



USER GUIDE

# Getting Started with Groundzero

Your first hour with Groundzero: connect a source, build a chart and dashboard, schedule a job and see where your data comes from.

Edition 2026-10-02

For new users of Groundzero

Screenshots show sample data for "Acme Retail".

# Contents

|     |                                            |    |
|-----|--------------------------------------------|----|
| 1   | <b>Welcome to Groundzero</b>               | 3  |
| 1.1 | What you can do with Groundzero            | 3  |
| 1.2 | A map of the main areas                    | 3  |
| 1.3 | The building blocks                        | 4  |
| 1.4 | Before you start                           | 5  |
| 1.5 | How to use this guide                      | 5  |
| 2   | <b>Sign in and find your way around</b>    | 6  |
| 2.1 | Sign in                                    | 6  |
| 2.2 | Start from the home page                   | 7  |
| 2.3 | Move around the console                    | 8  |
| 2.4 | Find anything with search                  | 9  |
| 2.5 | Get help and manage your account           | 10 |
| 2.6 | Name things the right way                  | 10 |
| 3   | <b>Your first workflow</b>                 | 11 |
| 3.1 | Step 1: Connect a data source              | 11 |
| 3.2 | Step 2: Explore it in the Catalog          | 14 |
| 3.3 | Step 3: Build a dataset                    | 16 |
| 3.4 | Step 4: Create a visualization             | 18 |
| 3.5 | Step 5: Compose and publish a dashboard    | 20 |
| 3.6 | Step 6: Automate with a job and a schedule | 22 |
| 3.7 | Step 7: Check lineage                      | 28 |
| 3.8 | What you built                             | 29 |
| 4   | <b>Notebooks and runtimes</b>              | 31 |
| 4.1 | Choose a runtime                           | 31 |
| 4.2 | Pick a runtime and compute size            | 31 |
| 4.3 | Run your first notebook                    | 32 |
| 4.4 | Write the same job for DuckDB              | 35 |
| 5   | <b>The AI assistant and agents</b>         | 36 |
| 5.1 | Ask Groundzero AI                          | 36 |
| 5.2 | Choose options and approve changes         | 37 |
| 5.3 | Get code help inside editors               | 38 |
| 5.4 | Create an AI agent                         | 39 |
| 5.5 | Use Groundzero from your own tools         | 40 |
| 6   | <b>Next steps</b>                          | 42 |
| 6.1 | Habits that save time                      | 44 |
| 7   | <b>Glossary</b>                            | 45 |
| 8   | <b>Troubleshooting and FAQ</b>             | 46 |
| 8.1 | Troubleshooting                            | 46 |
| 8.2 | FAQ                                        | 47 |

# 1 Welcome to Groundzero

Groundzero is a data platform that runs in your browser. You use it to bring your data together, understand it, turn it into charts and dashboards, keep it up to date automatically, and ask an AI assistant for help along the way.

This guide is your first hour with Groundzero. By the end you will have connected a data source, explored it, built a dataset, drawn a chart, published a dashboard, scheduled a job and checked where your data comes from.

## 1.1 What you can do with Groundzero

- **Connect** the systems that hold your data, such as a database, a data warehouse or cloud storage.
- **Find and understand** your data in a searchable catalog, with descriptions and tags.
- **Ask questions** of your data and save the answers as reusable datasets.
- **Show** the answers as charts and dashboards that you can share.
- **Automate** data preparation with jobs that run on a schedule, and explore with notebooks.
- **Trace** where every table comes from and what depends on it.
- **Ask for help** from an AI assistant that knows Groundzero.

## 1.2 A map of the main areas

Seven places do most of the work. Six sit in the bar at the top of every page, and the AI assistant is the home page.

| Area                       | Use it to                                                                                        | You will use it in   |
|----------------------------|--------------------------------------------------------------------------------------------------|----------------------|
| <b>Connections</b>         | Save links to your data sources                                                                  | Step 1               |
| <b>Catalog</b>             | Find data, read and write descriptions, see lineage                                              | Steps 2 and 7        |
| <b>Designer</b>            | Build datasets, filters, visualizations and dashboards                                           | Steps 3, 4 and 5     |
| <b>Workspaces</b>          | Share published dashboards in named folders, and browse your storage buckets ( <b>My Files</b> ) | Step 5               |
| <b>DE/ML</b>               | Keep jobs, notebooks, models and agents in projects, and schedule jobs                           | Step 6 and chapter 4 |
| <b>Lakehouse</b>           | Store governed tables in warehouses and query them with SQL                                      | Chapter 6            |
| <b>Home (AI assistant)</b> | Ask questions in plain language and get help writing code                                        | Chapter 5            |

Most areas also have a narrow menu on the left. Widen it with the arrow at the bottom left of the page.

## 1.3 The building blocks

Groundzero uses a small set of words. Each one is explained here the first time it appears, and again in the glossary at the end.

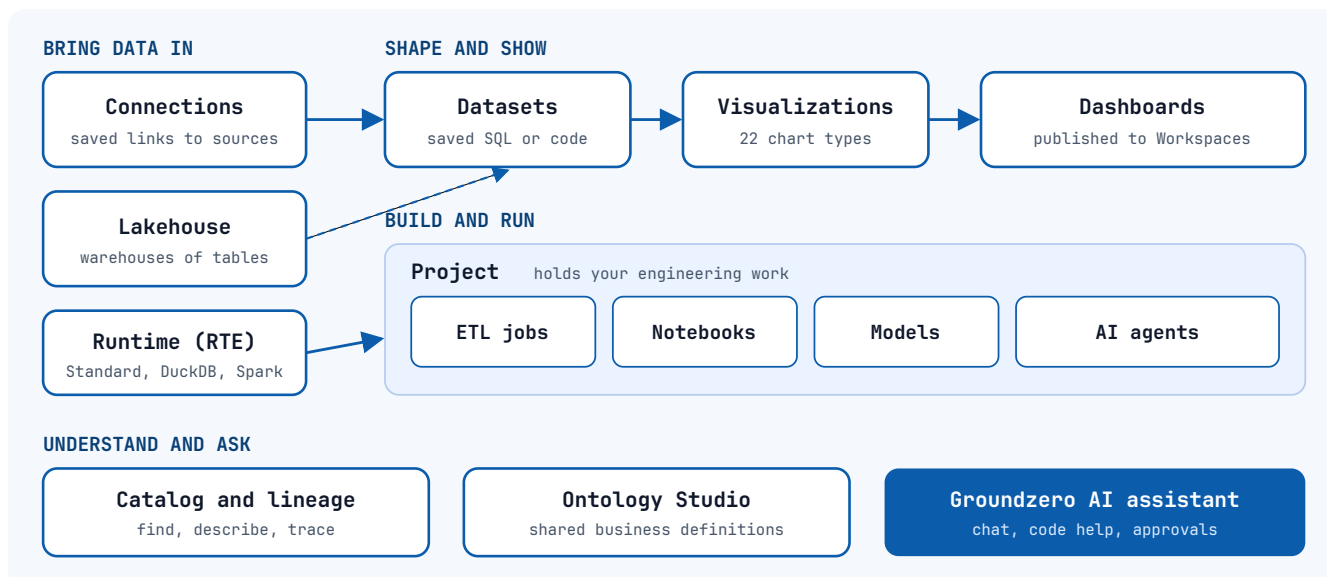


**Figure 1.** The Groundzero building blocks. Connections feed datasets, jobs and notebooks; runtimes run your code; the catalog and lineage describe what exists.

| Word                             | What it is                                                                                                                                                                                                       |
|----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Site</b>                      | Your organisation's Groundzero. Your administrator chooses its logo, colours and sign-in background.                                                                                                             |
| <b>Connection</b>                | A saved link to an outside system such as PostgreSQL, Snowflake, Amazon S3, a REST API, an AI provider or a Groundzero Lakehouse. Your code and datasets use the connection's name and never contain a password. |
| <b>Lakehouse and warehouse</b>   | A Groundzero Lakehouse stores tables in a <b>warehouse</b> . A warehouse holds databases, tables and columns that you can query with SQL.                                                                        |
| <b>Dataset</b>                   | A saved, named result. You define it with a SQL query (or Python code) on one or more connections. Charts and dashboards are built on datasets.                                                                  |
| <b>Visualization</b>             | A saved chart or table built on a dataset. Groundzero offers 22 chart types.                                                                                                                                     |
| <b>Dashboard</b>                 | A page of visualizations, arranged in tabs, with filters. You can publish a dashboard and import it into a <b>workspace</b> folder so others can read it.                                                        |
| <b>Project</b>                   | A container in <b>DE/ML</b> for your ETL jobs, notebooks, models and AI agents.                                                                                                                                  |
| <b>ETL job</b>                   | Python code that reads data from connections, changes it and writes the result. It runs when you ask or on a schedule. (ETL means extract, transform, load.)                                                     |
| <b>Notebook</b>                  | An interactive document of code and text cells. You run it in a session and see results as you go.                                                                                                               |
| <b>Runtime (RTE) and compute</b> | The runtime is the environment that runs your code (Standard, DuckDB or Spark). Compute is the memory and processor size you give it.                                                                            |
| <b>Catalog</b>                   | A searchable list of datasets, ETL jobs and warehouses, with descriptions and tags.                                                                                                                              |

| Word                           | What it is                                                                                                                             |
|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| <b>Lineage</b>                 | A map of how tables flow through jobs and datasets. Groundzero records it for you when work runs.                                      |
| <b>AI assistant and agents</b> | The assistant answers questions and drafts work in chat. An agent is a saved assistant with one fixed job that other systems can call. |

## 1.4 Before you start

| You need                                                    | Why                                                                                                                                                  |
|-------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| An account on your organisation's Groundzero site           | Your administrator creates accounts. The sign-in page has no sign-up link.                                                                           |
| An authenticator app (Google Authenticator or Authy)        | Groundzero asks you to set up two-step verification the first time you sign in.                                                                      |
| Permission for what you want to do                          | Your administrator grants access. When you lack it, Groundzero says so, for example <i>You do not have "All" or "Add" permissions on Connection.</i> |
| Details of one data source, for example a PostgreSQL server | You connect it in step 1.                                                                                                                            |

### NOT SURE WHAT YOU MAY DO?

Try each step. Nothing in this guide changes the data in your sources, and when a permission is missing Groundzero names it. Send that message to your administrator.

## 1.5 How to use this guide

Chapters 1 to 3 take about an hour and give you the first workflow from start to finish. Chapters 4 and 5 take about 20 and 15 minutes each, and you can do them whenever you are ready. Chapter 6 points you to what to learn next.

### ABOUT THE SCREENSHOTS

The screenshots show sample data for a company called **Acme Retail**, so a name in one figure can differ slightly from the same step in another. Follow the instructions and the labels on your screen rather than matching names exactly.

## 2 Sign in and find your way around

---

### 2.1 Sign in

You will sign in once, set up two-step verification, and land on the home page.

- 1 Open your site's address in a browser. The sign-in page asks for your **Username** and **Password**.
- 2 Enter them and choose **Sign in**. If you have forgotten your password, choose **Reset your password**.
- 3 The first time you sign in, Groundzero shows **Set up two-step verification** instead of the console. Scan the QR code with Google Authenticator or Authy, or type the code shown below it. Then type the six-digit **Authentication code** and choose **Verify**.
- 4 On later sign-ins Groundzero shows **Two-step verification**. Type the current code from your app.

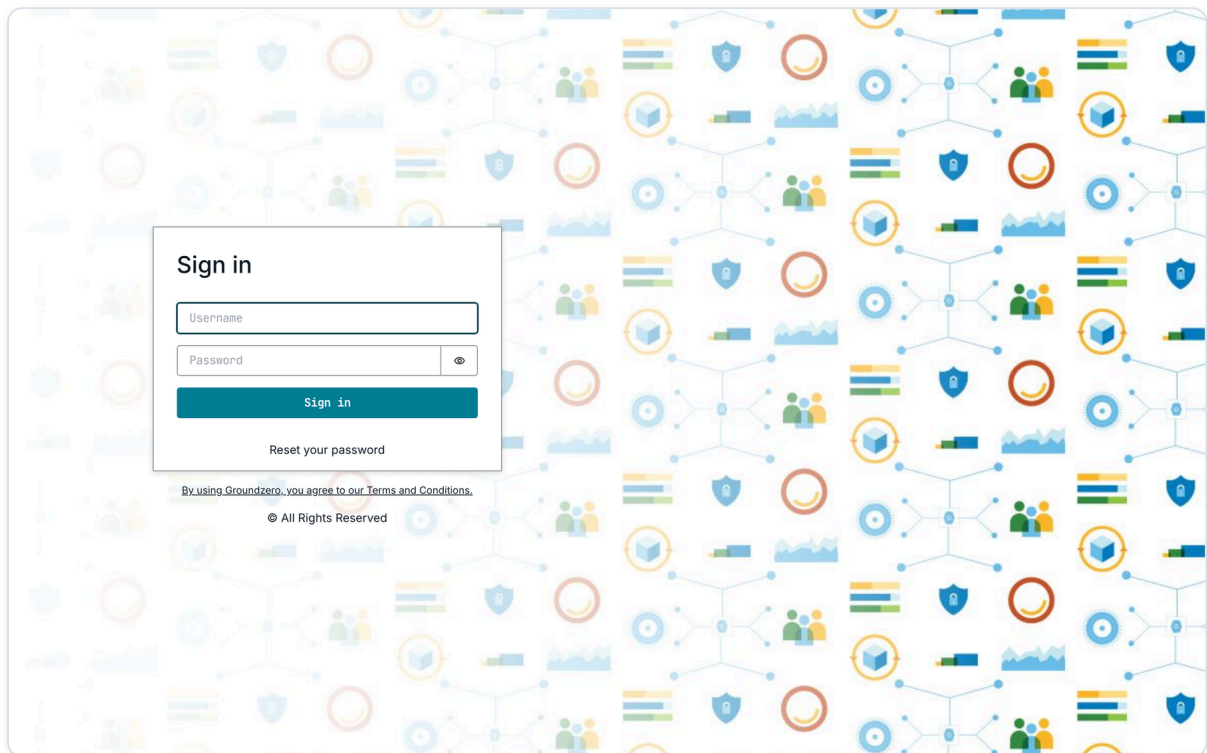
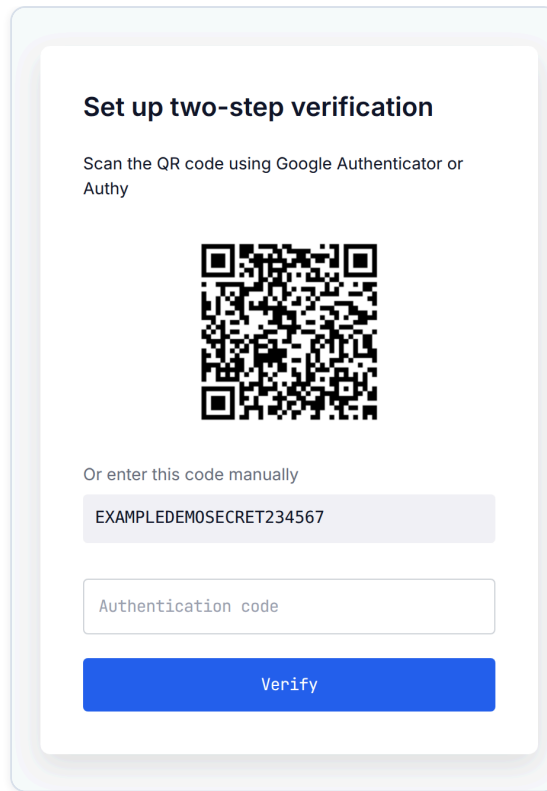


**Figure 1.** The Groundzero sign-in page: username, password and a Sign in button.



**Set up two-step verification**

Scan the QR code using Google Authenticator or Authy



Or enter this code manually

EXAMPLEDEMOSECRET234567

Authentication code

Verify

**Figure 2.** The first-time setup of two-step verification: a QR code, a manual code and the Authentication code field. The code shown here is a sample.

#### KEEP YOUR AUTHENTICATOR APP SAFE

If you lose access to your authenticator app, ask your administrator for help. You cannot reset two-step verification yourself.

## 2.2 Start from the home page

After you sign in you land on the home page, which is the **AI assistant**. Choose the Groundzero logo at the top left of any page to come back here.

You see a message box, six starter prompts (Debug SQL, Create dataset, Explain error, Build ETL job, Analyze logs and Trace lineage) and a **Chats** list of your earlier conversations. Chapter 5 shows how to use the assistant.

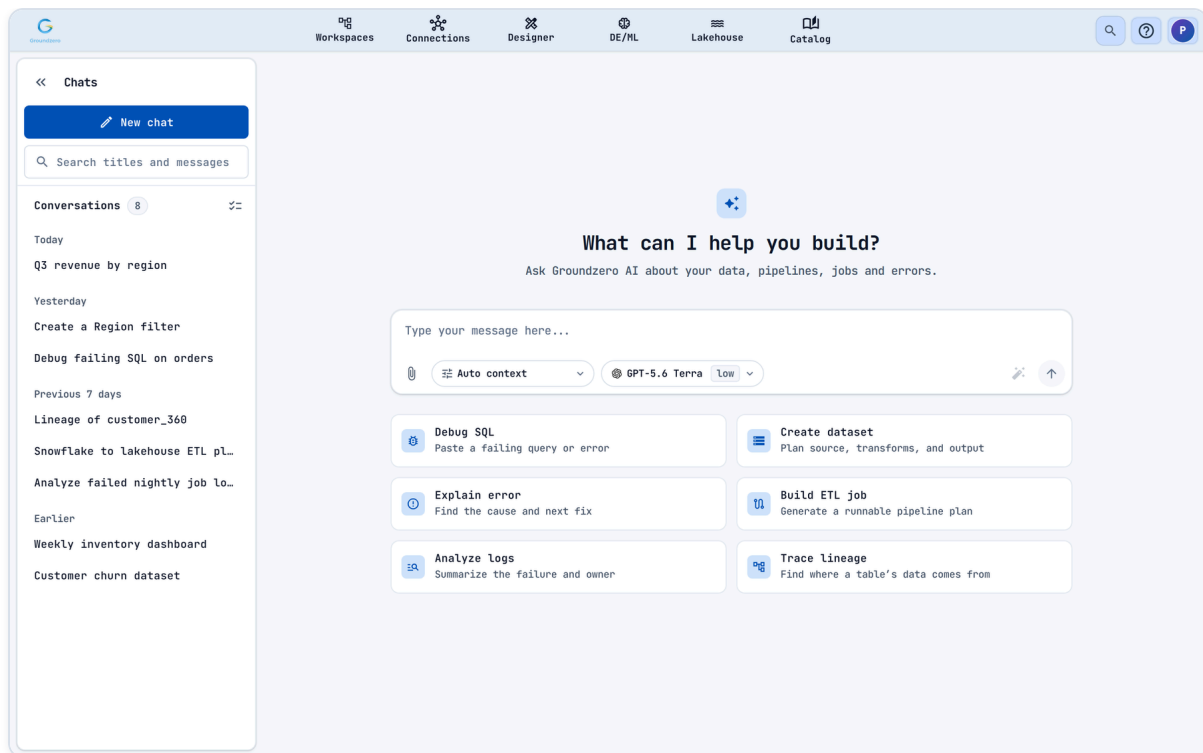


**Figure 3.** The home page: a message box with context and model controls, starter prompts, and the Chats list on the left.

#### IF THE HOME PAGE SAYS YOU HAVE NO ACCESS TO CHAT

Using the assistant needs the **AI Chat** permission. Without it the page shows *You don't have access to chat*. Everything else in this guide works without it.

## 2.3 Move around the console

Open any area from the top bar. Inside an area, use the menu on the left.

| Area               | What the left menu offers                                                                                                            |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| <b>Workspaces</b>  | <b>Workspaces</b> (folders of imported dashboards) and <b>My Files</b> (your storage buckets)                                        |
| <b>Connections</b> | The list of connections, with search and <b>Type</b> , <b>Owner</b> and <b>Updated</b> filters                                       |
| <b>Designer</b>    | <b>Datasets</b> , <b>Filters</b> , <b>Visualizations</b> and <b>Dashboards</b>                                                       |
| <b>DE/ML</b>       | <b>Projects</b> and the <b>Scheduler</b> . A project has the tabs <b>Jobs</b> , <b>Notebooks</b> , <b>Models</b> and <b>Agents</b> . |
| <b>Lakehouse</b>   | <b>Warehouses</b> and <b>Saved Queries</b>                                                                                           |
| <b>Catalog</b>     | <b>Catalog</b> , <b>Consumption Metrics</b> , <b>Data Lineage</b> and <b>Ontology Studio</b>                                         |

Most list pages look alike: a title, a search box, a table, and a three-dot menu on each row with the actions you can take.

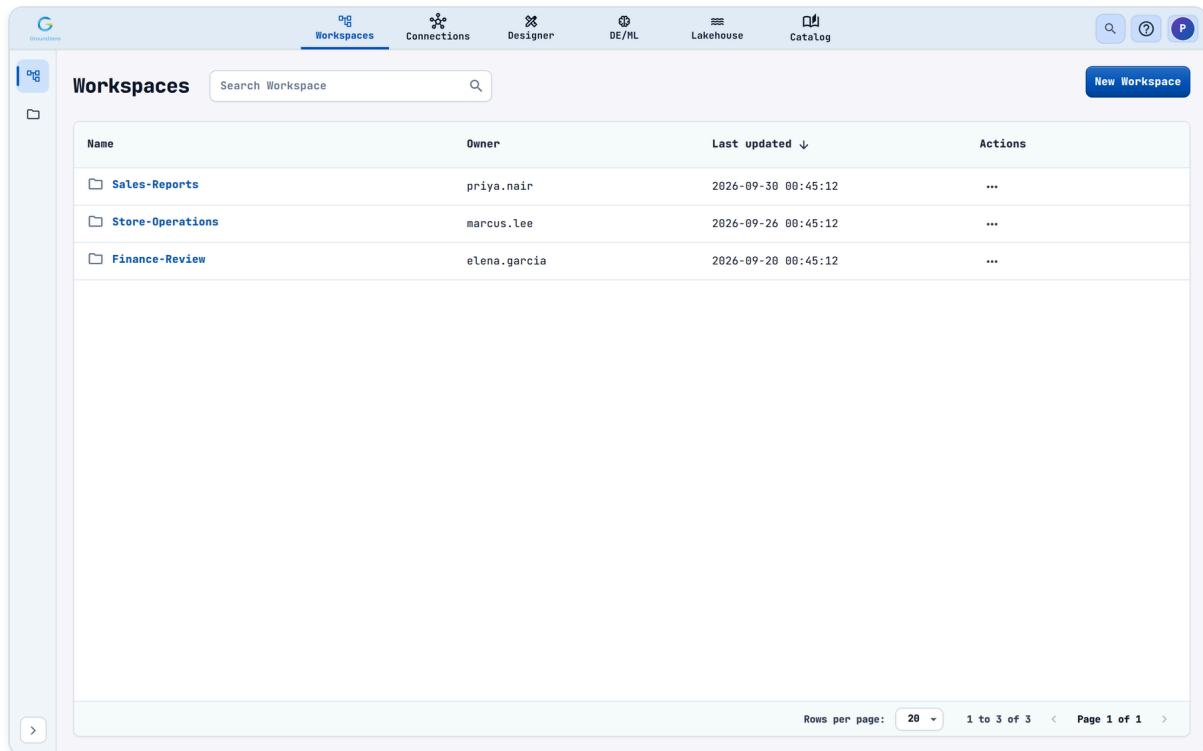


**Figure 4.** Workspaces is a typical list page: a title, a search box, a table with owners and dates, and a three-dot actions menu on each row.

## 2.4 Find anything with search

Press **Ctrl** + **K**, or choose the magnifier at the right of the top bar. Global search finds datasets, visualizations, dashboards, connections, workspaces, jobs, agents, saved queries, ontology models and pages such as Data Lineage.

To search one kind of item, start with a prefix: **ds:** (datasets), **viz:** (visualizations), **dash:** (dashboards), **conn:** (connections), **ws:** (workspaces), **job:** (jobs), **agent:** (agents), **q:** (saved queries) or **model:** (ontology models).

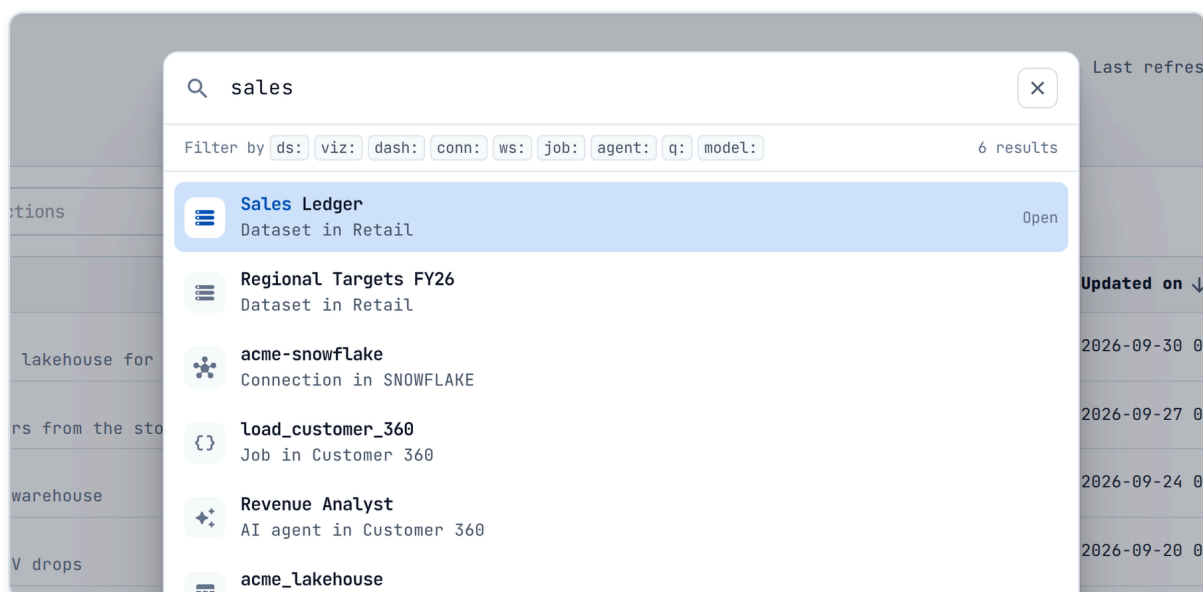


**Figure 5.** Global search with the query "sales": ranked results across kinds, and the prefix hints under the search box.

## 2.5 Get help and manage your account

Two more buttons sit at the right of the top bar.

The **Help** button (question mark) opens *Help and resources*: **Connect MCP**, **Community** and a support e-mail link. Chapter 5 explains **Connect MCP**.

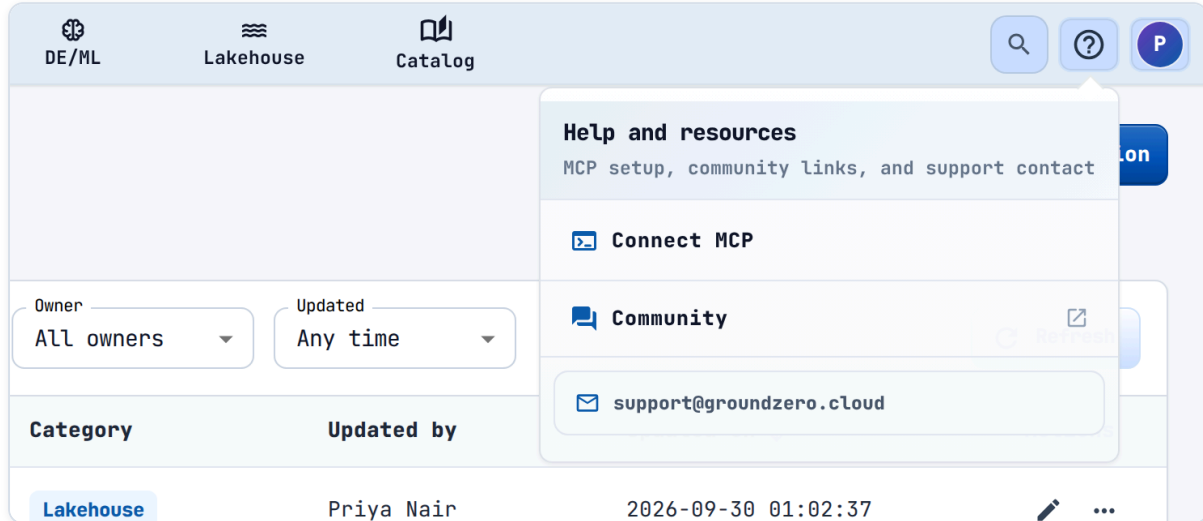


Figure 6. The Help and resources menu.

The **Account** button (your initial) shows your username and e-mail, with **Logout** and **Change Password**. A new password needs at least 8 characters, with an uppercase letter, a lowercase letter and a special character.

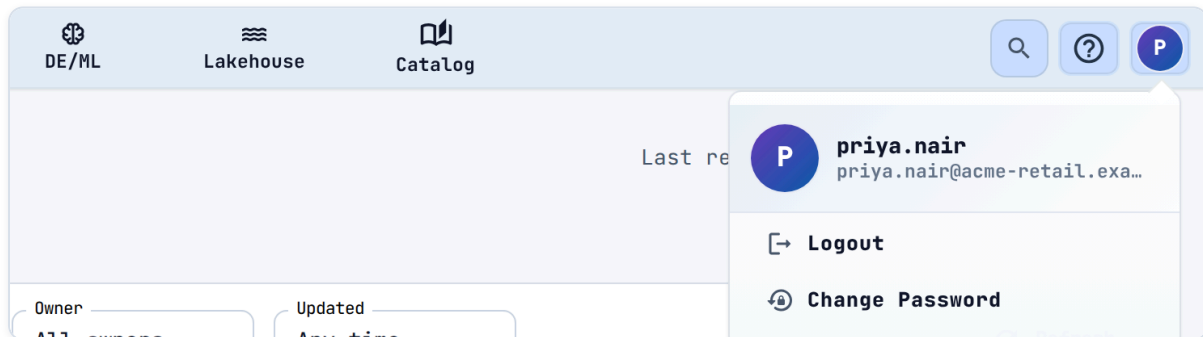


Figure 7. The account menu.

## 2.6 Name things the right way

Almost everything you create follows one naming rule: datasets, visualizations, dashboards, projects, jobs and notebooks. A name starts with a letter and uses only letters, numbers and hyphens. It cannot end with a hyphen, contain two hyphens in a row, or contain spaces.

`acme-weekly-sales` works. `Acme Weekly Sales` does not.

If an action is refused, the message names the permission you need, for example *You do not have "All" or "Add" permissions on Connection*. Send that message to your administrator, who grants permissions in the Access Manager.

## 3 Your first workflow

You will build one small data product from start to finish. Each step uses what the one before it made, and ends with something you can see.



**Figure 2.** The seven steps of the first workflow. Each step uses what the one before it made.

### WHAT YOU NEED FOR THIS CHAPTER

Permission to add connections, to create datasets, visualizations and dashboards, to create projects and run ETL jobs, and to use a compute size. If a step is refused, the message tells you which permission to ask for.

The examples use a PostgreSQL connection called `acme-postgres` (the storefront database) and a Lakehouse connection called `acme-lakehouse`. Use your own names wherever you see them.

### 3.1 Step 1: Connect a data source

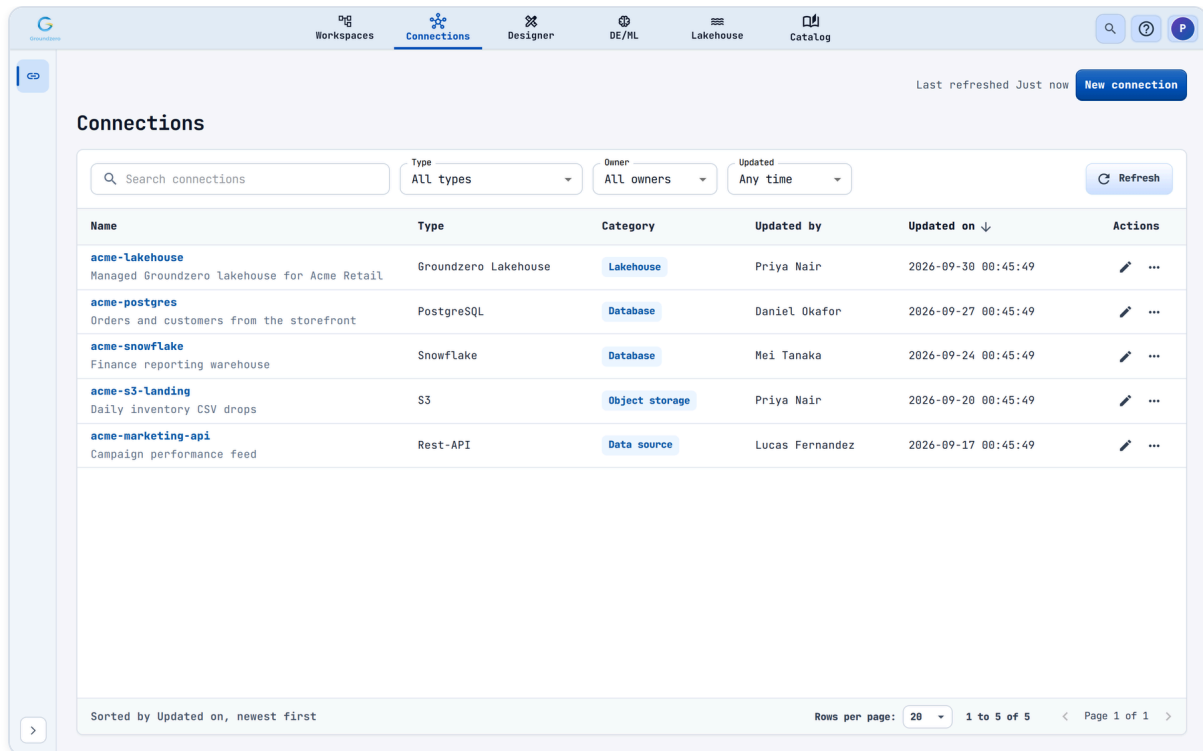
**1**

#### Connections

Tell Groundzero where your data lives. You do this once for each source.

**Why:** Everything else in this guide reads from a connection. When you finish this step, you have a saved link to your data that you can reuse.

Open **Connections** in the top bar. The page lists every connection you can see, with its type, a category (for example Lakehouse, Database or Object storage), who updated it and when. Use the search box and the **Type**, **Owner** and **Updated** filters to find one.



Connections

Search connections

Type: All types

Owner: All owners

Updated: Any time

Refresh

| Name                                                                  | Type                 | Category       | Updated by      | Updated on ↓        | Actions |
|-----------------------------------------------------------------------|----------------------|----------------|-----------------|---------------------|---------|
| <b>acme-lakehouse</b><br>Managed Groundzero lakehouse for Acme Retail | Groundzero Lakehouse | Lakehouse      | Priya Nair      | 2026-09-30 00:45:49 | ...     |
| <b>acme-postgres</b><br>Orders and customers from the storefront      | PostgreSQL           | Database       | Daniel Okafor   | 2026-09-27 00:45:49 | ...     |
| <b>acme-snowflake</b><br>Finance reporting warehouse                  | Snowflake            | Database       | Mei Tanaka      | 2026-09-24 00:45:49 | ...     |
| <b>acme-s3-landing</b><br>Daily inventory CSV drops                   | S3                   | Object storage | Priya Nair      | 2026-09-20 00:45:49 | ...     |
| <b>acme-marketing-api</b><br>Campaign performance feed                | Rest-API             | Data source    | Lucas Fernandez | 2026-09-17 00:45:49 | ...     |

Sorted by Updated on, newest first

Rows per page: 20 1 to 5 of 5 < Page 1 of 1 >

**Figure 8.** The Connections page lists connections with type, category, owner and last update. The pencil edits a connection; the three dots open more actions.

To add a connection:

- 1 Choose **New connection**. The **Add Connection** dialog opens.
- 2 Open **Connection Type** and choose your source. The list is alphabetical and includes databases, warehouses, object stores, streaming systems, business applications, AI providers and the Groundzero Lakehouse. Type to search it.
- 3 Fill in the form that appears. A PostgreSQL connection asks for **Connection Name**, **Port**, **Server**, **Username**, **Database** and **Password**.
- 4 Choose **Test Connection**. A green message confirms that Groundzero can reach the source. Testing saves nothing.
- 5 Choose **Save**.

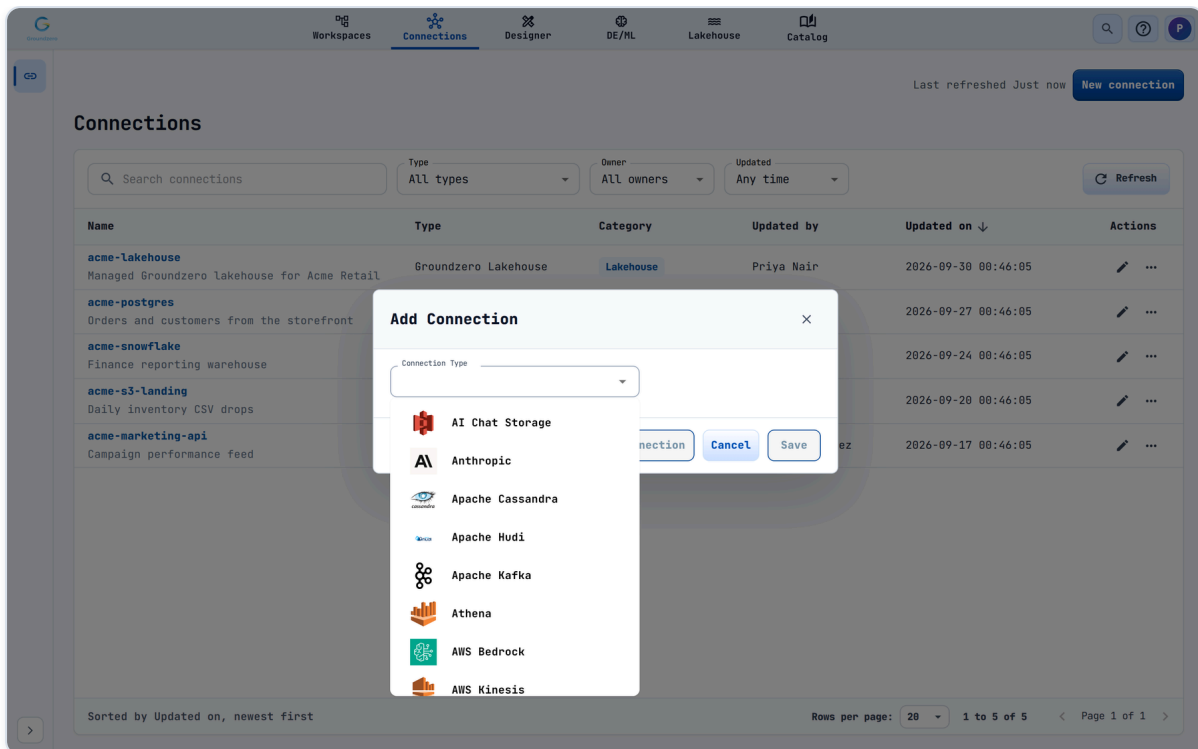


Figure 9. The Connection Type list. Choose the system you want to connect.

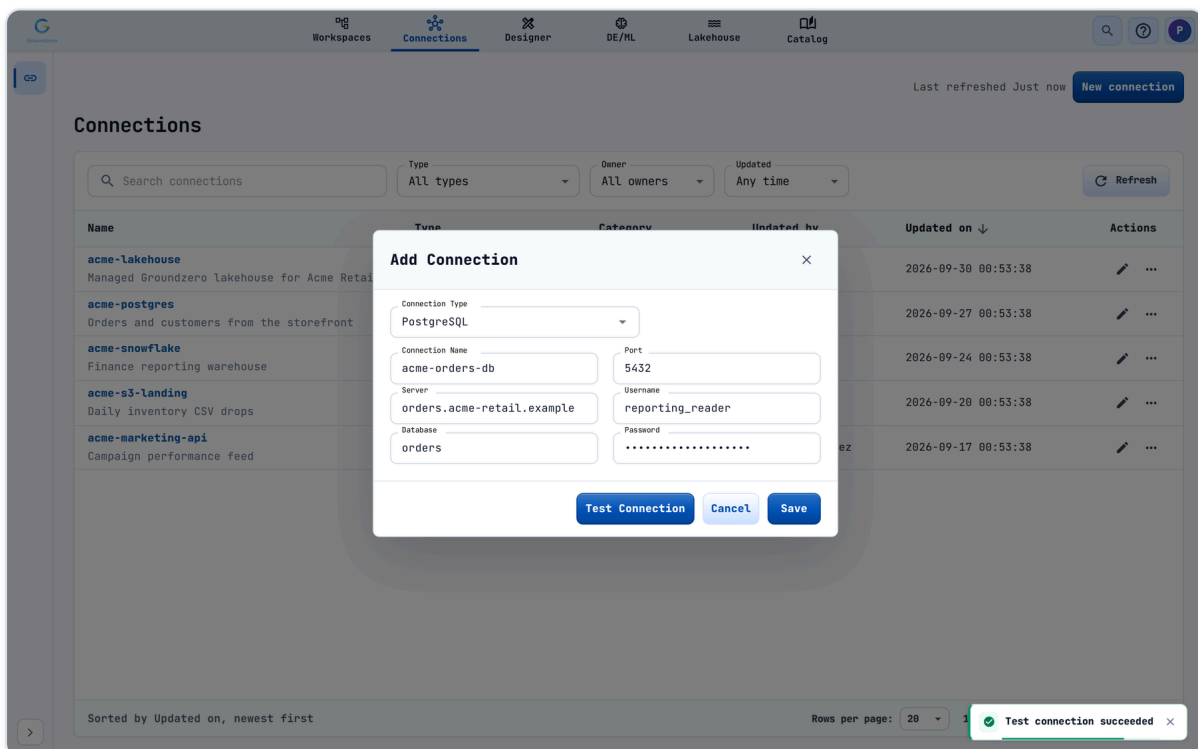


Figure 10. A PostgreSQL connection filled in with sample values. Test Connection and Save become available once every field has a value, and the test result appears as a toast.

**What you should see:** The message *Connection created successfully*, and your connection in the list.

**NAMING A CONNECTION**

A connection name starts with a letter and uses only letters, numbers and single hyphens, for example `acme-postgres`. A bad name is rejected when you choose **Save**.

**SECRETS STAY HIDDEN**

Choose a connection's name to see its **Configured details**. Passwords, tokens and keys appear as hidden values. Your code and datasets use the connection by name and never see the credentials.

**If something goes wrong:** If **Test Connection** and **Save** stay greyed out, a field is empty. Fill in every field.

## 3.2 Step 2: Explore it in the Catalog

**2****Catalog**

See what exists, who owns it and what it means.

**Why:** Before you build on data, check what it is. The Catalog is also where colleagues, and the AI assistant, look first. When you finish this step, you have found an item and described it.

Open **Catalog** in the top bar. The Catalog lists three kinds of things: **Datasets**, **Jobs** (ETL jobs) and **Warehouses** (lakehouse warehouses). You can change only descriptions and tags here. Warehouses appear only if you may list them.

- 1 Type in **Search catalog**, or choose **All**, **Datasets**, **Jobs** or **Warehouses** to narrow the list.
- 2 Select an item. Its header shows the type, owner and last update, with an **Open** button that takes you to the item itself.
- 3 On the **Overview** tab, read or edit the **Description**. You can write it yourself or choose **Draft with AI**. Add **Tags** too.
- 4 For datasets and jobs, the **Contract** tab shows how to call the item from outside Groundzero.
- 5 Select a warehouse and open its **Schema** tab to browse databases, tables and columns, with their sizes and tags.

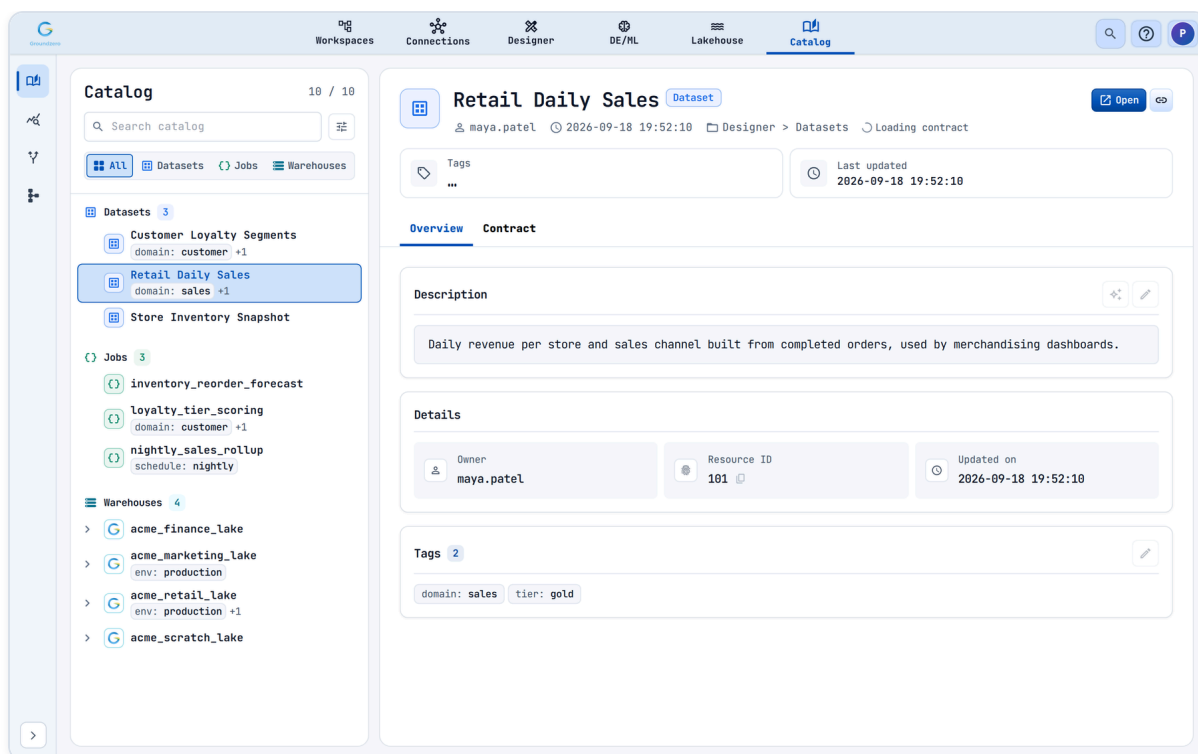


Figure 11. A dataset in the Catalog: description, owner and tags. The left pane lists datasets, jobs and warehouses.

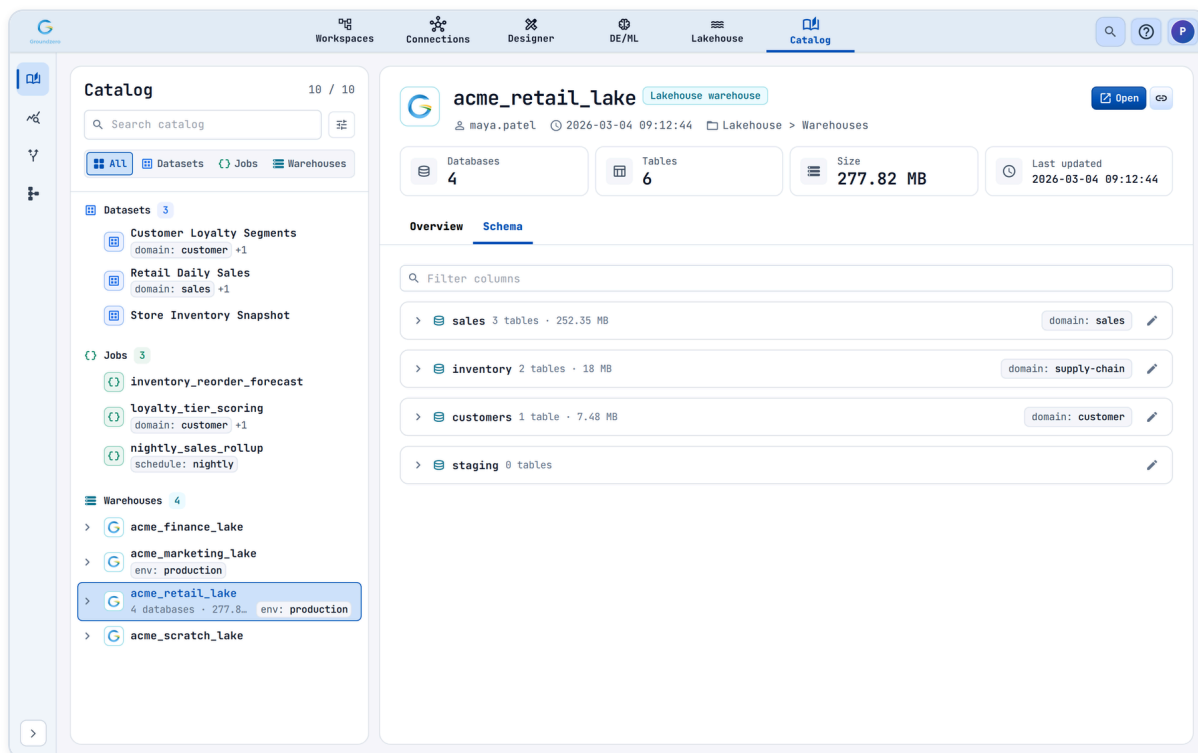


Figure 12. A lakehouse warehouse: databases, tables and size, and a schema tree you can filter by column.

**What you should see:** The description and tags you saved on the item's **Overview** tab.

#### GOOD DESCRIPTIONS PAY OFF

Add a one-sentence description to everything you publish. It is what colleagues find first.

### 3.3 Step 3: Build a dataset

3

#### Designer, Datasets

Turn a query into a named, reusable result.

**Why:** Charts and dashboards read datasets, so a good dataset is the foundation for everything after it. When you finish this step, you have a saved query that returns the rows you want.

- 1 Open **Designer**. The left menu has **Datasets**, **Filters**, **Visualizations** and **Dashboards**. Choose **Datasets**.
- 2 Choose **New Dataset**, type a name such as `acme-weekly-sales` in **Enter Name** and choose **Add**.
- 3 Open the new dataset from the list. The editor opens with the **Query** and **Prep** tabs.
- 4 In the left drawer, open **Connections** and tick the connection to query, for example `acme-postgres`. It expands into databases, tables and columns that you can browse.
- 5 On the **Query** tab, write your SQL. The editor suggests connection, table and column names as you type.
- 6 Leave the runtime picker on **Standard**. Standard runs one SQL query straight against the source and needs no compute size.
- 7 Choose **Run** (the play button). **Results** shows the response time, the row count, the page size and the rows, with each column's data type. **Messages** shows success or the error text.
- 8 Choose **Save** (the disk button).

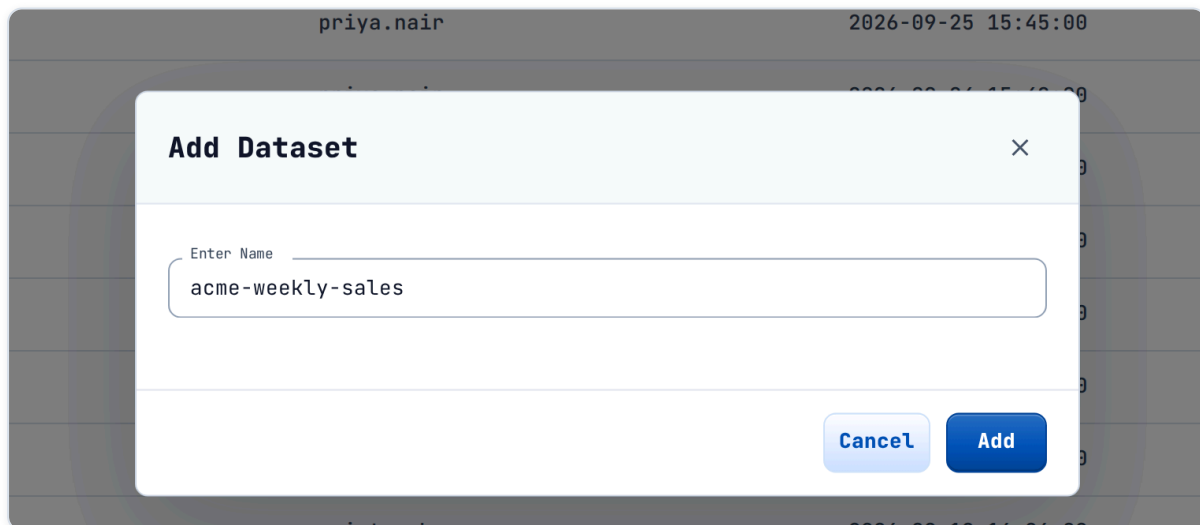


**Figure 13.** The Add Dataset dialog. Names use letters, numbers and hyphens.

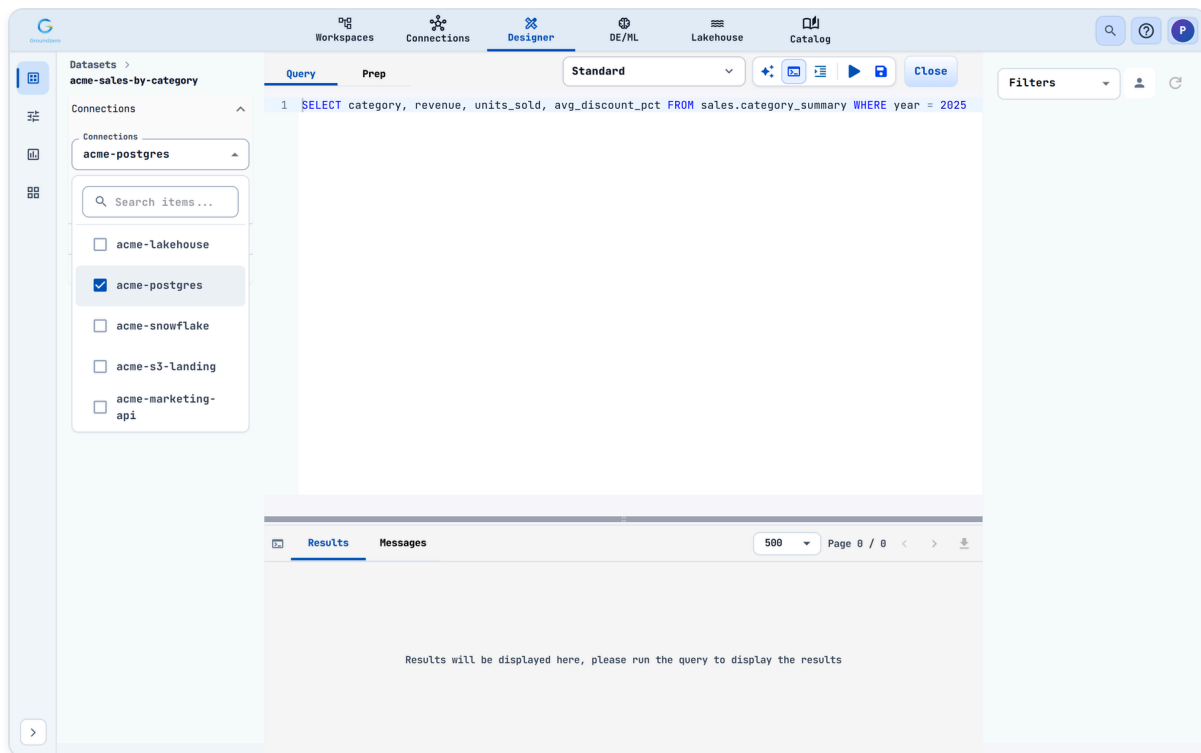


Figure 14. Pick the connection to query. The left drawer lets you choose more than one.

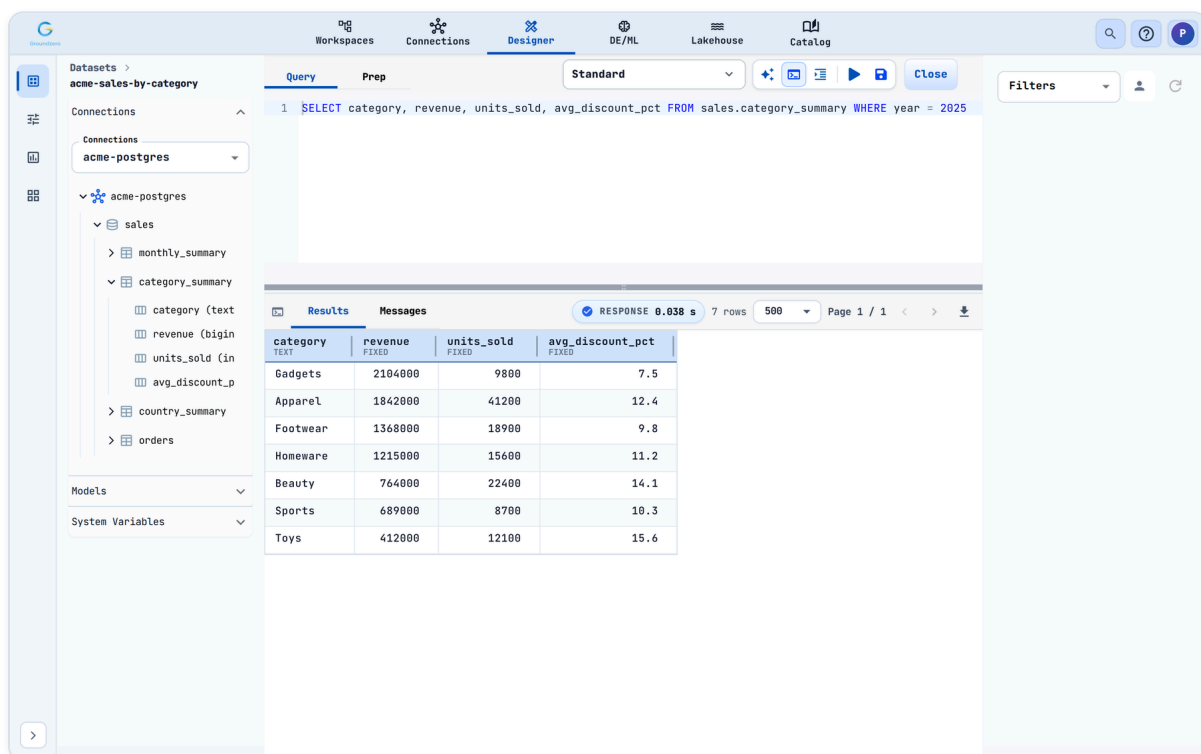


Figure 15. The dataset editor: connection tree and SQL on the left and centre, results below. The columns and 7 rows come from the query.

#### SQL

```
SELECT category, revenue, units_sold, avg_discount_pct
FROM sales.category_summary
WHERE year = 2025
```

**What you should see:** Rows under **Results** after you run the query, and the message *Dataset updated* after you save.

#### LET THE ASSISTANT HELP

The sparkle button in the toolbar (**Generate code with GZ**) opens a code assistant that knows this dataset. Describe the query you want, review what it writes, and run it. Chapter 5 has more.

#### GOOD TO KNOW

Standard works with SQL connections such as PostgreSQL, MySQL, SQL Server, Oracle, Snowflake, Redshift, Google BigQuery and the Groundzero Lakehouse. If you choose another runtime, or more than one connection, the dataset becomes a **code dataset** that you write in Python on the **Prep** tab. Groundzero keeps the results of an identical query in your browser for 10 minutes, so reload the page if you want a fresh start.

**If something goes wrong:** If Run says *Please select Connection*, tick a connection in the left drawer first.

## 3.4 Step 4: Create a visualization

4

### Designer, Visualizations

Turn rows into a chart.

**Why:** A chart makes a result easy to read at a glance. When you finish this step, you have a saved chart on your dataset.

- 1 In **Designer**, choose **Visualizations**, then **New Visualization**. Enter a name such as `revenue-by-category` and choose **Add**. Groundzero does not open it for you, so choose its name in the list.
- 2 In the **Setup** tab on the right, open **Chart Type** and choose a type. Hover over an icon to see its name. There are 22 types: Bar, Vertical Bar, Stacked Bar, Stacked Vertical Bar, Combo, Line, Pie, Half Donut, Donut, Gauge, Bubble, Scatter, Candle, Tree Map, Geo, Funnel, Table, Pivot, Sunburst, Sankey, Card and Graph.
- 3 In **Choose Dataset**, pick the dataset from step 3.
- 4 Map columns to the roles of the chart. A bar chart asks for a vertical axis (the category) and a series (the measure, here `revenue`). Add a **Chart Title**.
- 5 Open **Customize** to style the chart: axes and titles, colours, legends and labels.
- 6 Choose the **Save** icon (the disk at the top right). Groundzero does not save automatically, so save before you leave.

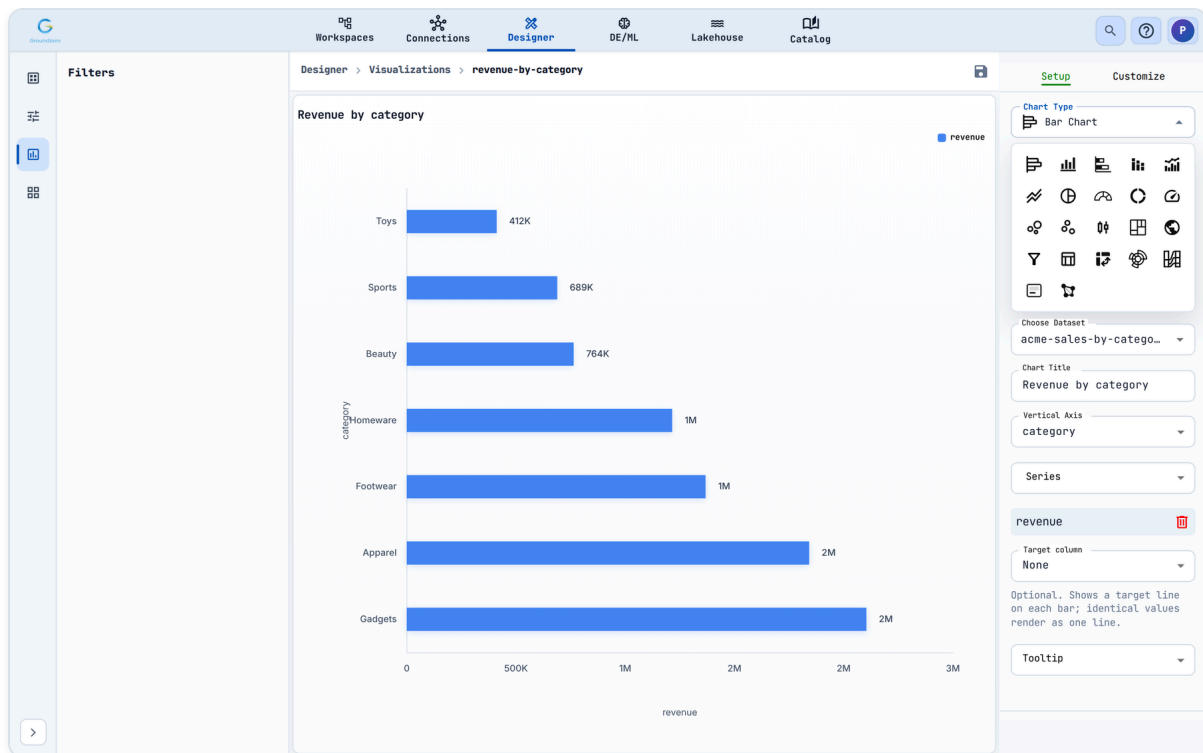


Figure 16. Choose a chart type: 22 icons, each with a name on hover.

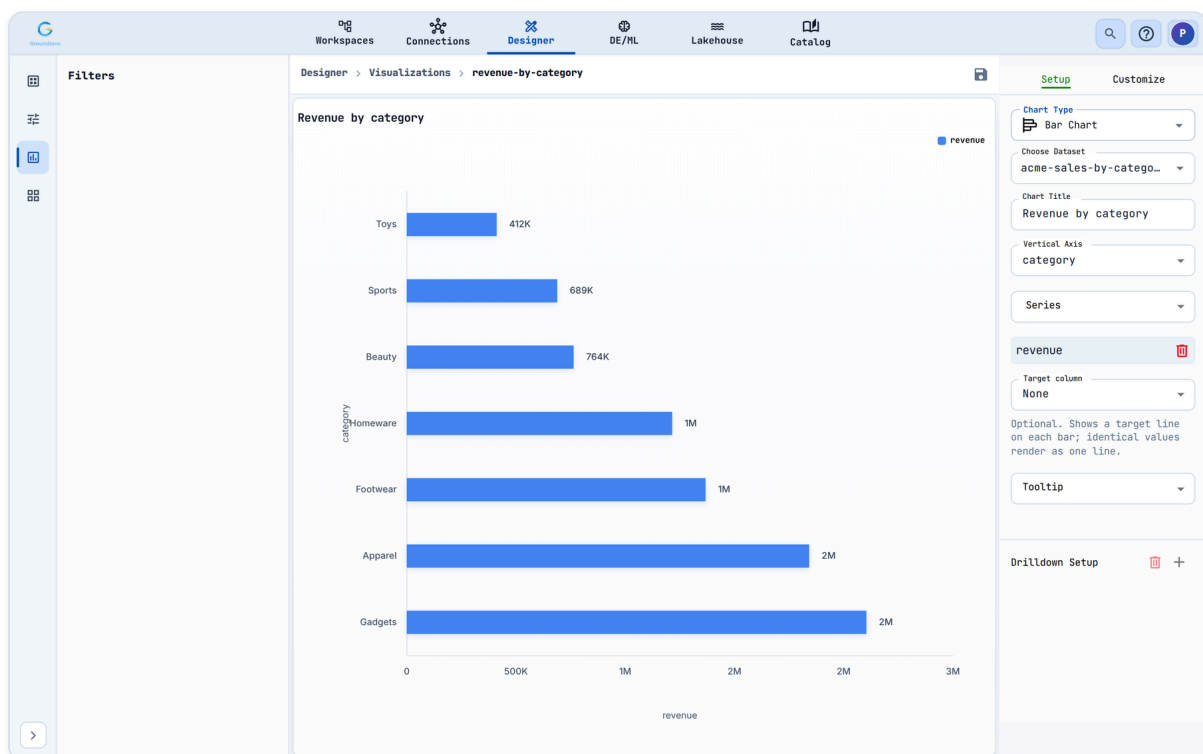


Figure 17. A bar chart of revenue by category. The Setup panel on the right shows the chart type, dataset, title, axis and series.

**What you should see:** Your chart, drawn from the dataset's rows.

**CHARTS ALWAYS SHOW CURRENT DATA**

Groundzero saves the design of your chart, not its rows. It runs the dataset again every time the chart opens, so a saved chart is always up to date.

### 3.5 Step 5: Compose and publish a dashboard

**5****Designer, Dashboards and Workspaces**

Place charts on a page and share it.

**Why:** A dashboard puts several charts on one page, and publishing lets other people open it. When you finish this step, your chart is on a dashboard in a workspace folder.

- 1 In **Designer**, choose **Dashboards**, then **New Dashboard**. Name it with the usual rule, for example `Acme-Sales-Overview`, and choose **Add**. Open it from the list.
- 2 In the right panel, choose the first icon, **Select visualizations**. Tick the visualizations to add. Each one becomes a tile on the canvas. Drag a tile to move it, and drag its edge to resize it. Untick a visualization to remove its tile.
- 3 Use the tabs at the bottom to organise the dashboard. Choose **+** to add a tab, and use the menu beside a tab name to rename, duplicate or delete it.
- 4 The **Setup** tab holds **Triggers** (click a chart to filter other charts or jump to another tab), **Text and Shapes** and **Dashboard Filters**. The **Customize** tab styles tiles, text and backgrounds.
- 5 Choose **Save dashboard** (the disk icon at the top right).
- 6 Choose **Publish** and confirm. Publishing makes the dashboard available to import into a workspace.
- 7 Open **Workspaces**, choose a folder, choose **Import Dashboard**, pick the published dashboard and choose **Import**.

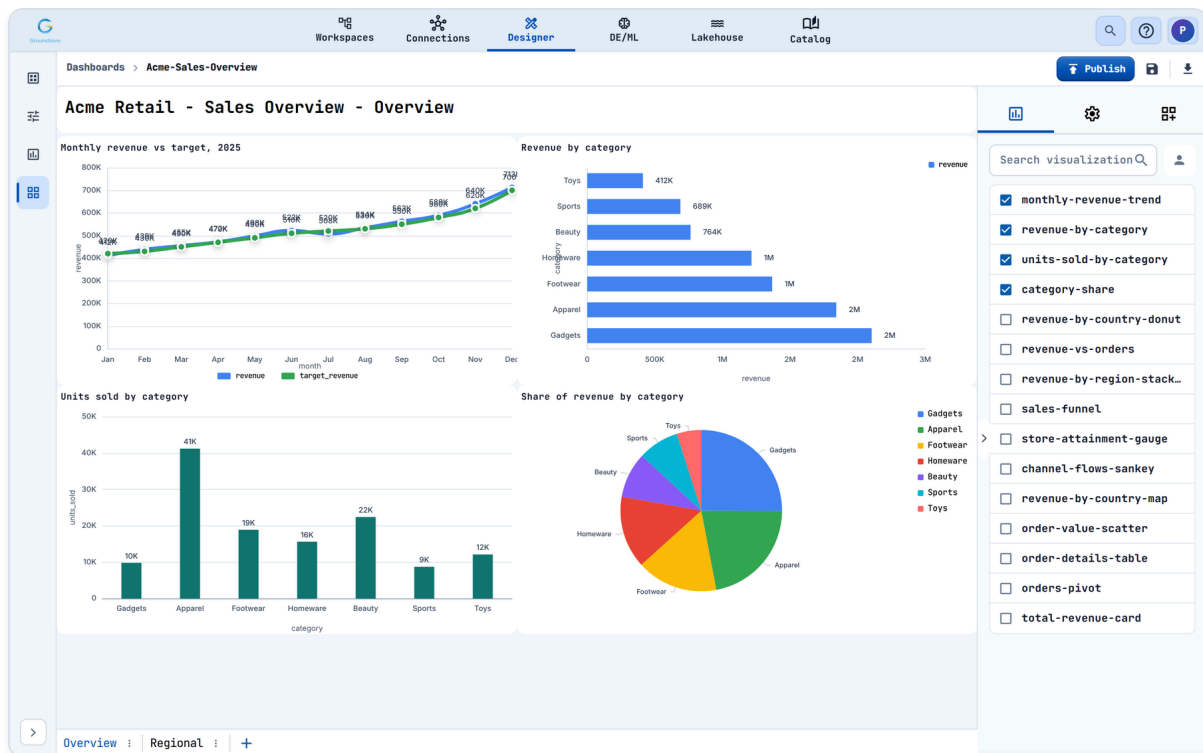


Figure 18. A dashboard in the editor. The canvas shows four tiles on the Overview tab; the right panel lists visualizations with check boxes; the tabs are at the bottom.

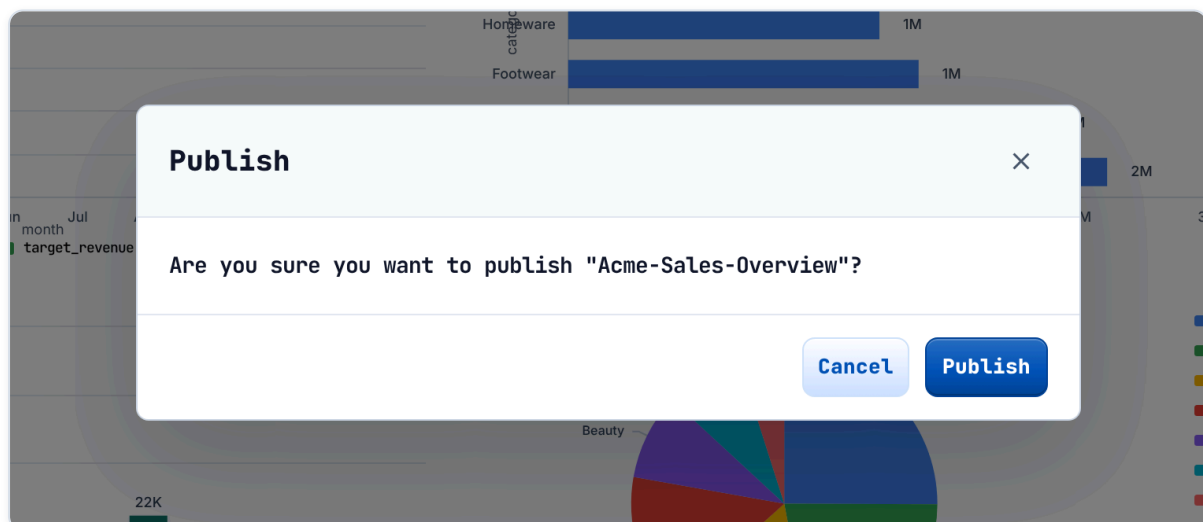


Figure 19. The Publish confirmation.

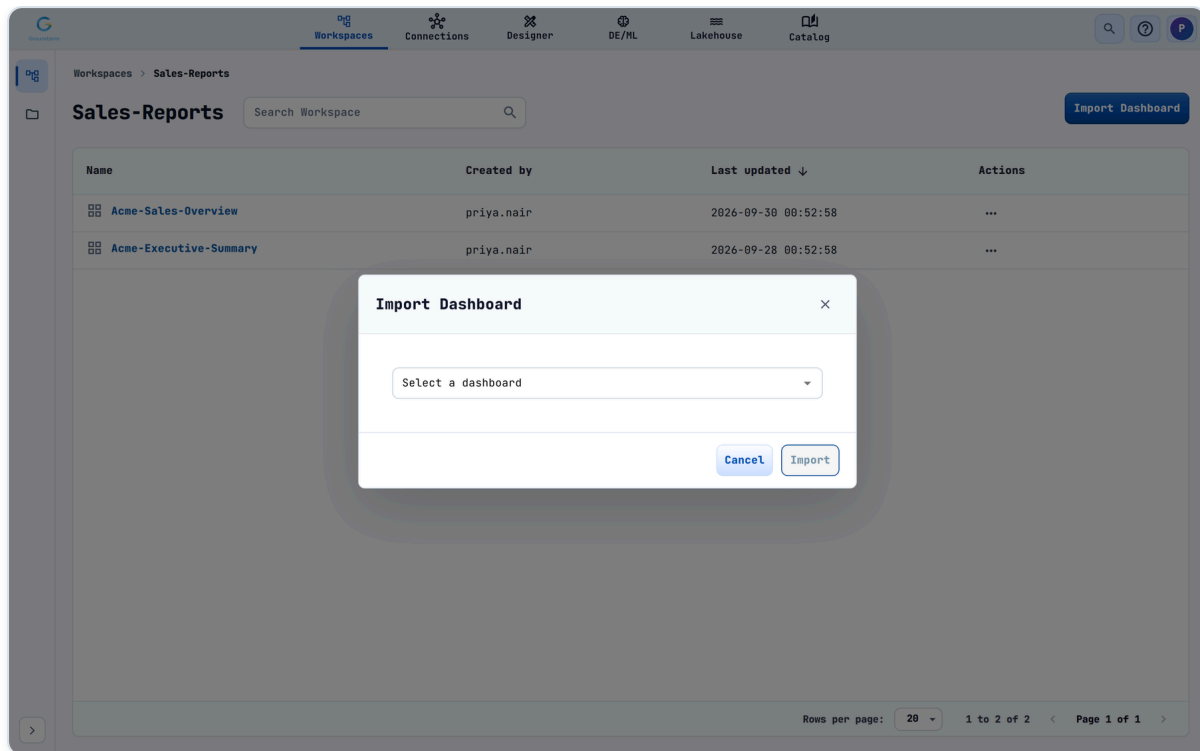


**Figure 20.** Inside a workspace folder, Import Dashboard lists the published dashboards.

**What you should see:** The dashboard in the workspace folder. Open it from there to read it.

#### SAVE FIRST, THEN PUBLISH

Publish does not save your changes. It publishes the last saved version, and it replaces a published dashboard that has the same name. Choose **Save dashboard** and then **Publish**.

#### TAKING A DASHBOARD WITH YOU

The export button in the dashboard header offers **Export To PDF** and **Export to Excel**. Each export covers the tab you are on.

**If something goes wrong:** If you see *Dashboard is empty, add some items and save.*, add at least one visualization first.

## 3.6 Step 6: Automate with a job and a schedule

6

### DE/ML: projects, jobs and the Scheduler

Make the data move on its own.

**Why:** A dataset answers a question when you ask it. A job prepares and moves data without you, for example every morning, so the data behind your dashboard stays current. Jobs live in a project. When you finish this step, you have a job that runs by itself every day.

#### Create a project

- 1 Open **DE/ML**. The **Projects** page lists your projects with their owner and last update.

- 2 Choose **New Project**, enter a **Project Title** (the usual name rule), and choose **Add**.
- 3 Choose the project name. Its tabs are **Jobs**, **Notebooks**, **Models** and **Agents**.

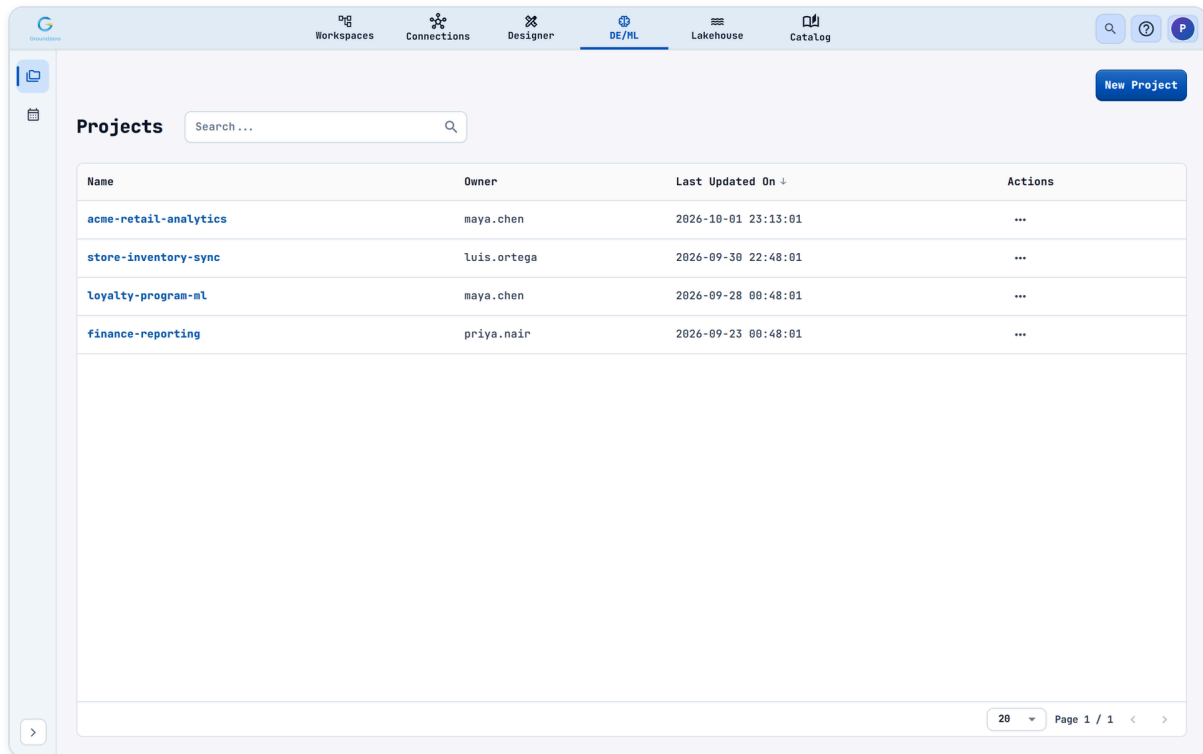


Figure 21. The Projects page.

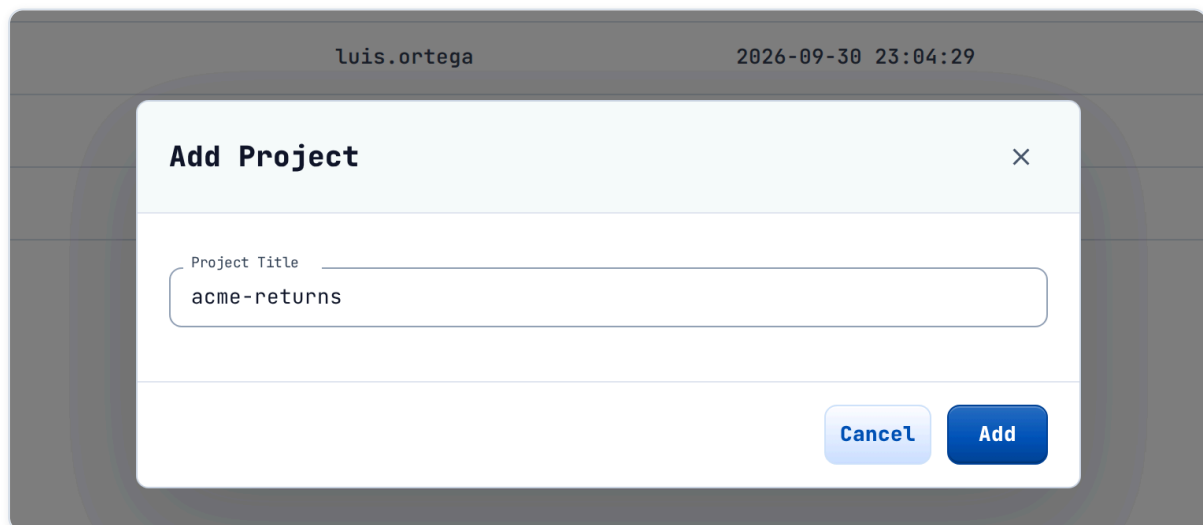


Figure 22. The Add Project dialog.

## Create and run a job

- 1 On the **Jobs** tab, choose **New Job**, enter a **Job Title** such as `daily-sales-rollup` and choose **Add**.
- 2 Choose the job name to open the editor. In the left drawer, tick the **Connections** the job uses. Your code can use only the connections you tick.
- 3 Write the Python code. Jobs run on a Spark or DuckDB runtime. The example below reads yesterday's orders from the lakehouse, adds them up by region and appends the result to a table.
- 4 Open the runtime picker in the toolbar. Choose a **Runtime** and a **Compute** size, then close the panel.

- 5 Choose **Run**. The **Messages** pane shows the log while the job runs and ends with the job status. Choose **Save** when you are happy with the job.

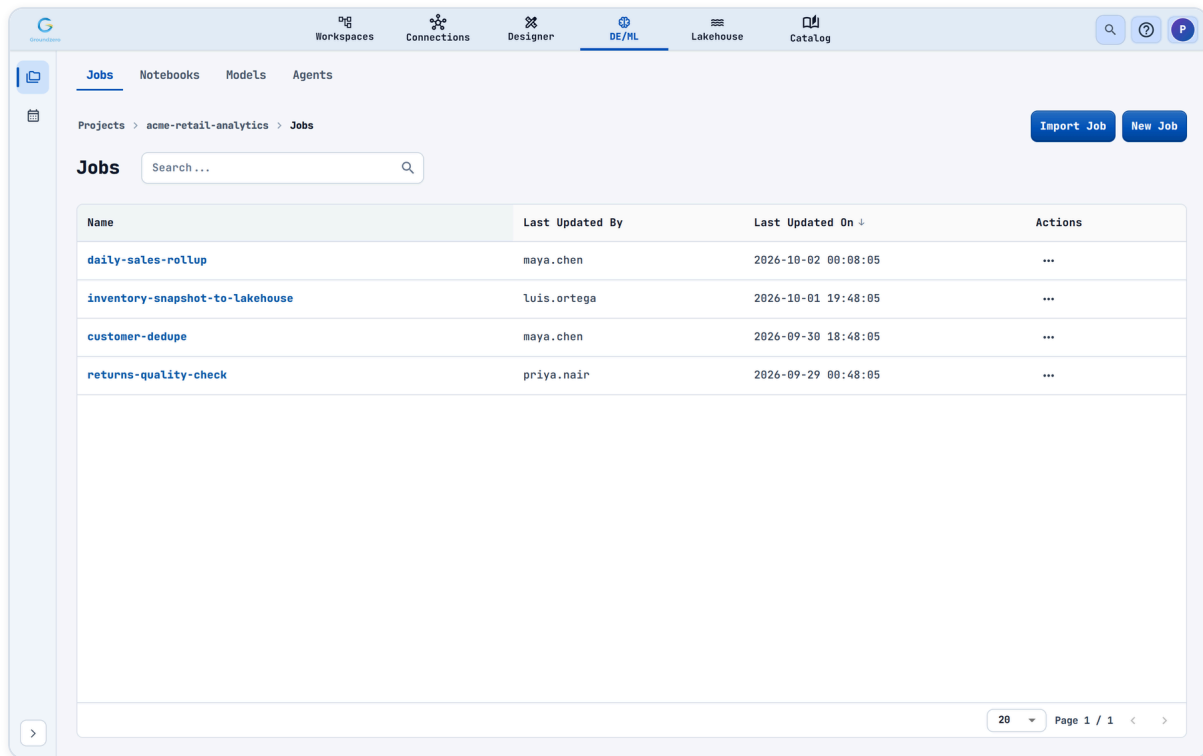


Figure 23. The Jobs tab of a project. Import Job and New Job are at the top right.

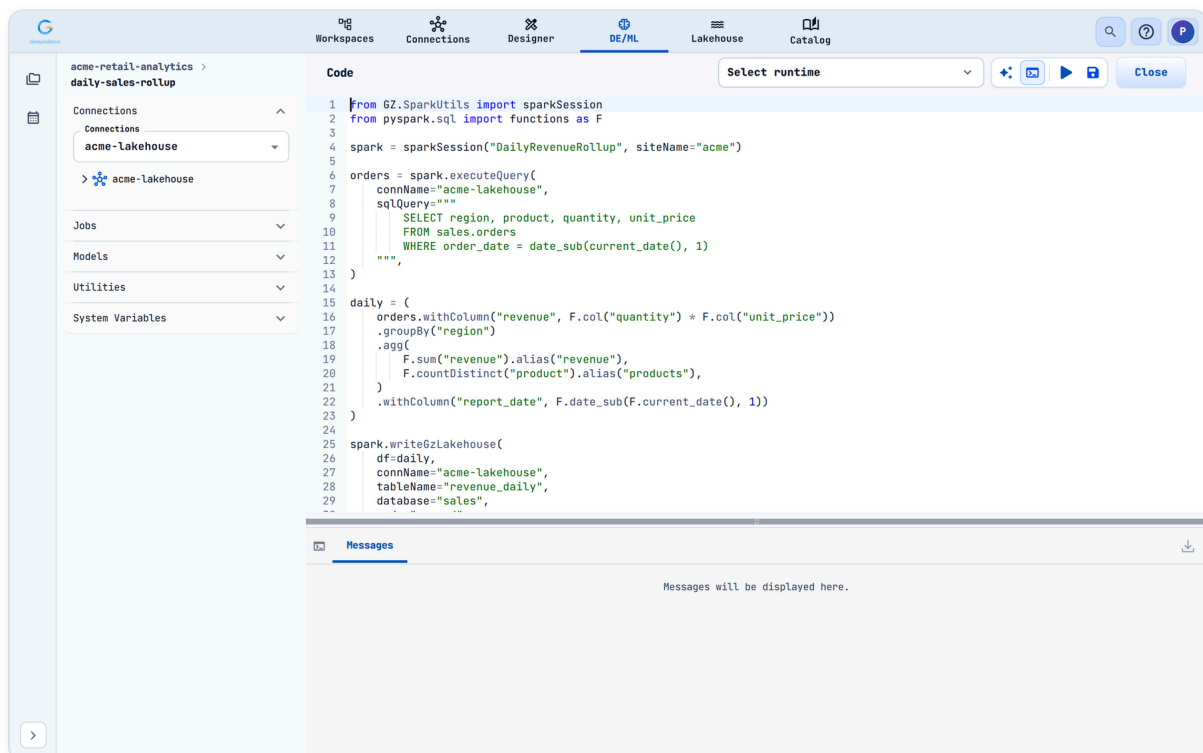


Figure 24. The job editor: connections and utilities on the left, code in the centre, the runtime picker and Run in the toolbar.

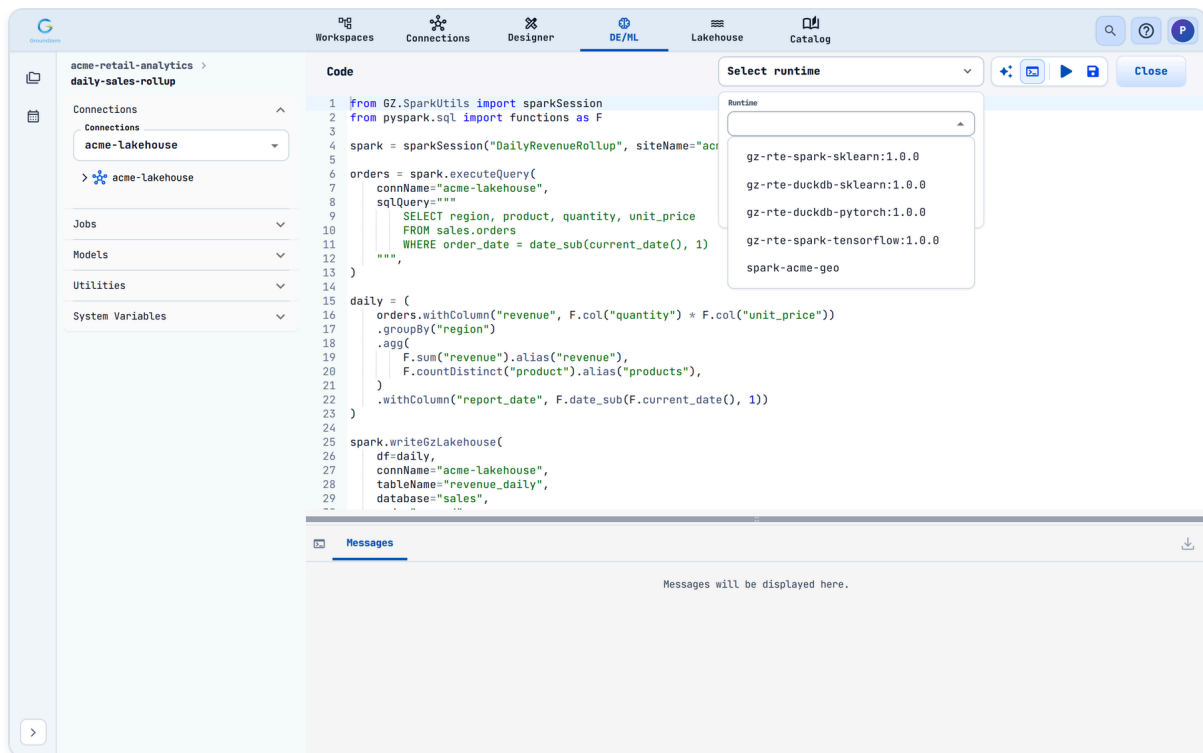


Figure 25. The runtime picker lists the runtimes your administrator made available.

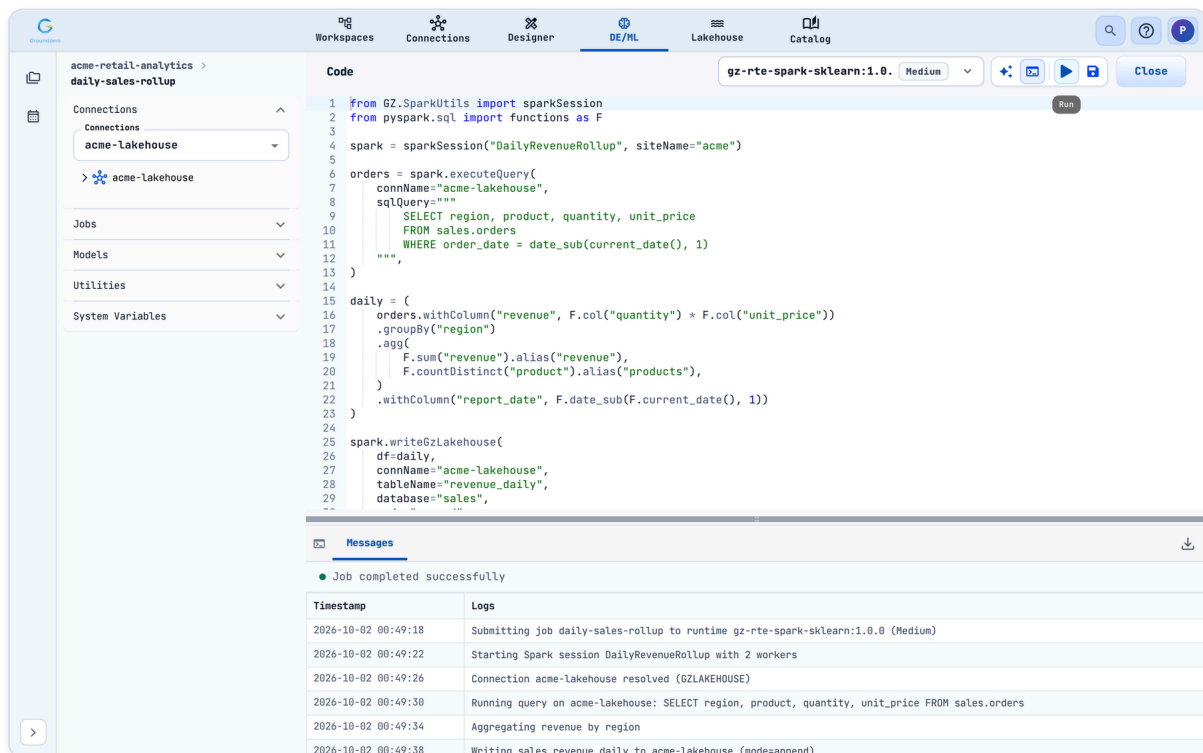


Figure 26. A finished run: Job completed successfully, followed by the timestamped log lines.

## PYTHON

```

from GZ.SparkUtils import sparkSession
from pyspark.sql import functions as F

spark = sparkSession("DailyRevenueRollup", siteName="acme")

orders = spark.executeQuery(
    connName="acme-lakehouse",
    sqlQuery="""
        SELECT region, product, quantity, unit_price
        FROM sales.orders
        WHERE order_date = date_sub(current_date(), 1)
    """,
)

daily = (
    orders.withColumn("revenue", F.col("quantity") * F.col("unit_price"))
    .groupBy("region")
    .agg(
        F.sum("revenue").alias("revenue"),
        F.countDistinct("product").alias("products"),
    )
    .withColumn("report_date", F.date_sub(F.current_date(), 1))
)

spark.writeGzLakehouse(
    df=daily,
    connName="acme-lakehouse",
    tableName="revenue_daily",
    database="sales",
    mode="append",
)

outputDF = daily.orderBy(F.desc("revenue"))

```

**What you should see:** *Job completed successfully* at the end of the log.

## WHY OUTPUTDF?

When your script finishes, Groundzero looks for a variable named `outputDF` at the top level of your code. A Spark DataFrame, a pandas DataFrame or a list of dictionaries works. If it is missing, the job still succeeds but shows no preview.

## Schedule the job

Open **DE/ML**, then choose the calendar icon in the left menu (**Scheduler**).

- 1 Choose **New Task**.
- 2 Choose **Select Project** and **Select Job**.
- 3 Choose the **RTE** box, pick a **Runtime** and a **Compute** size, and choose **Save** in the panel.
- 4 Choose **Batch** for a repeating schedule. **Streaming**, the default, runs the job once.
- 5 Type a **Name** (letters, digits, dot, hyphen and underscore; no spaces) and pick the **Start date** and **Start time**.

- 6 Under **Recurrence**, set **Repeat** (Minutes, Hourly, Daily, Weekly, Monthly or Yearly) and **Ends** (Never or On a date).
- 7 Choose **Save**.

Figure 27. The RTE panel inside New Task records the runtime and compute as one value.

Figure 28. A Batch task that repeats daily. Save is available once every required field is set.

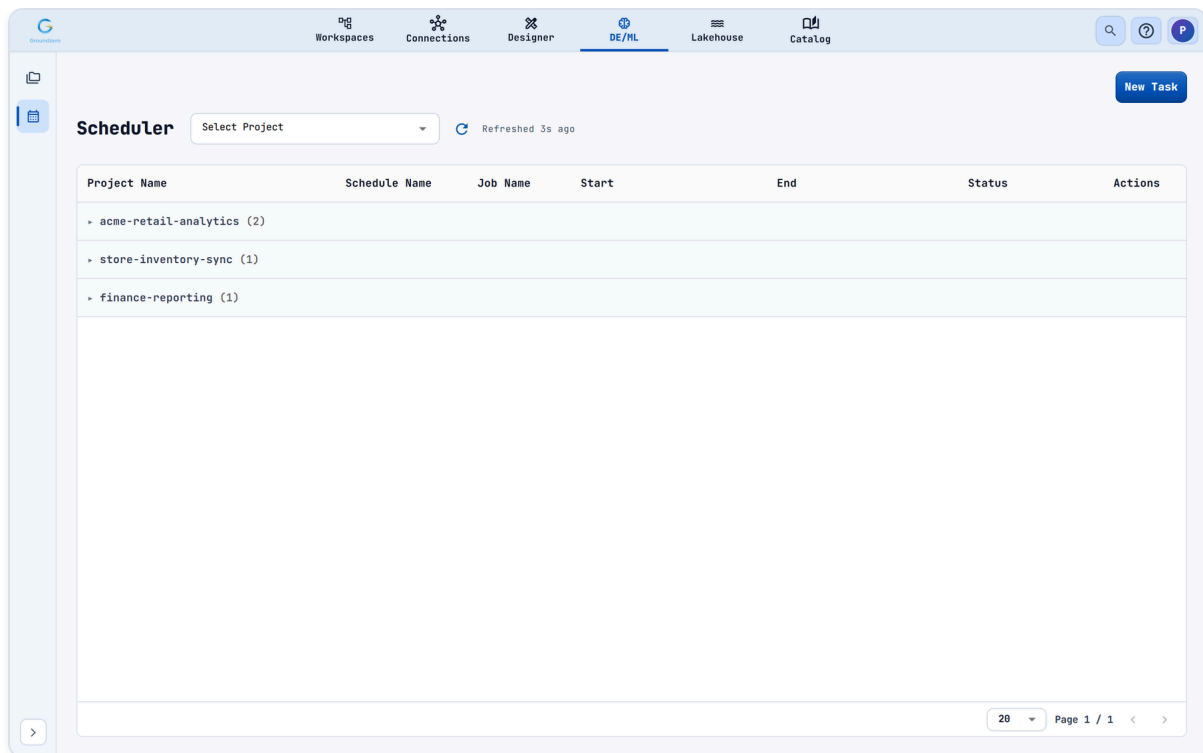


**Figure 29.** The Scheduler lists tasks grouped by project, with status and actions such as Run Once, Abort, Last 10 runs and View Logs.

**What you should see:** The message *Job scheduled successfully*, and your task in the Scheduler. Choose **Run Once** on its menu to check that it works without waiting, then **View Logs** to read the result.

#### SCHEDULES RUN IN UTC

Groundzero converts the start time you pick from your own time zone, and the repeating pattern then follows UTC. A daily task started at 06:00 in India runs at 00:30 UTC every day.

**If something goes wrong:** If you see *Please select Compute* or *Please select RTE and Compute*, open the runtime picker and choose both a runtime and a compute size. If the **Compute** list says *Insufficient Permission*, ask your administrator for access to compute.

## 3.7 Step 7: Check lineage

7

### Catalog, Data Lineage

See where a table comes from and what depends on it.

**Why:** Lineage shows which jobs and datasets write a table and which ones read it. You will use it to check your work, and later to see what a change would affect. Groundzero records lineage when datasets and jobs run, so there is nothing to set up. When you finish this step, you can see how your job fits into the bigger picture.

- 1 Open **Catalog**, then **Data Lineage** in the left menu.
- 2 Under **Explore**, open **Connection** and choose a connection that supports lineage. The **Tables** tab lists its tables. The **Columns** tab searches by column name.

- 3 Select a table. The map shows the table in the middle, the jobs and datasets that write it on the left (upstream), and what reads it on the right (downstream).
- 4 Use the **Upstream** and **Downstream** steppers to widen or narrow the view. If a table has more run history than the map shows, choose **Show older runs**.
- 5 Select a table, job or flow in the map to read its details in the right panel (**Overview**, **Schema**, **Insights**, **Impact**). Evidence is labelled **Derived from code** (found by analysing your queries and scripts) or **Runtime observed** (seen in real runs).
- 6 Choose **Export** to download the lineage. Choose **Hide lineage details** (the **Details** handle at the right edge) when you want more room for the map.

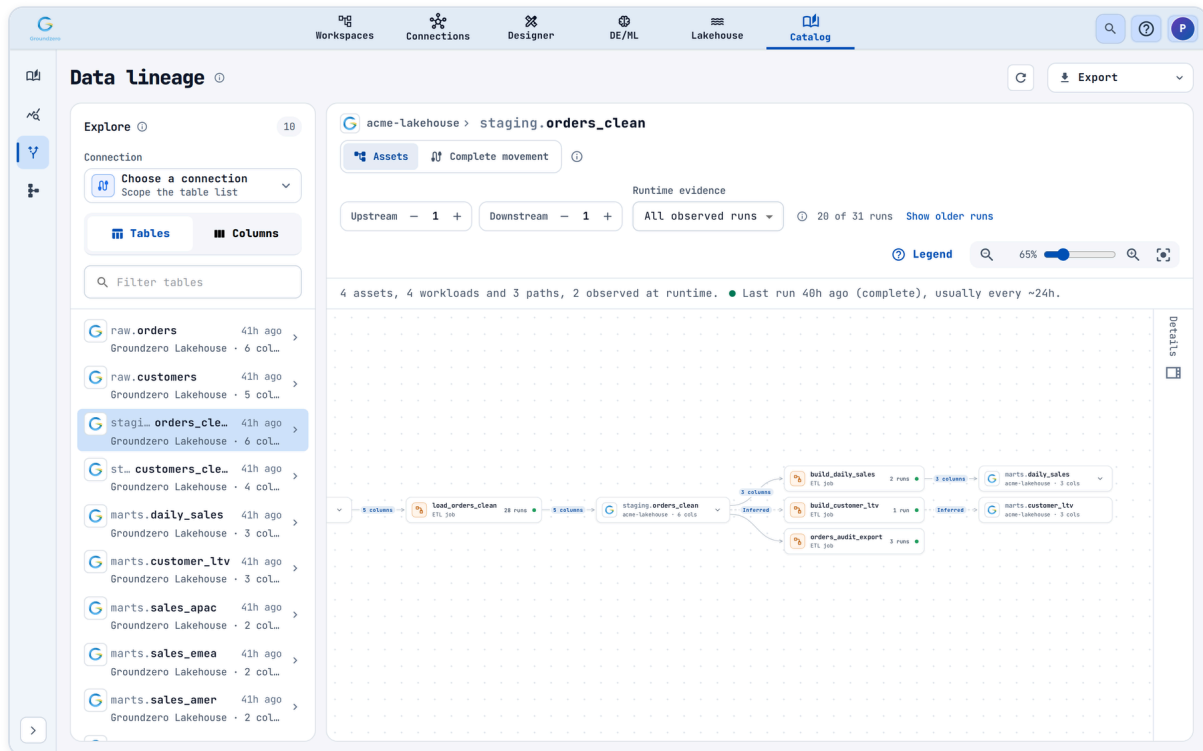


**Figure 30.** The lineage map of a table with the details panel hidden: the job that loads it upstream, and the jobs and tables that read it downstream. The strip above the map summarises assets, workloads and runs.

**What you should see:** The table you chose, with the job that writes it on one side and the jobs and tables that read it on the other.

#### USE LINEAGE BEFORE YOU CHANGE SOMETHING

Before you rename a column or retire a table, open its lineage and its **Impact** tab to see the jobs and tables that depend on it.

**If something goes wrong:** If the page says *Choose a table to trace*, pick a connection and then a table.

## 3.8 What you built

You have now used every main area of Groundzero.

| Step | Where                            | What you made                            |
|------|----------------------------------|------------------------------------------|
| 1    | Connections                      | A saved link to a data source            |
| 2    | Catalog                          | A description and tags on what you found |
| 3    | Designer, Datasets               | A named SQL dataset                      |
| 4    | Designer, Visualizations         | A chart on that dataset                  |
| 5    | Designer, Dashboards; Workspaces | A published dashboard in a workspace     |
| 6    | DE/ML                            | A project, a job and a daily schedule    |
| 7    | Catalog, Data Lineage            | A map of how it fits together            |

## 4 Notebooks and runtimes

Notebooks are for exploring: you run code a cell at a time and see the result straight away. First you choose where your code runs.

### 4.1 Choose a runtime

Groundzero runs your code on a **runtime**. You pick one wherever you create a dataset, a job or a notebook. There are three kinds.

|                      | Standard                             | DuckDB                                                       | Spark                                          |
|----------------------|--------------------------------------|--------------------------------------------------------------|------------------------------------------------|
| Good for             | Charts, dashboards and quick lookups | Fast analysis, joins and ETL on data of small to medium size | Large data and heavy ETL                       |
| Compute size         | None to choose                       | You choose                                                   | You choose, plus the minimum number of workers |
| Connections          | One SQL source or the lakehouse      | Several                                                      | Several                                        |
| Can write data       | No, it reads only                    | Yes                                                          | Yes                                            |
| Where you can use it | Datasets                             | Datasets, jobs, notebooks                                    | Datasets, jobs, notebooks                      |

Work down this list and stop at the first match.

- 1 One query on one SQL source or the lakehouse, for a chart or dashboard: choose **Standard**.
- 2 Several connections, writes, or Python beyond a few lines of pandas: go on to the next question.
- 3 The data is small to medium in size: choose **DuckDB**. It starts quickly.
- 4 The data is large, or you need joins across a lot of data: choose **Spark**.
- 5 You need scikit-learn, PyTorch or TensorFlow: choose the matching machine-learning version of DuckDB or Spark.

#### START SMALL

Try your idea on Standard or DuckDB, and move to Spark when a job outgrows it. The helper code for DuckDB and Spark has the same shape, so moving a job over is quick.

### 4.2 Pick a runtime and compute size

The runtime picker is a compact control that shows the runtime name and a small badge for the compute size. Select it to open a panel.

| Field           | What to choose                                                                                                                                                                                |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Runtime         | The environment to run in. <code>Standard</code> is for datasets only. The rest are the runtimes your administrator made available. If the list is empty, it reads <i>No RTEs Available</i> . |
| Compute         | A named size with memory and processors. If it reads <i>Insufficient Permission</i> , ask your administrator for access to compute.                                                           |
| Minimum Workers | Spark only: how many workers to start with, never fewer than 1.                                                                                                                               |
| GPU Size        | How many GPUs to use.                                                                                                                                                                         |

If you need extra Python packages, ask your administrator to build a custom runtime. It then appears in the same list.

### 4.3 Run your first notebook

By the end of this task you have run code in a notebook and seen a data table appear under it.

- 1 Open a project and choose the **Notebooks** tab. Choose **New Notebook**, enter a name, and choose **Add**.
- 2 Open the notebook. Open the runtime picker, and choose a DuckDB or Spark runtime and a compute size.
- 3 In the left drawer, choose the **Connections** the notebook should use.
- 4 Choose **Start session**. Groundzero shows *Waiting for notebook runtime to become ready* while the runtime starts.
- 5 Type code in a cell and press `Shift` + `Enter`, or choose **Run**. The output, including data tables, appears under the cell.
- 6 Choose **Stop session** when you are done, so the compute is released.

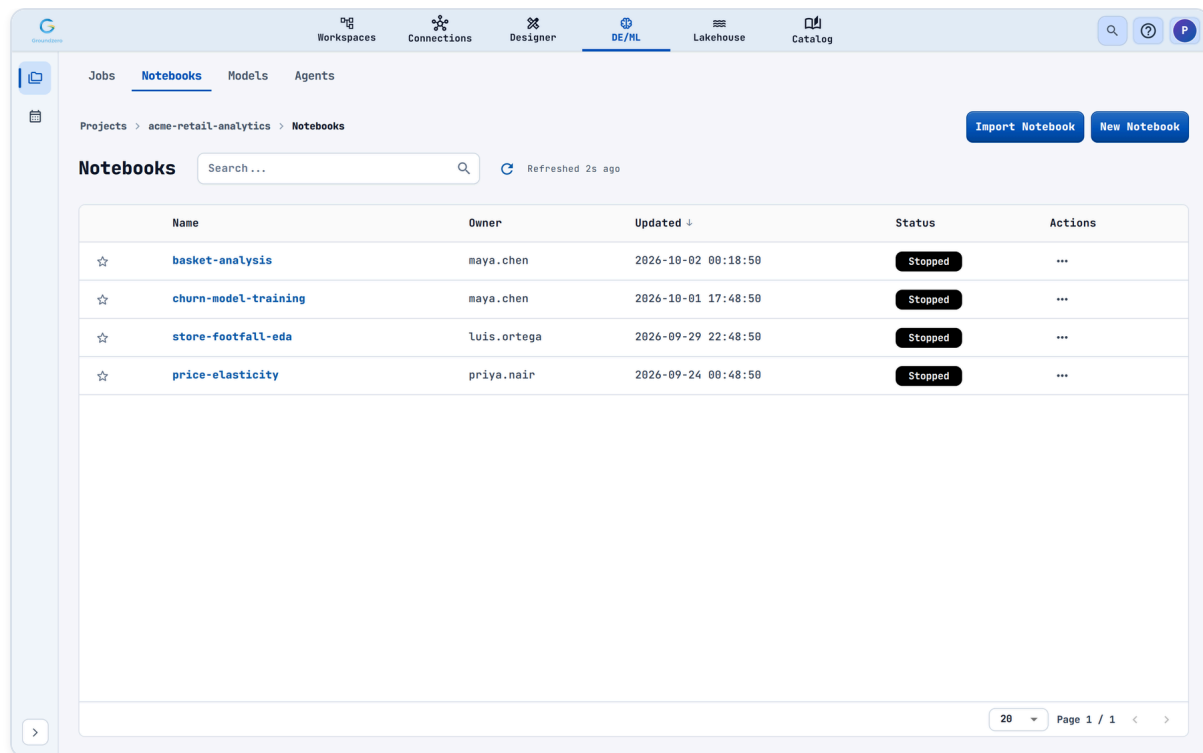


Figure 31. The Notebooks tab lists notebooks with owner, update time and status.

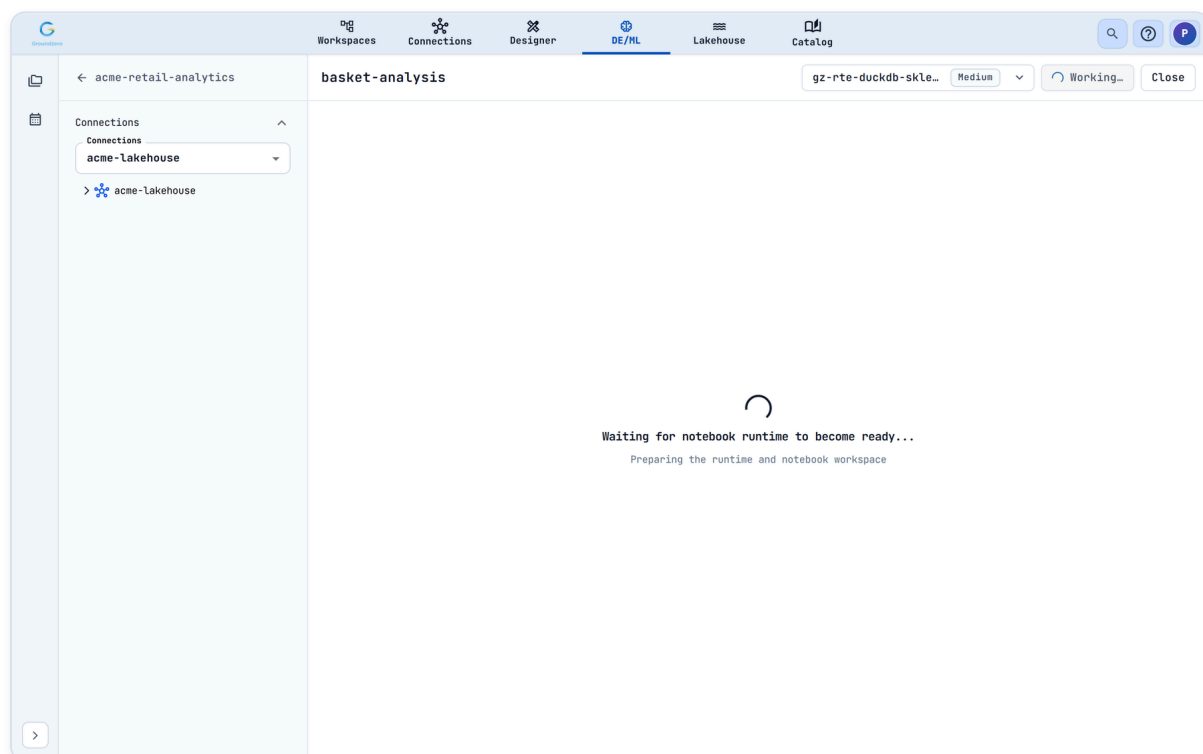


Figure 32. A new session is starting: a runtime and compute are chosen, and a connection is selected.

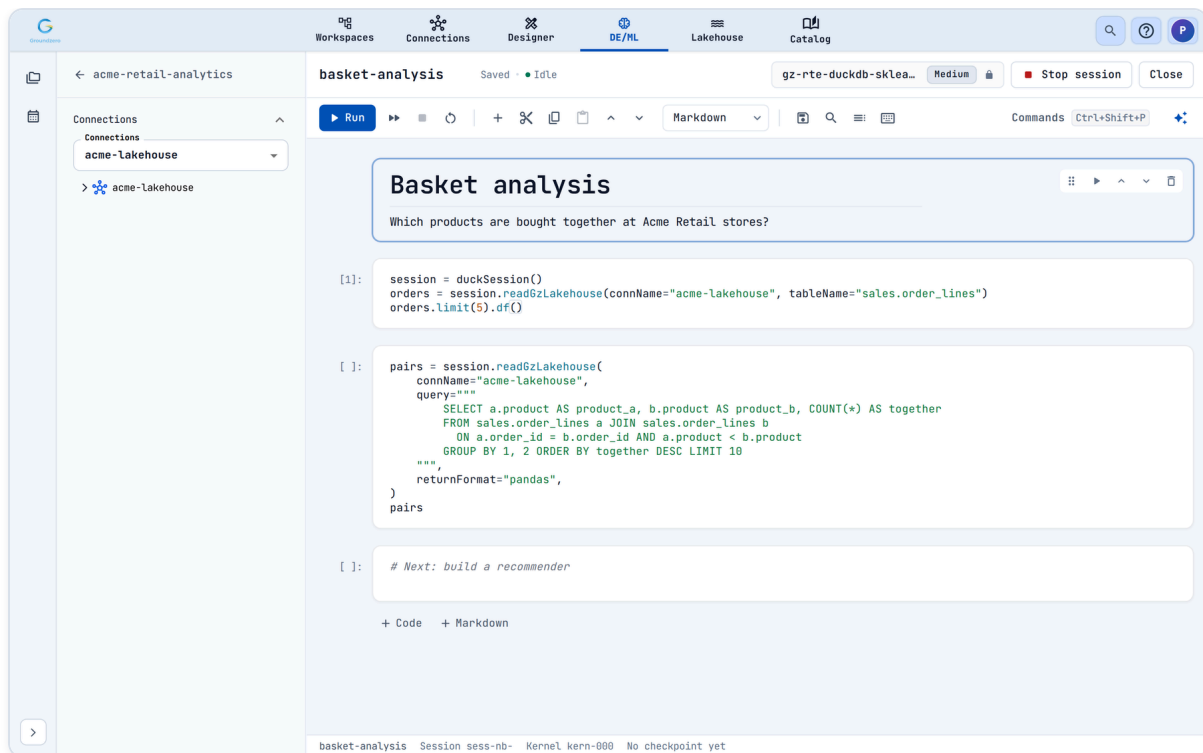


**Figure 33.** A running DuckDB notebook. The toolbar has Run, cell actions and the notebook commands; Stop session is at the top right.

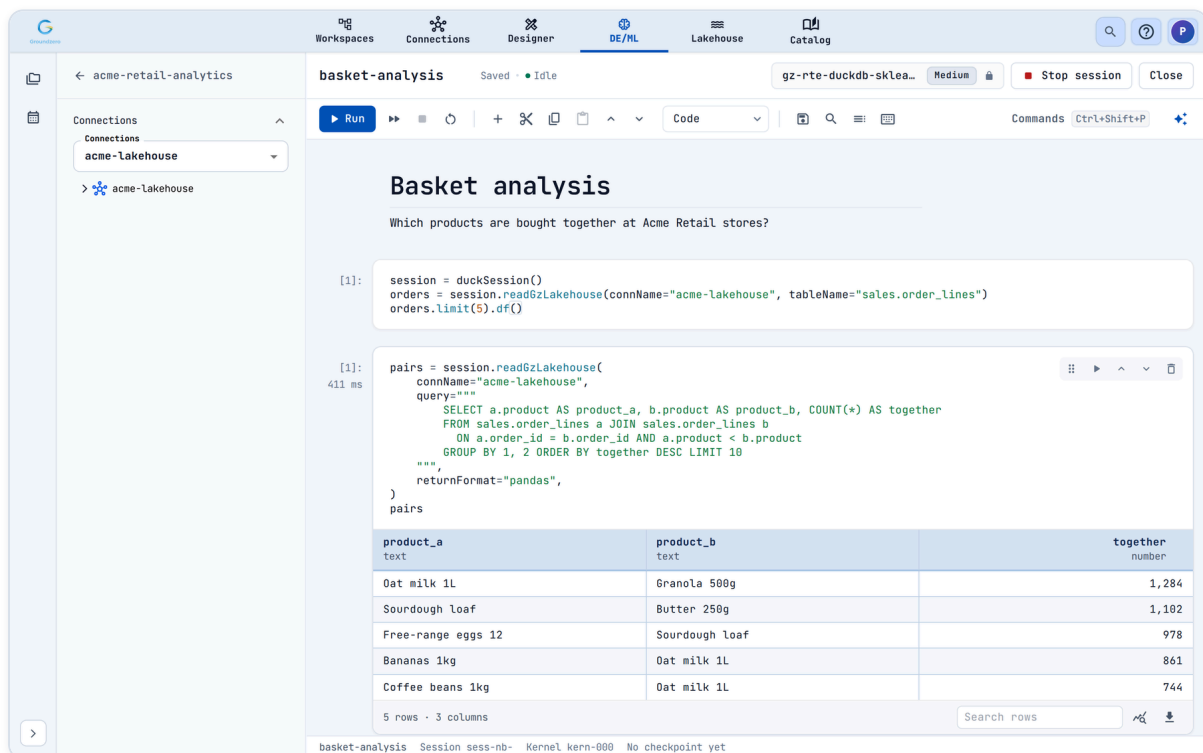


**Figure 34.** Running a cell returns a data table under it.

In a DuckDB or Spark notebook, the helper library is already loaded when the session starts ( `duckSession` for DuckDB, `sparkSession` and `executeJob` for Spark), together with NumPy, pandas and boto3. This cell finds the products that are bought together most often.

## PYTHON

```

session = duckSession()

pairs = session.readGzLakehouse(
    connName="acme-lakehouse",
    query="""
        SELECT a.product AS product_a, b.product AS product_b, COUNT(*) AS together
        FROM sales.order_lines a JOIN sales.order_lines b
            ON a.order_id = b.order_id AND a.product < b.product
        GROUP BY 1, 2 ORDER BY together DESC LIMIT 10
    """,
    returnFormat="pandas",
)
pairs

```

## NOTEBOOK OR JOB?

Use a notebook to explore. It stays interactive until you stop it. Use a job to repeat work: it runs your code once, stops when the code ends, and can run on a schedule.

## 4.4 Write the same job for DuckDB

The helper library for DuckDB works much like the Spark one. This job reads a lakehouse table, adds up revenue by region with SQL and prints the result.

## PYTHON

```

from GZ.DuckDBUtils import duckSession

session = duckSession(siteName="acme")

top = session.readGzLakehouse(
    connName="acme-lakehouse",
    query="""
        SELECT region, sum(quantity * unit_price) AS revenue
        FROM sales.orders
        GROUP BY region
    """,
    returnFormat="pandas",
)
print(top)
session.close()

```

Your code only ever names a connection. It never handles keys or passwords, and secret environment variables show as `****`.

## 5 The AI assistant and agents

The assistant helps you get things done faster: it answers questions, drafts SQL and jobs, and explains errors.

### 5.1 Ask Groundzero AI

The home page is a chat with an assistant that knows Groundzero. Describe what you want in plain language. It answers with text, tables, charts and links to the things it creates, such as datasets, dashboards, ETL jobs and filters. Before it changes anything, it stops and asks you.

- 1 Open the home page and click **Message Groundzero AI**, or pick a starter prompt. A starter prompt fills the box without sending it.
- 2 Type your request. **Enter** sends, and **Shift** + **Enter** adds a line.
- 3 To ask about a file, attach it with the paperclip. You can attach CSV, Excel, PDF, text and common image files, up to 50 MB each.
- 4 Read the answer. Open **Run details**, **Context used** and **Response details** under it to see which tools ran, what context it used, and its token use, speed and cost.
- 5 Use the **Chats** list to search earlier conversations or go back to one.

The screenshot shows the Groundzero AI chat interface. On the left is a sidebar with a 'Chats' section containing a 'New chat' button and a search bar. Below this is a list of conversations, with 'Q3 revenue by region' selected. The main chat area shows a user message: 'How did we do in Q3 by region? Compare against the FY26 targets file I attached.' with an attached file 'regional\_targets.csv'. The AI response, from 'Groundzero', states: 'Acme Retail closed Q3 with \$12.24M in net revenue, up 5.2% on Q2. APAC grew fastest (+9.4%) while LATAM is the only region that shrank. [1]'. Below the text is a table with 4 columns: Region, Q3 revenue, vs. Q2, and vs. target. The table lists data for North America, EMEA, APAC, and LATAM. Below the table, the AI provides additional context and a SQL query to reproduce the headline number. The interface includes a top navigation bar with icons for Workspaces, Connections, Designer, DE/HL, Lakehouse, and Catalog. At the bottom, there is a text input field, an 'Auto context' dropdown, and a model selection dropdown set to 'GPT-5.6 Terra'.

| Region        | Q3 revenue | vs. Q2 | vs. target |
|---------------|------------|--------|------------|
| North America | \$4.82M    | +6.1%  | +2.4%      |
| EMEA          | \$3.37M    | +3.8%  | -1.2%      |
| APAC          | \$2.91M    | +9.4%  | +5.8%      |
| LATAM         | \$1.14M    | -2.3%  | -7.9%      |

```
sql
SELECT region, SUM(net_revenue) AS q3_revenue
```

Figure 35. A saved conversation: a question, a table, a citation and the run details under the answer.

Good first prompts: *List the datasets I can access*, *Explain this SQL error*, *Trace lineage for sales.revenue\_daily* and *Build an ETL job that rolls up daily revenue by region*.

### ENHANCE PROMPT

The wand next to the send button (**Enhance prompt with AI**) rewrites your draft with the selected model before you send it.

## 5.2 Choose options and approve changes

The pill next to the paperclip, normally **Auto context**, opens the chat options.

| Option                  | What it does                                                                                                              |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------|
| <b>Plan mode</b>        | The assistant builds and tracks a plan, shown as a plan card.                                                             |
| <b>Business context</b> | Chooses whether answers use your published ontology (your shared business definitions): automatic, off, or chosen models. |
| <b>Approvals</b>        | Shows the standing approvals you have granted, if your site has them turned on.                                           |

The second pill shows the **model** and its **thinking effort** (Low to Max, depending on the model). Open it to pick another model.

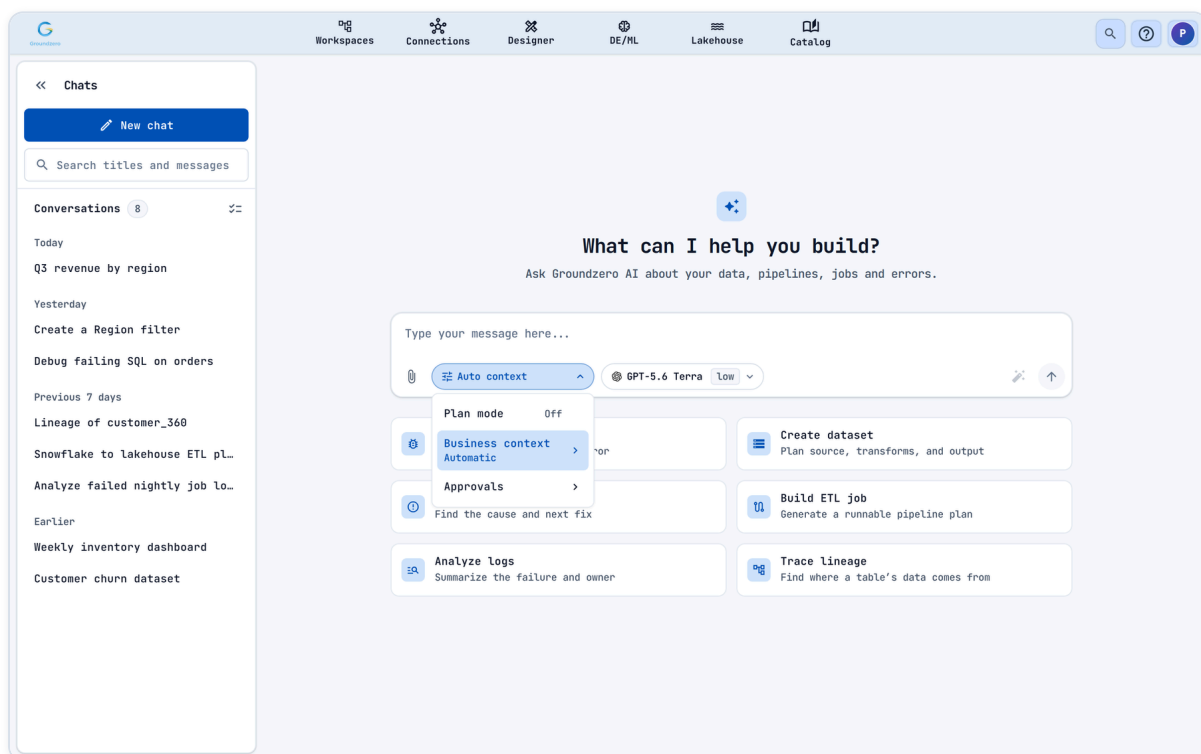


**Figure 36.** The chat options: Plan mode, Business context and Approvals.

When the assistant wants to **change** something, such as creating a filter, it shows an approval card with the proposed input, a countdown and the buttons **Deny** and **Allow once**. Nothing happens until you decide.

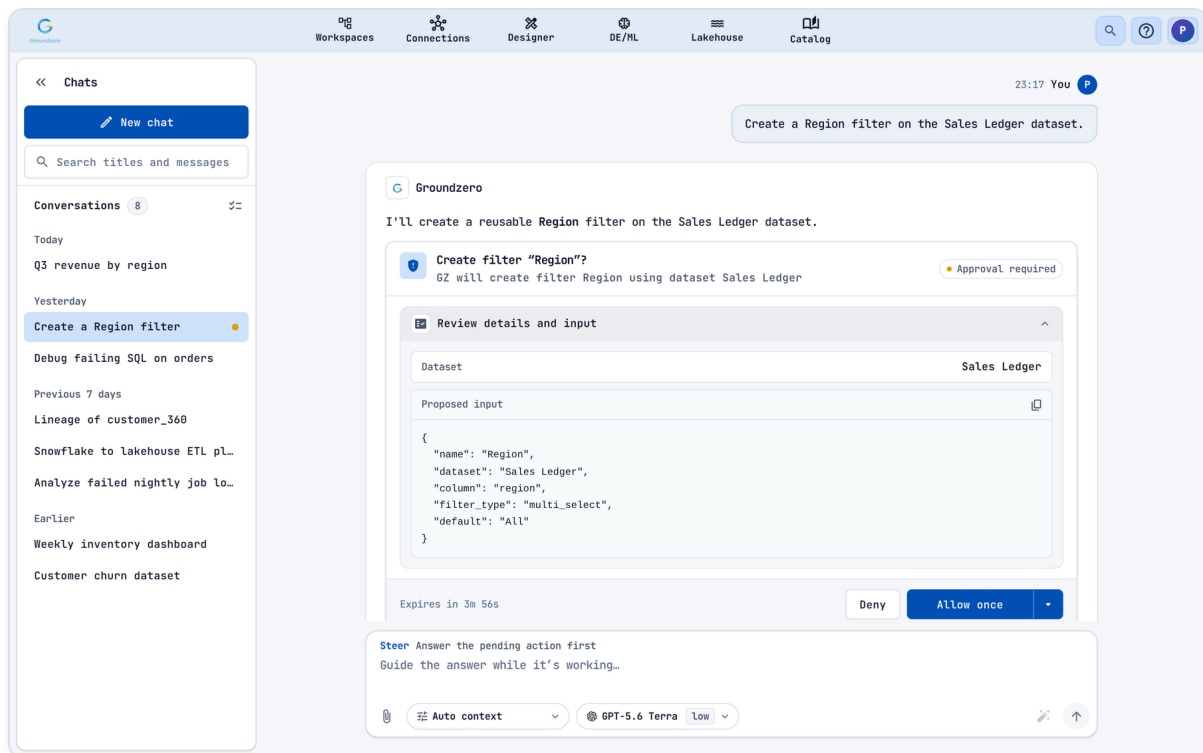


**Figure 37.** An approval card: the assistant proposes creating a filter and waits for Deny or Allow once.

#### REVIEW BEFORE YOU ALLOW

Open **Review details and input** and read what the assistant proposes before you choose **Allow once**. The assistant acts with your permissions.

## 5.3 Get code help inside editors

The dataset, job and notebook editors each have a sparkle button in the toolbar (**Generate code with GZ**). It opens a **CodeGenie** panel that knows what you are editing. Ask it to write SQL, Spark or notebook code, to explain an error from the editor, or to build a focused transformation. Then insert the result and run it. In a job, type `@logs` to give it the latest log lines.

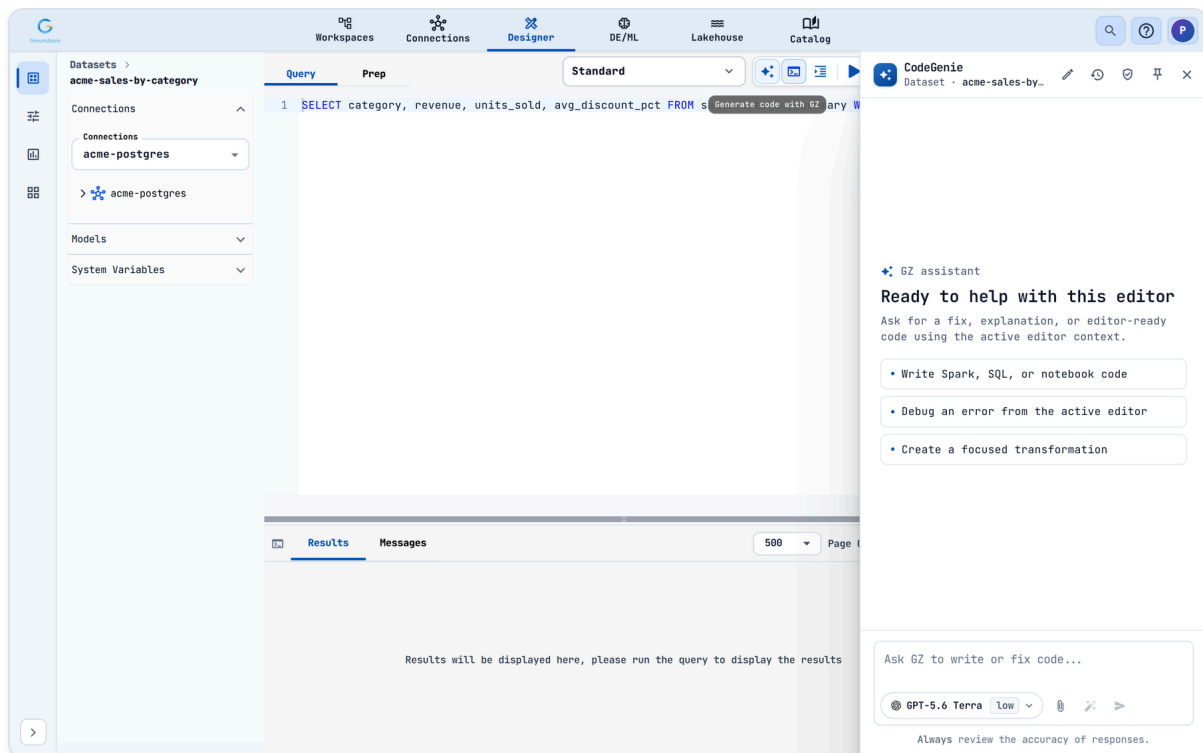


**Figure 38.** The code assistant opened beside a dataset editor, ready to write or fix code.

#### ALWAYS REVIEW GENERATED CODE

The panel reminds you to check the answers. Run the code on a small input first.

## 5.4 Create an AI agent

An **agent** is a saved assistant with one fixed job: instructions, a model and a narrow set of tools. Once a reviewer approves a version, other systems you control can call it. Use chat to explore, and use an agent for work that repeats and must behave the same way each time.

- 1 Open **DE/ML**, choose a project and open the **Agents** tab. The list shows each agent's **Status** (Draft, In review, Approved or Changes requested) and **Availability** (Live or Not live).
- 2 Choose **New agent**. The builder has six steps: **Identity**, **Model & behaviour**, **Tools & resources**, **Governance**, **Response** and **Review & create**. A guide panel on the right explains each field.
- 3 Test the agent with a prompt, then submit it for approval.

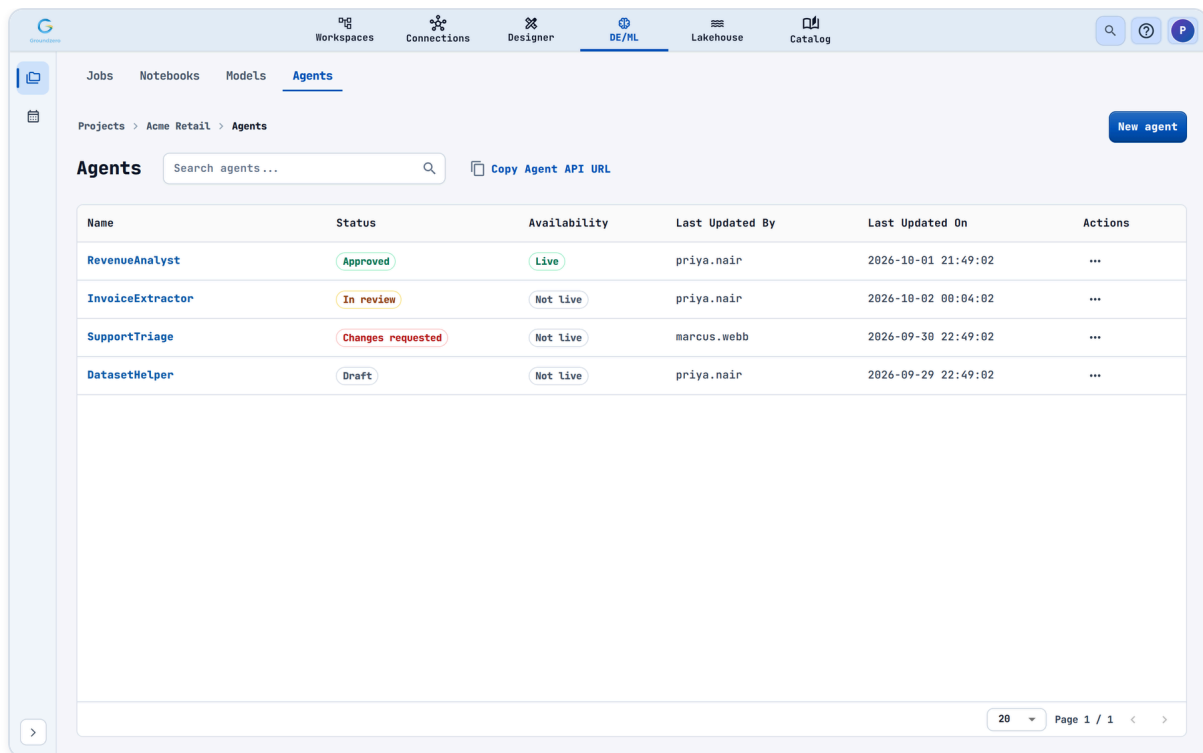


Figure 39. The Agents tab of a project.

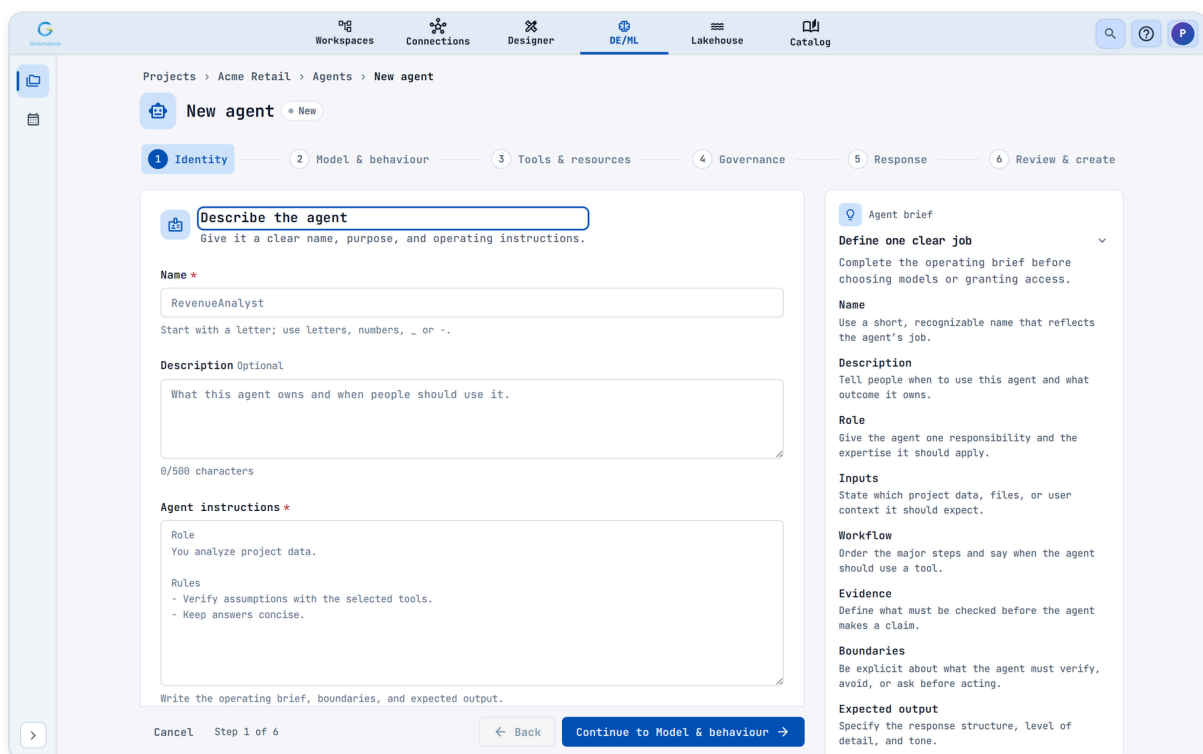


Figure 40. Step 1 of the agent builder: Identity, with a guide panel that explains each field.

## 5.5 Use Groundzero from your own tools

MCP (Model Context Protocol) lets coding assistants and similar tools work with Groundzero for you.

- 1 Open the **Help** menu and choose **Connect MCP**.
- 2 Pick your client: Codex, Claude Code, Cursor, VS Code or Generic HTTP MCP.

- 3 Copy the commands shown and run them in your tool.
- 4 Sign in through your browser when asked. There is no key to paste.

Tools that change data still ask for your approval.

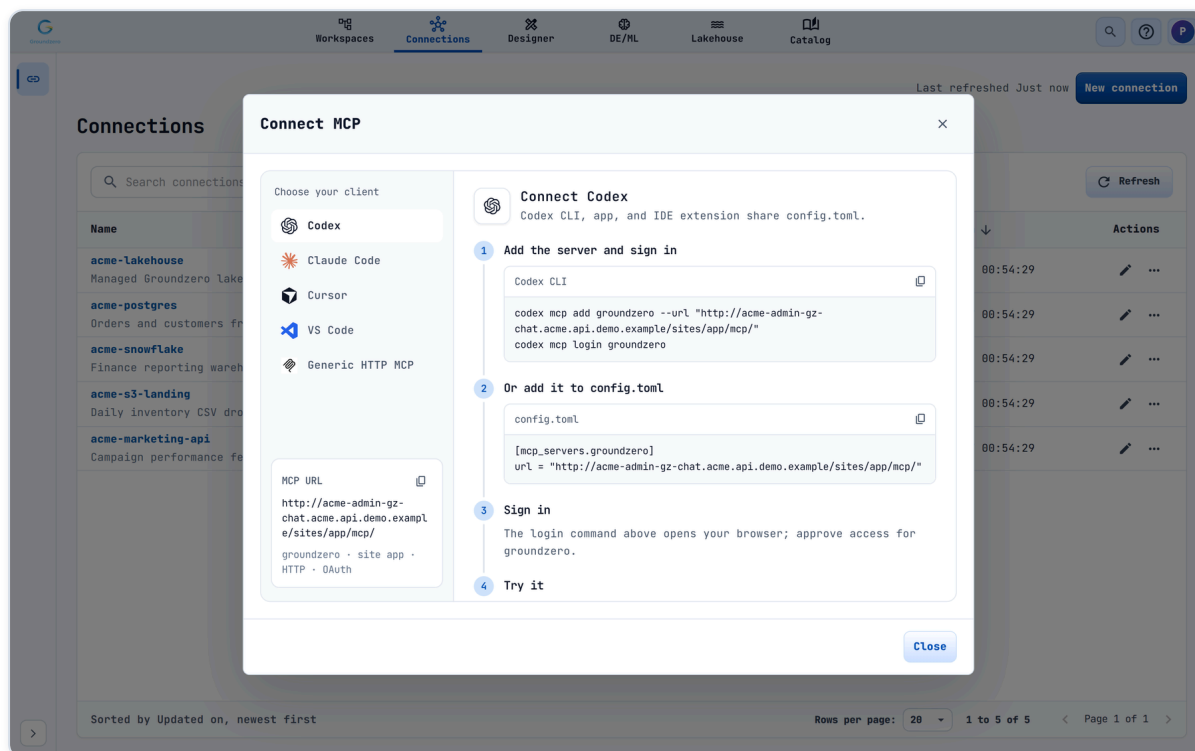


Figure 41. The Connect MCP dialog, with the server address and the steps for the chosen client.

## 6 Next steps

You now know how Groundzero fits together. Pick the area that matches your work.

| If you want to...                                             | Go to                                                                                                               | Where              |
|---------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------|--------------------|
| Store governed tables and query them with SQL                 | <b>Lakehouse:</b> create a warehouse, start a session with a compute size, write SQL in query tabs and save queries | Lakehouse          |
| Give data a shared business meaning                           | <b>Ontology Studio:</b> define entities, metrics and terms, review suggestions and publish releases                 | Catalog            |
| See what chat, agents and workloads cost and how they perform | <b>Consumption Metrics:</b> AI Chat, Agents, Workloads and APIs views with time ranges                              | Catalog            |
| Share a dashboard with a team                                 | <b>Workspaces:</b> folders of imported dashboards; <b>My Files</b> for storage buckets                              | Workspaces         |
| Track experiments and models                                  | The <b>Models</b> tab of a project                                                                                  | DE/ML              |
| Build a reusable filter                                       | <b>Designer, Filters</b>                                                                                            | Designer           |
| Manage people, permissions and sites                          | The <b>admin console</b> (Users & Groups, Access Manager, Sites, Email, Budgets)                                    | Your administrator |

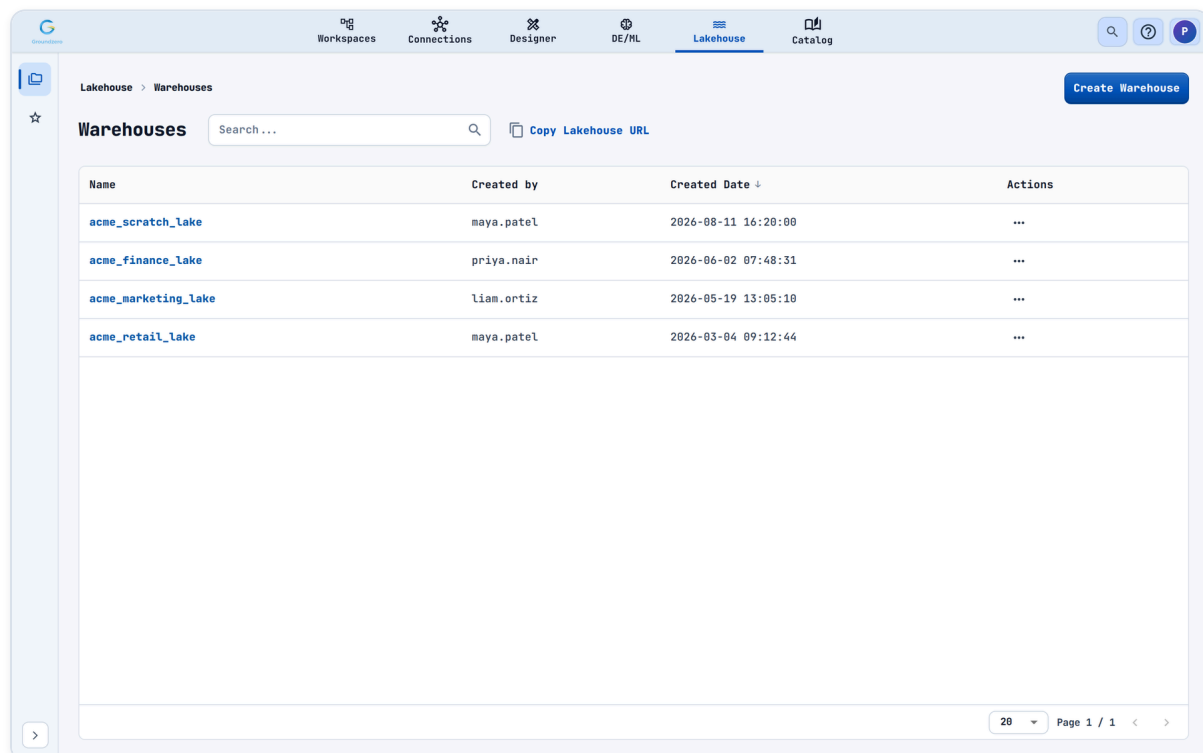


**Figure 42.** Lakehouse warehouses: a table of warehouses with owner and creation date. Open one to start a session and run SQL.

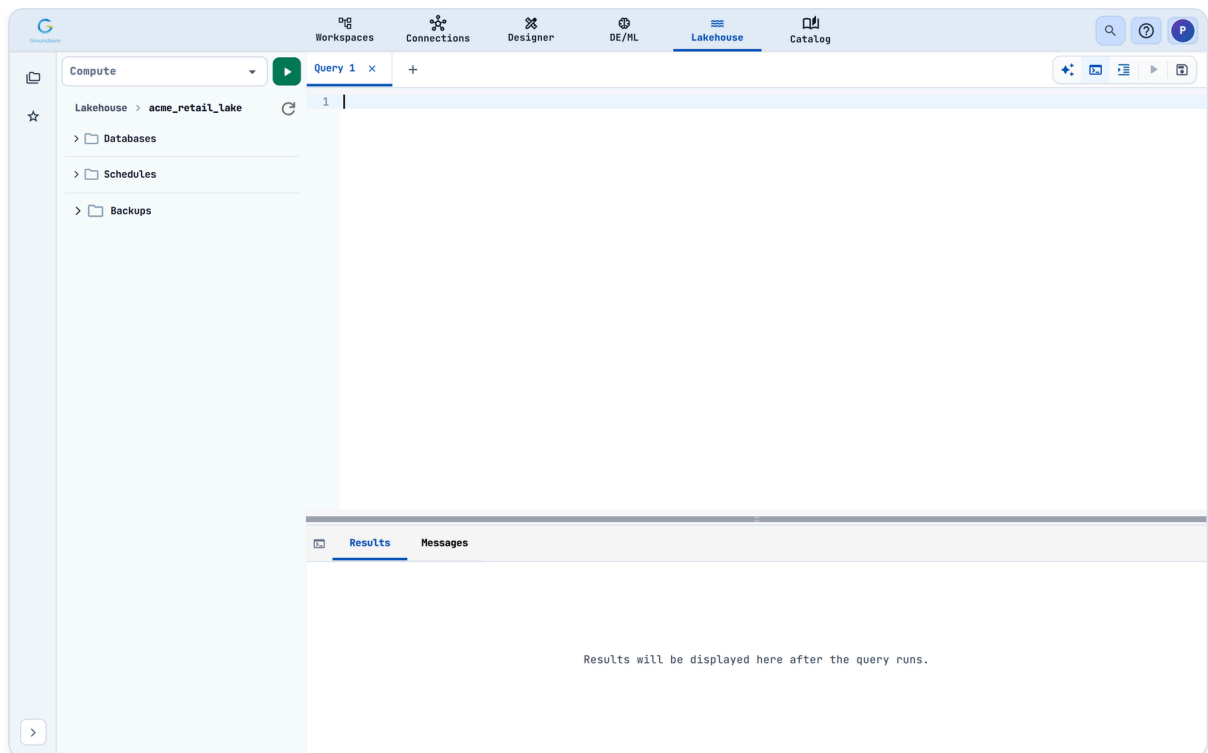


Figure 43. A lakehouse workspace: pick a compute, start a session, and run SQL in query tabs next to the schema tree.

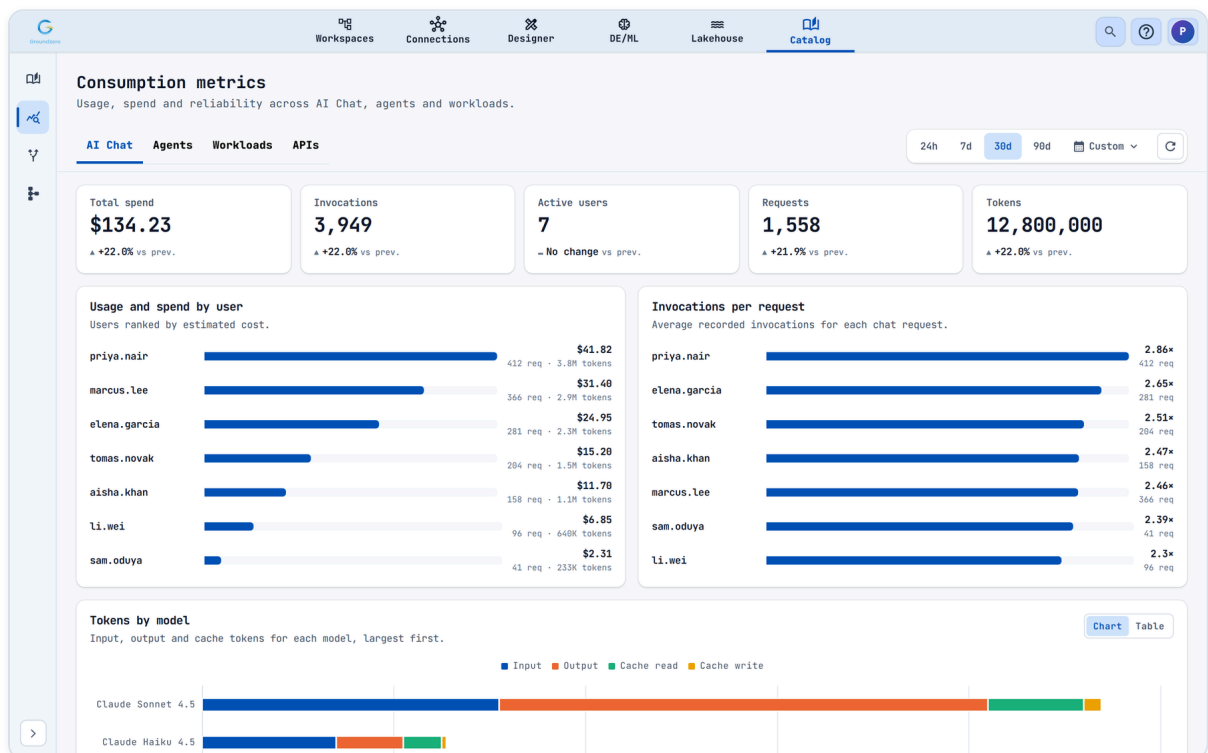


Figure 44. Consumption Metrics: usage, spend and reliability across AI Chat, Agents, Workloads and APIs.

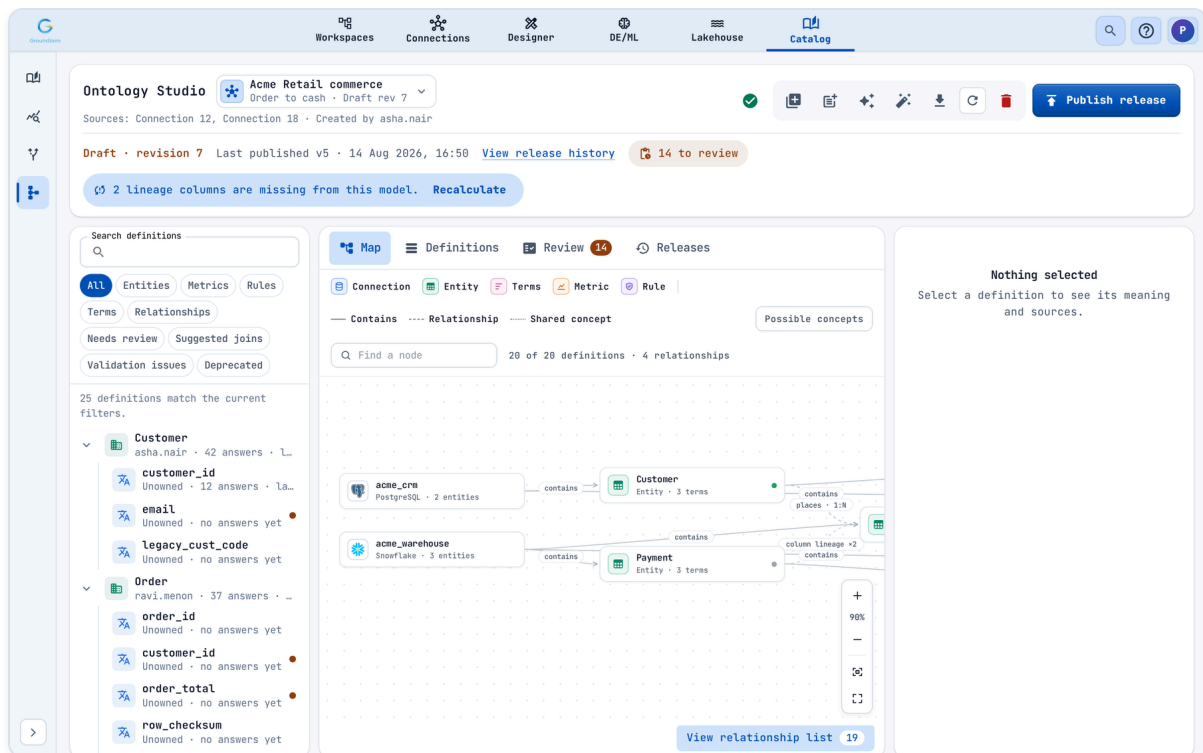


Figure 45. Ontology Studio: definitions on the left, a map of how connections, entities and terms relate, and a review queue.

## 6.1 Habits that save time

- Use hyphenated names everywhere. They are easy to find in global search with the `ds:`, `viz:` and `dash:` prefixes.
- Add a description and tags in the Catalog to everything you share.
- Tick only the connections a job or notebook needs.
- Save in the editor, then publish. Publishing a dashboard does not save it first.
- Stop notebook and lakehouse sessions when you finish.

## 7 Glossary

| Term                         | Meaning                                                                                                                               |
|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| <b>Agent</b>                 | A saved assistant with instructions, a model and a fixed set of tools. Other systems can call it after a reviewer approves a version. |
| <b>Approval</b>              | A confirmation the assistant asks for before it changes something.                                                                    |
| <b>Batch / Streaming</b>     | The two schedule types in the Scheduler. Streaming (the default) runs once; Batch repeats.                                            |
| <b>Catalog</b>               | The searchable list of datasets, ETL jobs and warehouses.                                                                             |
| <b>Compute</b>               | The memory and processor size given to a runtime.                                                                                     |
| <b>Connection</b>            | A saved link to an outside system. Code uses it by name and holds no credentials.                                                     |
| <b>Contract tab</b>          | A Catalog tab that shows how to call a dataset or job from outside Groundzero.                                                        |
| <b>Dashboard</b>             | A page of visualizations in tabs, with filters. You can publish it and import it into a workspace.                                    |
| <b>Dataset</b>               | A saved SQL or code query whose result charts and dashboards use.                                                                     |
| <b>ETL job</b>               | Python code in a project that reads, changes and writes data.                                                                         |
| <b>Lakehouse / Warehouse</b> | Groundzero storage for governed tables, organised in databases and tables.                                                            |
| <b>Lineage</b>               | A map of how data flows between tables, jobs and datasets.                                                                            |
| <b>MCP</b>                   | Model Context Protocol, a standard way for tools and assistants to work with Groundzero.                                              |
| <b>Ontology</b>              | Shared business definitions (entities, metrics, terms) that the assistant can use as context.                                         |
| <b>Project</b>               | A DE/ML container for jobs, notebooks, models and agents.                                                                             |
| <b>Runtime (RTE)</b>         | The environment that runs your code: Standard, DuckDB or Spark, and their variants.                                                   |
| <b>Session</b>               | A running notebook. You start it, work in it and stop it.                                                                             |
| <b>Site</b>                  | Your organisation's Groundzero, with its own logo and theme.                                                                          |
| <b>System variable</b>       | A placeholder such as <code>\${JOBNAME}</code> that Groundzero fills in before your code runs.                                        |
| <b>Visualization</b>         | A saved chart or table on a dataset.                                                                                                  |
| <b>Workspace</b>             | A named folder that holds imported (published) dashboards.                                                                            |

## 8 Troubleshooting and FAQ

### 8.1 Troubleshooting

| What you see                                                     | Cause                                                 | Fix                                                                                    |
|------------------------------------------------------------------|-------------------------------------------------------|----------------------------------------------------------------------------------------|
| The sign-in page has no sign-up link                             | Administrators create accounts                        | Ask your administrator for an account                                                  |
| <i>Set up two-step verification</i> appears after sign-in        | Your account does not use an authenticator app yet    | Scan the QR code and enter the code                                                    |
| <i>You don't have access to chat</i>                             | Your account lacks the AI Chat permission             | Ask your administrator for it                                                          |
| <i>Usage limit reached</i> in chat                               | A spending limit set for you or your group is used up | Ask your administrator about the limit                                                 |
| <i>You do not have "All" or "Add" permissions on Connection.</i> | A permission is missing                               | Send the message to your administrator                                                 |
| <b>Test Connection</b> and <b>Save</b> stay greyed out           | A required field is empty                             | Fill in every field. A name starts with a letter and uses letters, numbers and hyphens |
| <i>Please select Connection</i> when you run a dataset           | No connection is ticked                               | Tick a connection in the left drawer                                                   |
| <i>Please select Compute or Please select RTE and Compute</i>    | A runtime is chosen without a compute size            | Open the runtime picker and choose both                                                |
| The Compute list says <i>Insufficient Permission</i>             | You cannot use compute sizes                          | Ask your administrator to grant access                                                 |
| A job succeeds but shows no preview                              | The script did not set <code>outputDF</code>          | Assign a DataFrame to <code>outputDF</code> at the end                                 |
| <i>Dashboard is empty, add some items and save.</i>              | You tried to save or publish an empty dashboard       | Add visualizations first                                                               |
| A published dashboard looks old                                  | Publish does not save your changes                    | Choose <b>Save dashboard</b> , then <b>Publish</b> again                               |
| A scheduled job runs at an unexpected hour                       | Schedules follow UTC                                  | Convert your time to UTC                                                               |
| <i>No schema available for this warehouse yet.</i>               | The warehouse has no tables, or you may not list them | Create tables or ask for access                                                        |
| Lineage shows <i>Choose a table to trace</i>                     | No table is selected                                  | Pick a connection, then a table                                                        |
| <i>Access restricted</i> on the admin address                    | The admin console is for administrators               | Use your normal site address                                                           |

## 8.2 FAQ

**Where are my passwords kept?** In connections only. Your code and datasets use a connection by name and never contain credentials. A connection's details show secrets as hidden values.

**Can I change the theme or colours?** An administrator sets them for the whole site.

**How do I refresh a dashboard?** Reopen it. Use **Export To PDF** or **Export to Excel** to take a copy. A scheduled job can refresh the data a dashboard reads.

**What is the difference between a dataset, a job and a notebook?** A dataset answers a question when you ask and feeds charts. A job moves data without you and can run on a schedule. A notebook is for exploring interactively.

**Why does a saved chart show current data?** Groundzero saves the chart design, not the rows. It runs the dataset each time the chart opens.

**How long are dataset results kept?** Groundzero keeps the result of an identical query in your browser for 10 minutes. Reload the page to start fresh.

**Where can I get help?** Open the **Help** menu. **Community** opens the Groundzero community chat, and the support address is listed there. Your administrator can help with permissions and access.