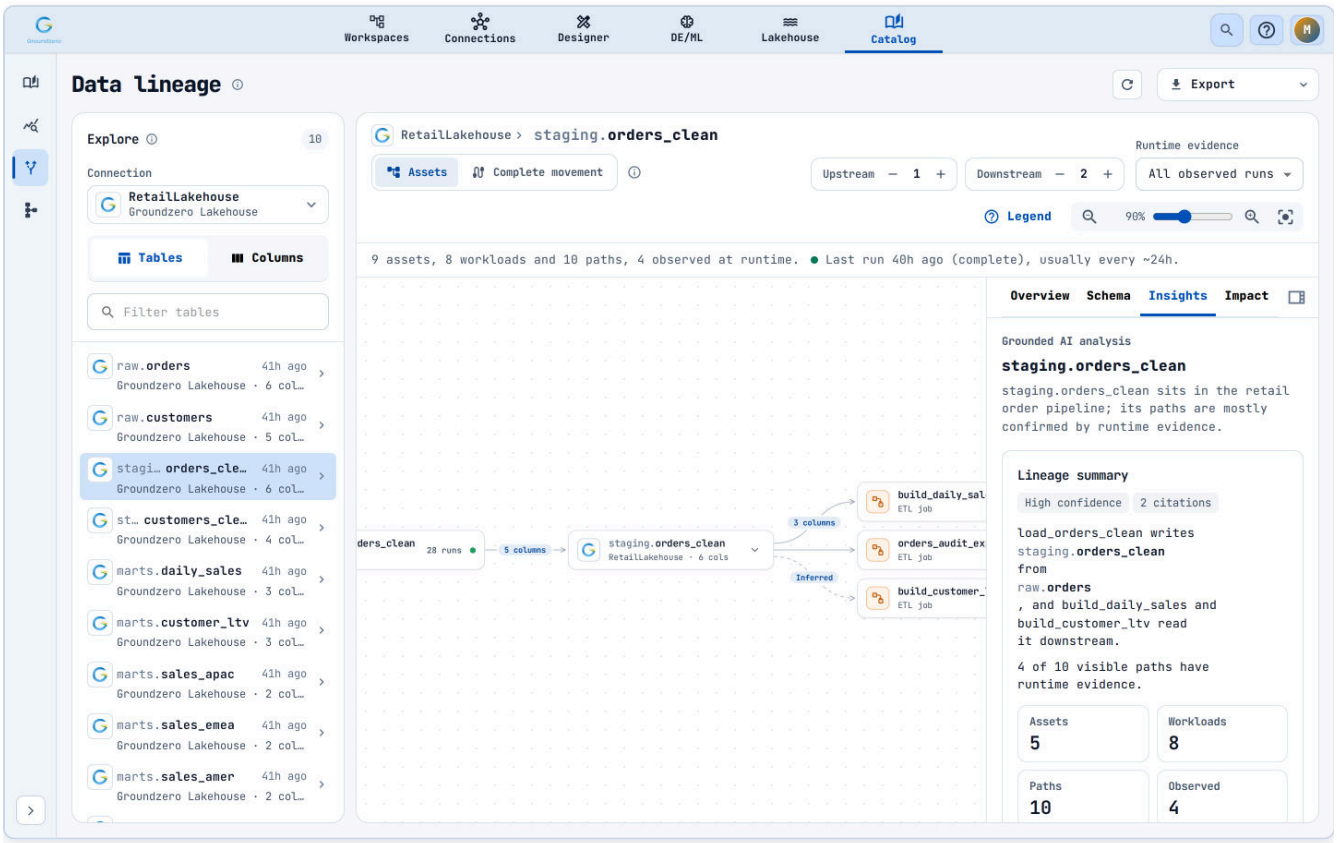


Figure 127. Show older runs loads the complete run history.

6.10 Use the details panel

The right panel has the tabs **Overview**, **Schema** (for tables), **Insights** and **Impact**. The toggle **Hide lineage details** collapses it to a thin strip so the map gets the full width.

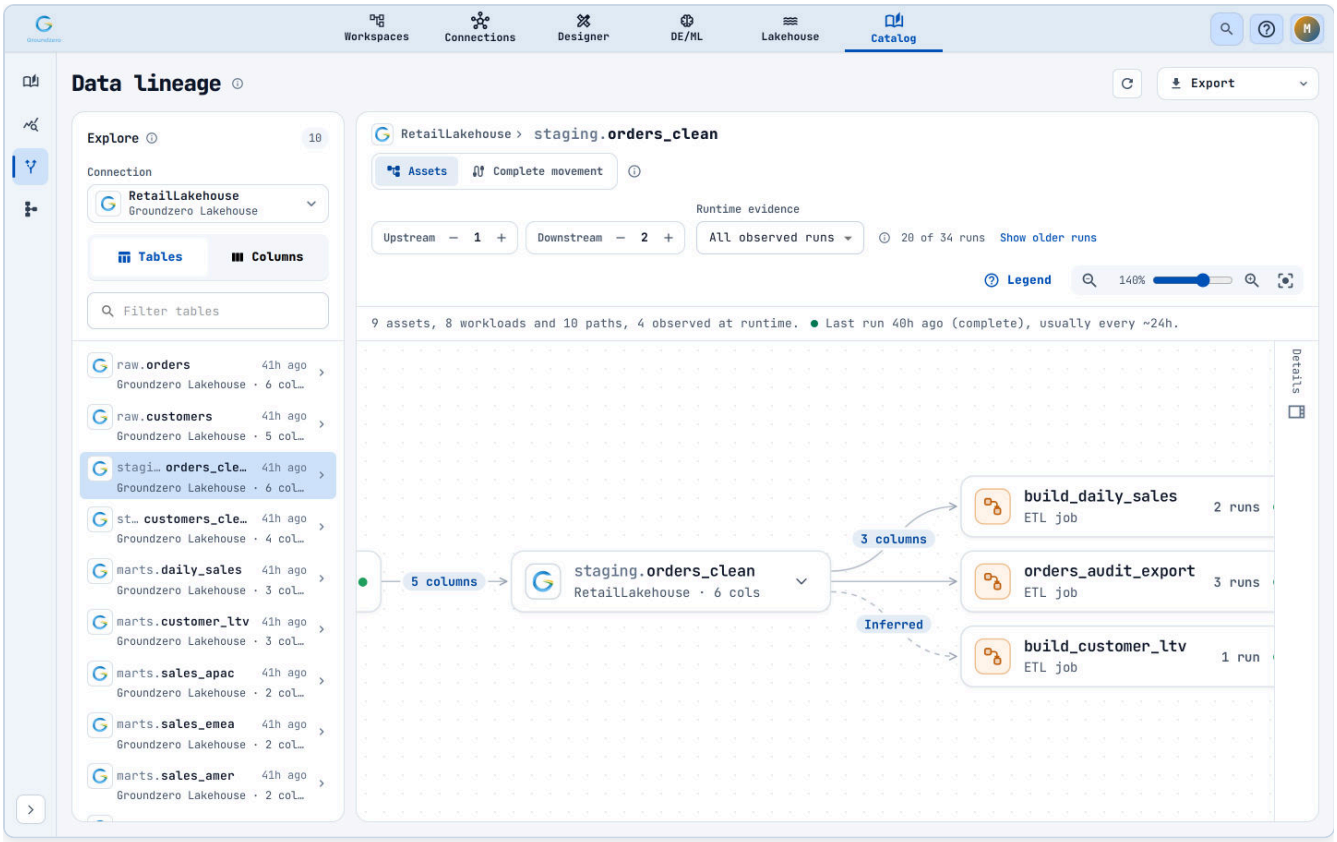


Figure 128. With the panel collapsed the map has the full width.

6.10.1 Overview

The header shows what you selected (Table, Dataset, ETL job or "Lineage path"), its name, the **Upstream** and **Downstream** counts and a row of trust signals: the owner (or "Unowned"), when it was last observed ("Not observed" if never) and how many runs.

- For a **table**: its qualified name, platform, connection, owner, runs, columns, schema version, last observed time, and the jobs that write it (**Written by**) and read it (**Read by**). **From code** marks a job that has never run.
- For a **job or dataset**: owner, type, last mapped and last observed times, run count and a **Diagnostics** section with its analysis status and warnings. A dataset also lists the **Dashboards** built on it.
- For a **line**: source, target, its evidence (**Runtime observed** or **Derived from code**), the columns, how often it was observed, and **Relationships** to jump between lines.

6.10.2 Owners

Groundzero works out owners for you. A job's owner is the person who owns or created it. A table's owner is whoever owns the latest job that wrote it; if none, the dataset's or connection's creator. When none can be found the panel reads **Unowned**.

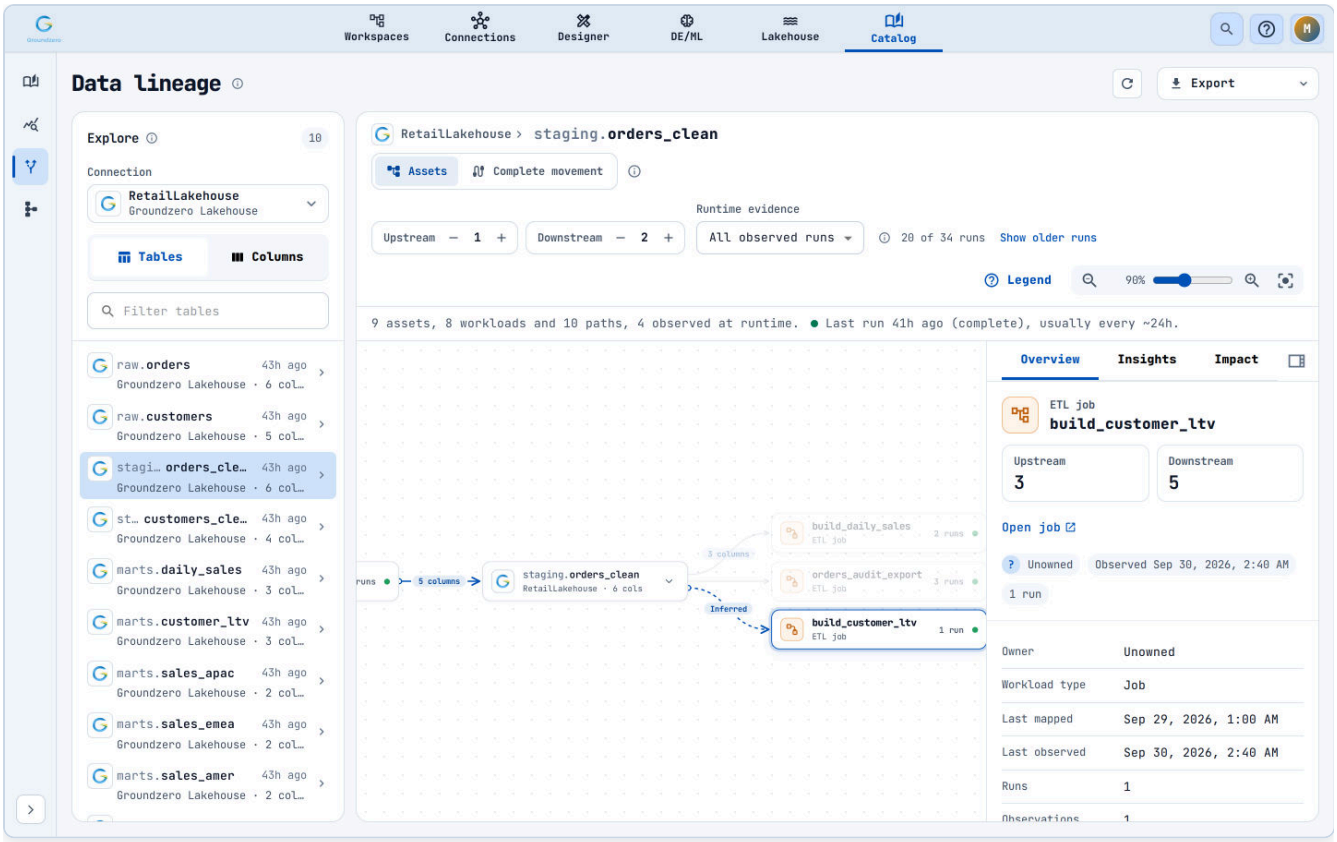


Figure 129. A job with no owner reads "Unowned" in the trust row and the Owner row.

6.10.3 Schema

The **Schema** tab lists the table's **Columns** with their types, then **Schema changes**: the five most recent versions with chips for columns added, removed or changed. If you see "Schema not observed. Run a job or dataset that reads this table to record its columns.", run one.

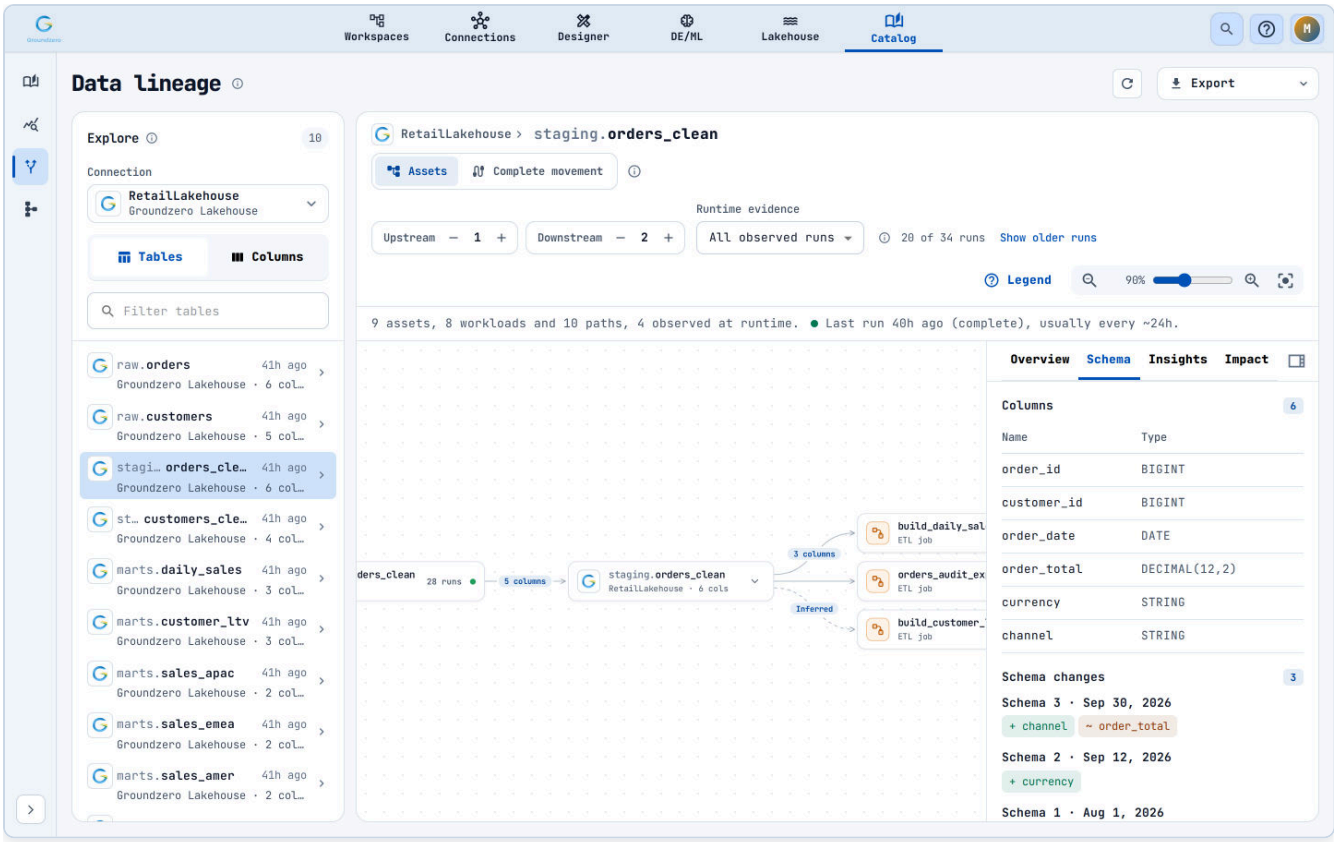


Figure 130. Schema and recent schema changes.

6.10.4 Insights

The **Insights** tab gives a plain-language reading of the map in four cards: **Lineage summary**, **Breaking-change impact**, **Column transformations** and **Schema drift and evidence gaps**. It starts with **Evidence analysis**, then changes to **Grounded AI analysis** when the AI answer arrives. AI cards show a confidence level and the number of sources they cite; **Show more** expands long text.

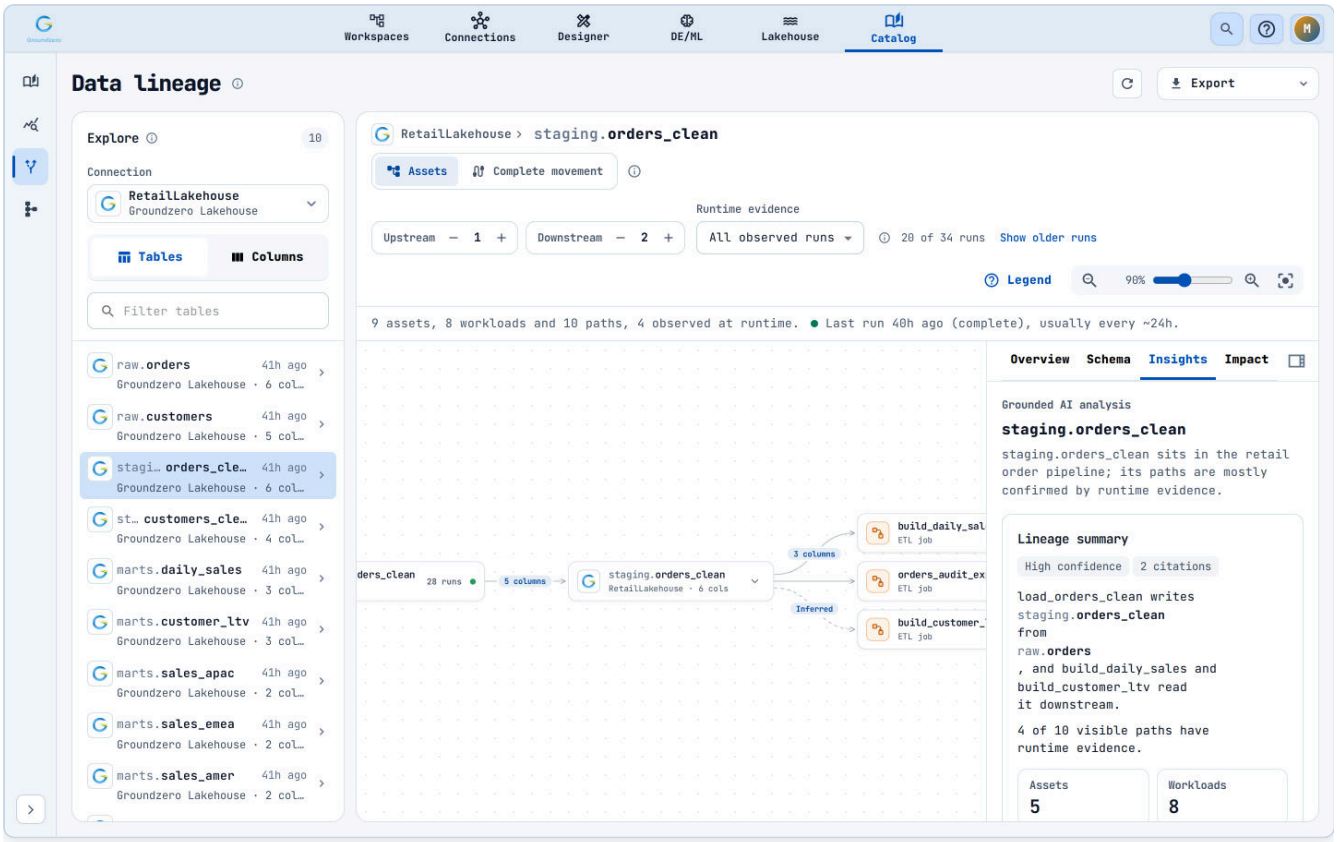


Figure 131. Insights: grounded analysis with confidence and citations.

If AI is not available you see "AI analysis is unavailable. Verified graph evidence is shown instead." The cards built from recorded evidence still appear.

6.10.5 Dashboards

Dashboards are drawn in the rightmost lane, after the datasets they are built on. Raise **Downstream** to bring them in. Select a dataset to see its **Dashboards** list in Overview.



Figure 132. Dashboards appear downstream of the datasets they are built on.

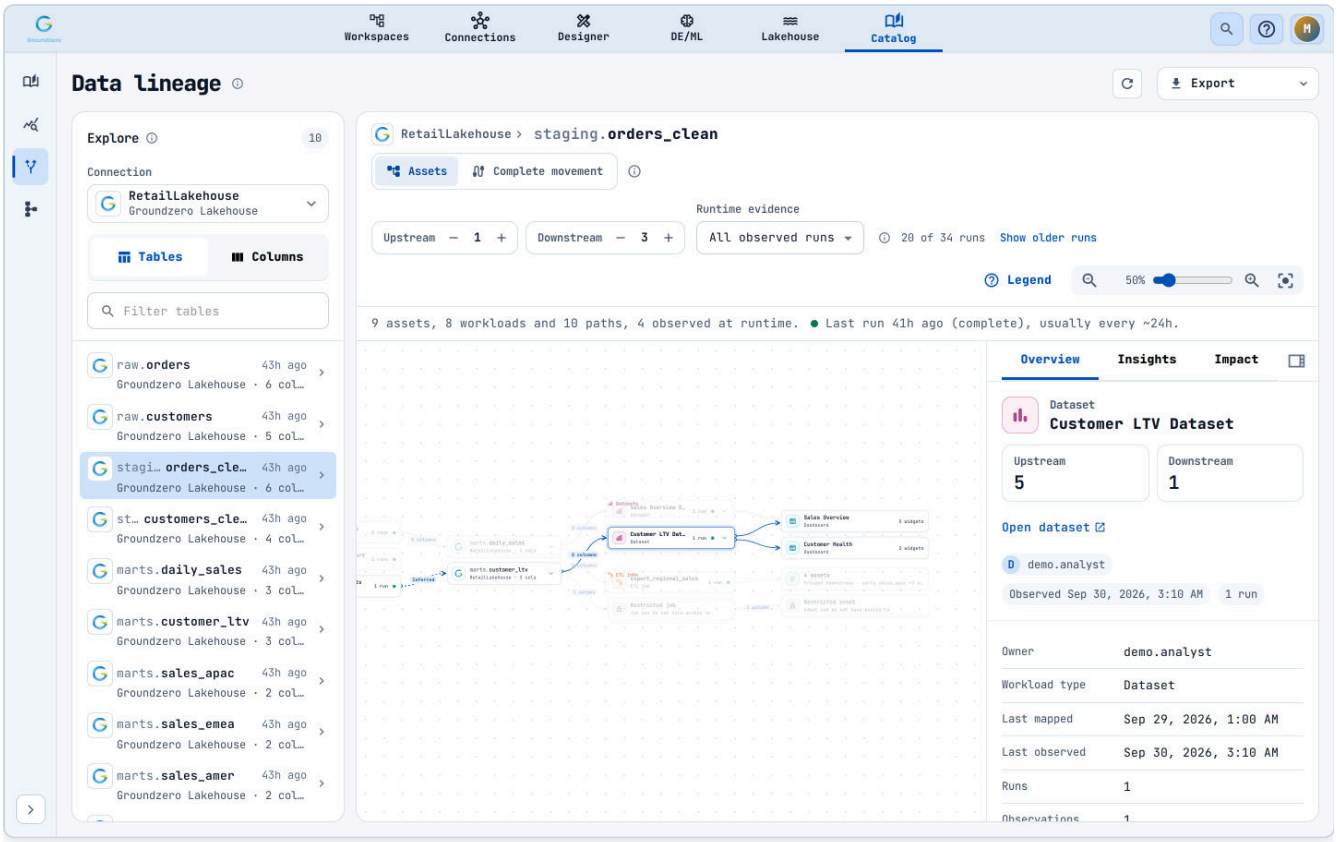


Figure 133. A dataset's Overview; scroll the panel to the Dashboards section that lists the dashboards built on it.

6.10.6 Restricted items

Anything you may not describe shows only as a placeholder, and the rest of the map still works.

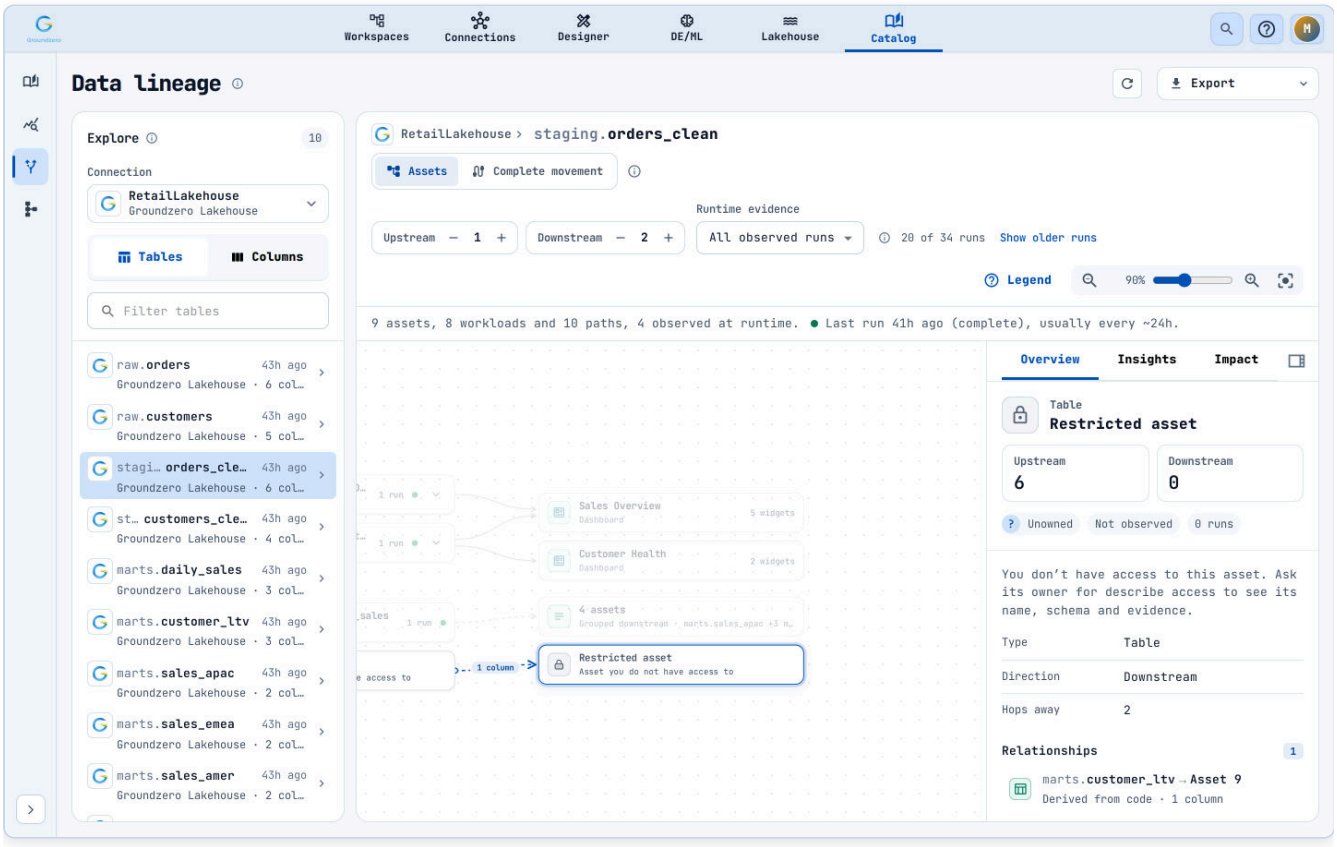


Figure 134. A restricted asset reveals only its type and distance.

The panel says "You don't have access to this asset. Ask its owner for describe access to see its name, schema and evidence." If the table you started from is restricted, you see **Lineage access is restricted**.

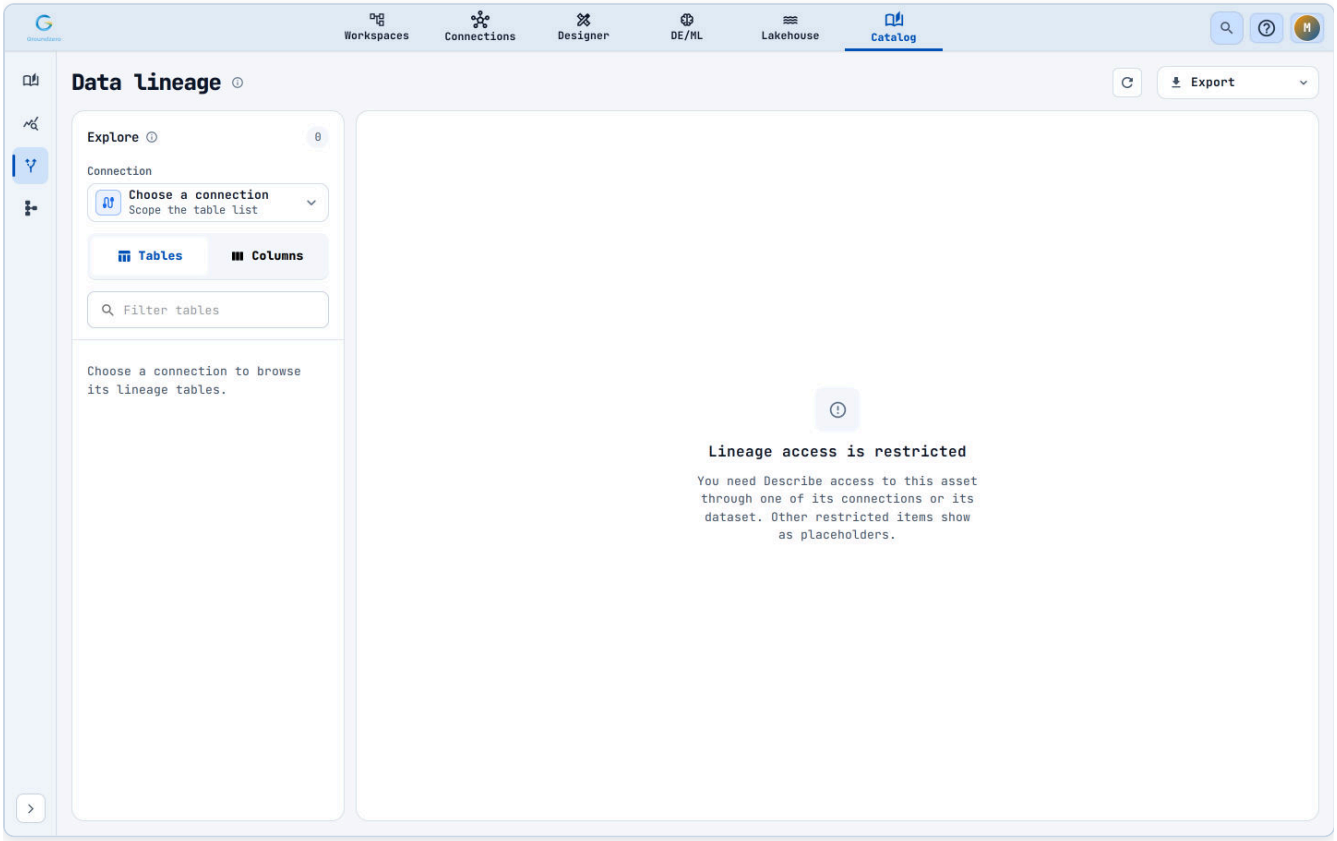


Figure 135. Restricted starting point.

6.11 See every hop: complete movement

Use the switch at the top of the map to change from **Assets** to **Complete movement**. This view adds every recorded hop and the saved datasets, visualizations and dashboards built on the table.

A bar lets you filter by kind (Table, ETL job, Dataset, Visualization, Dashboard) and by line style (**Runtime observed**, **Derived from code**, **Saved reference**), search with **Find a node**, and shows a count such as "23 nodes · 25 links".

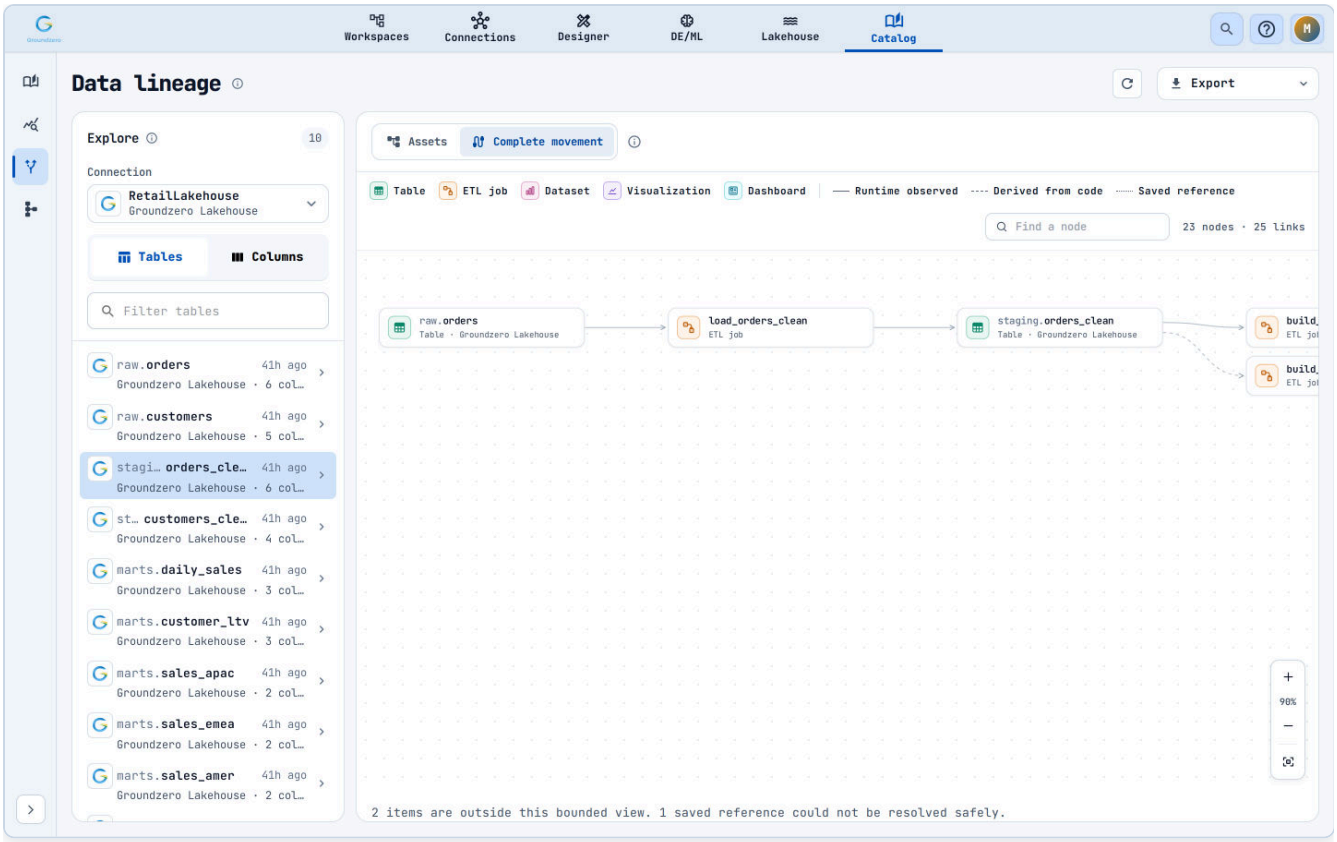


Figure 136. The complete movement view.

Click a card for details: what it **Reads from**, what it **Feeds** and what is **On its path**. **Close details** closes them.

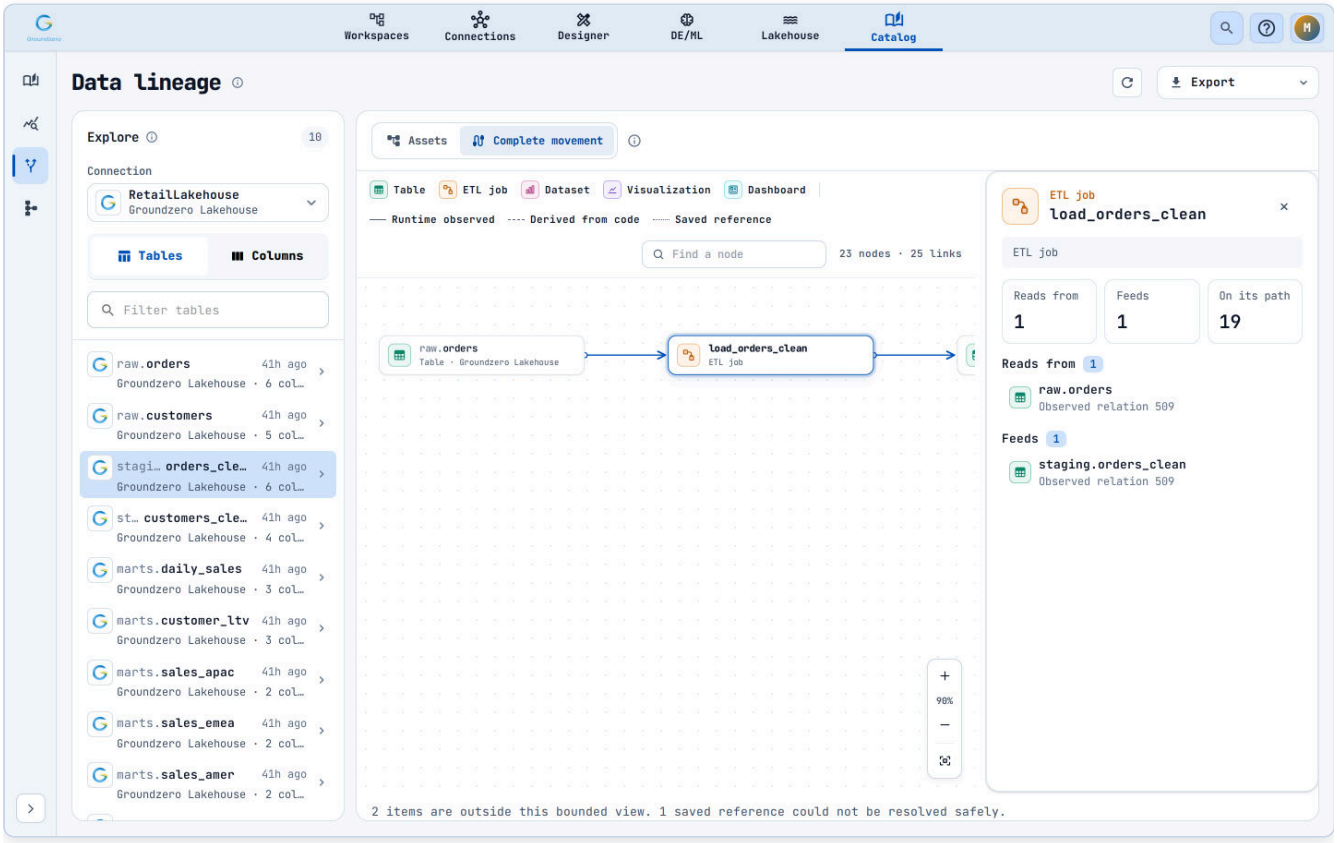


Figure 137. A movement card's details.

The view is limited to a set number of items; the footer says how many fall outside it. "No connected data movement is available for this mapped source." means nothing is recorded yet.

6.12 Start from a job or dataset

You can also centre the map on one job or dataset instead of a table; the breadcrumb then reads "ETL job > name" or "Dataset > name". The usual way is to follow **Open job** or the lineage link from a job or dataset.

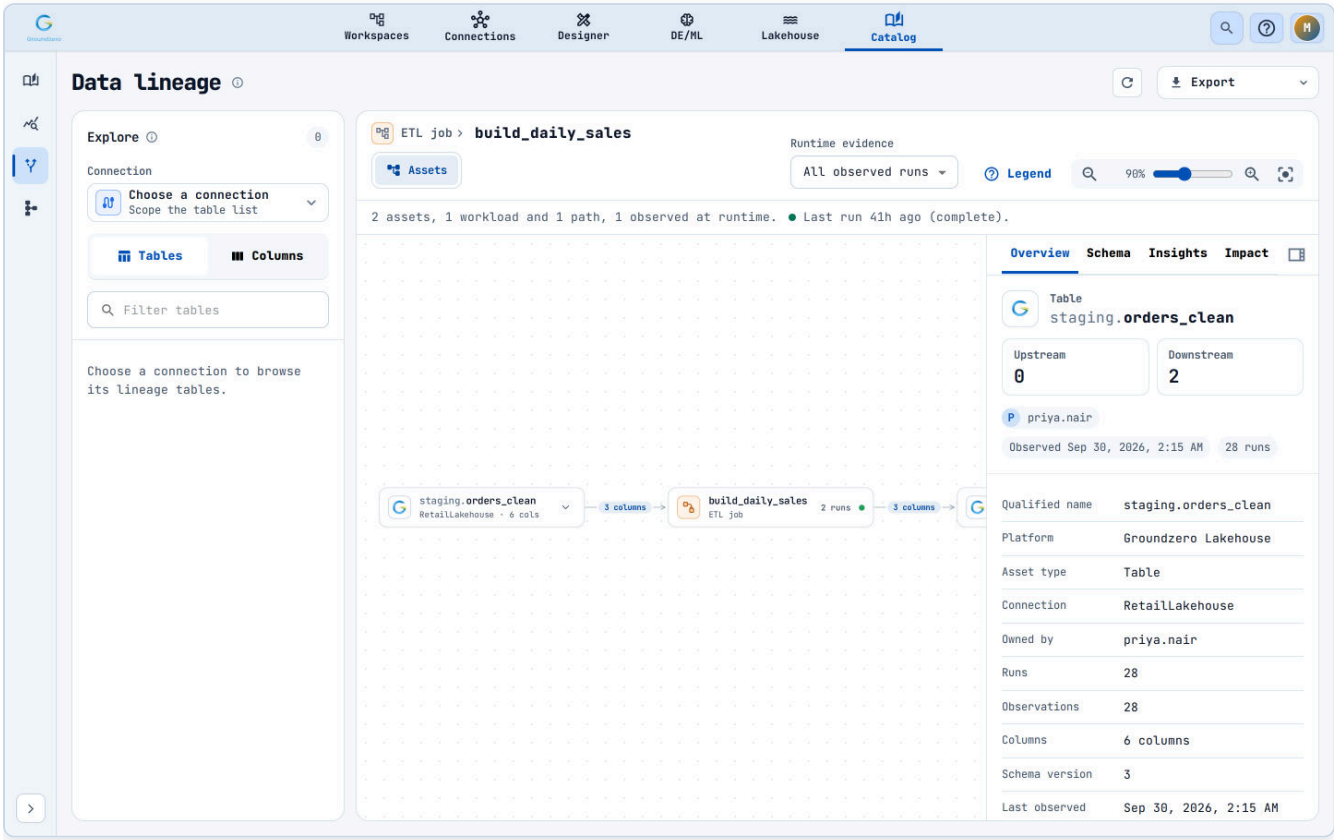


Figure 138. Job scope.

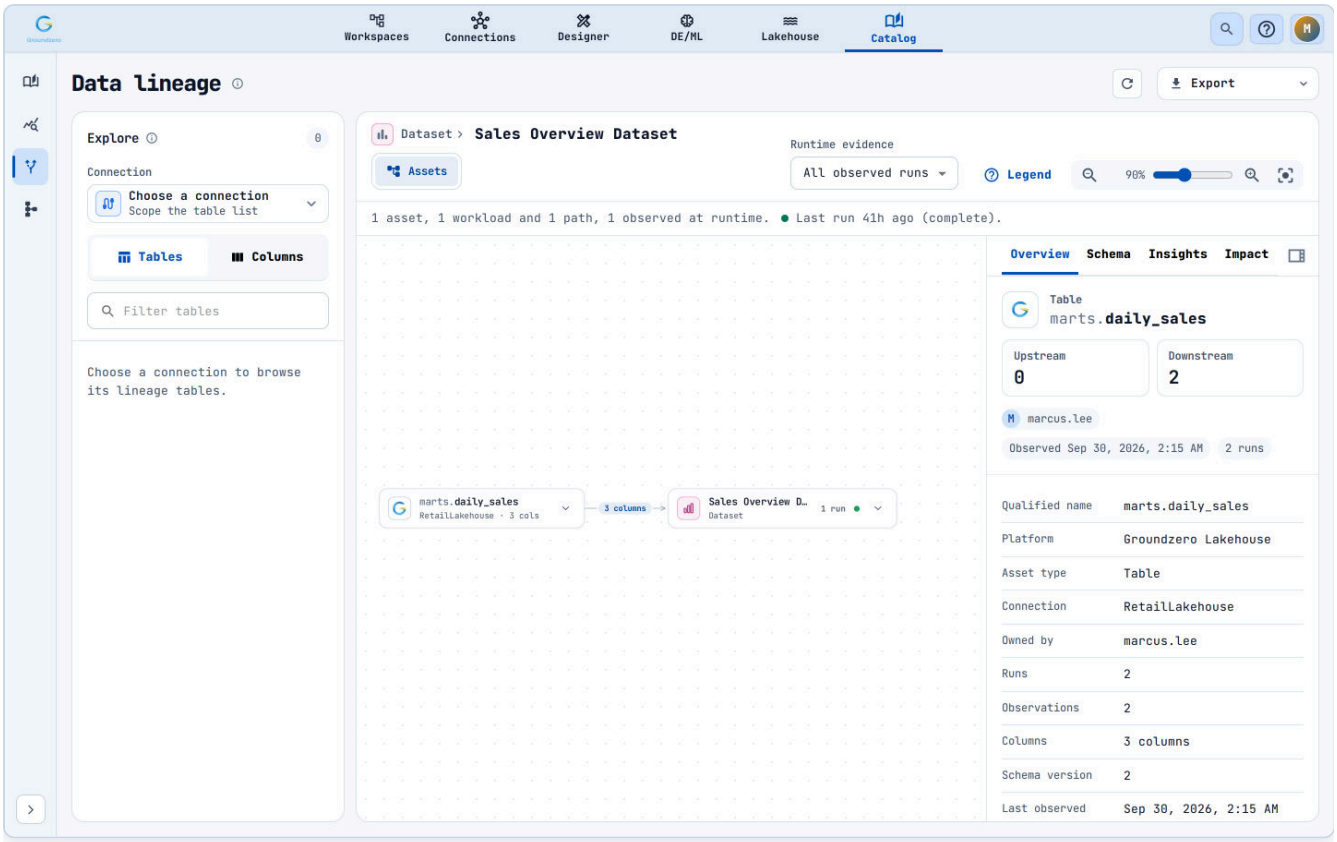


Figure 139. Dataset scope.

6.13 Export lineage

Outcome: a file with the lineage you are looking at, or every connection for audit.

Open the **Export** menu.

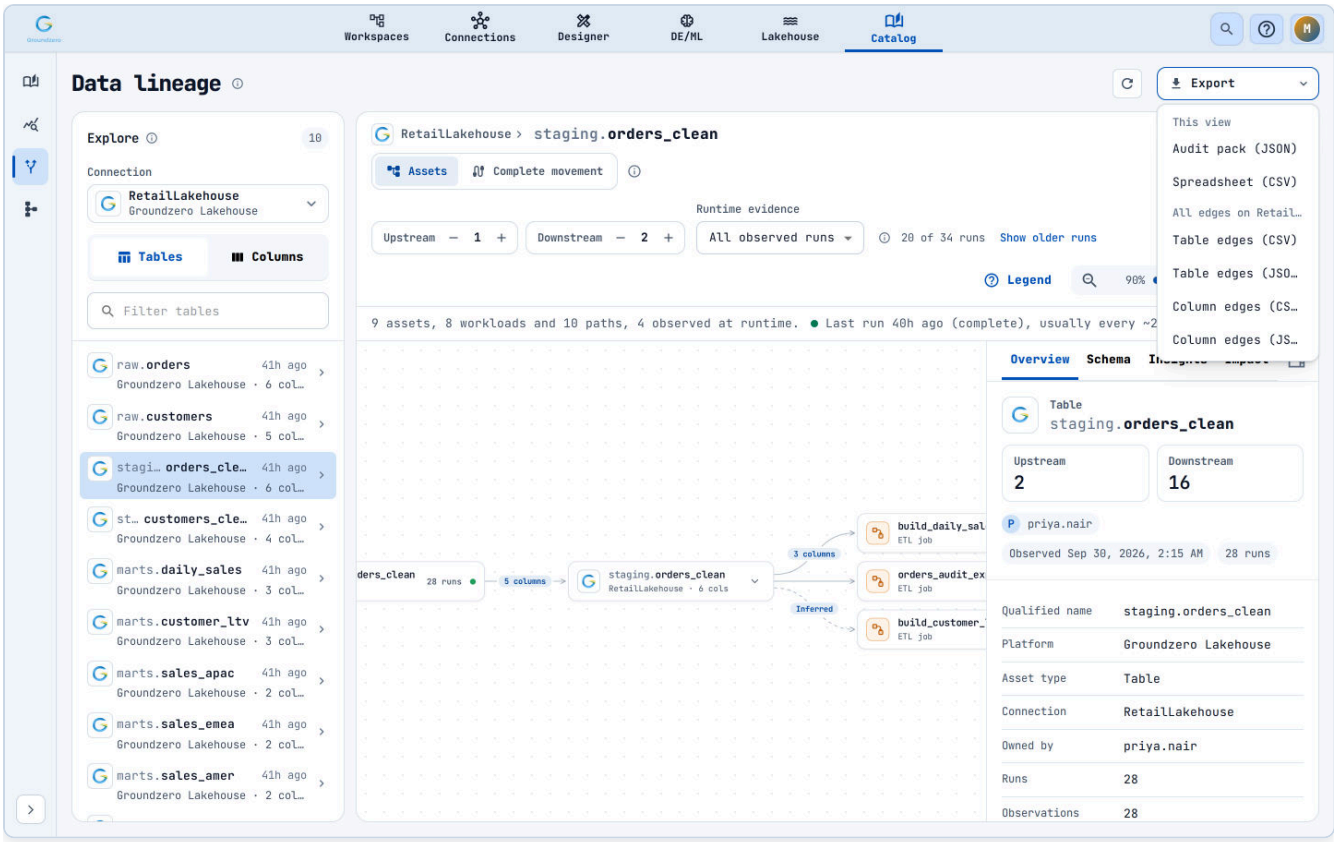


Figure 140. The Export menu.

Group	Item	What you get
This view	**Audit pack (JSON)**	The full picture of the current map: counts, evidence, assets, relations, job versions and runs
This view	**Spreadsheet (CSV)**	One row per line: source, target, columns, job, owner and run
All edges on a connection	**Table edges** and **Column edges** (CSV or JSON)	Every link on the connection, up to 10,000 rows

The first two are on once a map has loaded. A status line reports "Exporting lineage…" then "Exported {n} edge(s)." with notes such as "{n} edge(s) hidden by access rules." or "Only the first 10,000 were included." You only get rows for tables and jobs you may see.

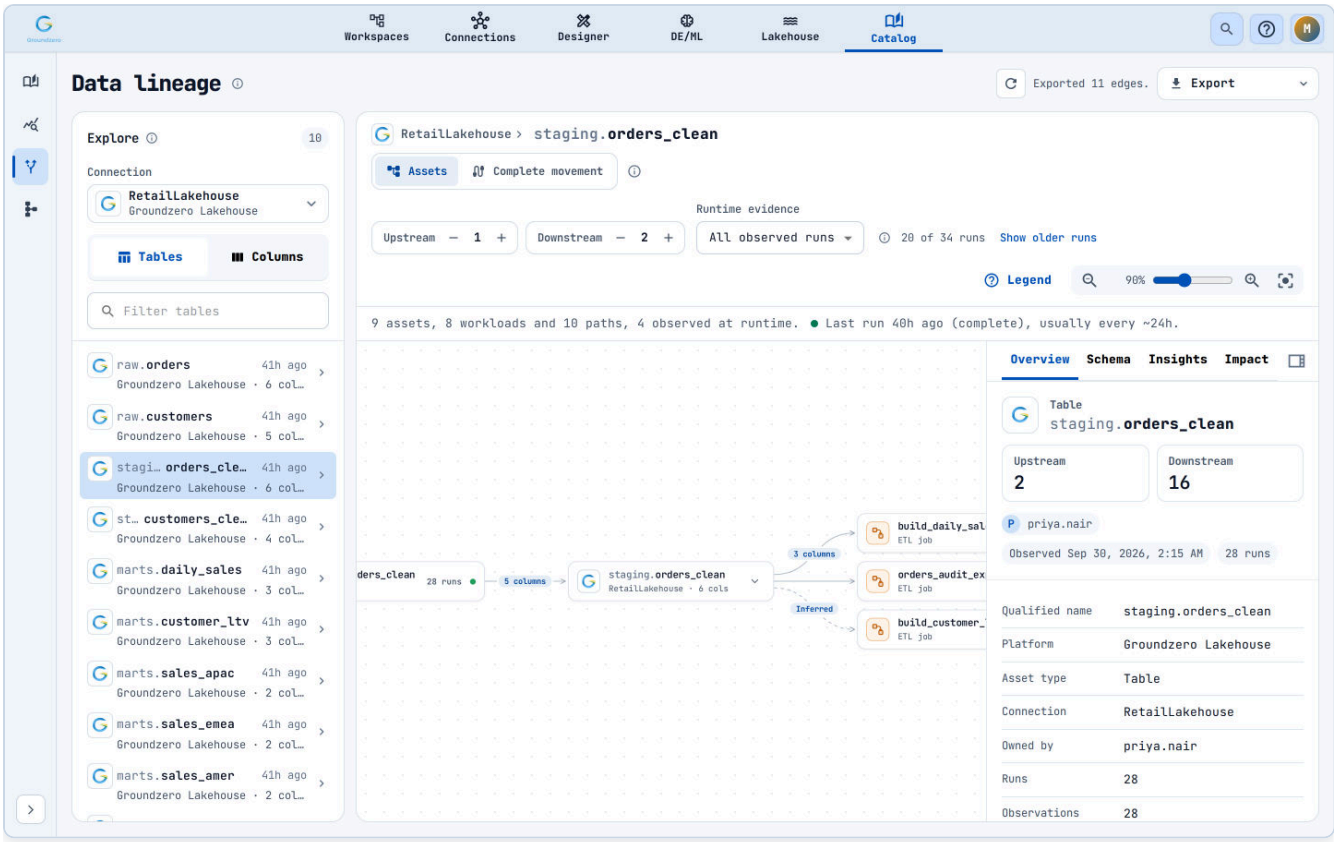


Figure 141. The export status line beside the button.

6.14 Legend, refresh and the "catching up" banner

Legend explains the cards (Asset, ETL job, Dataset, Group), direction, status, paths (Runtime observed, Derived from code) and columns (Mapped columns, Unknown mapping).

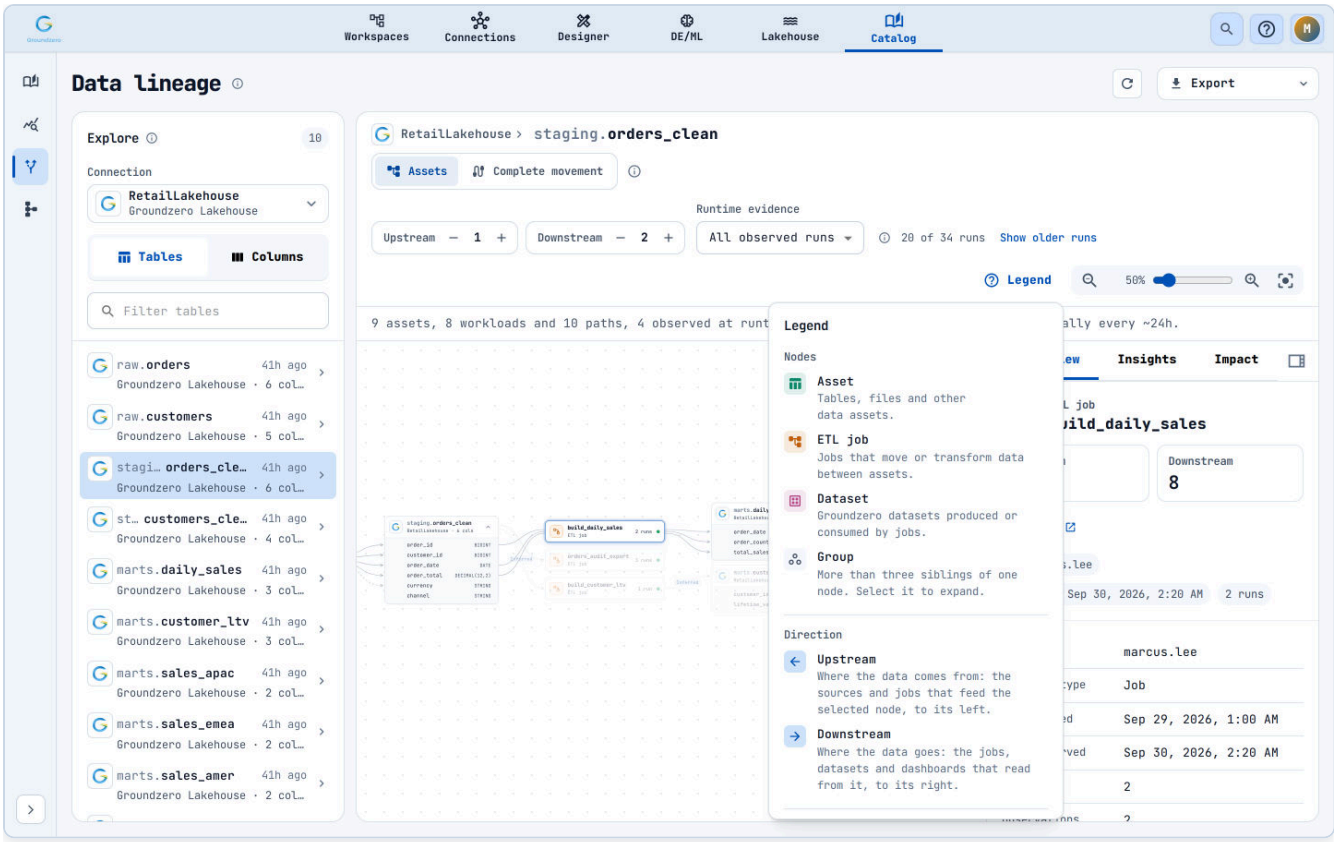


Figure 142. The legend explains cards, direction and line styles.

Refresh lineage (the circular arrow) reloads the connections, the table list and the current map.

If recent runs are still being processed, a banner appears:

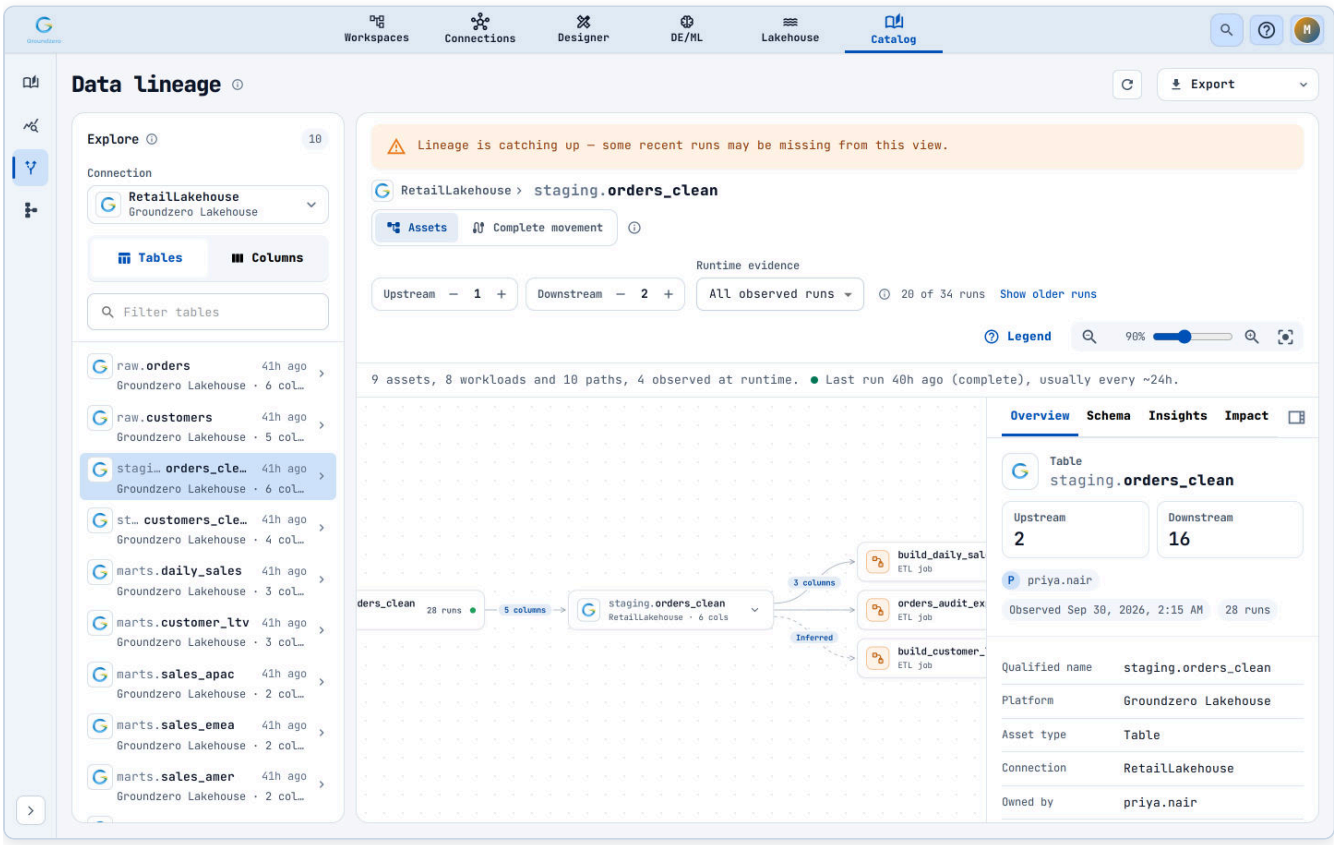


Figure 143. The "catching up" banner.

"Lineage is catching up — some recent runs may be missing from this view." Wait a little, then press **Refresh lineage**.

6.15 When the page shows a message

You see	What it means and what to do
"Choose a table to trace"	Nothing is selected. Pick a connection and a table.
"No lineage recorded for this table"	No job or dataset has been seen writing or reading it. Run one that uses it.
"No lineage compiled yet"	Run this job or dataset after connecting its source and target.
"Lineage compilation failed"	The code could not be analysed. Review the code and run it again.
"Lineage access is restricted"	Ask the owner for Describe access.
"This lineage scope is too large"	Lower the Upstream and Downstream steps, then **Try again**.
"Lineage is not available"	The item no longer exists.
"Lineage could not be loaded"	Press **Try again**.

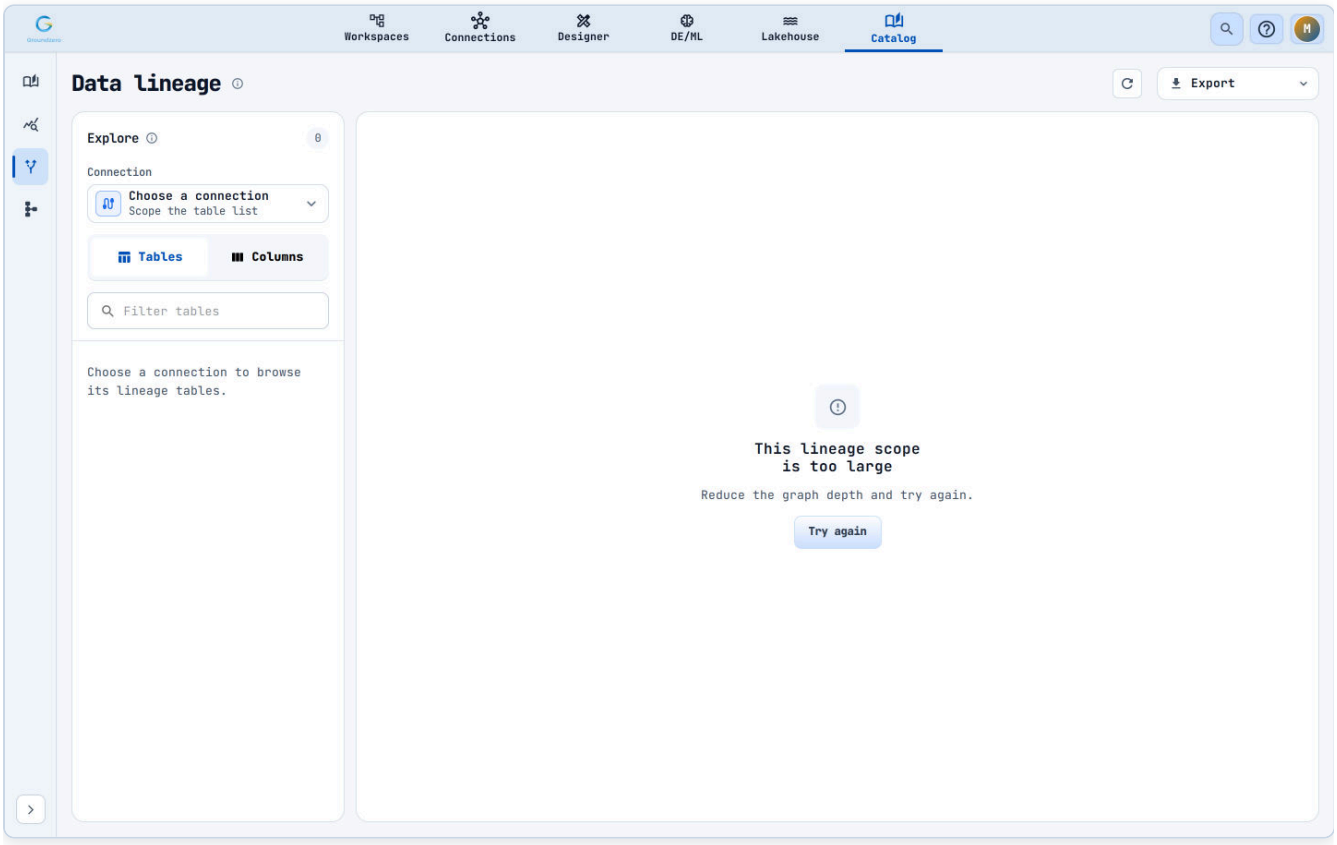


Figure 144. A scope that is too large offers Try again.

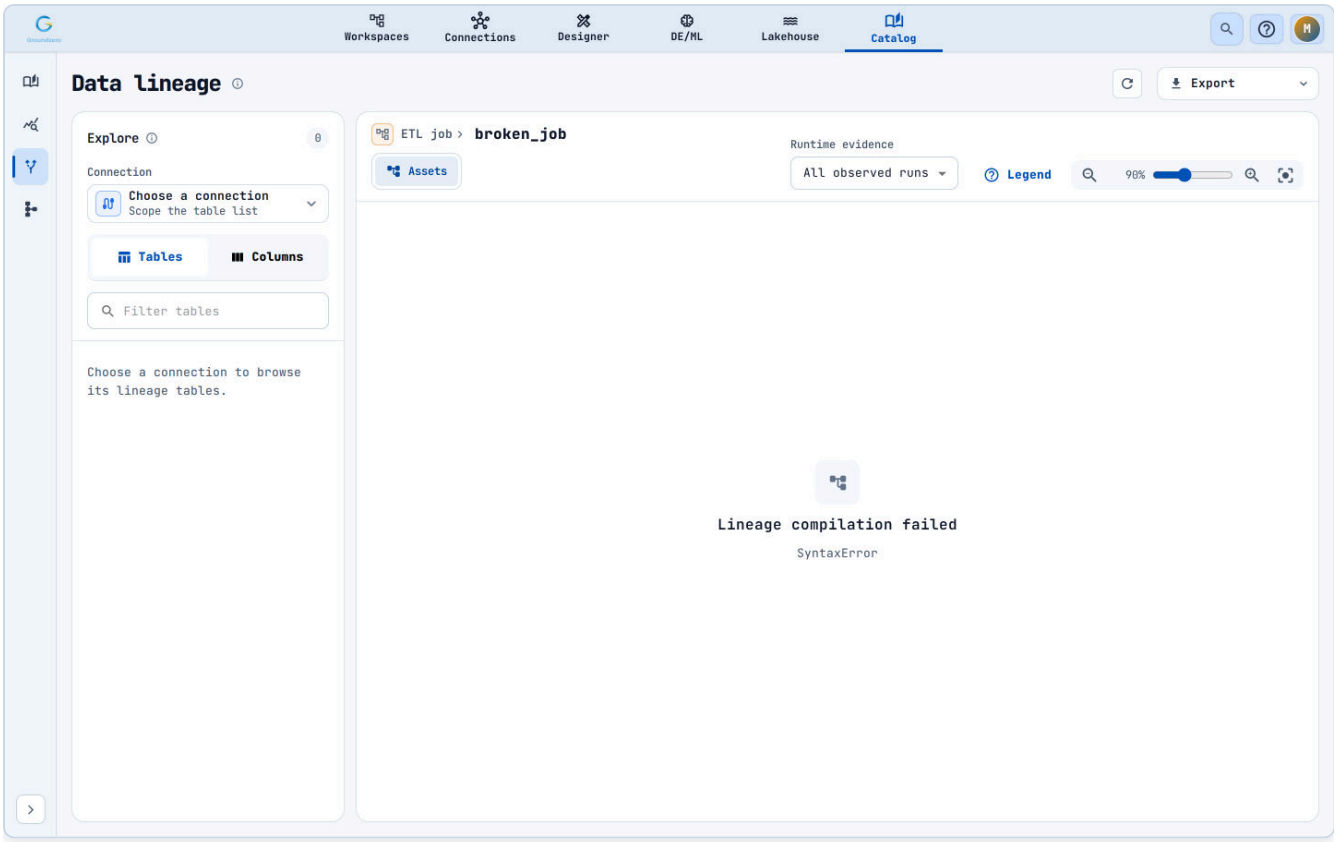


Figure 145. A workload whose analysis failed.

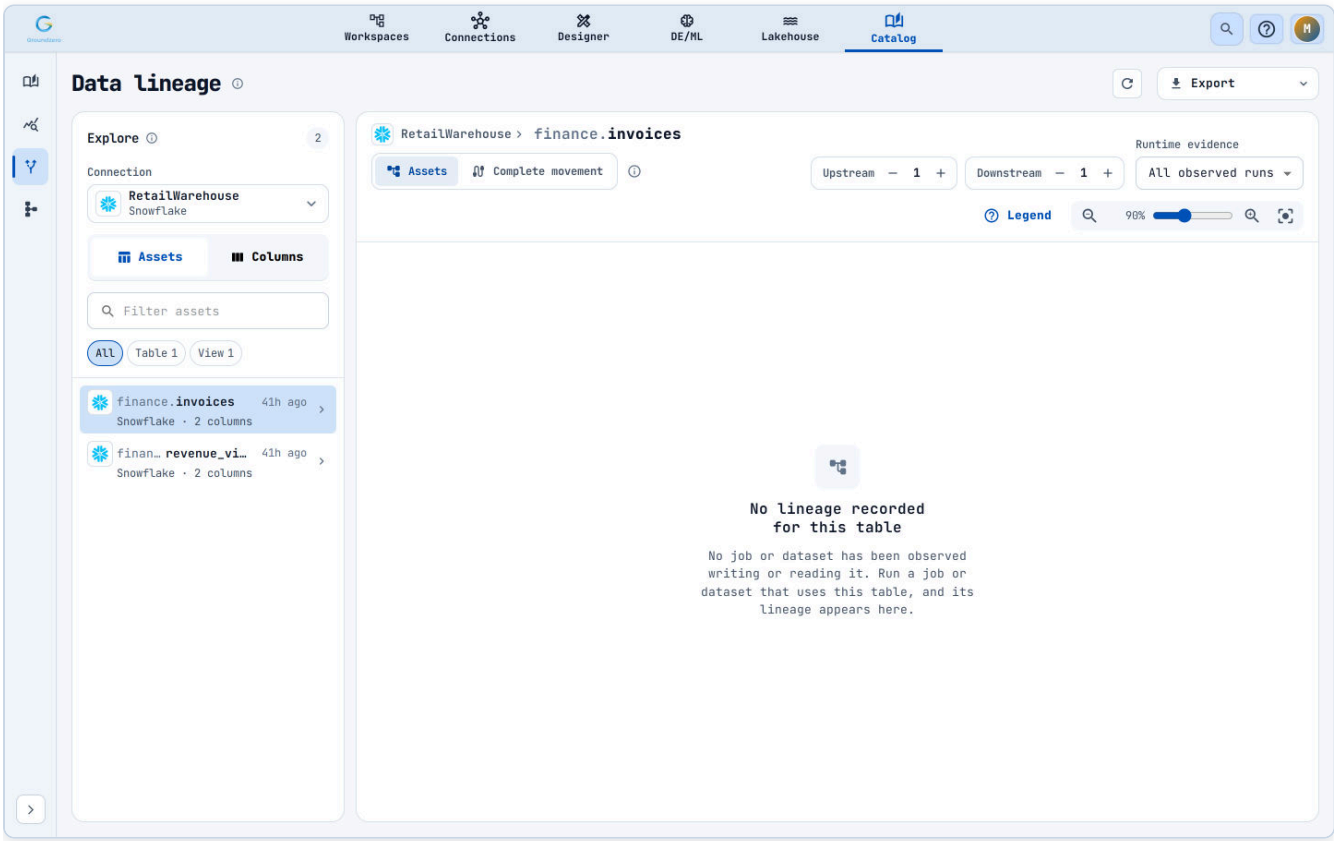


Figure 146. A table nothing has used yet.

6.16 Troubleshooting

Symptom	Cause	Fix
The explorer is empty	No connection chosen, or its type has no lineage	Pick a supported connection; run a job once
Lines are only dashed	Only the code was analysed	Run the job so a run confirms it
Inferred pills	Columns could not be worked out (for example `SELECT *`)	List the columns in the code
Grey placeholders everywhere	You lack Describe access	Ask the owner of the connection, job or dataset
"This lineage scope is too large"	Too many steps	Lower the Upstream and Downstream steps
An AI warning in Insights	AI is unavailable or you lack permission	The evidence cards still work
Recent runs are missing	Only 20 are loaded	Use **Show older runs**

6.17 Limits

Item	Limit
Steps each way	0 to 5 (default 1); impact depth 1 to 5 (default 3)
Cards drawn	250; groups form when more than 3 hang off one card
Runs loaded	20 by default, up to 100
Columns shown per table	15
Tables listed per page	50
Export	10,000 rows

6.18 Related

- **Data catalog**: browse the same tables with descriptions and tags.
- **Ontology Studio**: business definitions that appear in the Impact tab.
- **Lakehouse**: the warehouses and tables lineage tracks.
- **Designer and DE/ML** (Build documentation): the datasets and jobs that produce lineage when they run.

7

Ontology Studio

An **ontology** is a reviewed list of what your business words mean: what an "Order" is, how "Net revenue" is calculated, how customers relate to orders. **Ontology Studio** helps you build one from the tables and columns Groundzero already knows about through lineage. Groundzero proposes the entries, your team reviews and approves them, and you **publish** a numbered **release**. AI Chat and Agent Builder use published releases to understand your business; drafts never reach them.

Open **Catalog** in the top navigation bar, then **Ontology Studio** in the module menu.

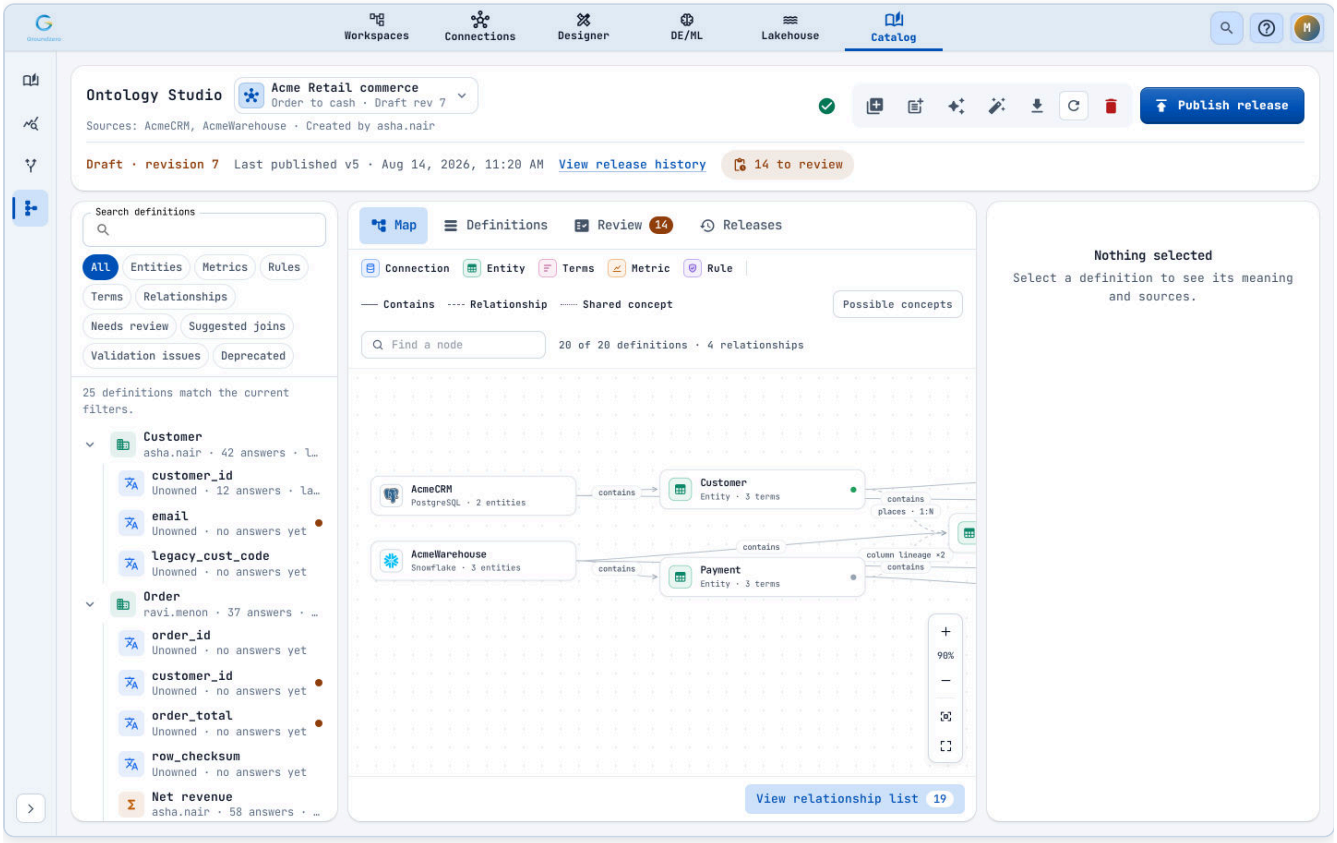


Figure 147. Ontology Studio: status bar, definition navigator, the map and the definition inspector.

I want to…	Go to
Start an ontology for a business area	"Create a model"
See how things connect	"See how things connect: the map"
Approve or reject entries	"Review definitions"
Let AI draft wording for me	"Draft with AI"
Make it available to AI Chat and agents	"Publish a release"
Change something already published	"Change a published model"

7.1 The words used here

Term	Meaning
Model	One ontology for a business area, built from the sources you choose
Definition	One entry in the model, with a business meaning, an owner, synonyms and optional guidance for AI
Entity	A table, such as Order or Customer
Term	A column of an entity, such as `order_total`
Metric and **Rule**	A calculated measure (Net revenue) or a business rule, with one SQL expression that defines it
Relationship	A link between two definitions with a business verb, such as "Customer places Order", and optionally its cardinality (for example 1:N) and join columns
Concept	A group of columns from different tables that share a name, such as `customer_id` on three tables
Release	A frozen, numbered copy of the approved definitions

7.1.1 Review status

Status	Meaning
Needs review	Proposed or newly written; nobody has decided yet
Approved	Accepted. Only approved definitions are published.
Rejected	Declined. Never published; stays in the draft.
Deprecated	Once approved, now retired. It needs a reason, leaves the next release and no longer guides AI answers.

A definition also records whether its source still exists. **Missing** means its table, column or relationship is no longer in lineage. An approved definition with a missing source blocks publishing (see "Fix a missing source").

7.1.2 Draft and published

Every change to a model creates a new revision. A model is **Published** only while nothing has changed since its last release. Any later change makes it **Draft** again ("Draft · revision N"). A published model is read-only; use **Create draft** to start editing again.

7.2 Permissions

You need permission on the Ontology resource (`All` , `Create` , `List` , `Read` , `Modify` or `Publish`) and permission on the model's source connections (`Describe` to see, `Edit` to change).

To do this	You need
See models	`List` or `Read` , and `Describe` on a source connection
Create a model	`Create` , and `Edit` on the source connections
Review, recalculate, curate with AI, delete a model	`Modify` (and `Edit` on the sources) while it is a draft

To do this	You need
Publish or delete a release	`Publish` (and `Edit` on the sources)
Draft with AI	The above, permission to use AI Chat, and AI usage budget left

A missing permission shows a message such as `You do not have "All" or "Modify" permissions on GZ Ontology.` With only read access you can look around; the page says "You can inspect this draft, but ontology modify permission is required to review definitions or recalculate candidates."

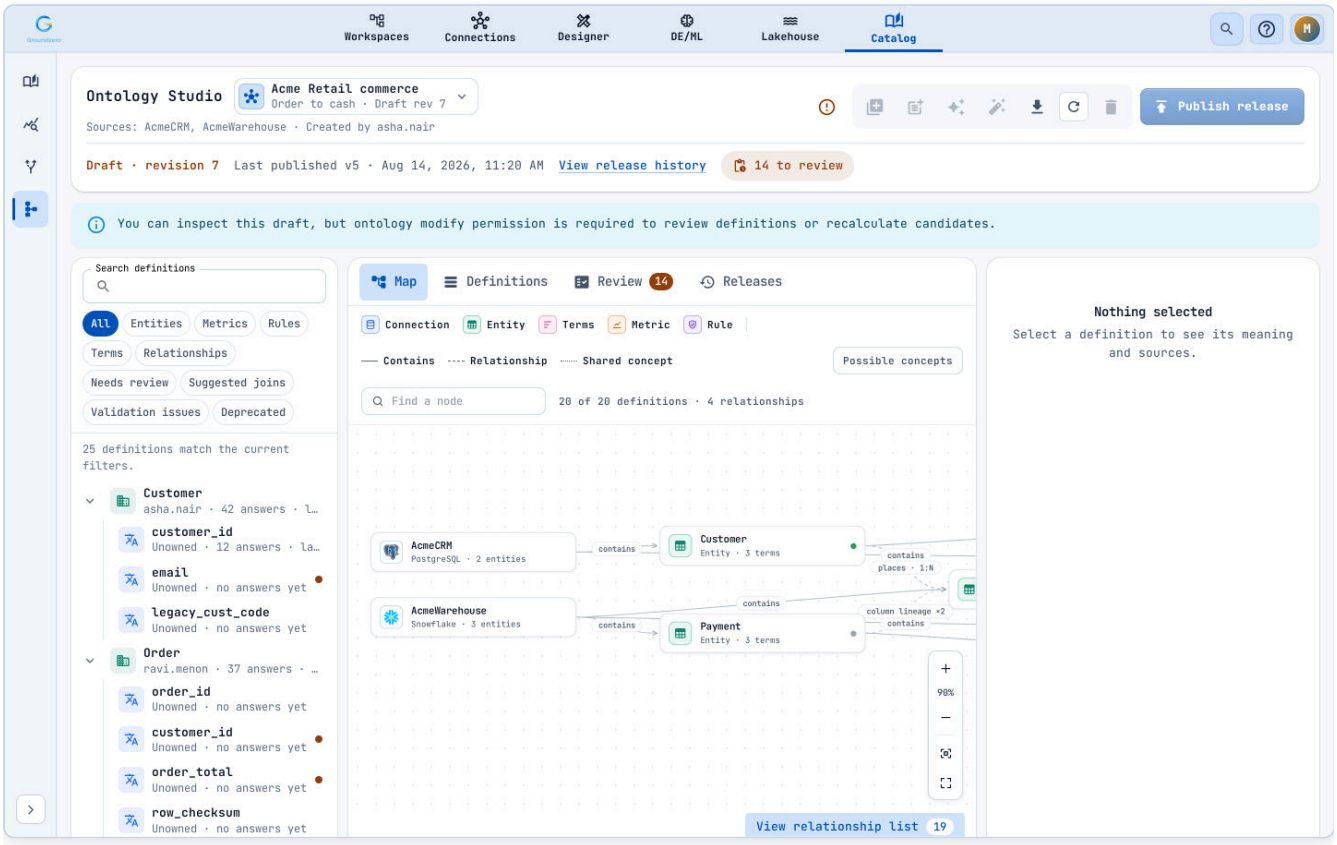


Figure 148. With read-only access, review and recalculate are unavailable and the page says why.

7.3 Find your way around

At the top is the status bar. Use the **Ontology model** picker to switch models (type in **Search models**; the footer has **Create model**).

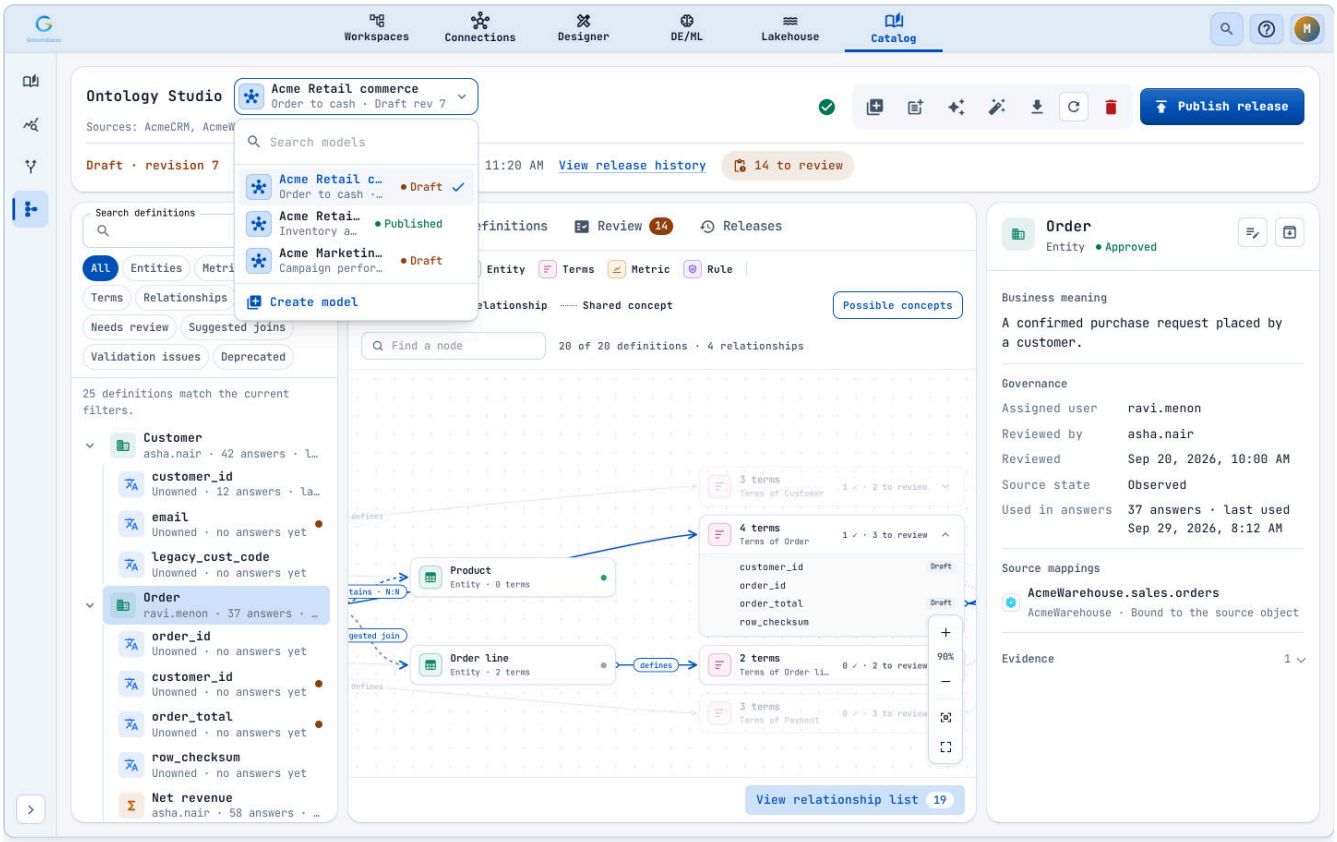


Figure 149. The model picker lists models with their domain, sources, status and last update.

The status bar also shows the sources and creator, whether the model is **Published · v9** or **Draft · revision 7**, when it was last published, **View release history**, a button such as **14 to review**, and any lineage drift notice. On the right are a readiness icon, the toolbar and **Publish release**.

The four tabs are **Map**, **Definitions**, **Review** and **Releases**. The left navigator lists definitions as a tree.

Toolbar button	Use it to
Create ontology model	Start a new model
New definition	Add a term, metric, rule or relationship
Curate draft with AI	Have AI draft wording for items waiting for review
Recalculate candidates	Pick up new tables and columns (reads **Create draft** on a published model)
Export audit pack	Download the model
Refresh ontology	Reload
Delete ontology model	Delete the model

A button that is off still shows why in its tooltip. If a refresh fails, the page keeps what it loaded and says "Showing the last loaded ontology because the latest refresh failed."

7.4 Create a model

Outcome: a model with a first set of proposed definitions, ready to review. Lineage must already exist for your tables, so run the jobs that use them first.

1. Click **Create ontology model** (or **Create model** in the picker, or **Create first model** if there are none).
2. Fill **Model name** and **Business domain**, both required, for example "Order to cash".

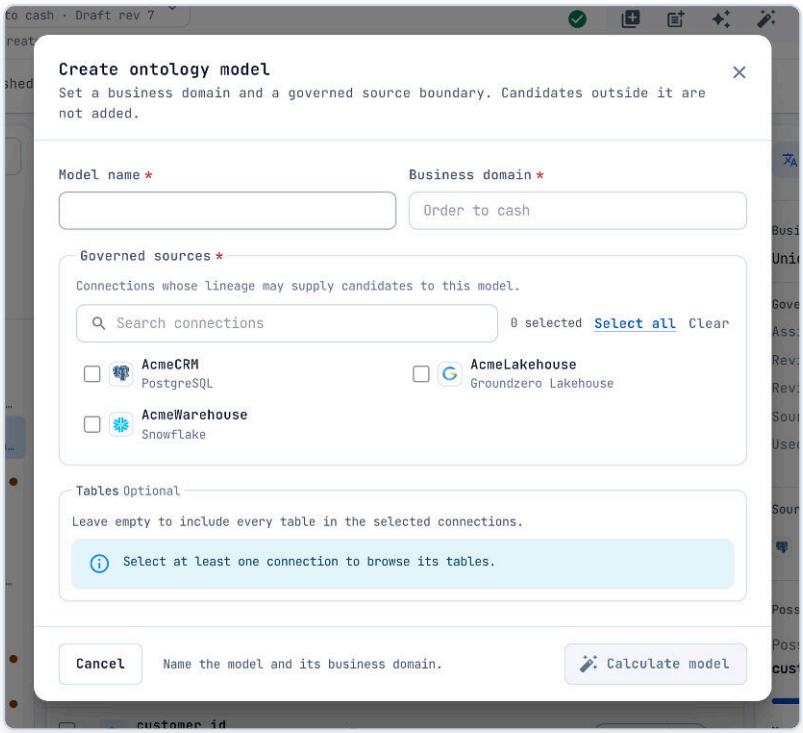


Figure 150. The Create ontology model dialog.

3. Under **Governed sources**, tick at least one connection. **Search connections**, **Select all** and **Clear** help. Only connections with tables in lineage are listed.
4. Optionally narrow the **Tables**. Leave it empty to include every table of the chosen connections. Search to find a table.

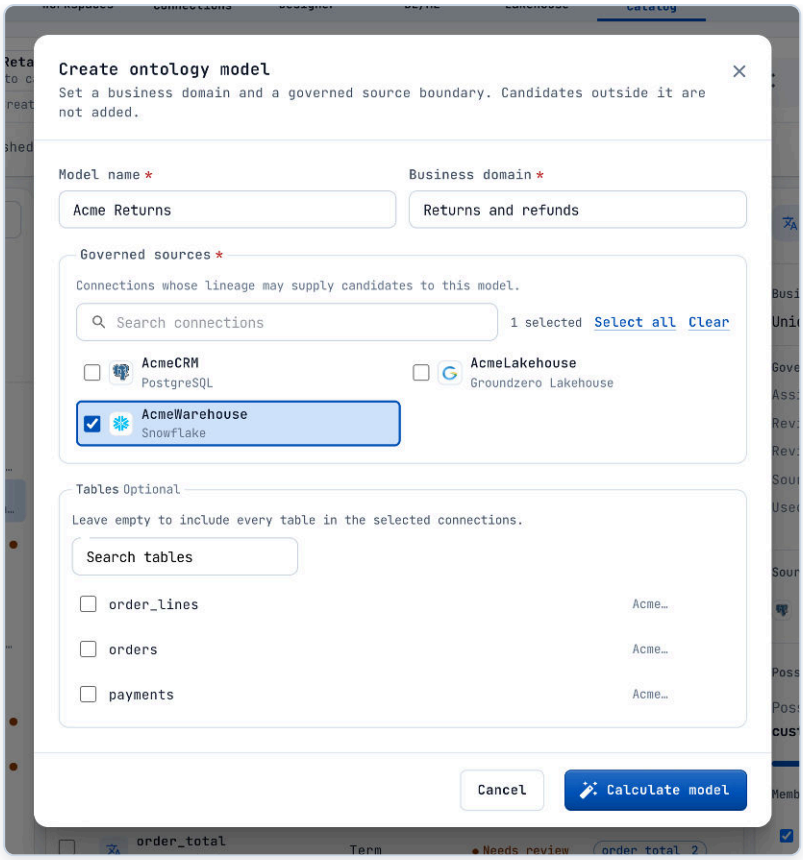


Figure 151. A connection and an optional table scope chosen.

5. Click **Calculate model**. It stays off until name, domain and a source are set.

You see "Scoped ontology model calculated and ready for review." and the model opens on **Definitions** with every item marked **Needs review**.

If you see "No governed lineage sources match the requested scope", nothing is recorded in lineage for your choice. Run the jobs that produce lineage first.

7.4.1 Delete a model

Click **Delete ontology model** and type the model name to confirm. This removes the draft, its review history, its releases and the saved AI answers that cited them.

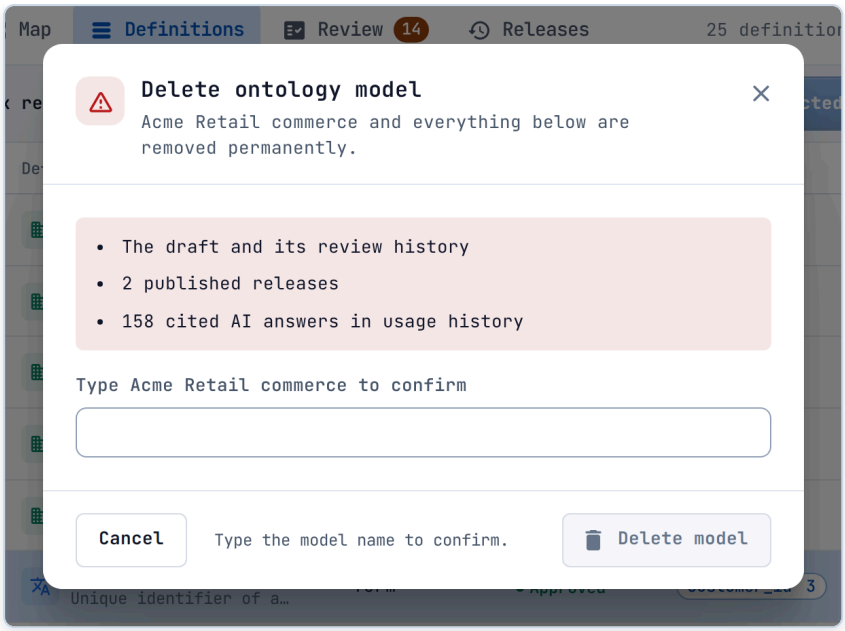


Figure 152. Deleting a model is permanent and needs the name typed.

You see "Ontology model deleted."

7.5 See how things connect: the map

The **Map** tab shows connections, entities, their terms, and metrics and rules, from left to right.

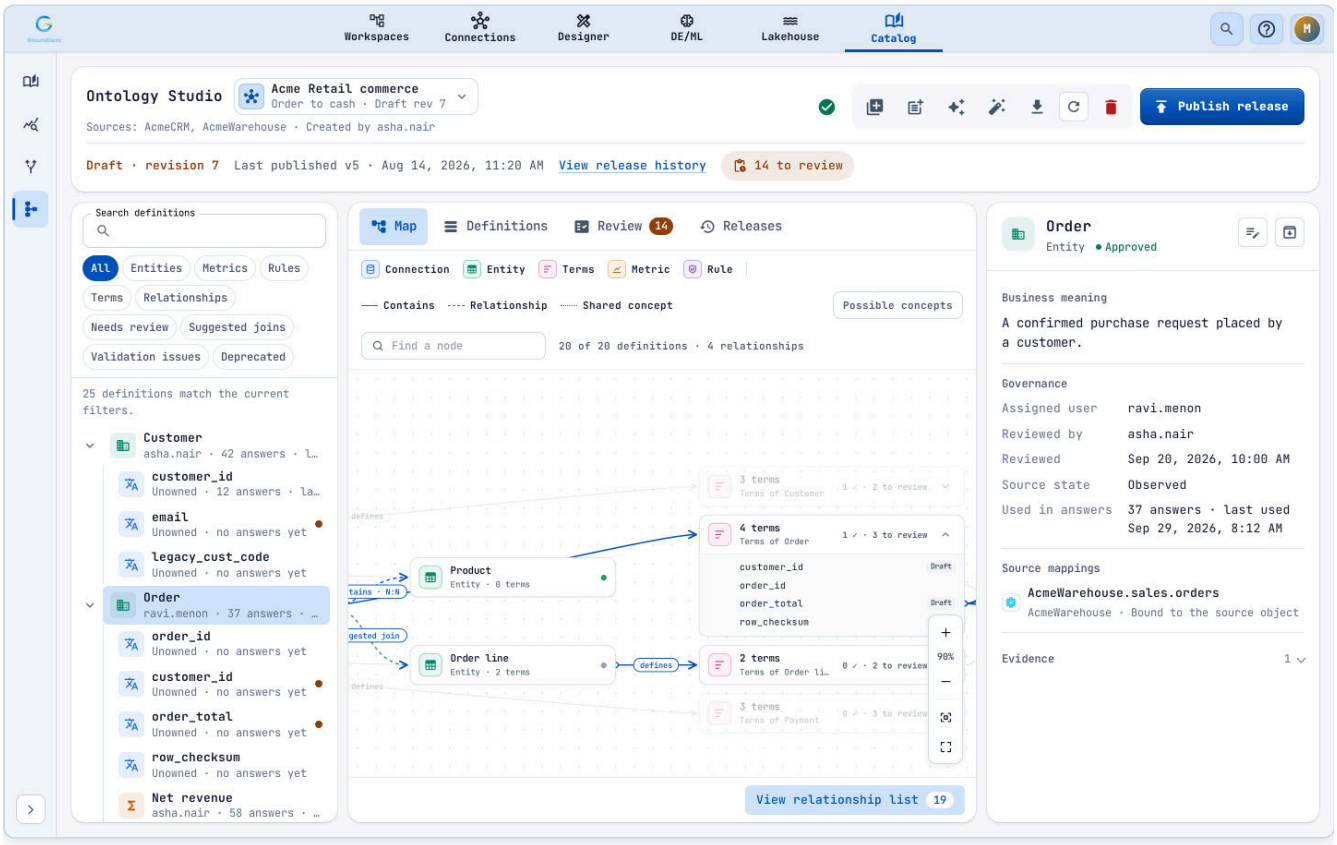


Figure 153. A selected entity highlights its relationships; its terms card is open.

- A solid line means **Contains**, a dashed line **Relationship**, a dotted line **Shared concept**. Verbs such as "places · 1:N" show when you zoom in.
- **Possible concepts** draws dotted arcs between columns that could form a shared concept.

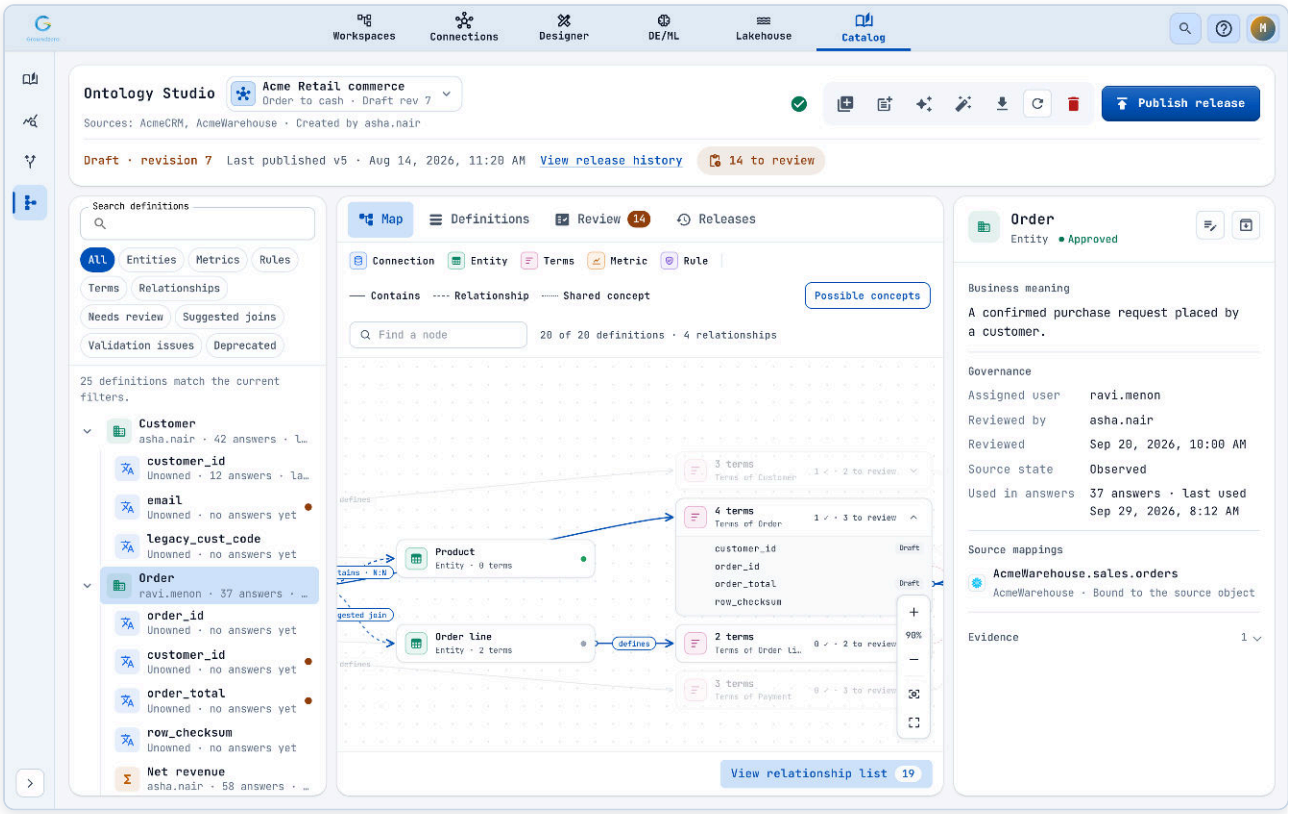


Figure 154. Possible concepts shows candidate shared-concept arcs.

- **Find a node** searches by name. The count reads, for example, "20 of 20 definitions · 4 relationships".
- Click a card to select it; click empty space or press Esc to clear. Drag to move cards or pan. **Zoom in**, **Zoom out** and **Fit to view** change the zoom.
- **Expand map** opens a full-screen map.

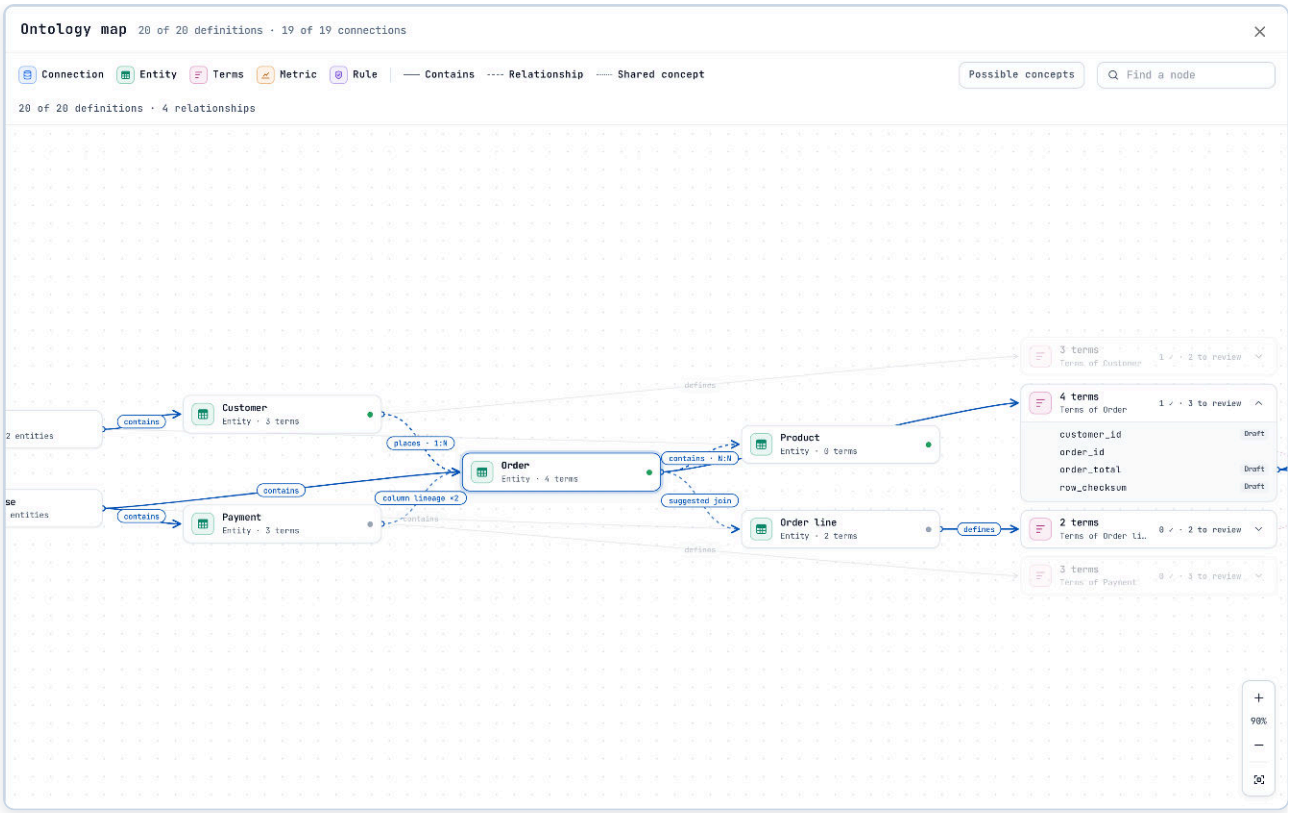


Figure 155. The full-screen map always shows the whole model.

- In a very large model, selecting a card shows it and its neighbours only ("Showing the selection and its direct neighbours."). **Show overview** returns to everything.
- **View relationship list** opens a searchable list of every relationship: source, verb, target and cardinality.

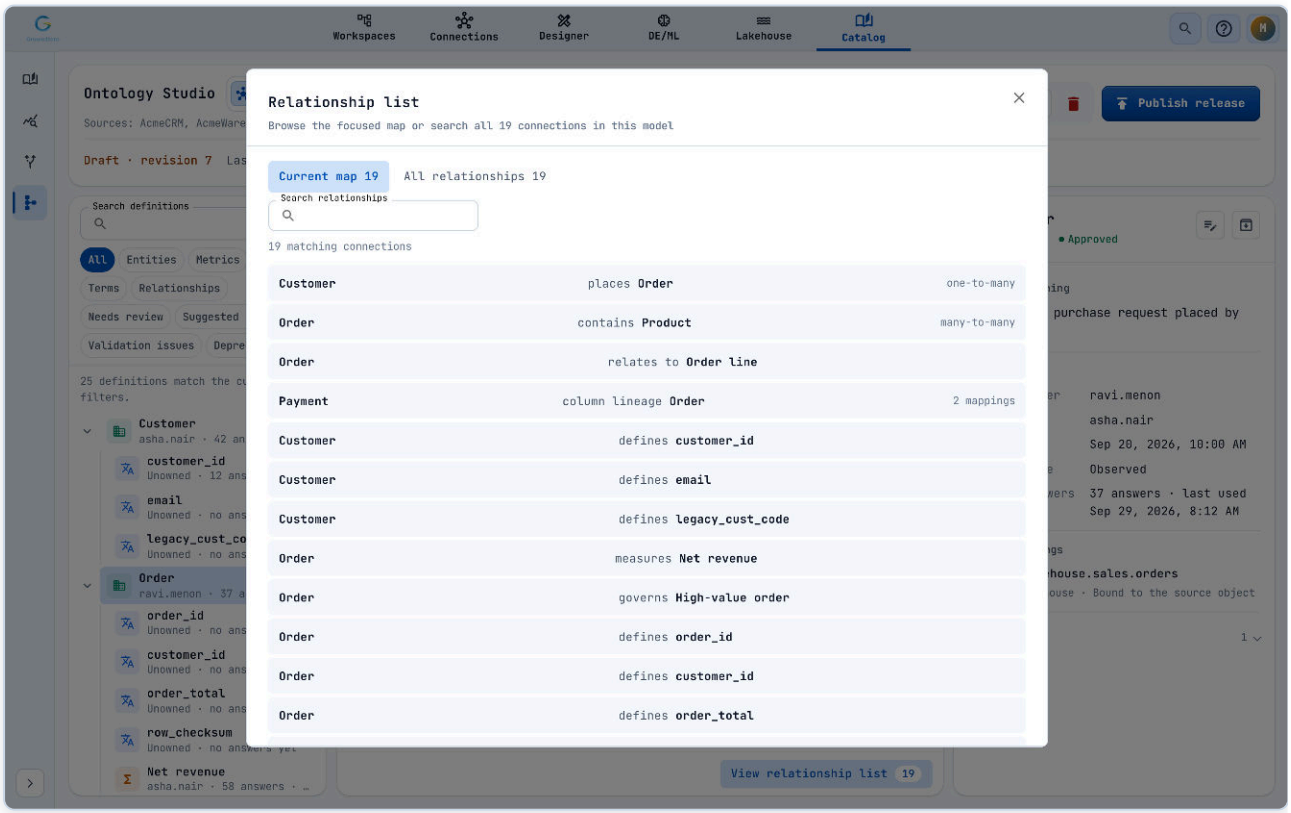


Figure 156. The relationship list: each row reads source, verb, target and cardinality.

Relationship verbs include defines, measures, governs, references, joins on, known as, describes and derives from. You can also write your own (up to 160 characters).

7.6 Browse and find definitions

The **Definitions** tab lists everything in the model.

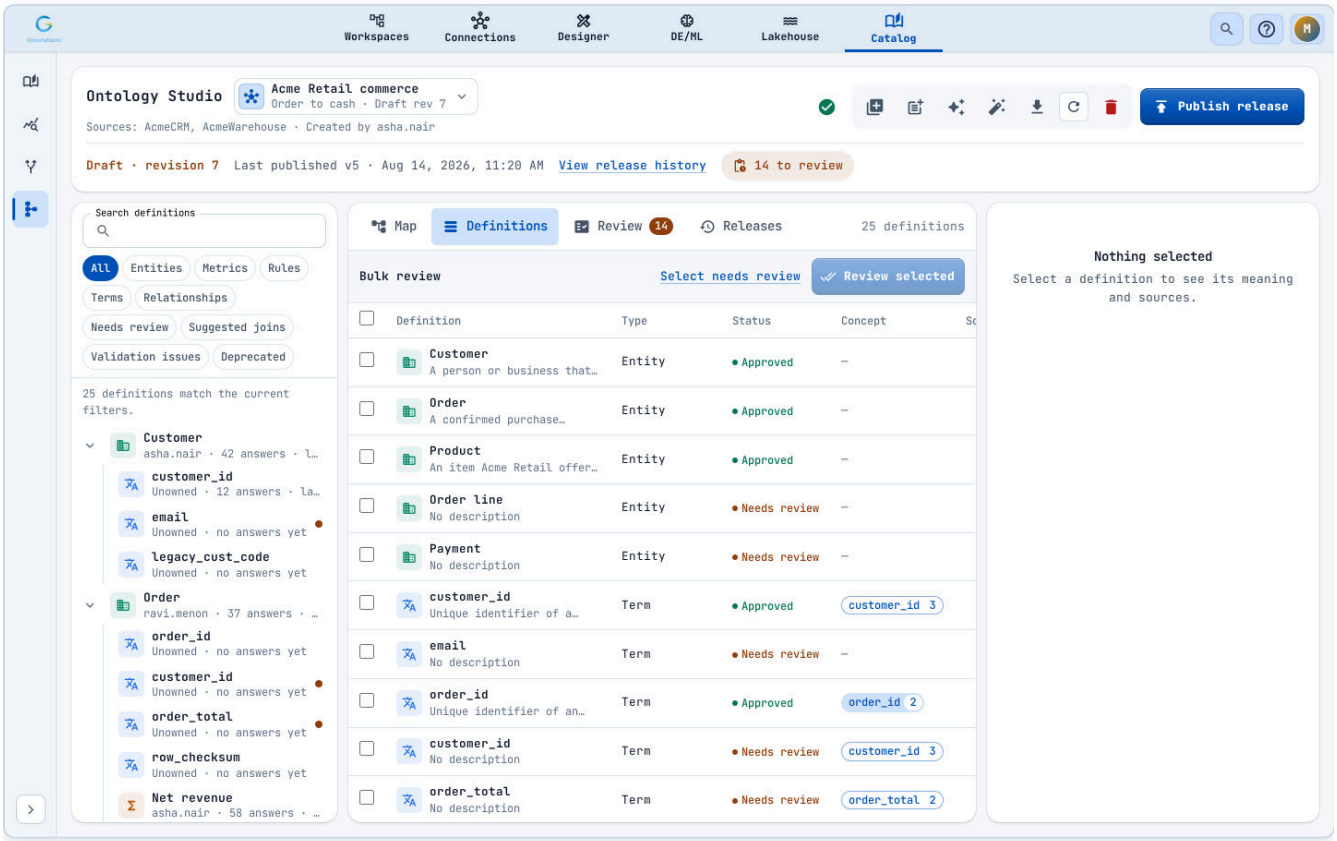


Figure 157. The Definitions table with bulk review controls and type, status and concept columns.

Use **Search definitions** (name, description, owner and synonyms) and the chips **All**, **Entities**, **Metrics**, **Rules**, **Terms**, **Relationships**, **Needs review**, **Suggested joins**, **Validation issues** and **Deprecated**. The text "<n> definitions match the current filters." shows the count. An amber dot marks items that need review. Only the first 250 matches are listed; refine your search to see others.

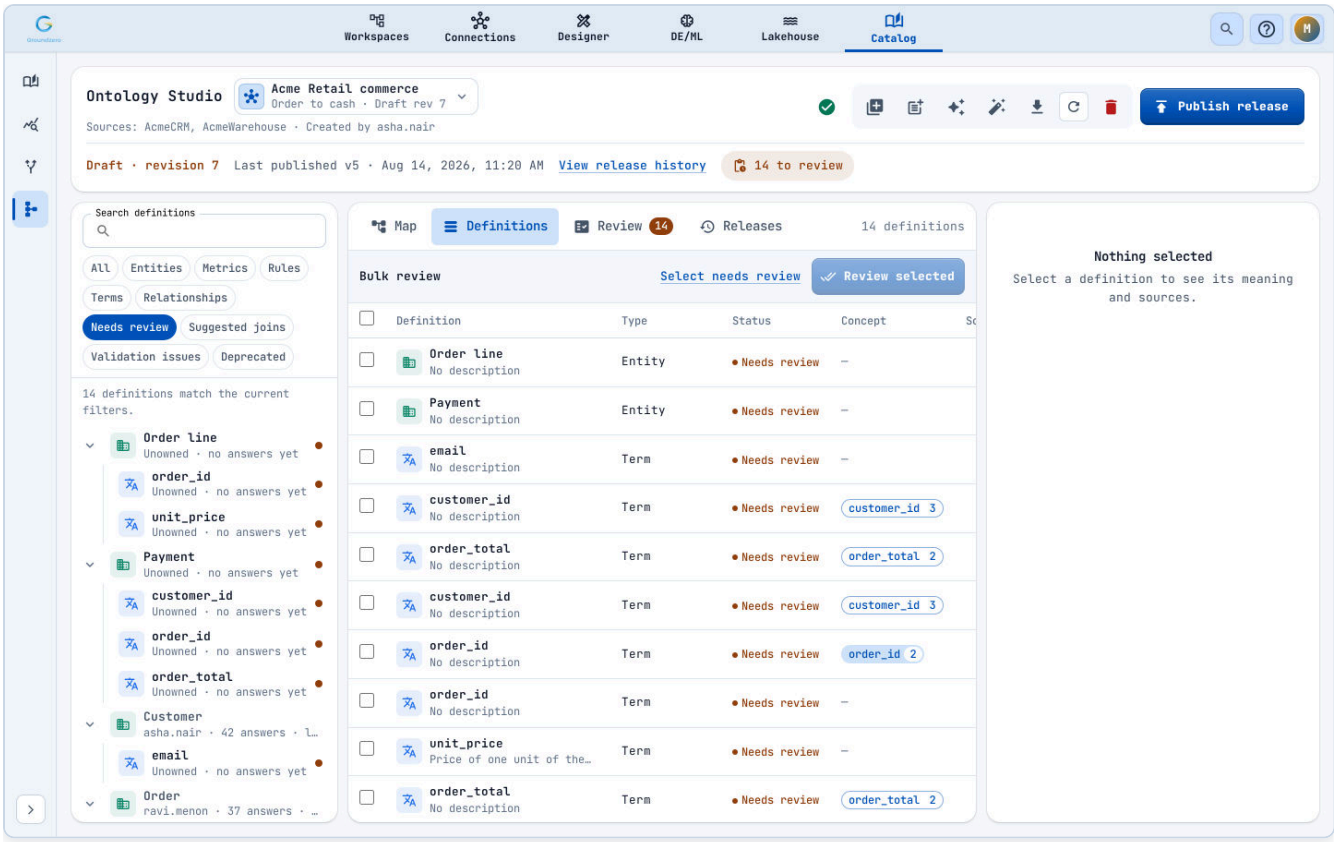


Figure 158. The Needs review filter lists the pending items.

Click a definition to see it in the inspector on the right.

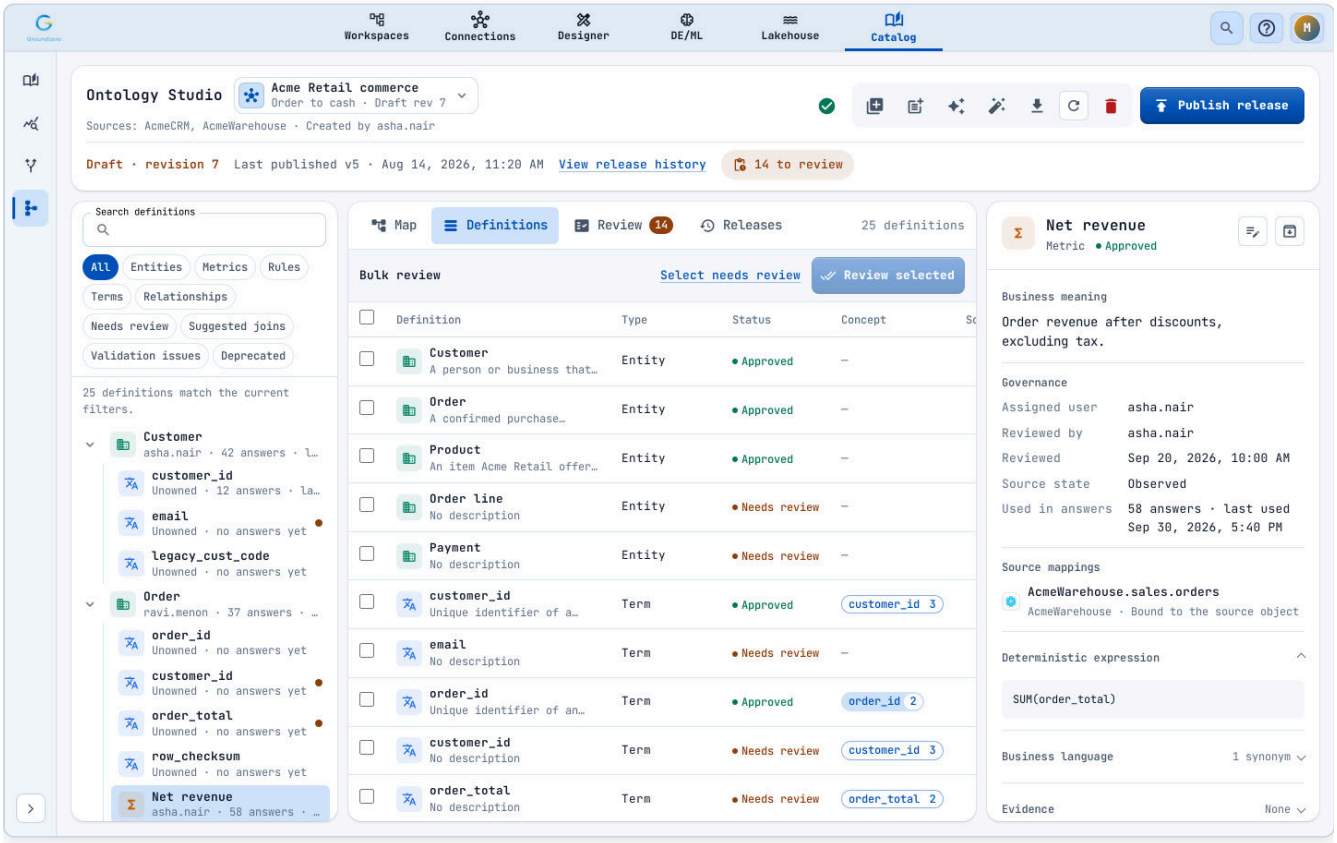


Figure 159. A metric: business meaning, governance, source mappings and its deterministic expression.

The inspector shows its **Business meaning**, **Governance** (assigned user, reviewer, confidence, source state, use in answers), **Source mappings**, the **Relationship** (from, verb, to, cardinality, join columns), **Deterministic expression** for metrics and rules, **Shared concept**, **Business language** (synonyms), **Evidence** and **AI guidance**.

7.7 Review definitions

Outcome: each definition is approved, rejected or still waiting, with a clear meaning and an owner.

7.7.1 Review one definition

1. Select it and click **Review definition**.
2. Change only what you need.

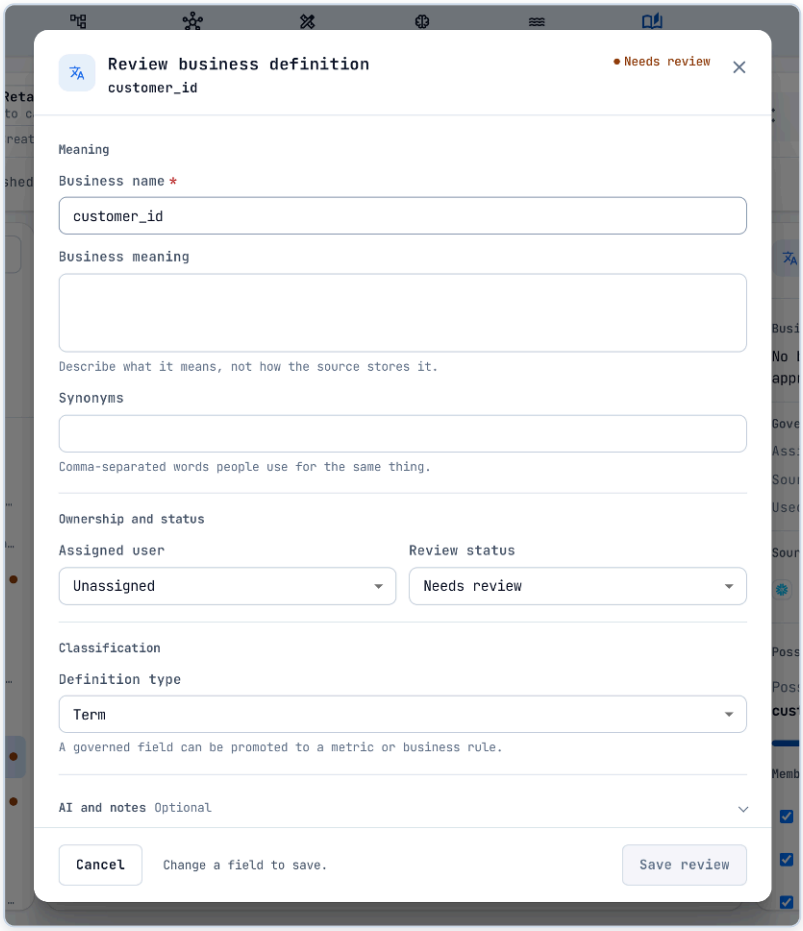


Figure 160. The Review business definition dialog.

Field	What to enter
Business name	Required, up to 200 characters
Business meaning	Up to 4,000 characters. Required to approve ("Required before approval.")
Synonyms	Comma separated; up to 20
Business verb, **Cardinality**, **Source column**, **Target column**	Relationships only
Assigned user	The person responsible for it

Field	What to enter
Review status	Needs review, Approved or Rejected
Definition type	A term can become a metric or a rule; a metric and a rule can swap
Deterministic expression	Metrics and rules only. Required to approve. One valid SQL expression (sub-queries are refused). Up to 8,000 characters
AI usage guidance	Up to 2,000 characters. It guides AI; data access is still enforced by permissions.
Review notes	Up to 4,000 characters

3. Click **Save review** (it turns on once you change something). You see "Business definition review saved."

7.7.2 Review many at once

1. On **Definitions**, click **Select needs review** (up to 1,000 items) or tick rows.
2. Click **Review selected**. Choose a **Review decision** (Keep current status, Needs review, Approved or Rejected), an **Assigned user** and **Review notes**.

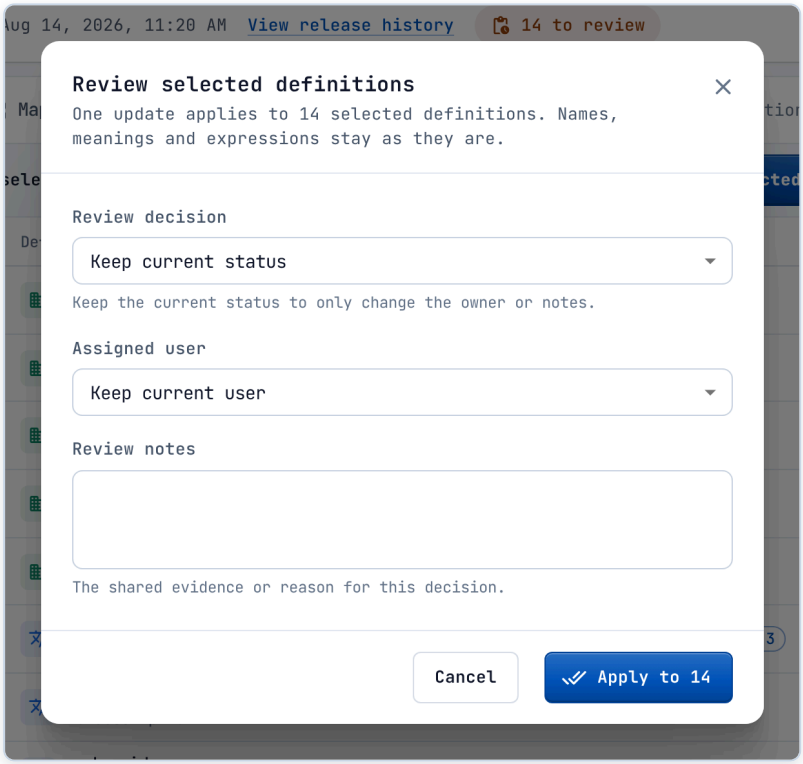


Figure 161. Bulk review applies one decision to many definitions.

All items change together or none do. Approving items that have no meaning yet is refused with a count.

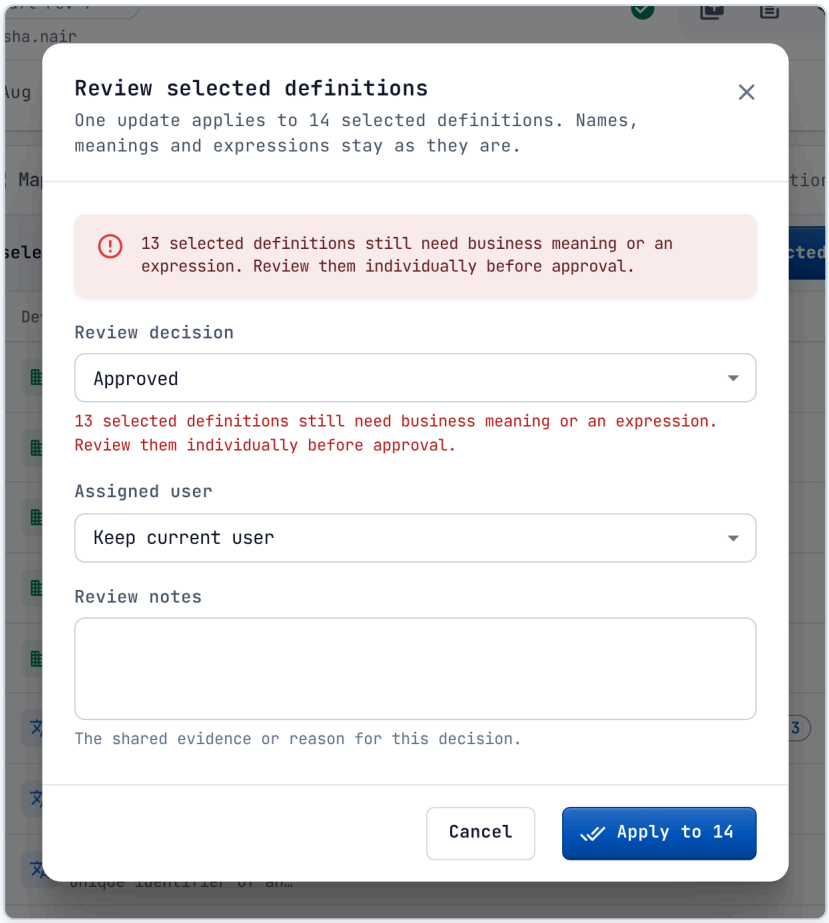


Figure 162. Approving items that have no meaning yet is refused with a count.

7.7.3 Work through the review queue

The **Review** tab takes you through waiting items one by one. A progress bar reads, for example, "11 of 25 reviewed".

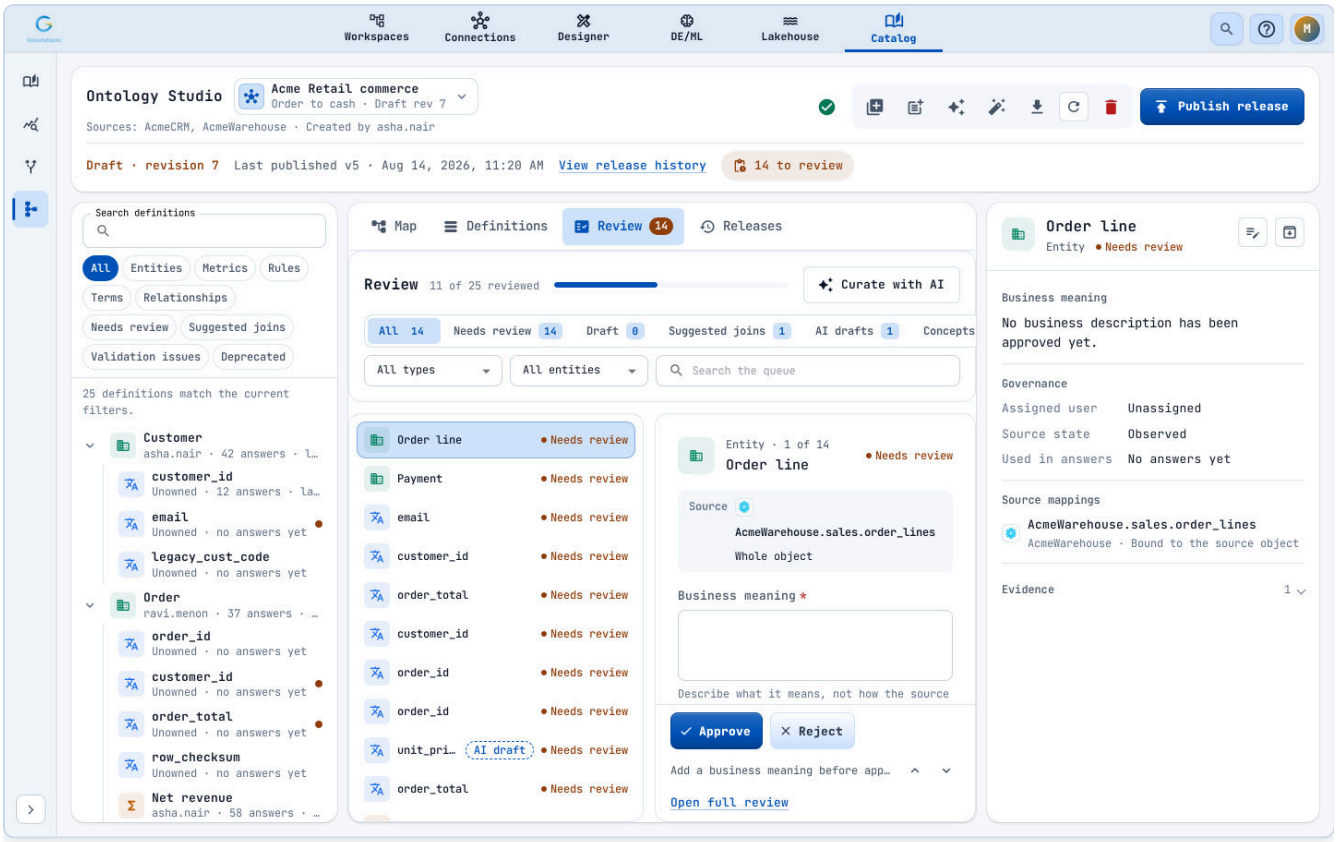


Figure 163. The review queue: a list on the left and the review card with inline fields and a decision bar.

1. Choose a filter: **All**, **Needs review**, **Draft**, **Suggested joins**, **AI drafts** or **Concepts**. You can also filter by **Definition type** and **Entity**, or **Search the queue**.
2. On the review card, fill in **Business meaning**, **Deterministic expression** (metrics and rules), **Synonyms** (press Enter after each) and **AI usage guidance**.
3. Click **Approve** or **Reject**. **Previous** and **Next** move along. **Open full review** opens the full dialog.

Without a meaning, approving shows "Add a business meaning before approving."

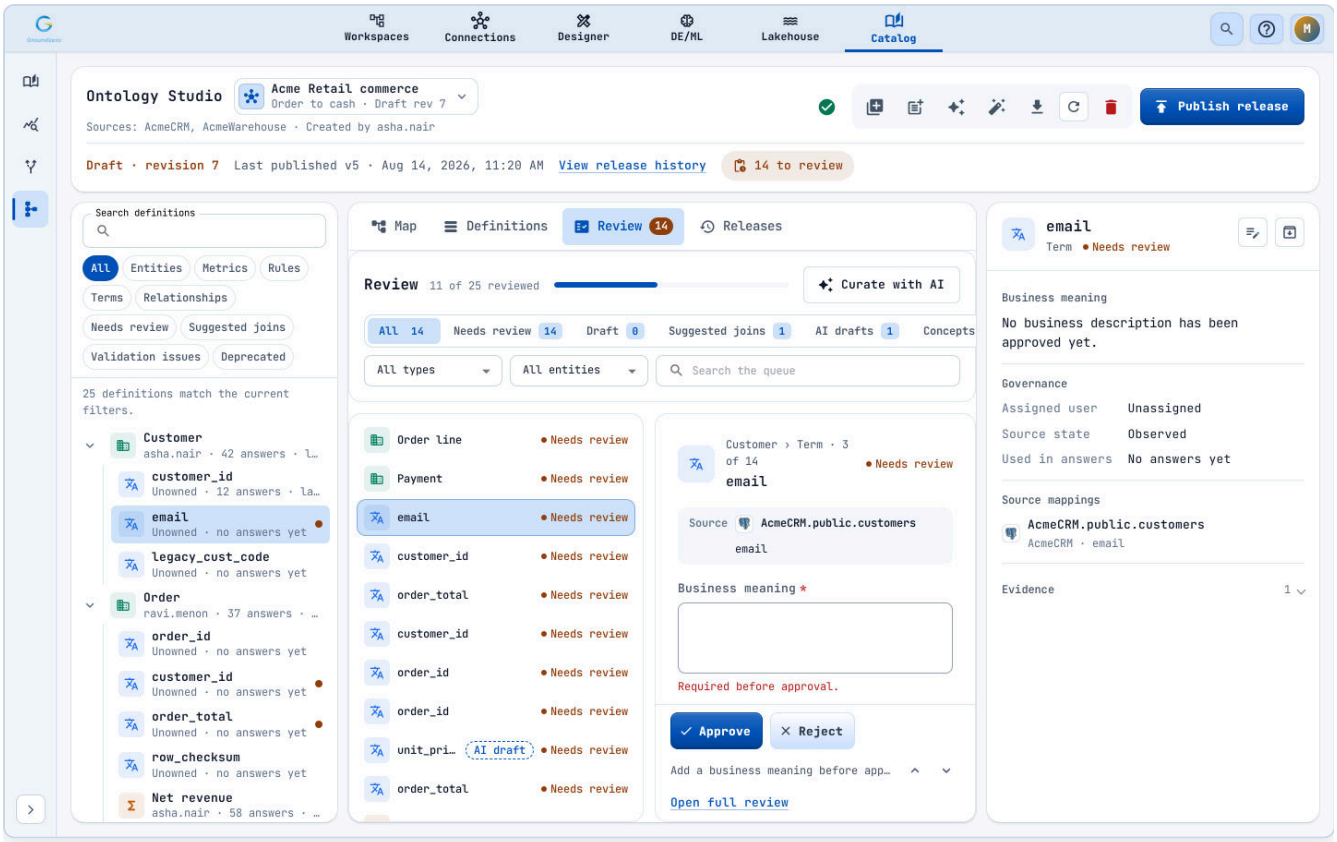


Figure 164. Approve is blocked until the business meaning is filled in.

Keyboard shortcuts work while you are not typing in a field: A approves, R rejects, E opens the full review, J or Down goes to the next item, K or Up to the previous. Your unsaved edits are kept while you move through the queue, but not after a reload.

Items drafted by AI carry an **AI draft** marker. Use the **AI drafts** filter to check AI-written wording first.

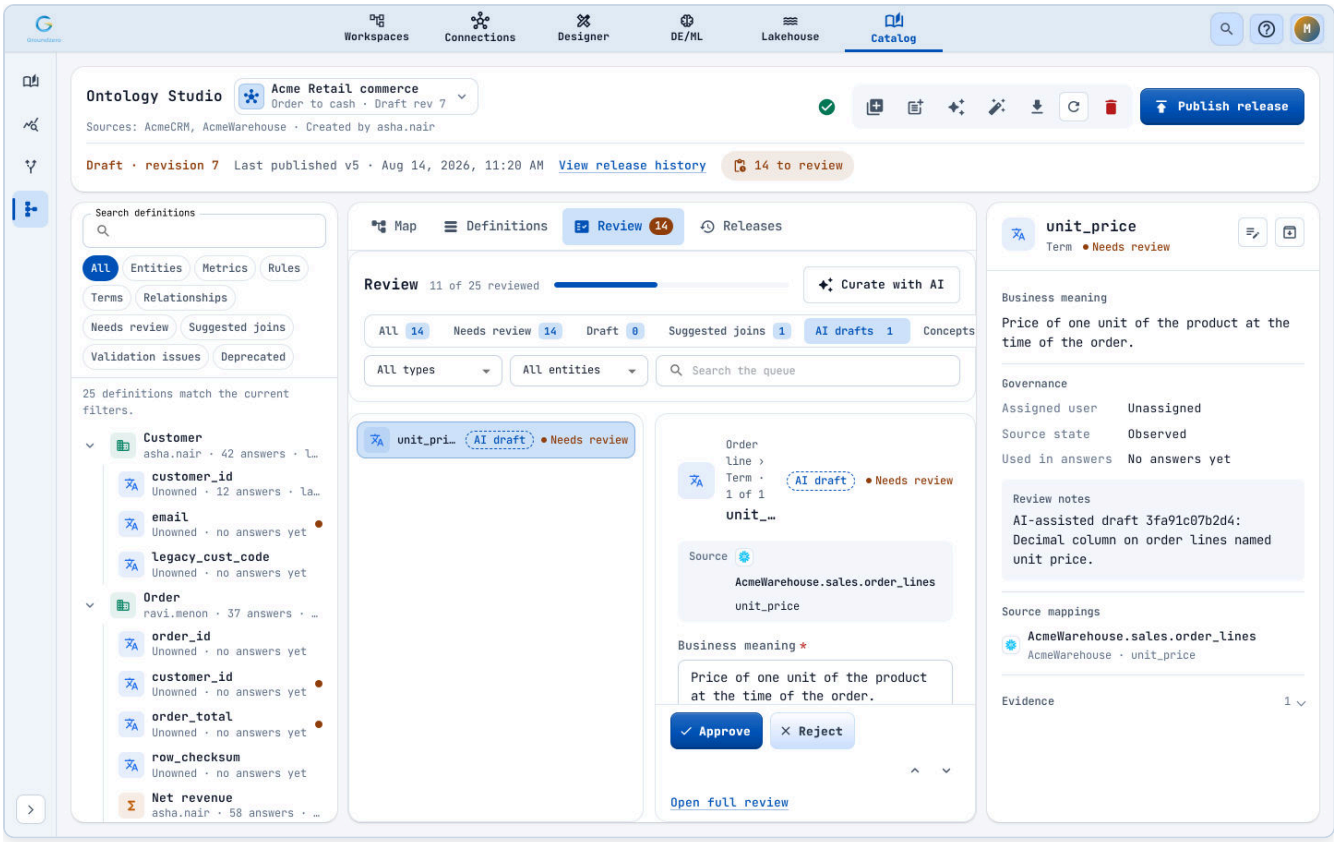


Figure 165. The AI drafts filter lists the items AI has worded, marked AI draft.

7.7.4 Add a definition yourself

Click **New definition** to write a term, metric, rule or relationship on an approved entity.

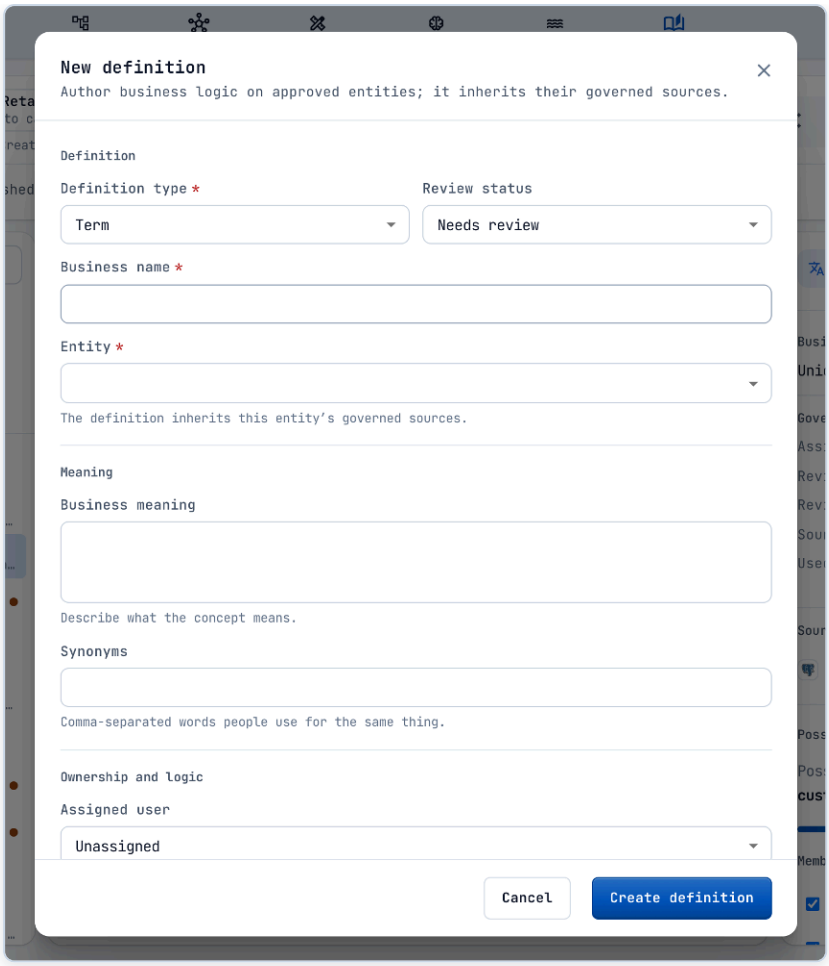


Figure 166. The New definition dialog.

Give it a type and a business name, and choose its **Entity**. To approve it straight away you also need a business meaning (and an expression for a metric or rule). A relationship needs **From definition** and **To definition**; if you leave the verb empty, Groundzero suggests one, for example "references" for entity to entity.

7.7.5 Retire a definition

Click **Deprecate definition**, enter a **Reason** (required) and, if there is one, a **Successor**. You see "Business definition deprecated." **Restore definition** brings it back as Approved.

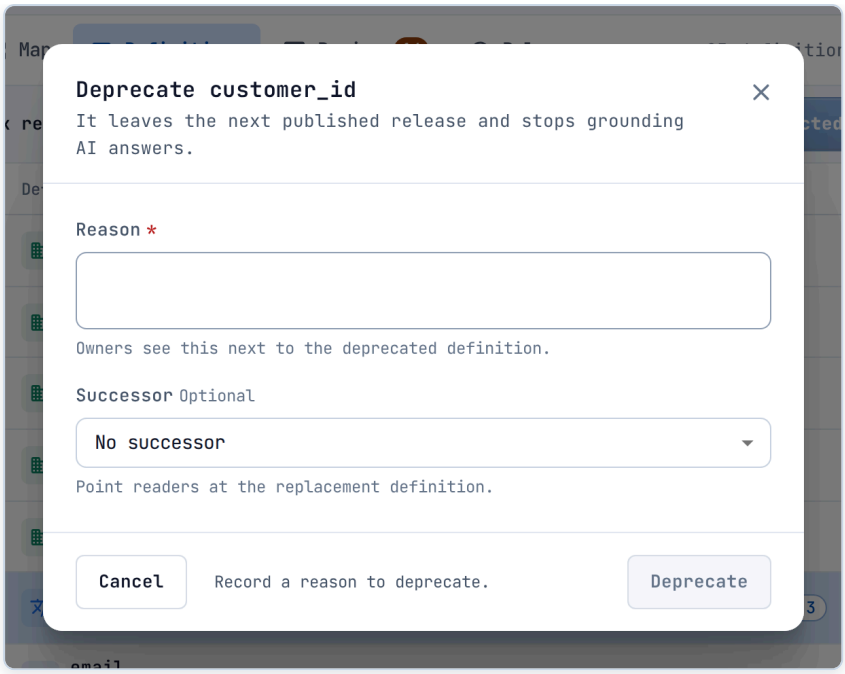


Figure 167. The Deprecate dialog.

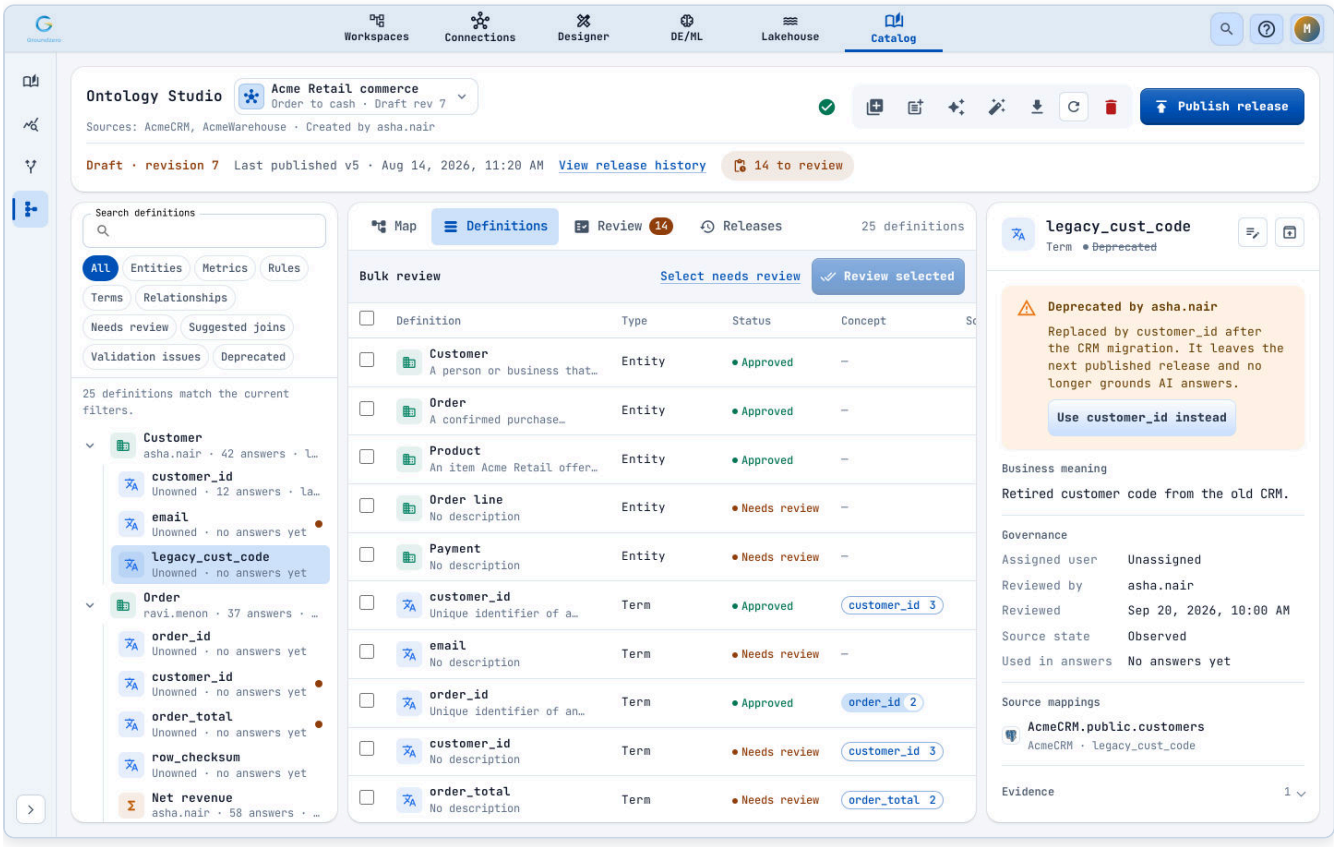


Figure 168. A deprecated definition shows who retired it, why and its successor.

7.7.6 Confirm shared concepts

Columns from different tables whose names match once capital letters and punctuation are ignored (`customer_id`, `customerId`, `CUSTOMER_ID`) form a **candidate concept**. Very generic names such as `id` or `status` count only after a reviewer approves the term. Each concept has a confidence from 0 to 100 percent, based on matching names, compatible data types, lineage and joins. Only **confirmed** concepts reach a release.

1. In the **Review** tab choose **Concepts**.
2. Select a concept. Tick the columns that really mean the same thing. The detail shows **Members**, **Evidence** and a confidence meter. "Members have incompatible data types." warns of a mismatch.
3. Click **Confirm** (needs at least two kept members) or **Reject**. **Move unchecked to a new concept** splits off the rest.

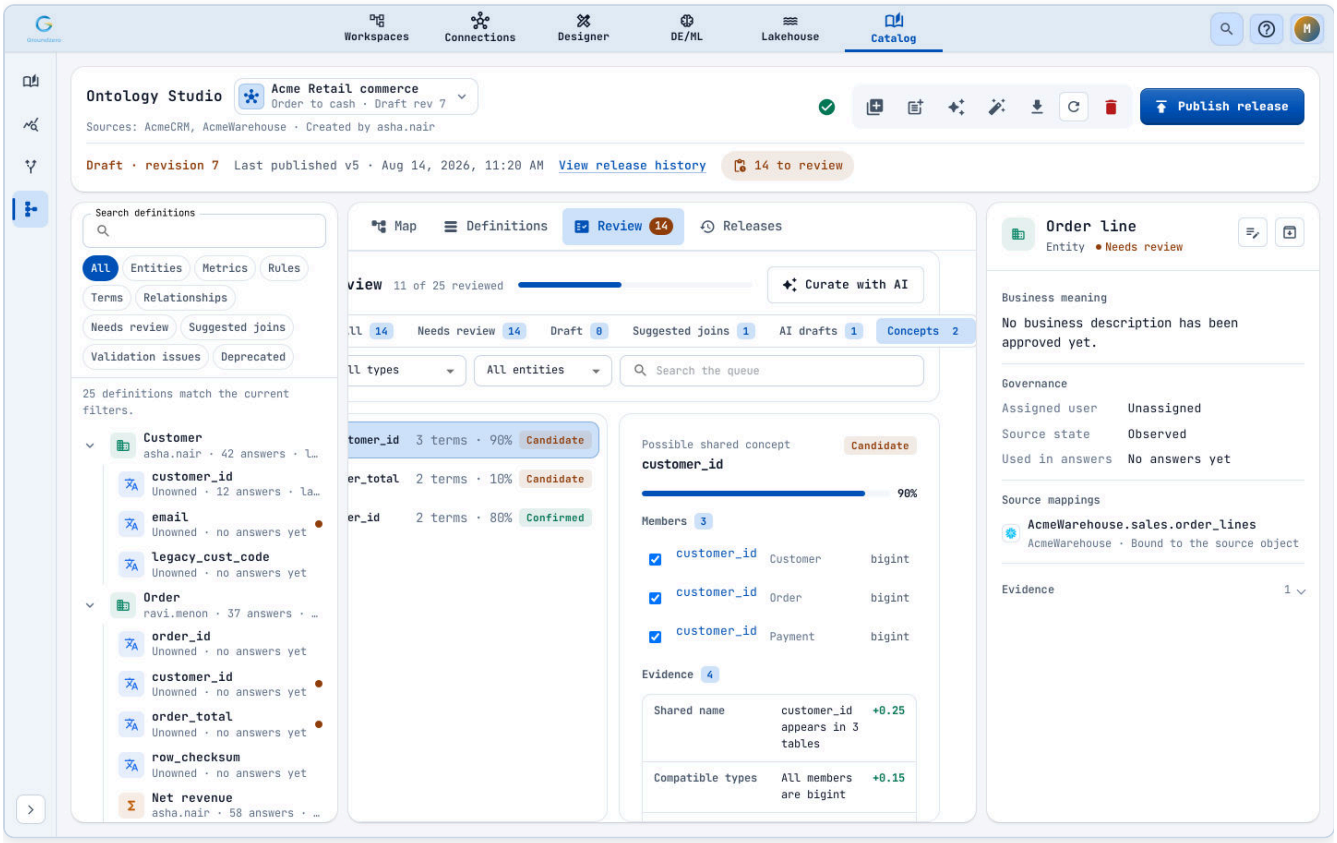


Figure 169. Concept review: members, evidence and Confirm, Reject or split actions.

A decided concept can be reopened with **Reopen**. Your decisions survive a recalculation.

7.7.7 Accept suggested joins

When a job reads two tables and they share a column, Groundzero suggests a join between their entities and marks it **Suggested**.

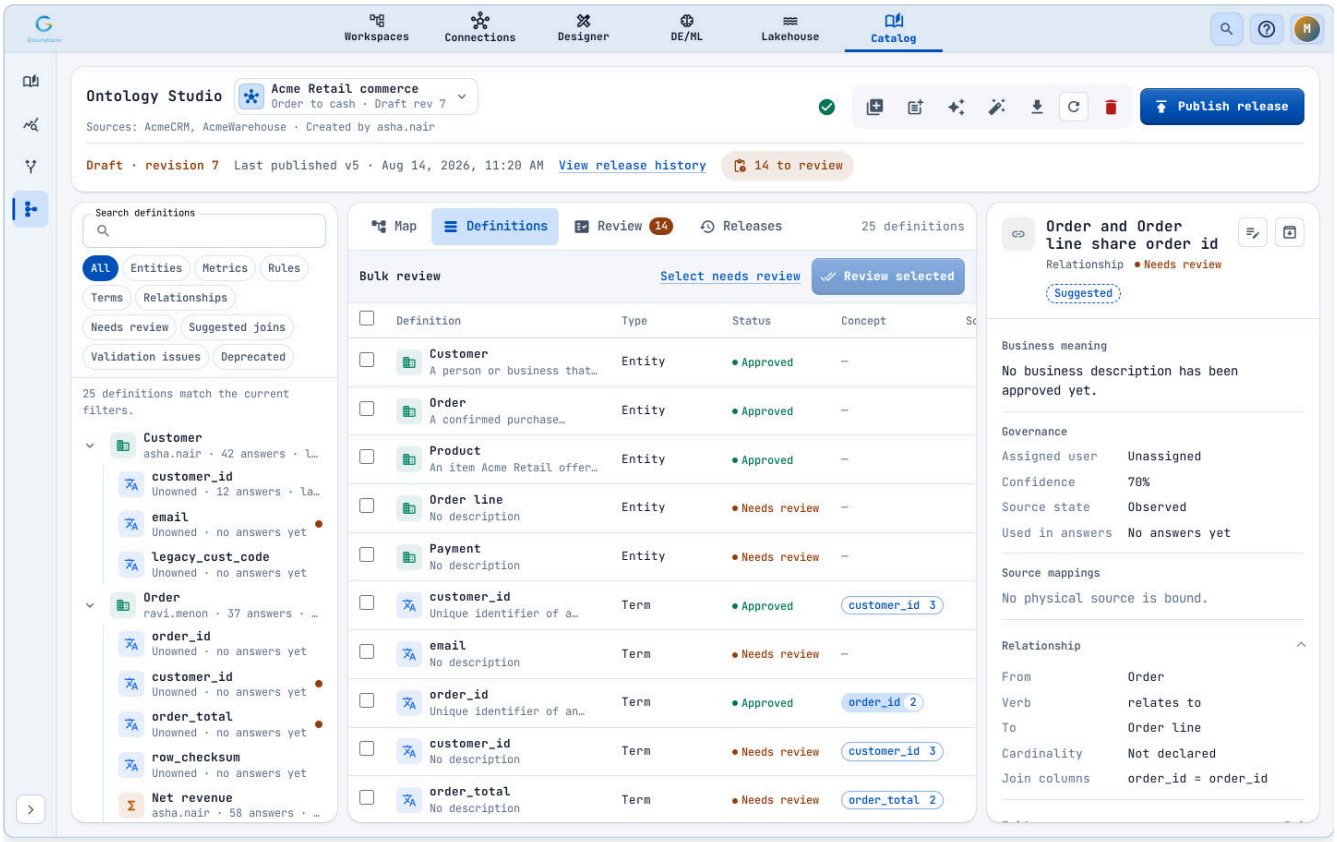


Figure 170. A suggested join is marked "Suggested" and shows its evidence.

Approve it (with a description) to make it a normal relationship. Use the **Suggested joins** filter to find them.

7.8 Draft with AI

Curate draft with AI writes first drafts of names, descriptions, synonyms and guidance for items that need review. It never approves anything, publishes, assigns owners or invents sources or formulas. A person always makes the decision.

You need `Modify`, a draft model, permission to use AI Chat and AI usage left. Without AI curation on your site you see "AI curation is not enabled in this environment. Ask an administrator to deploy the latest AI Chat service." At your limit you see "You have reached your current AI usage limit. Contact your administrator to request additional access."

1. Click **Curate draft with AI** (toolbar) or **Curate with AI** (Review tab). Optionally describe your **Business use case** in a sentence, for example "Help sales operations understand orders, customers and products."

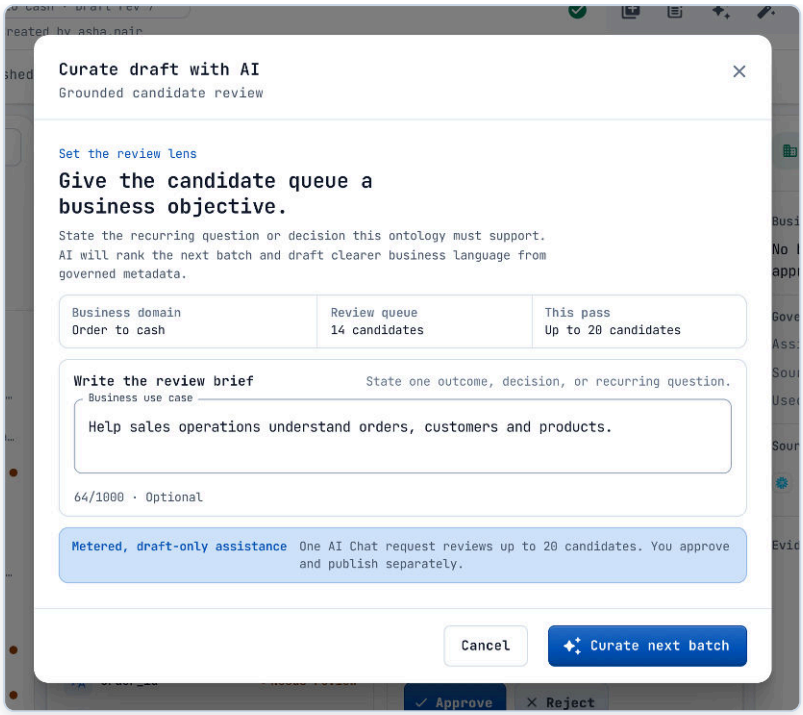


Figure 171. Setting the review lens for an AI curation pass.

2. Click **Curate next batch**. A pass handles up to 100 waiting items, entities first.

> **DIALOG WORDING**
>
> The dialog may say "Up to 20 candidates". A pass actually handles up to 100.

3. Check the results. The counters show **Recommended**, **Needs review** and **Leave out**. Each card has the proposed name, description, synonyms, **AI guidance**, a reason, a confidence level and the **Evidence used**. Only "Recommended" items are ticked for you; "Leave out" items cannot be chosen.



Figure 172. AI suggestions with confidence and evidence; nothing is applied yet.

4. Click **Apply <n> to draft**. This writes only the wording and a note ("AI-assisted draft ..."). **Statuses stay Needs review.** The review dialog opens so you can record the decision. **Regenerate batch** goes back to the start.

Amber warnings, such as "Some drafts were inferred only from technical names and are marked for review.", tell you where to look more carefully. If the draft changes while you work, the results are discarded; start again. In the Review tab, an AI suggestion can be taken field by field with **Accept**, all at once with **Accept all**, or ignored with **Dismiss**. Nothing is saved until you approve or reject. Suggestions are lost on reload.

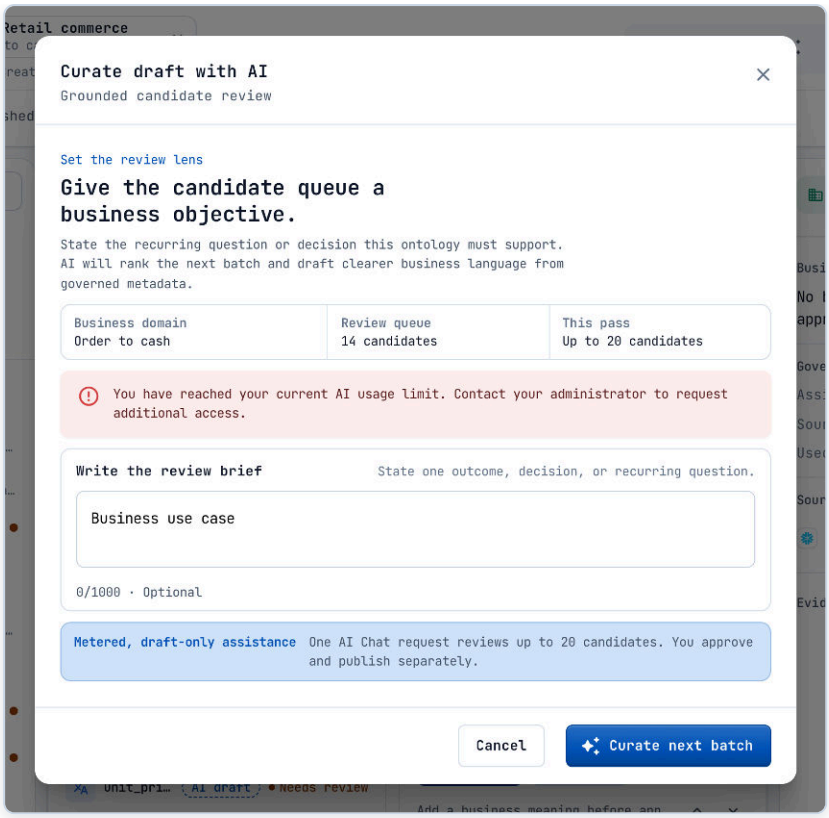


Figure 173. An exhausted AI budget stops curation with a clear message.

7.8.1 Fix a missing source

A definition whose table or column disappeared from lineage shows **Governed source missing**.

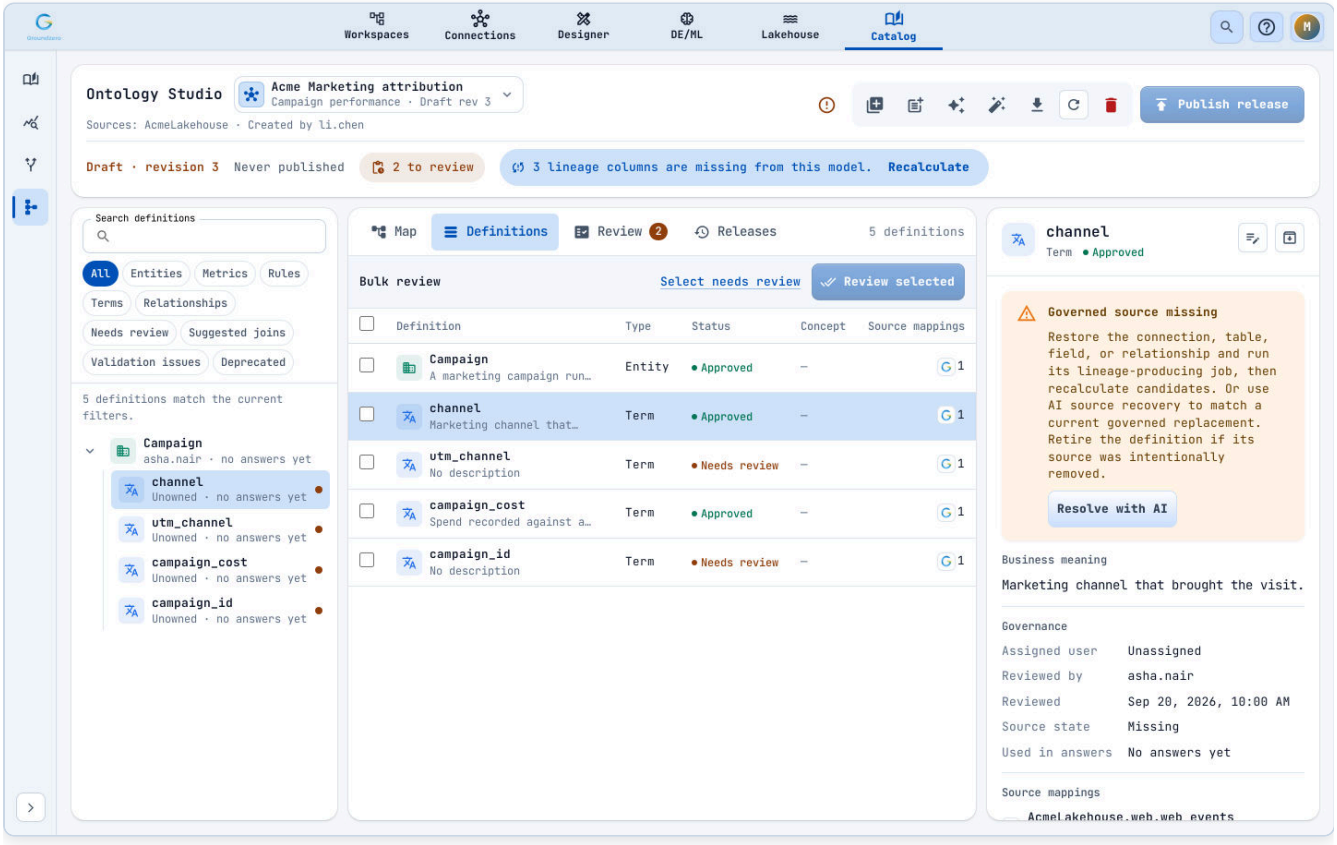


Figure 174. A definition whose source vanished from lineage can be fixed with Resolve with AI.

Your options:

- Restore the source, run its job, then **Recalculate**.
- Click **Resolve with AI** (in the inspector or the readiness popover). AI matches up to 20 such approved definitions with current sources. Each card proposes **Replace** (use the new source and keep your reviewed text), **Needs review** or **Retire**. Click **Apply <n> source resolution(s)**, then check validation before publishing.
- Deprecate or reject the definition.

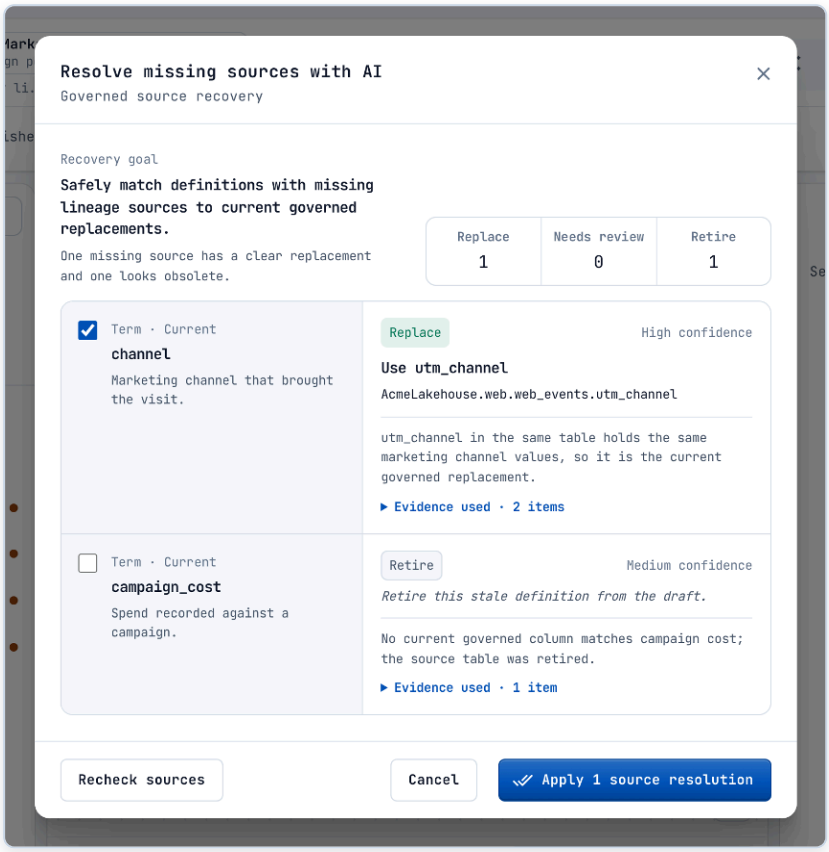


Figure 175. AI proposes replacing or retiring definitions whose sources vanished.

7.9 Publish a release

Outcome: your approved definitions become the version AI Chat and Agent Builder use.

7.9.1 Check readiness

While the model is a draft, an icon in the status bar shows **Ready to publish** (green) or **Release needs attention** (red). Click it for details.

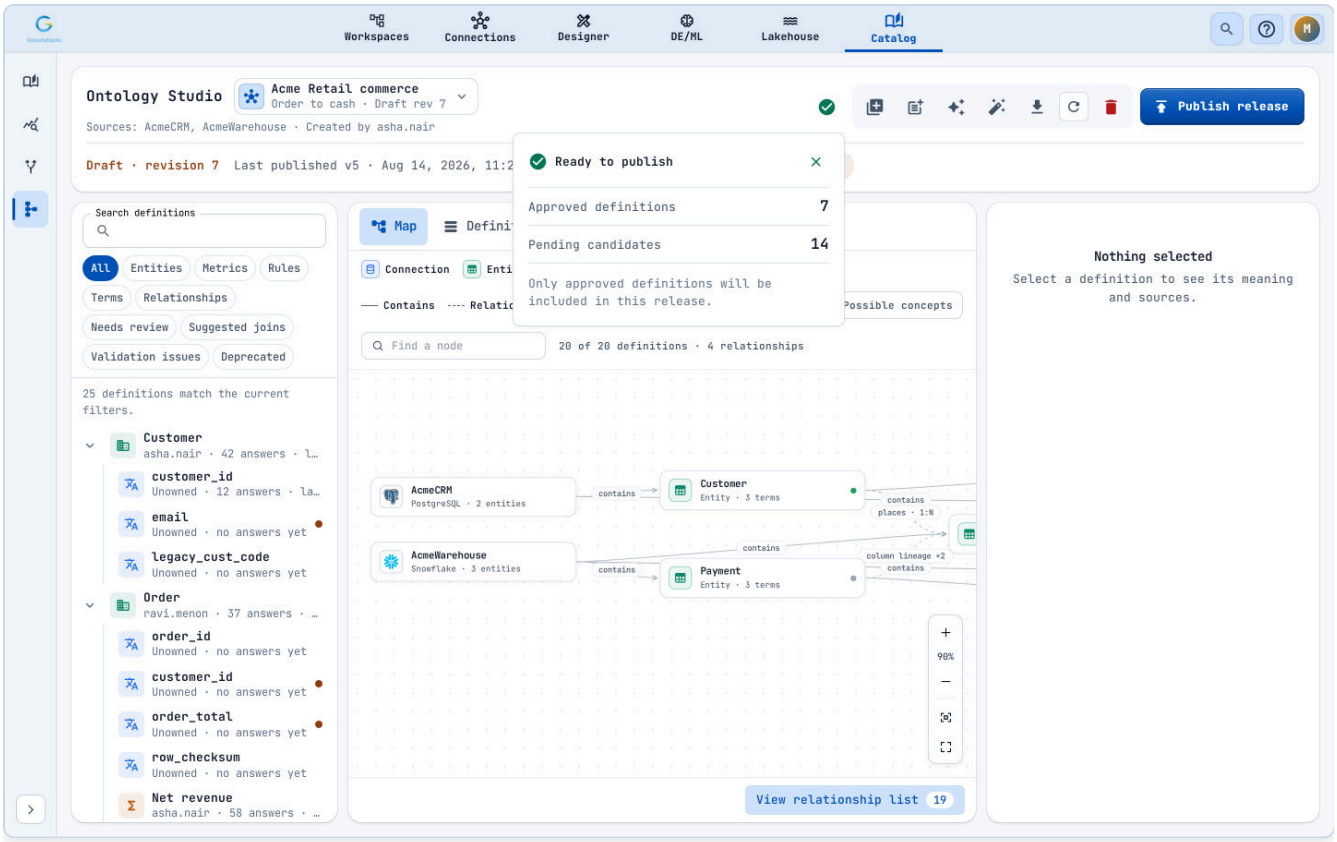


Figure 176. Ready to publish: approved definitions and pending candidates are counted.

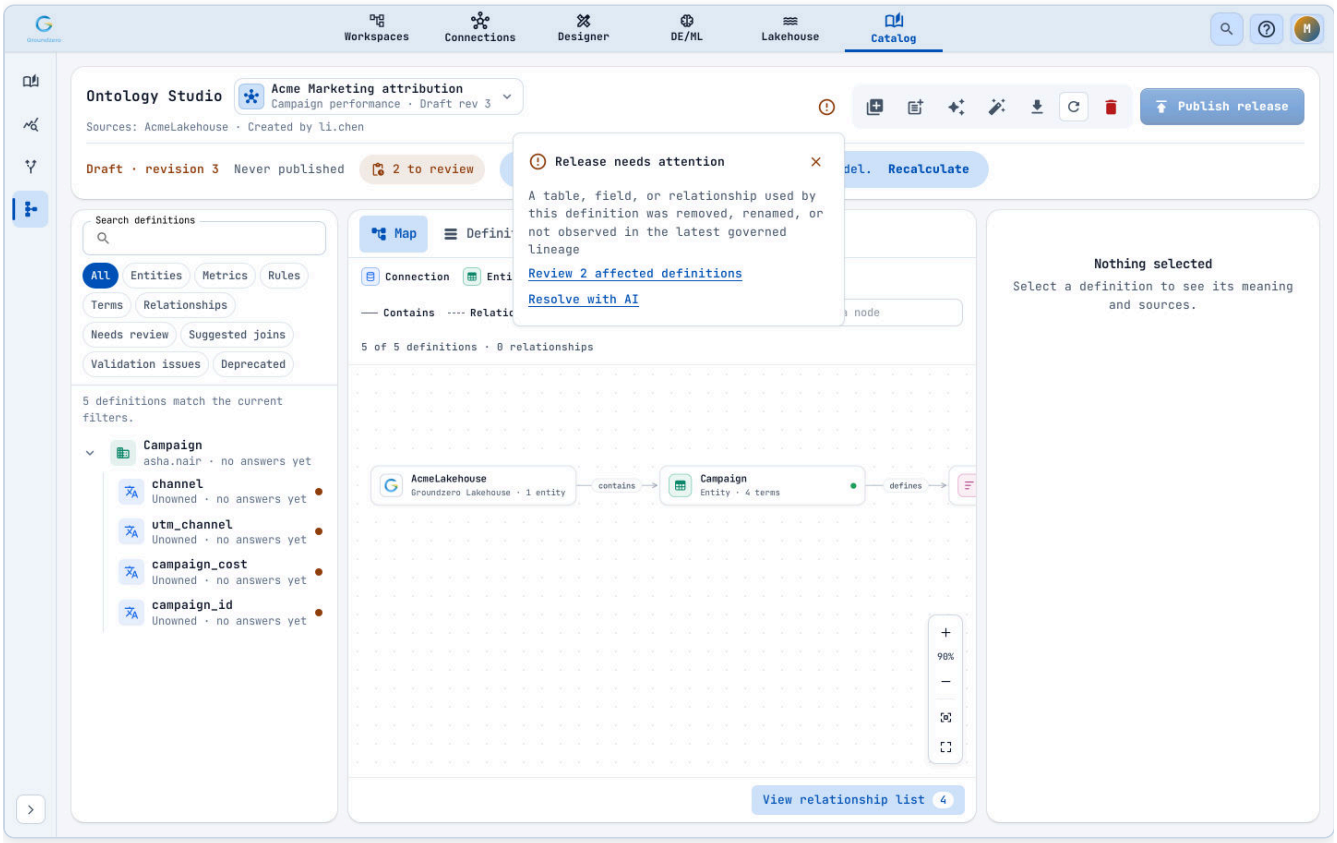


Figure 177. Blockers list what must be fixed, with links to the affected definitions.

Publishing is blocked when:

- you may not publish ("You do not have permission to publish this model.");
- nothing is approved ("Approve at least one business definition.");
- an approved definition has a problem: no business meaning, a missing source, no governed connection, no expression (for a metric or rule), an entity that is not approved, or a relationship whose ends are not approved.

Items still waiting for review do **not** block publishing. They are left out and carry over to the next draft.

7.9.2 Publish

1. Click **Publish release**. The dialog shows the **Version**, the **Approved scope** (for example "7 definitions and 2 relationships"), **AI access** ("AI Chat and Agent Builder") and how many pending items will not be published.



Figure 178. The publish dialog summarises exactly what goes into the release.

2. Click **Publish release** again. You see "Ontology release published to its configured AI targets."

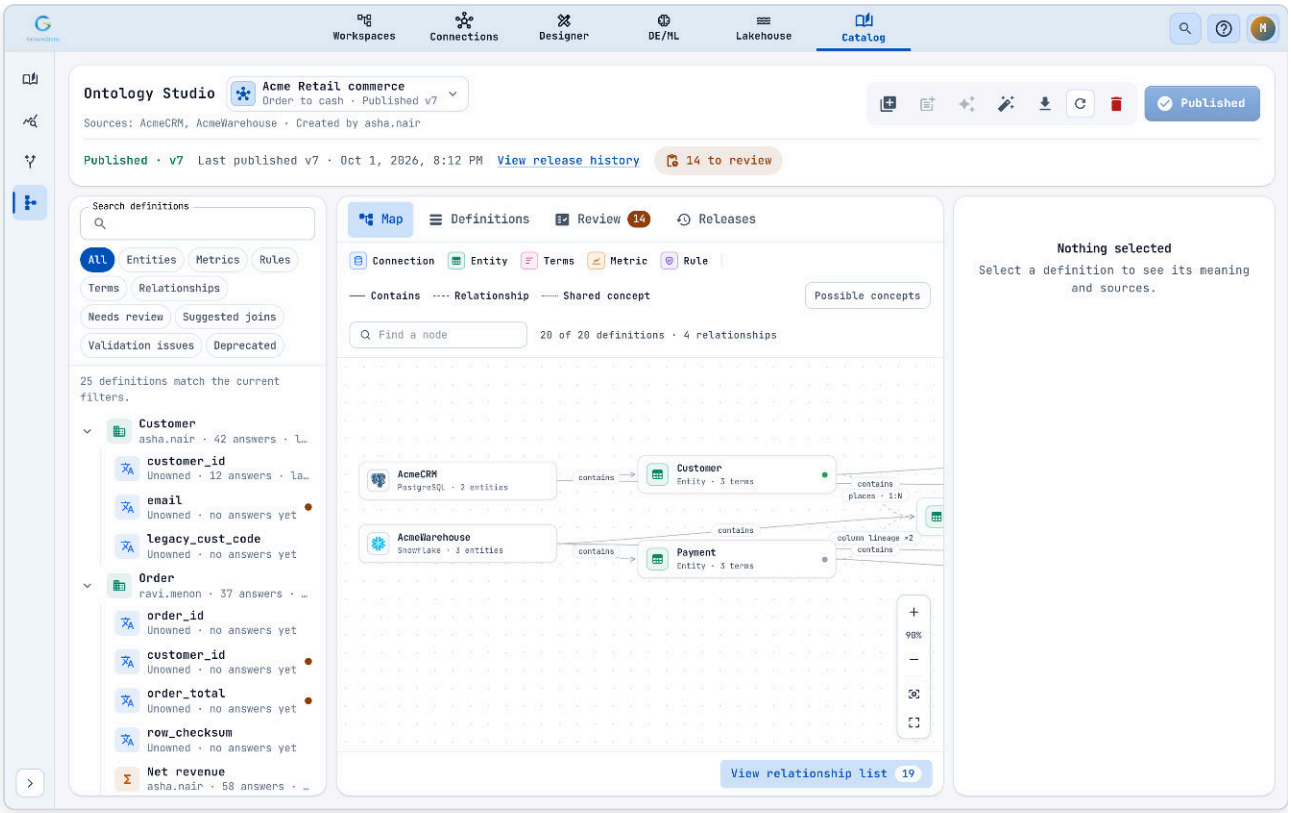


Figure 179. A published model is locked; the toolbar offers Create draft.

The model now reads "Published · v<version>". Version numbers rise with each change to the draft, so they skip (for example v3, v6, v9).

7.9.3 Change a published model

A published model is read-only: definitions show a lock instead of Review and Deprecate.

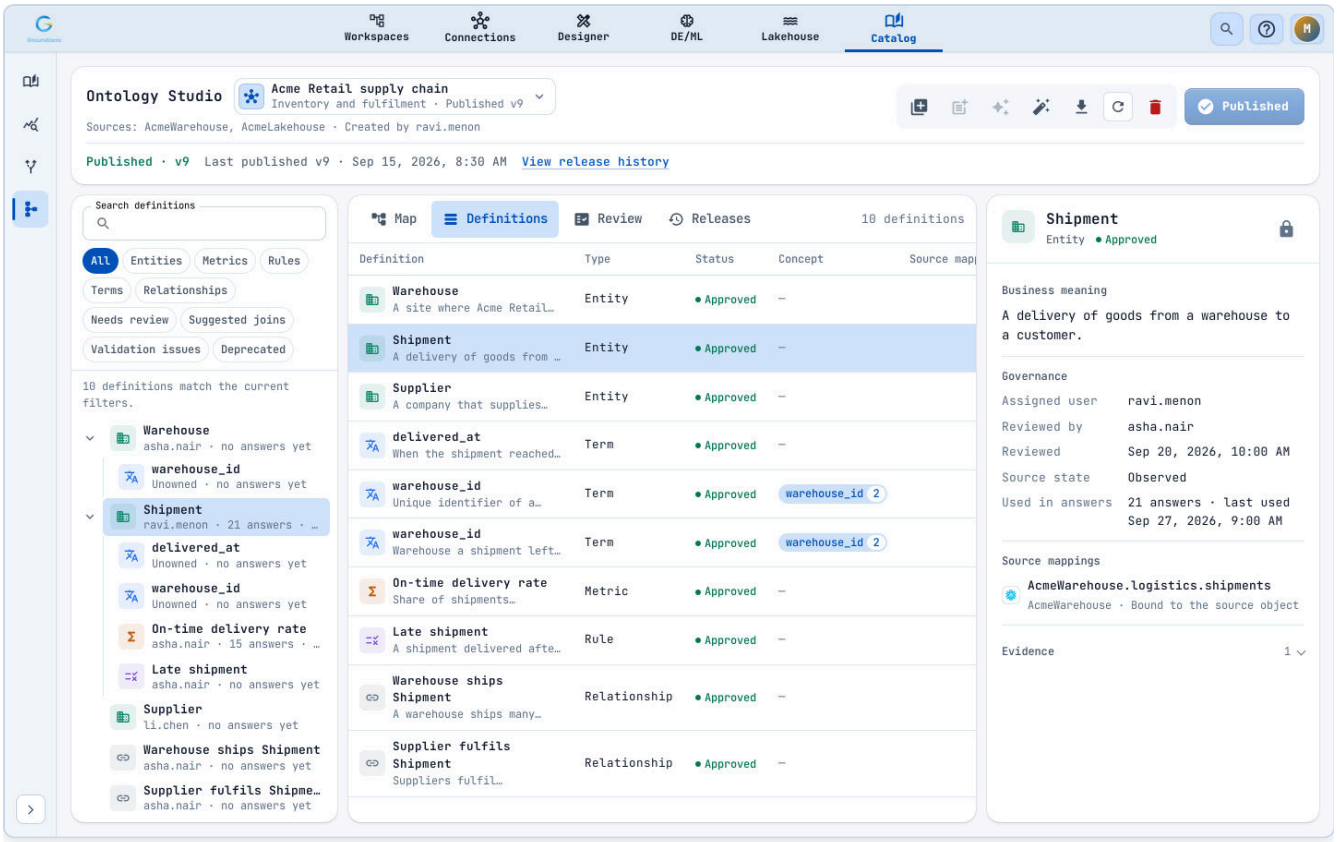


Figure 180. A published model's definition shows a lock instead of Review and Deprecate.

Click **Create draft** to start editing. The published release keeps serving AI until you publish a new one.

7.10 Look at past releases

The **Releases** tab lists published versions, newest first, with a **Current** chip, the publisher, counts and a short fingerprint (hash) of the release.

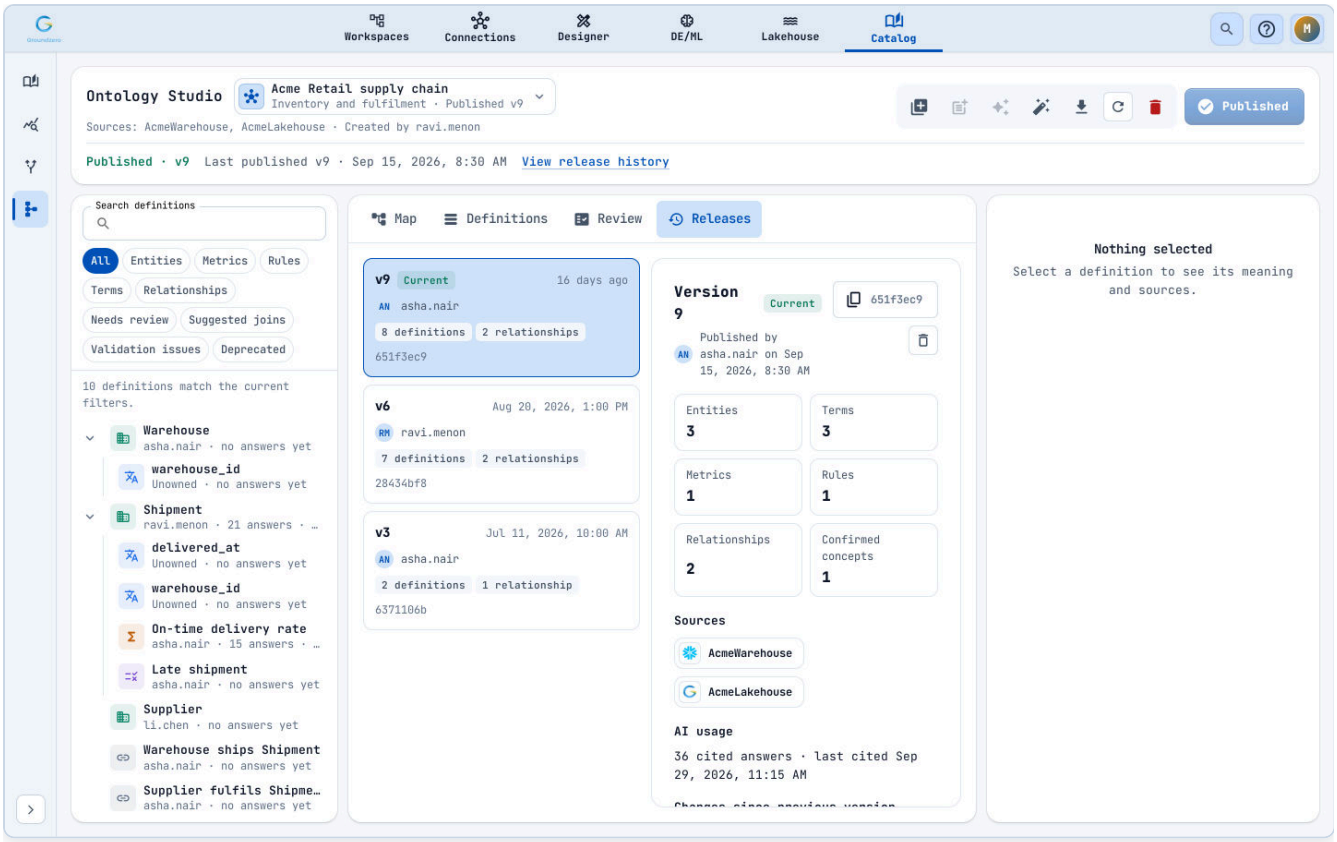


Figure 181. Release history: the current version, its statistics, sources, AI usage and the changes since the previous version.

Open one to see its counts, its **Sources**, **AI usage** ("<n> cited answers · last cited ..."), **Changes since previous version** (Added, Changed, Removed) and a searchable list of its definitions. Click the hash to copy it.

To remove a release, click **Delete release** (needs `Publish`) and type its version, for example `v9`. Agents pinned to it will report an error, AI answers go back to the previous release, and answers already given stay in usage history.

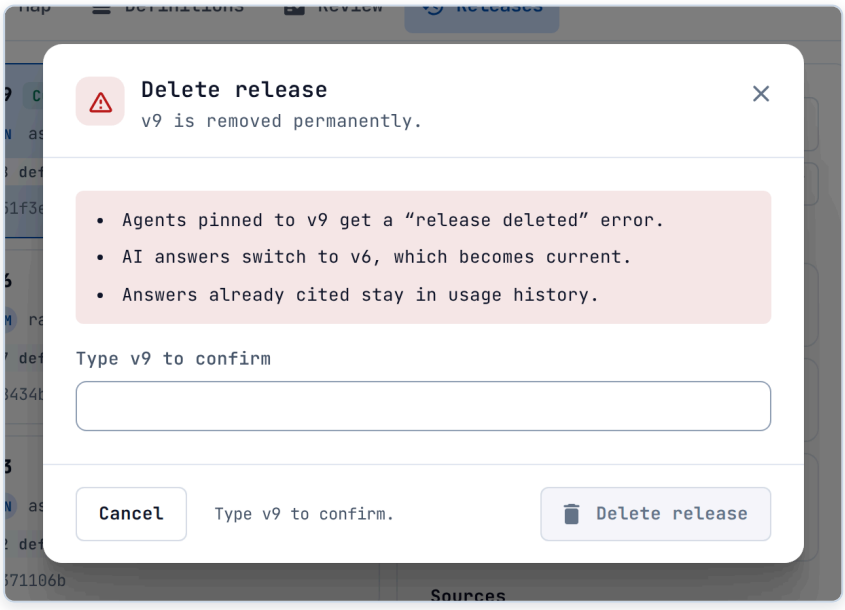


Figure 182. Deleting a release lists its consequences.

An empty history reads "No release has been published yet." with **Publish release**, or "Approve definitions to publish the first release."

7.11 Keep up with new columns: lineage drift

When your tables gain columns the model does not know, the status bar says "<n> lineage columns are missing from this model." Click **Recalculate**.

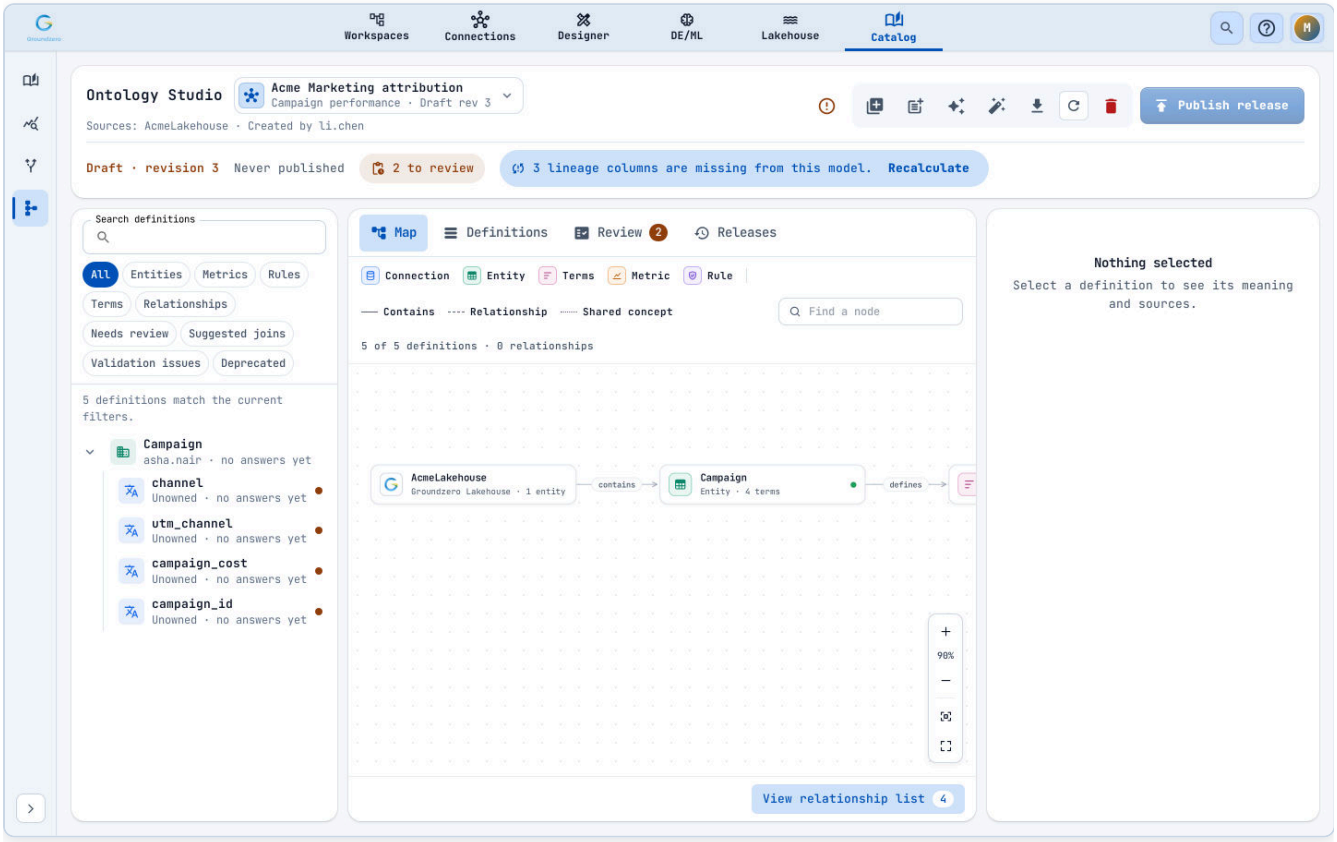


Figure 183. The drift notice offers Recalculate.

Recalculate adds new tables, columns and relationships as items to review. It keeps everything you have already reviewed, marks reviewed items whose source vanished as missing, drops unreviewed candidates that vanished, and refreshes join and concept suggestions. You see "Ontology candidates recalculated from the latest governed lineage." If the check cannot run you see "Lineage drift is unavailable right now."

7.12 Export a model

Click **Export audit pack** and choose **Audit pack (JSON)** (the whole model, counts and usage) or **Spreadsheet (CSV)** (one row per definition with its status, owner, reviewer and use in answers).

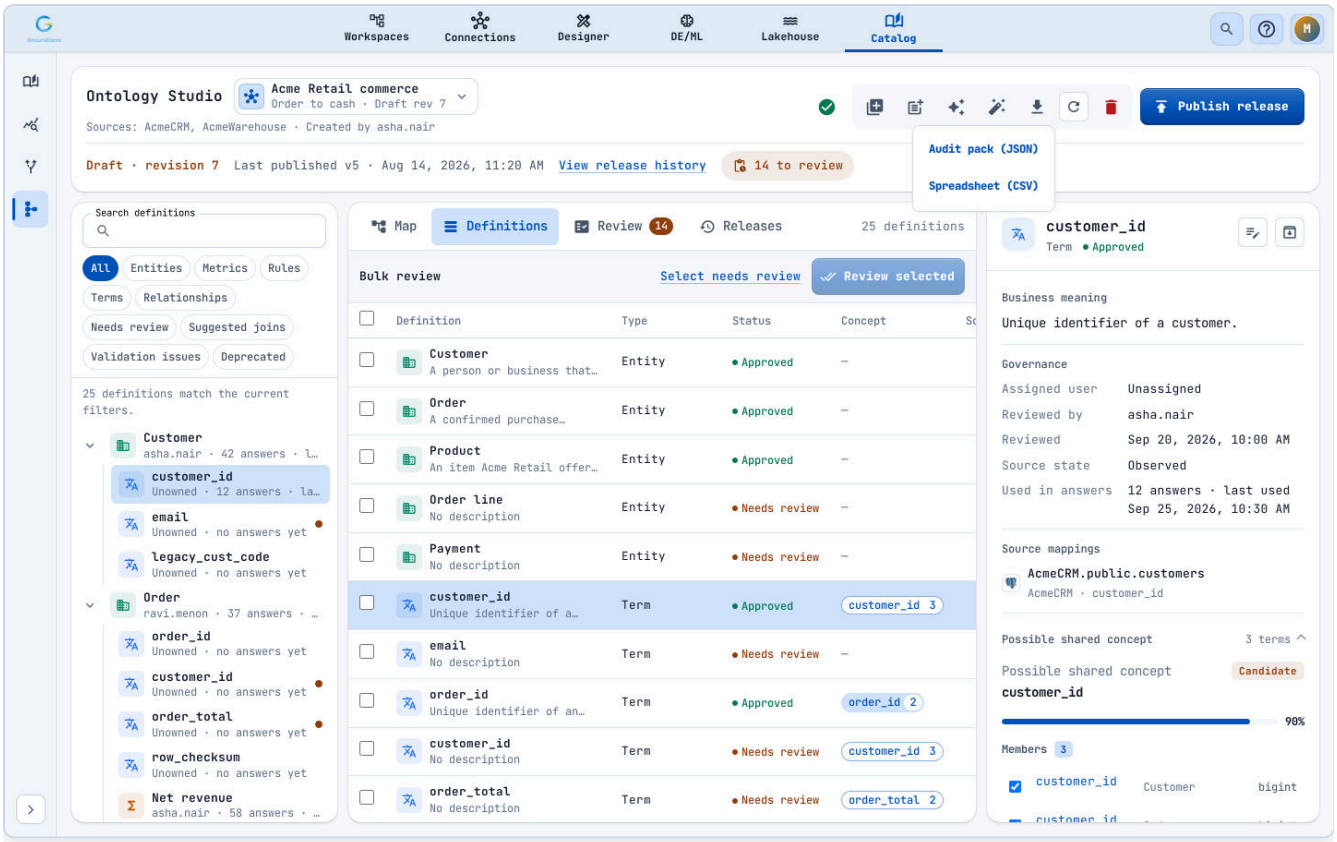


Figure 184. The export menu.

7.13 If someone else edits at the same time

If a colleague saves first, the page loads their version and shows "This draft changed in another session. The latest version is loaded; review the current state and retry." If you had unsaved work you also see that it is preserved; save again to apply it.

7.14 When the page shows a message

You see	What it means
"No ontology model yet"	None exists. Click **Create first model**. If you lack permission: "You can view ontology models, but you cannot create one for this site."
"Ontology models are unavailable"	The list failed to load. Click **Retry**.
"Your account cannot view ontology releases for this site."	You lack list permission
"This ontology model is unavailable"	It failed to load or does not exist. Click **Retry**.
"You do not have permission to inspect this model."	Ask for access to its source connections
"The ontology service could not be reached. Check your connection and retry."	Try again

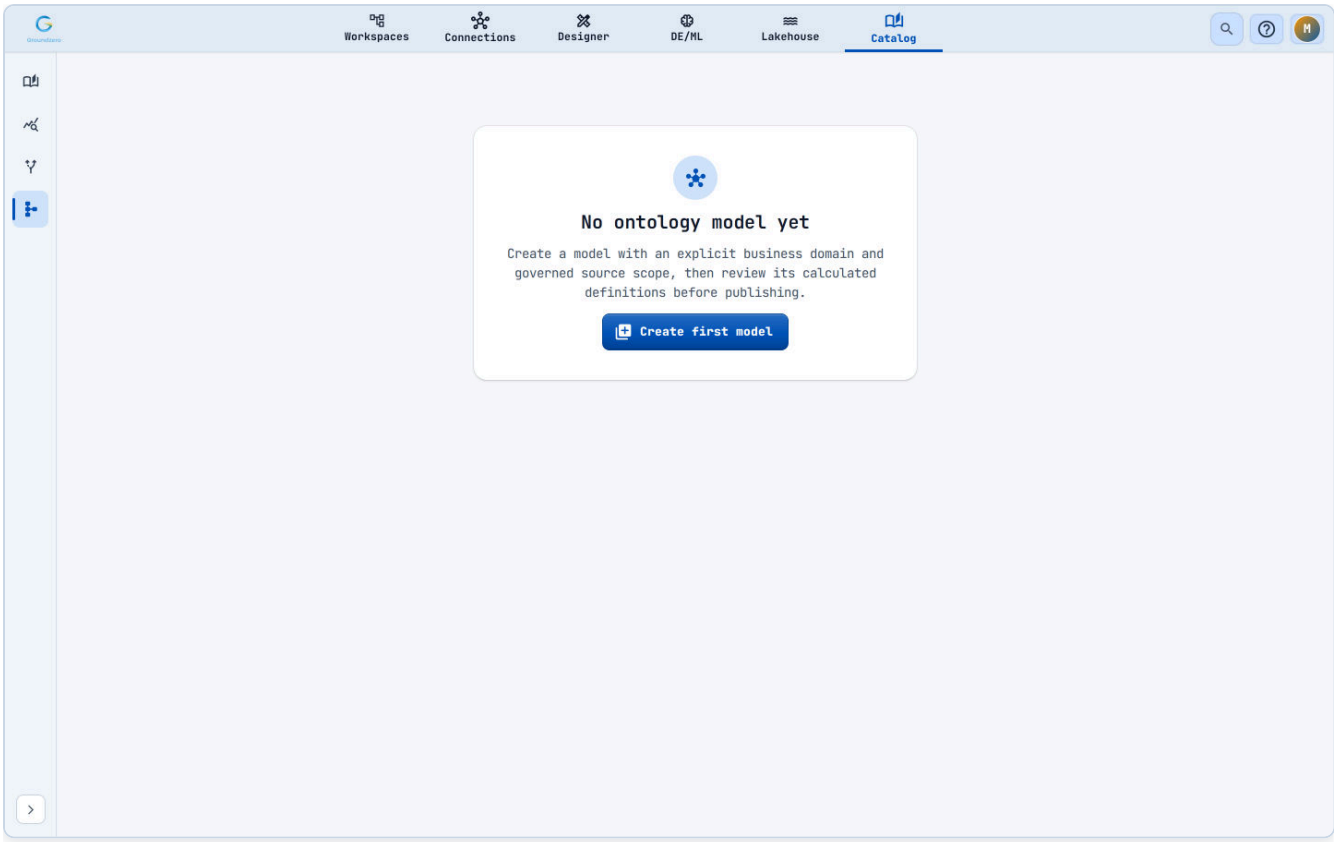


Figure 185. With no models the page offers to create the first one.

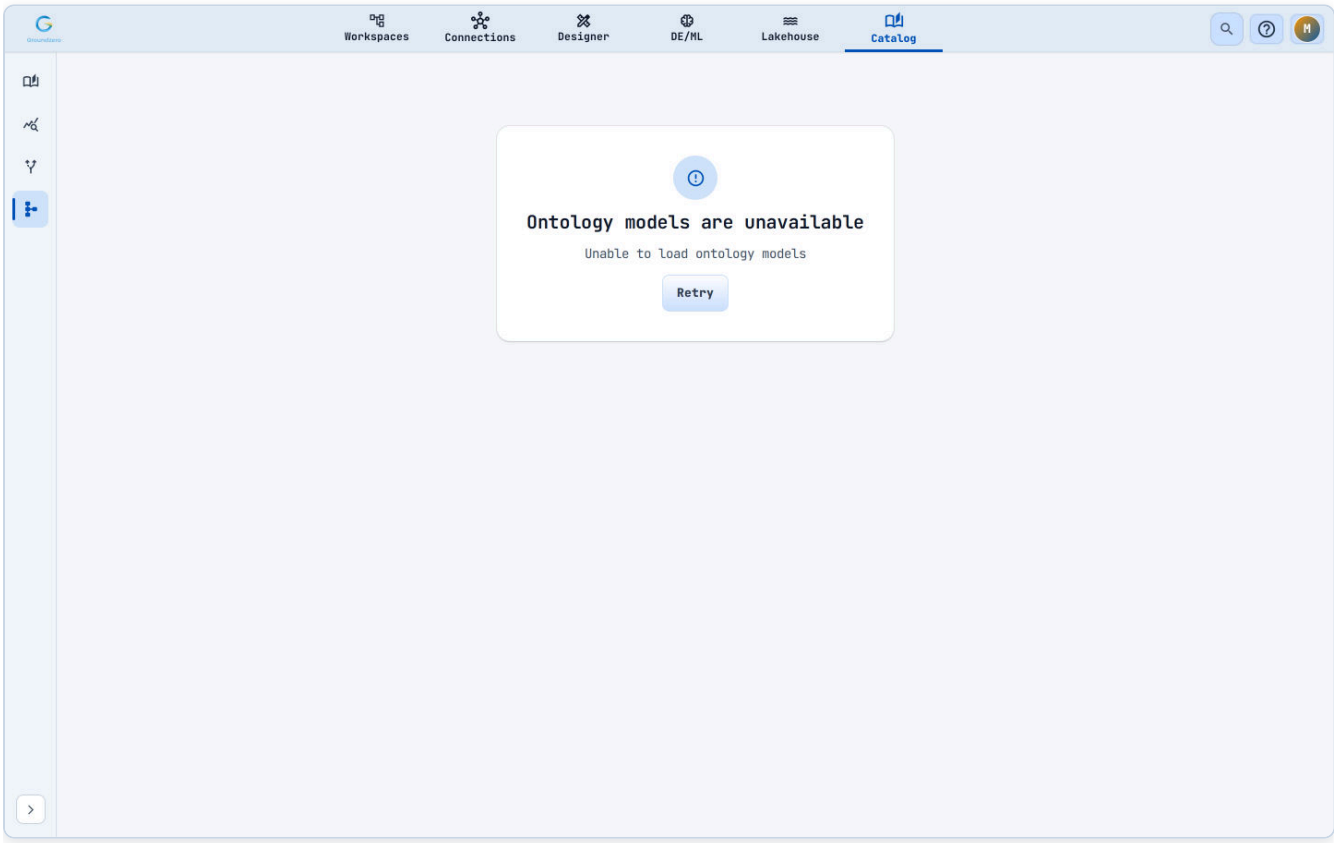


Figure 186. A failed model list can be retried.

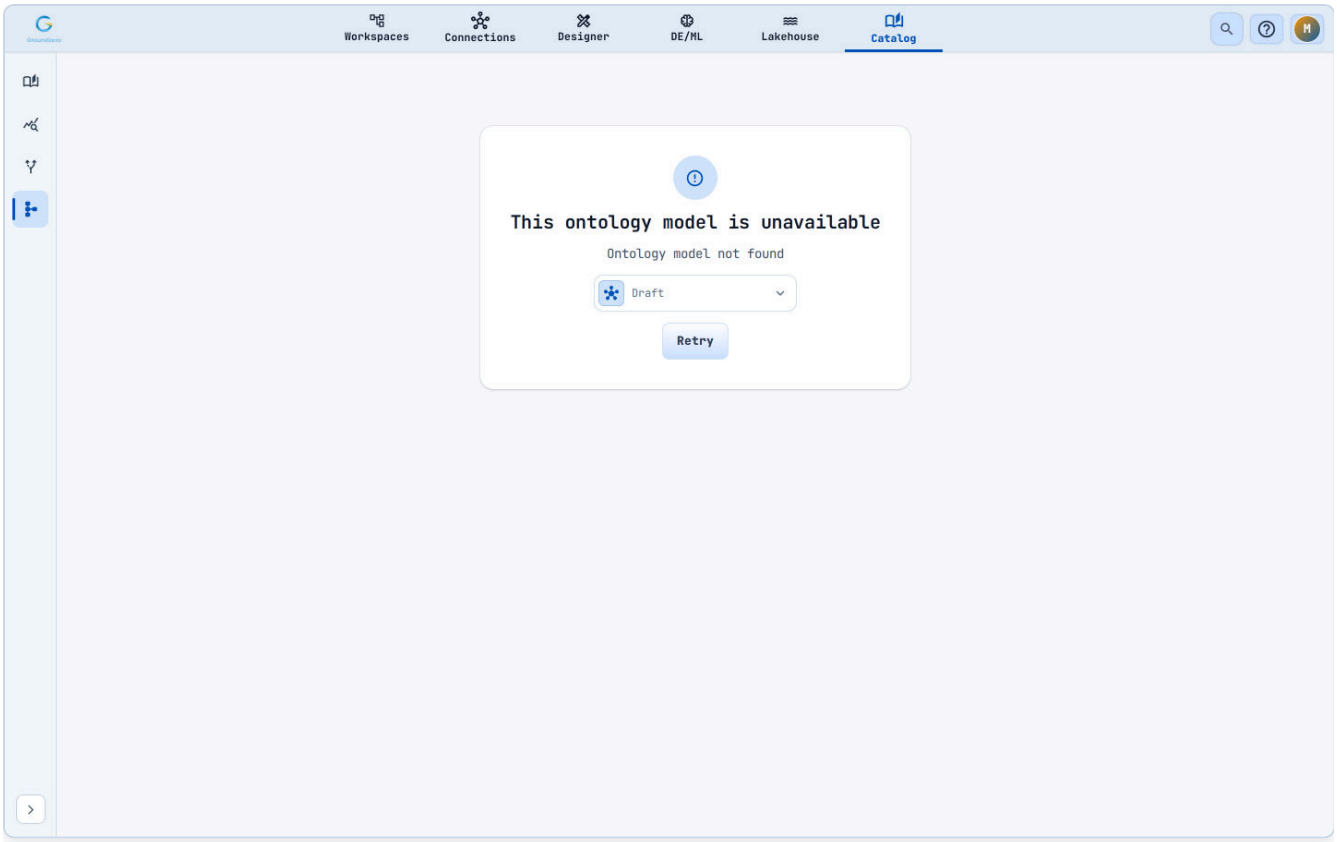


Figure 187. A model that does not exist or cannot be opened.

7.15 Tips

TIP

Start with the **Needs review** filter and the **Review** tab. The progress bar shows how close you are.

TIP

Give AI a specific business use case; it ranks and words candidates, but you approve each one.

TIP

Fix missing sources before you publish. The readiness popover links to the affected definitions.

7.16 Troubleshooting

Symptom	Cause	Fix
"No ontology model yet", but you expect one	You lack `List` or `Read`, or `Describe` on its connections	Ask an administrator
Create ontology model is greyed out	No `Create` permission	Ask an administrator
"No governed lineage sources match the requested scope"	The chosen tables have no lineage	Run the jobs that use them

Symptom	Cause	Fix
Publish release is off	Something blocks it	Open **Release needs attention** and clear the blockers
Governed source missing	The source vanished	See "Fix a missing source"
Curate draft with AI is greyed out	Nothing to review, no `Modify`, or the model is published	Use **Create draft**
AI curation errors	Not enabled, limit reached, or the draft changed	Refresh and try again
A lock on every definition	The model is published	Use **Create draft**

7.17 Limits

Item	Limit
Model name, domain, definition name	200 characters
Meaning, review notes	4,000 characters
Expression	8,000 characters
AI guidance, deprecation reason	2,000 characters
Synonyms	20 per definition
Bulk review	1,000 definitions
AI curation pass	100 items; source resolution 20
A new model	Up to 100 connections and 1,000 tables

7.18 Related

- **Data lineage**: supplies the candidates; its Impact tab lists ontology definitions.
- **Data source connections**: the sources and their `Describe` and `Edit` permissions.
- **AI Chat and Agent Builder** (Analyze and AI documentation): use the current published release.
- **Users and roles** and **Budgets** (Admin documentation): permissions and AI usage limits.

A

Appendix

A.1 Naming rules

Object	Rule
Connection	Starts with a letter; letters, digits and single hyphens; unique on your site; `GZ-CHAT-STORAGE` is reserved
Workspace	Starts with a letter; letters, digits and single hyphens
Warehouse	Starts with a letter; letters, digits and single hyphens (no underscore); its bucket is named `gz-lakehouse-<name>`
Database backup	Letters, digits and underscores; 3 to 50 characters; starts with a letter
Ontology model and definition	Up to 200 characters

A.2 Limits at a glance

Area	Limit
Lists	20, 50 or 100 rows per page
Lakehouse query	4 at a time per session; 30 minutes at most; 100,000 rows paged at once; export up to 5 GB; results up to 50,000,000 rows or 20 GiB
Lakehouse session	Starts within 3 minutes; stops after 30 idle minutes
Lineage	0 to 5 steps each way; 250 cards drawn; export 10,000 rows
Ontology	100 items per AI pass; bulk review 1,000

A.3 Glossary

Term	Meaning
Asset	A table, view, topic, file or other data object that lineage tracks
Compaction	Joining many small table files into fewer large ones so queries run faster
Concept	A group of columns from different tables that mean the same thing
Connection	Saved details for reaching an outside system
Data policy	A row filter and column masks applied to every read of a table
Dataset	A reusable data selection built in Designer
Definition	One entry in an ontology: a business meaning for an entity, term, metric, rule or relationship
Downstream	Everything that uses a table: jobs, datasets, dashboards

Term	Meaning
Entity	A table in an ontology
ETL job	A job that moves or transforms data
Iceberg	The open table format used by lakehouse tables
Lineage	The recorded flow of data between tables through jobs and datasets
Mask	A rule that hides or changes a column value for some readers
Ontology	A reviewed set of business definitions and how they relate
Release	A frozen, numbered copy of an ontology's approved definitions
Session	A query engine started for you and one warehouse
Snapshot	A saved version of a table at one moment
Tag	A short `key: value` label on a resource, table or column
Upstream	Everything that feeds a table: sources and jobs
Warehouse	An S3 bucket registered to hold lakehouse tables
Workload	An ETL job or a dataset that moves or transforms data
Workspace	A named folder of imported dashboards