Building Governed Data Models With Sigma

A Practical Guide
For The Data Professional

Phil Ballai
Enterprise Architect
Sigma

Table of Contents

1 Introduction

The cloud data warehouse has become the de facto system of record for most modern enterprises—making governance, compliance, security, and a unified source of truth more critical than ever. But simply centralizing data in the warehouse doesn't guarantee consistent, trustworthy analytics. The real challenge is ensuring that every dashboard, report, and application draws from the same definitions and relationships, without creating bottlenecks for data teams.

That's where Sigma Data Models come in. Sigma's modeling layer is designed to be the governed, reusable semantic layer for your analytics environment—sitting directly on top of your warehouse, preserving its performance and security, and enabling trusted self-service for every business user. With Sigma Data Models, you can define joins, metrics, and permissions once, and reuse them everywhere—from dashboards and operational workflows to AI-driven analytics.

Sigma Data Models offer a path forward, enabling teams to:

- **Create and reuse** governed data assets that power dashboards, operational workflows, and AI-driven analytics.
- **Define and manage metrics, joins, and permissions** in one place, ensuring every analysis uses consistent business logic.
- **Adapt quickly to change** with versioning, lineage, and dependency awareness that prevent breakage as schemas evolve.

By putting governance, reusability, and agility in the same layer, Sigma makes it possible to deliver consistent insights without slowing innovation, allowing data teams to maintain control while enabling true self-service for business users.

Who is this eBook for?

This guide is for SQL-fluent, warehouse-first practitioners—data engineers, analytics engineers, BI leads, and architects—who are tasked with scaling governed analytics environments.

We'll explore how to model with confidence in Sigma, bring consistency to your data layer, and align your governance strategy with modern cloud architecture.

The goal is simple: **keep your logic in one place, run it in your warehouse, and make it accessible to everyone who needs it—all without losing control.**

2: Sigma's Philosophy On Data Modeling & Governance

Sigma Data Models are built on a simple truth: **governance and agility don't need to be at odds.**

The best modeling environments deliver trusted outputs without slowing innovation. That means giving data teams the tools to centrally define relationships, metrics, and security, while still enabling business users to explore, extend, and apply that logic in their own workflows.

Our approach blends governed modeling with real-time exploration, combining the flexibility of spreadsheets with the control of a semantic layer. Because models run directly on your cloud data warehouse, Sigma preserves your existing security posture, enforces role-based access, and scales seamlessly with your data.

Core Principles

Security First – All queries run in your warehouse. Sigma supports private deployments, works natively with warehouse row-level and column-level security (RLS/CLS) and OAuth authentication, and never stores data.

True Self-Service – Modeling features are opt-in. Data teams can define reusable calculations, metrics, and joins, while business teams leverage them without breaking governance.

Beyond Dashboards – Sigma Data Models power end-to-end workflows, from reconciliations and approvals to forecasting and write-back applications.

AI That Speaks SQL – Logic defined in your models becomes the knowledge base for "Ask Sigma," an AI assistant that improves query accuracy, search, and data labeling.

This modeling strategy aligns with the future of governed analytics: a modular, direct-to-warehouse layer that serves both technical and non-technical users without forcing a rigid process.

3. Key Functional Capabilities To Look For

Modeling Capabilities That Matter:

- **Simple Setup** – No separate modeling server or deployment step. All logic lives in Sigma Data Models and runs **directly on your warehouse**, preserving its performance, security, and cost controls.
- **Visual Relationships** – Define joins between datasets with an intuitive visual builder. Supports complex joins, one-to-many lookups, and fanout prevention to keep metrics consistent.
- **Security Controls** – Native support for your warehouse's RLS/CLS policies, private cloud deployments, column-level permissions, and join pruning—all enforced in real time because queries run where your data lives.
- **Reusable Metrics** – Define KPIs like revenue, margin, or churn once and use them consistently across workbooks, dashboards, and applications without redefining them for each project.
- **Ask Sigma (AI)** – Models are queryable through natural language. Defined metrics and relationships improve context and accuracy for AI-generated queries, so answers match your governed definitions.
- **Limitless Exploration** – Users can interact with governed models flexibly and drill, filter, sort, and pivot without requiring new datasets for every question.
- **Writeback & Input Tables** – Users can enter data directly into governed workflows like reconciliations, forecasts, QA flags, and more, all while the logic still runs against live warehouse data.

Together, these features deliver on the promise of governed self-service—letting teams build confidently while maintaining consistency and trust, all in a **direct-to-warehouse** modeling environment.

4. Let's Make It Work

In this example, we'll build a Sigma Data Model ("model") for a fictional retail chain to answer core business questions about sales, profitability, and promotion effectiveness.

The model starts with point-of-sale transactions and enriches them with product, store, and calendar details.

From this foundation, we'll define governed metrics such as gross margin, units sold, and gross margin percentage that can be reused across dashboards, reports, and AI-driven queries.

We'll also apply column-level security (CLS) to protect sensitive cost and margin data, ensuring that only authorized roles can see it.

Finally, we'll demonstrate how Sigma's content validation prevents downstream breakage when model changes are published, and how governed metrics make it simple for anyone—from analysts to AI assistants—to explore sales performance confidently and consistently.

Key Business Questions This Model Will Answer

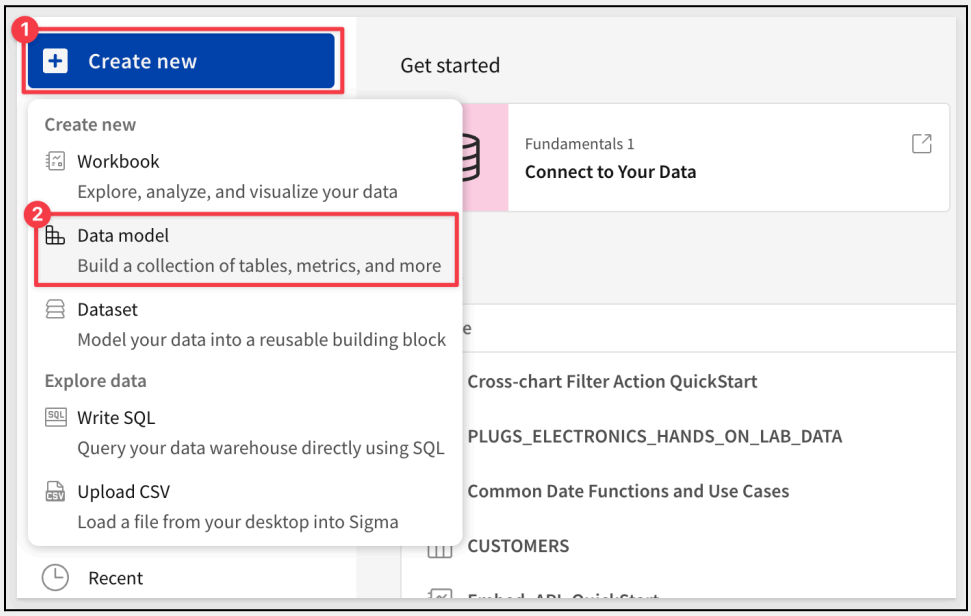
- Which product categories drive the highest revenue and margin?
- How does sales performance vary by store, region, or channel?
- What is our average order value (AOV) and units per transaction (UPT)?
- Which promotions delivered the largest sales lift?
- Are there seasonal patterns in revenue and margin across product lines?
- How does performance compare during promotional vs. non-promotional periods?

By answering these questions with a single, governed model, the retail team can trust that every metric—whether viewed in a dashboard, AI query, or ad hoc analysis—comes from the same consistent definitions.

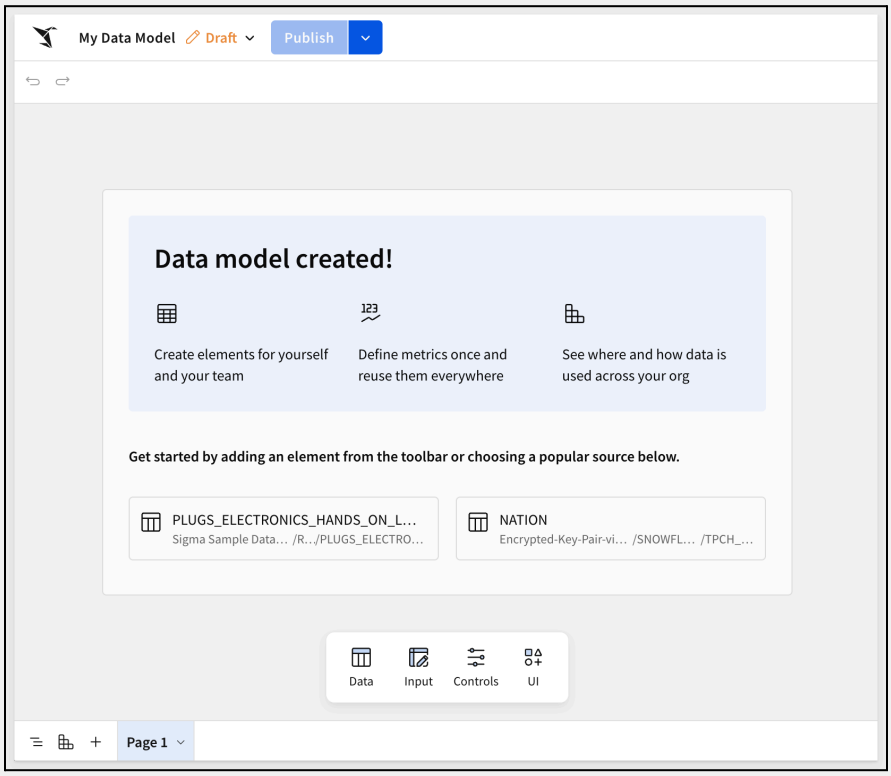
NOTE
All demonstrations are performed using a trial environment. Sigma recommends that you use non-production resources when completing this section. Sigma Trial

1. Log into Sigma and create a new Data Model

We will use the Sigma provided sample database so that this can be recreated in any Sigma environment. Click the `Create new` button and select `Data model`:

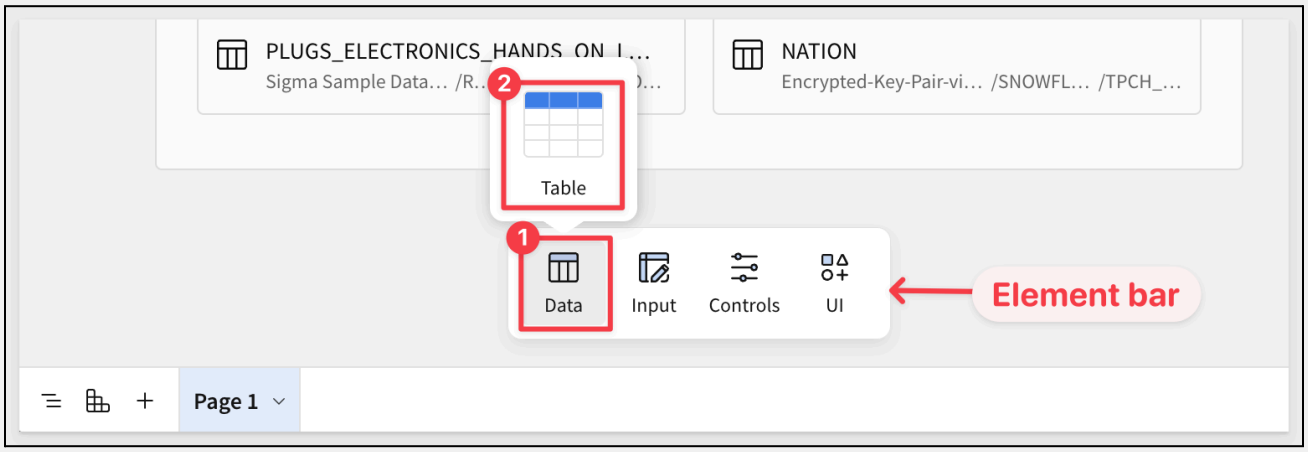


The modeling page looks similar to the workbook page so that model builders are working in a familiar interface:



2. Add the fact table

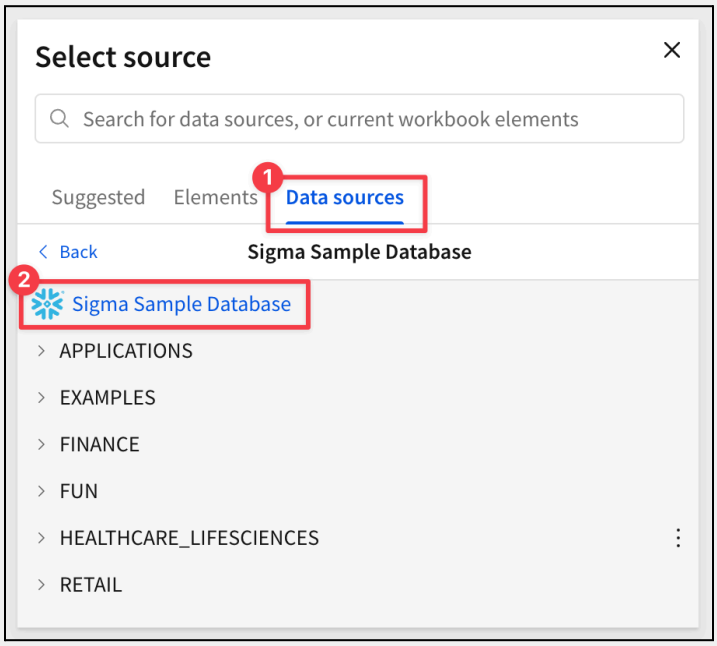
Using the `Element bar` click on `Data` and select `Table`:



Sigma presents us with three options on where to obtain our source data:

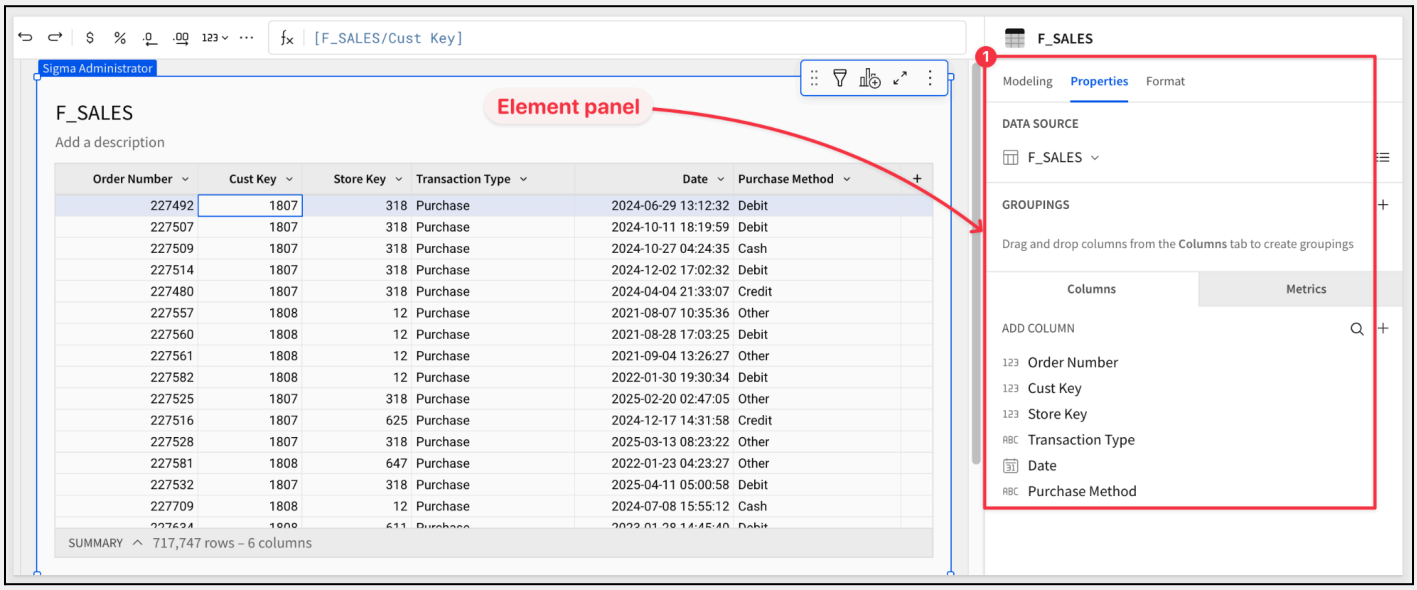
1. **Suggested:** a list of frequently used sources
2. **Elements:** elements that already exist in the data model we are working with
3. **Data Sources:** any connection that the user has access to (ie: Snowflake, Databricks, SQL)

Since we are using sample data, select the `Data sources` tab and the `Sigma Sample Database`:



Expand the `RETAIL` > `PLUGS_ELECTRONICS` schema and select the `F_SALES` table.

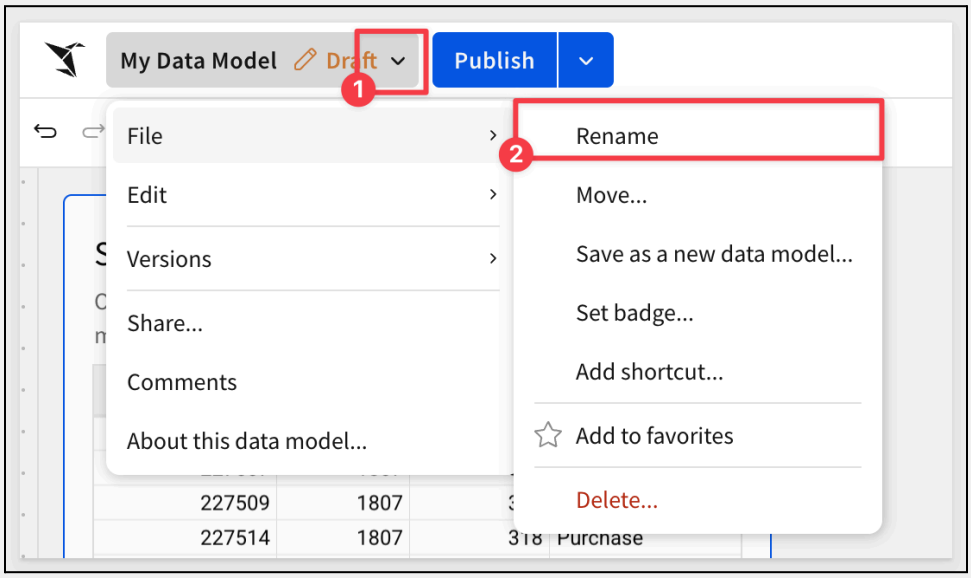
Once selected, it will load onto the workspace and we can access the `Element panel` which allows us to fully customize the table in a variety of ways:



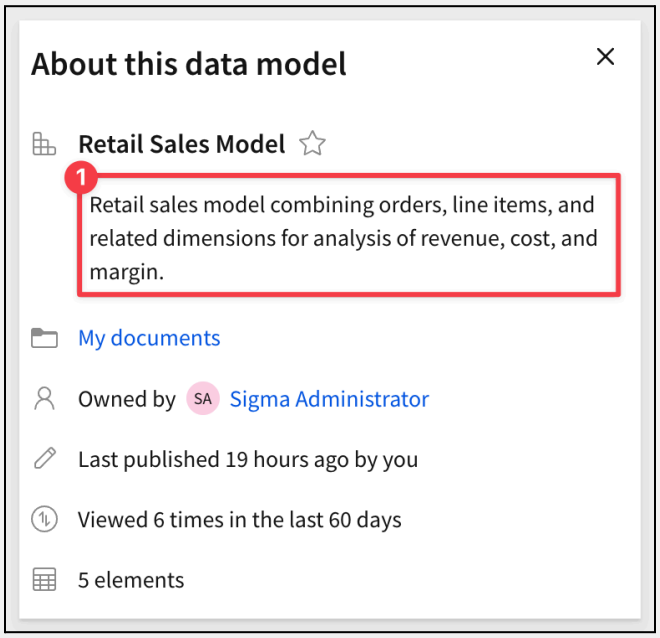
Change the table name to `Sales Orders` and the description to `Order-level sales data, one row per order`

This fact table contains one row per product sold per order. We'll join this to the `Point of Sale`, `Store Locations` and `Customer List` dimensions, then define metrics directly in the model.

Open the data model's menu and select `Rename`. Use the name `Retail Sales Model`:



Open the same menu again and select `About this data model`. Here we can provide a useful description for users to better understand the model when it is first opened:



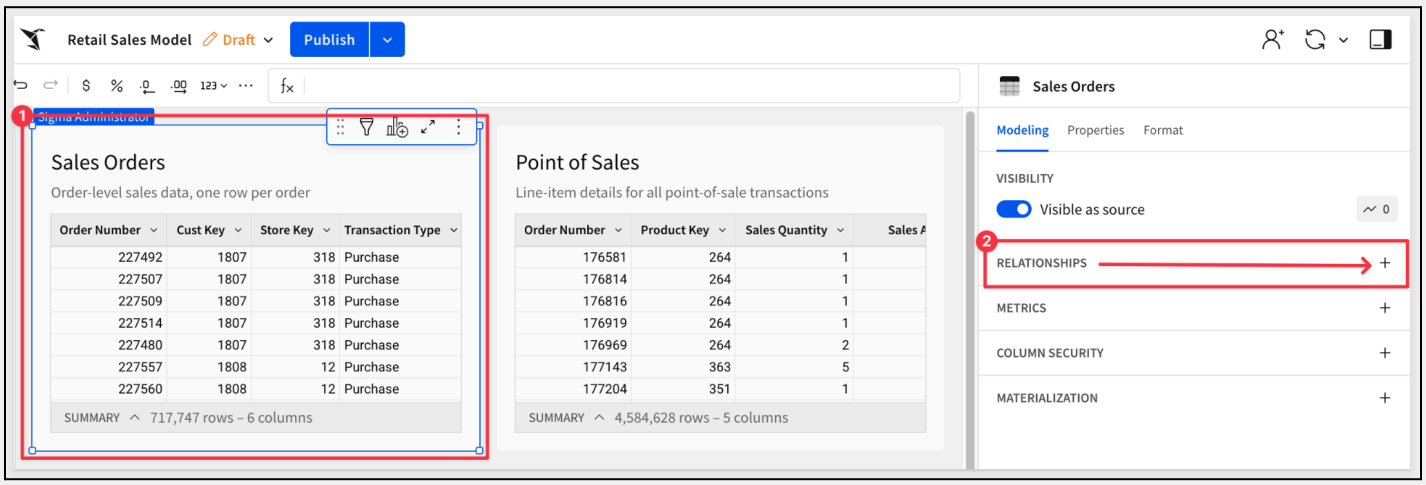
Click `Publish`.

3. Add dimension tables

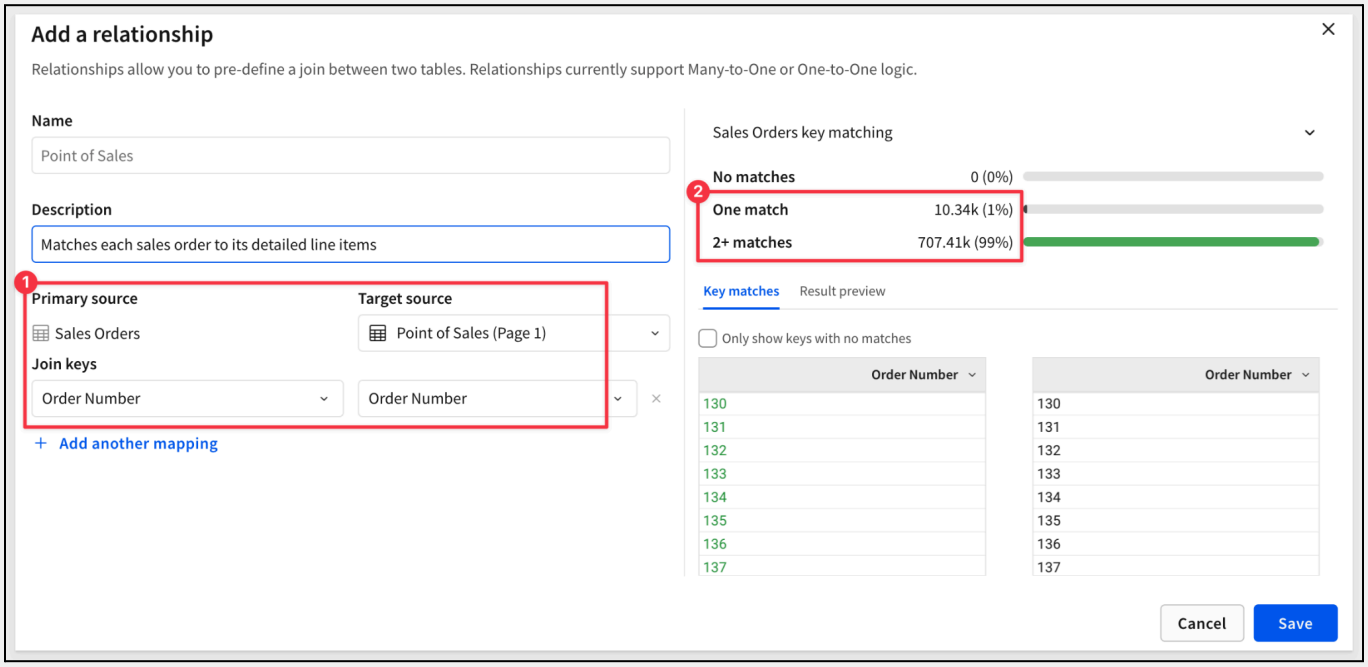
Using the same workflow add the `F_POINT_OF_SALE` table to the model.

Rename it `Point of Sales` and set the description to `Line-item details for all point-of-sale transactions`:

Next we will relate the two tables. Click `Sales Orders` to select the table and then click the `+` as shown:

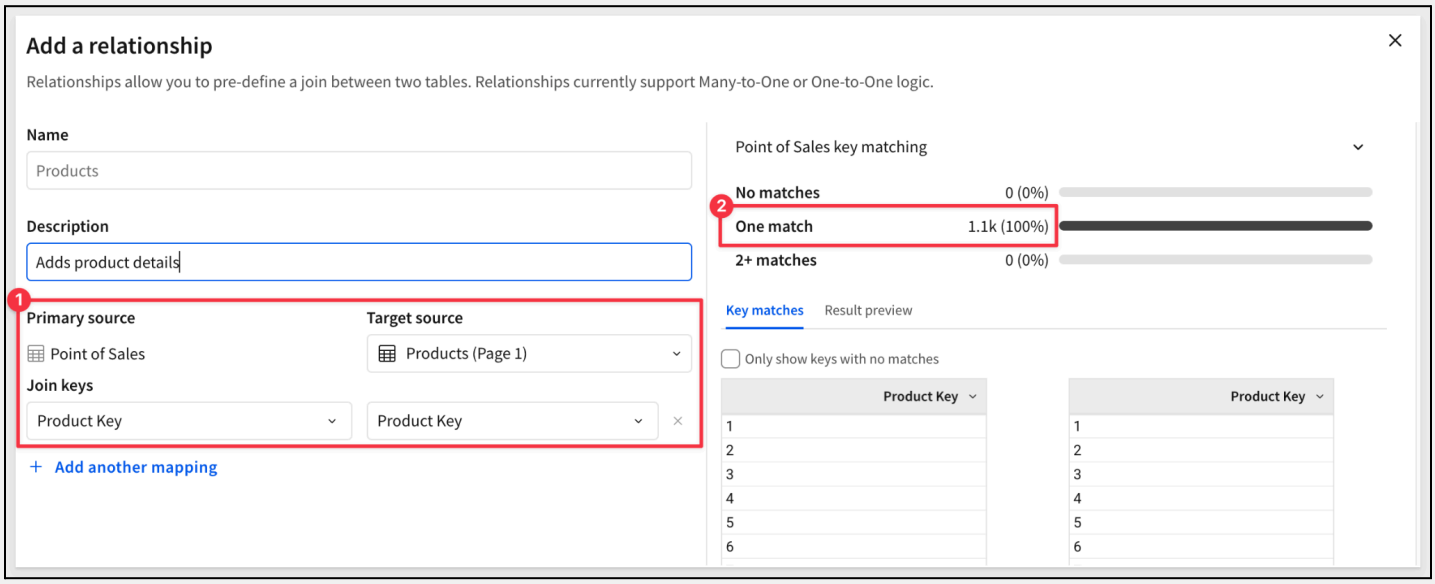


Configure the relationship as shown. Most orders have more than one line item which is expected:



We also want to know which products we purchased.

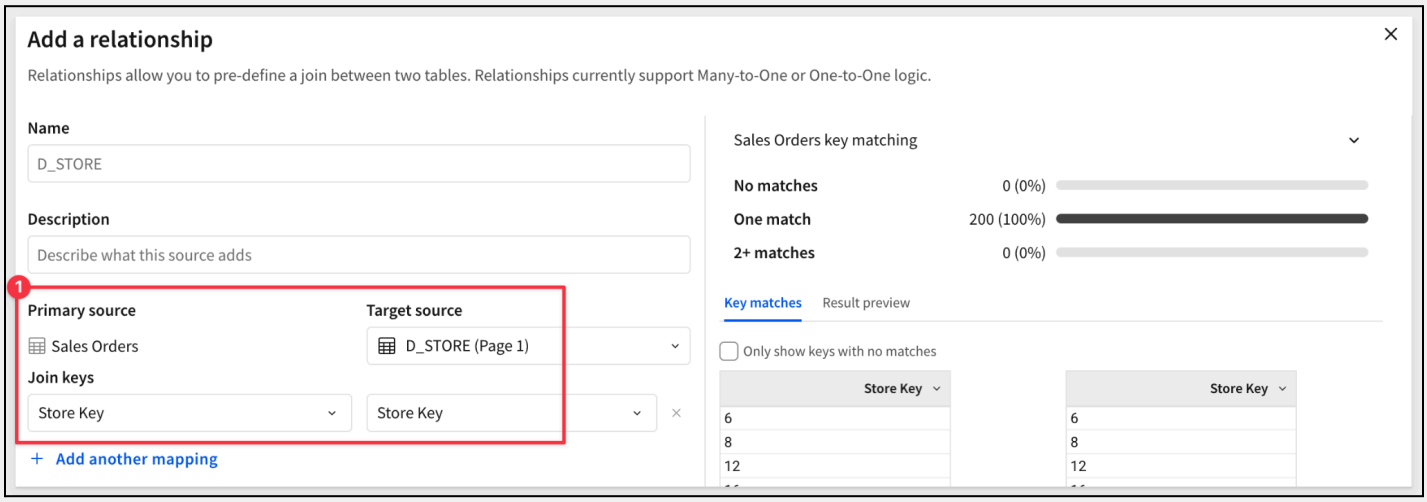
Repeat the workflow to add the `Products` table, **but this time join it to the `Point of Sales` table:**



We have 100% match because every sale has to have an accompanying product.

Add the `D_STORE` table and join it to the `Sales Order` table.

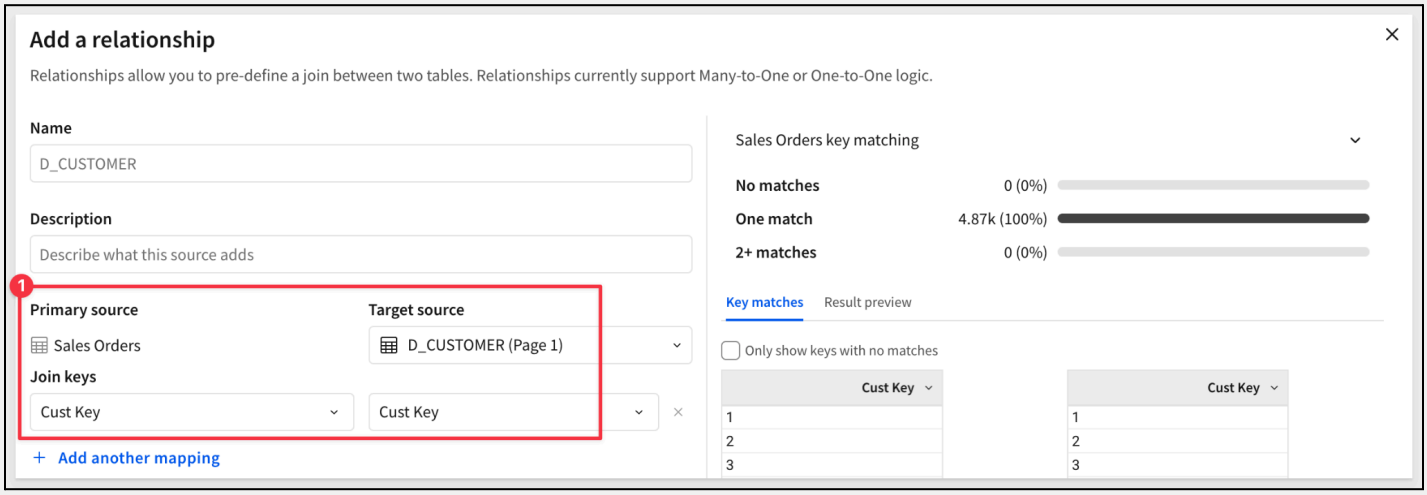
Every order is associated with a store location in our data:



Rename the table `Stores Locations`.

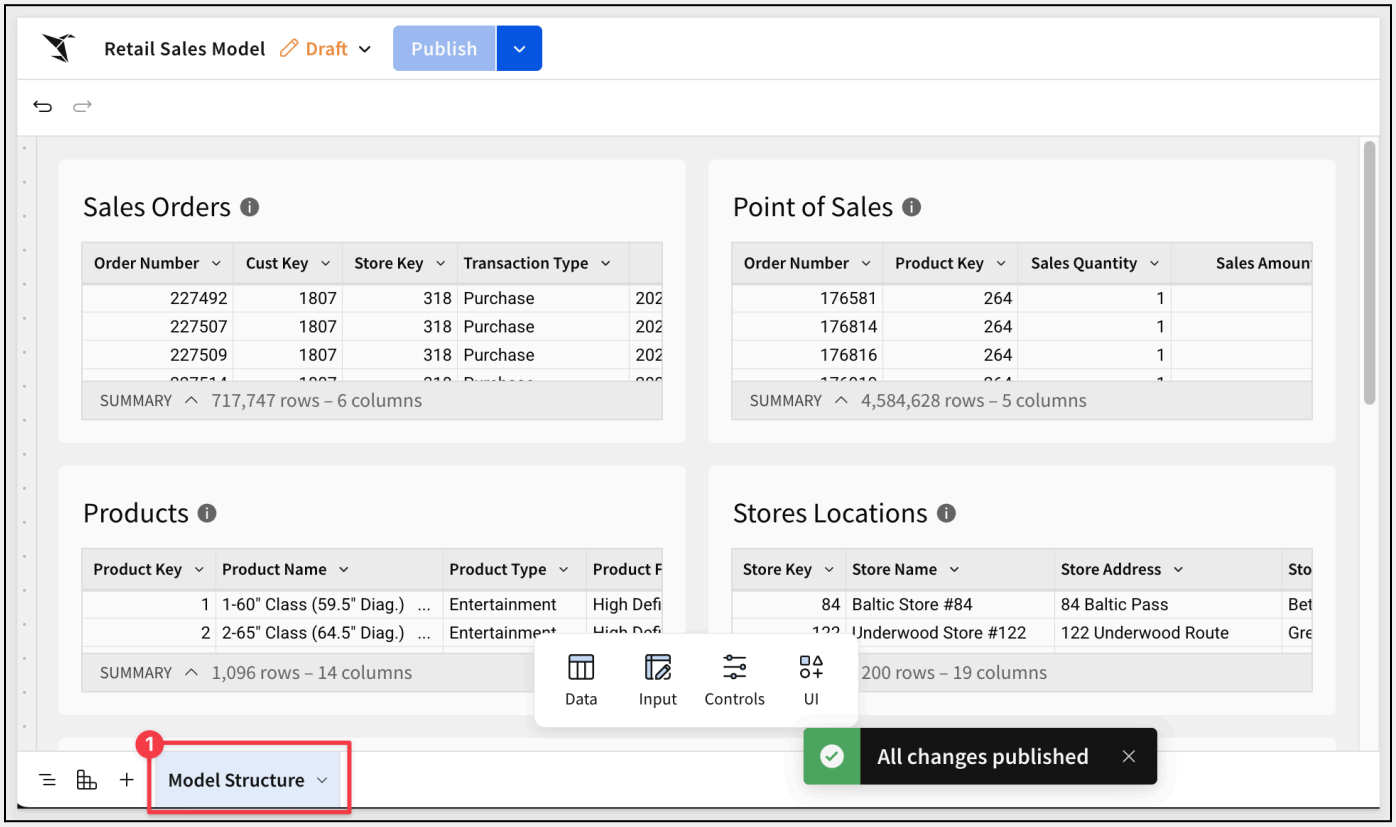
Lastly, add the `D_CUSTOMER` table and join it to the `Sales Order` table.

Every order has a customer associated with it:



Click `Publish`.

Now that we have all our source tables, let's rename the workbook page to `Model Structure` to help future editors understand the model better:

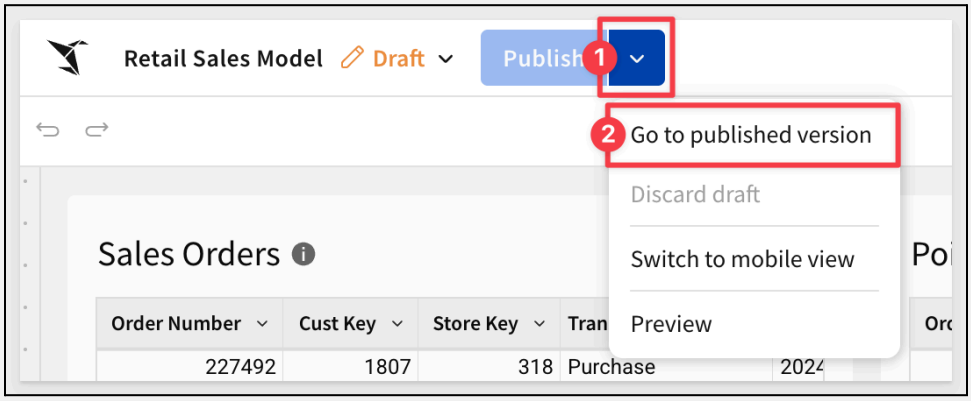


For more information on navigating the data model workbook page, see [Data model overview page]

4. Data Model landing page

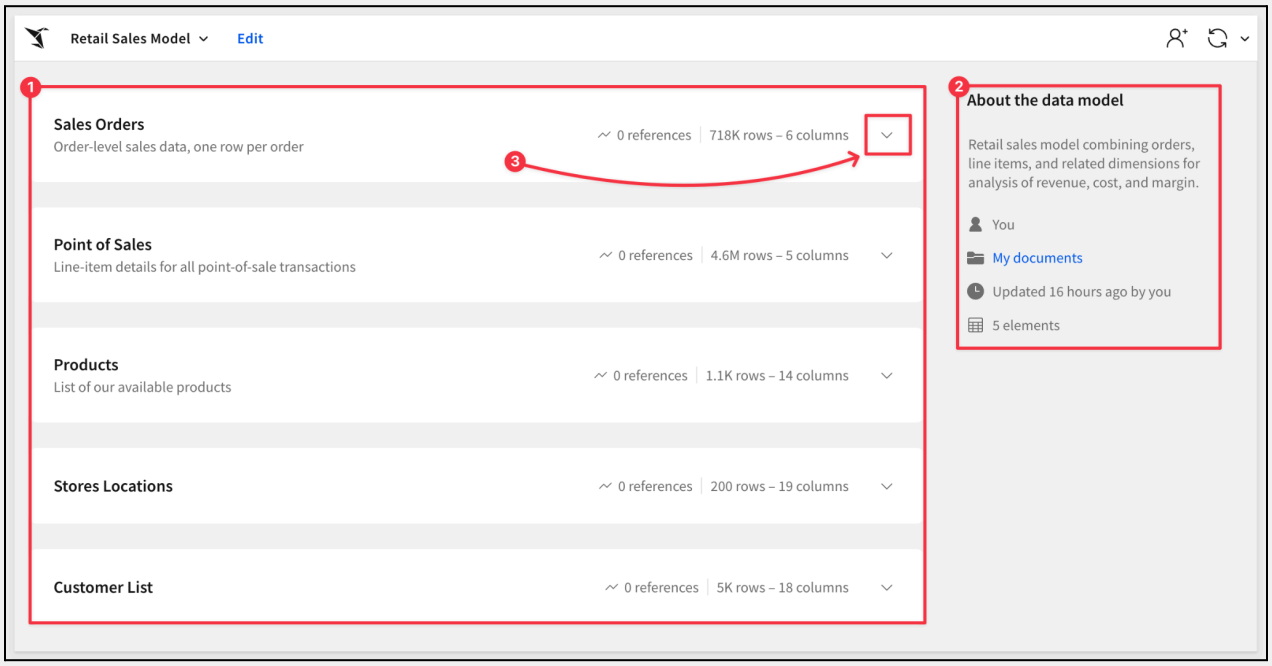
Before we go further, let's take a moment to understand how users will access the model when they want to build content from its resources.

From the `Publish` menu, select `Go to published version`:



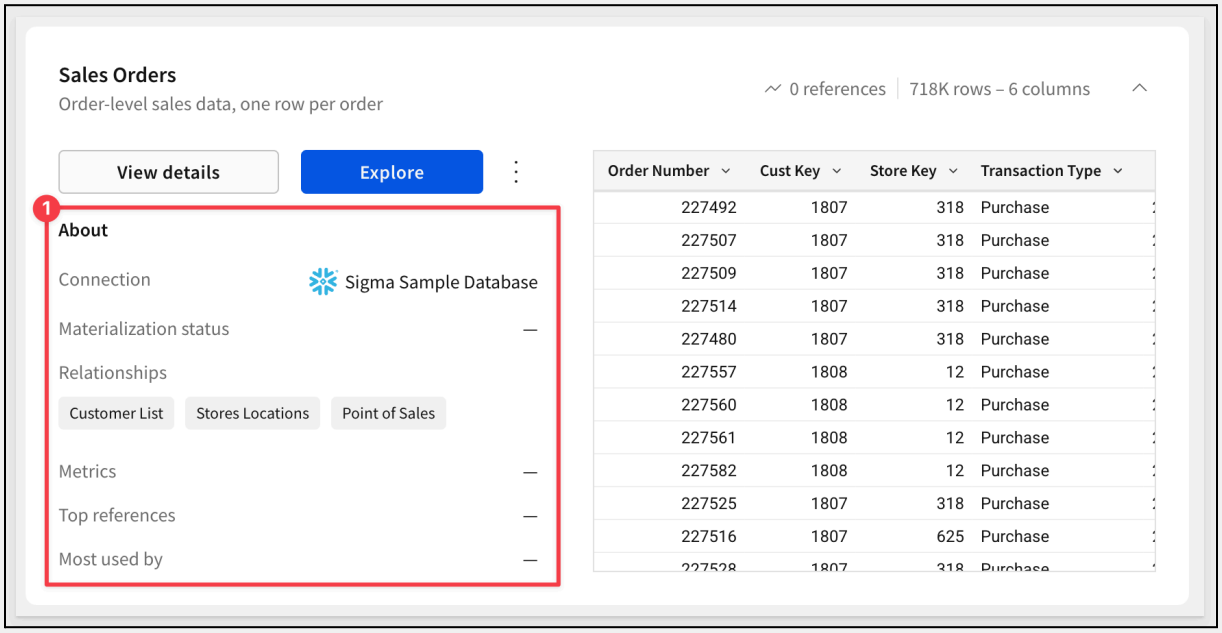
We can see all the tables in the model and some basic information about it like a description and when it was last updated.

We can get detailed information about each table, for example expanding the `Sales Orders` table:



Here we can see all the detailed information about the table, its relationships, metrics, and so on. We also can see the columns in the table and how many rows there are.

Users can click `Explore` and jump straight into a workbook with the table pre-loaded so that they can start building immediately:



This is just a preview to orient you while we build out the data model more.

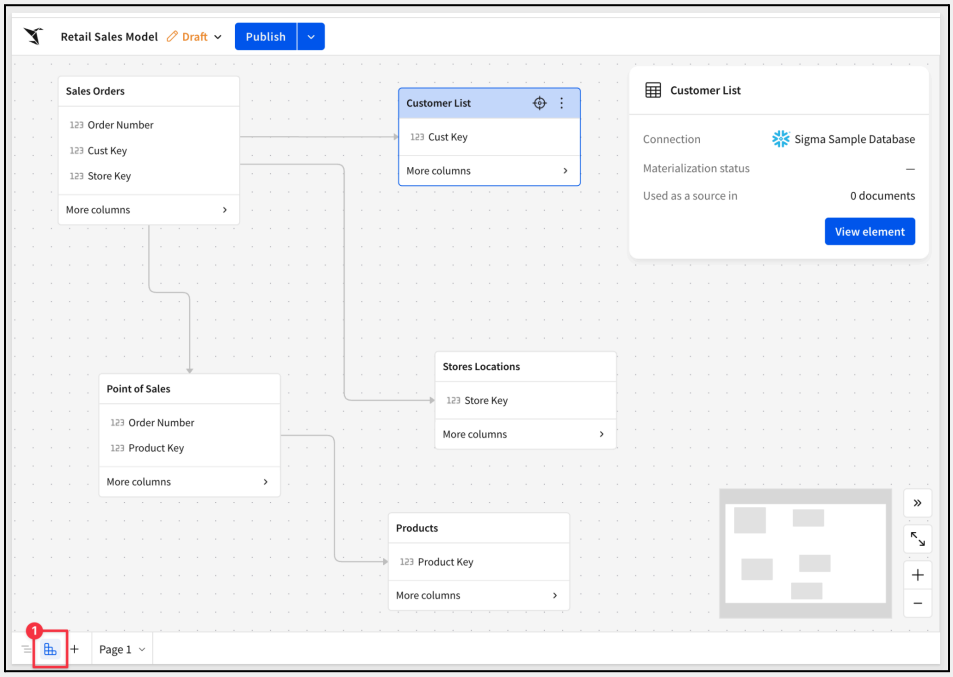
For more information on navigating the landing page, see [Data model overview page]

Click `Edit` so we can build more functionality into the model.

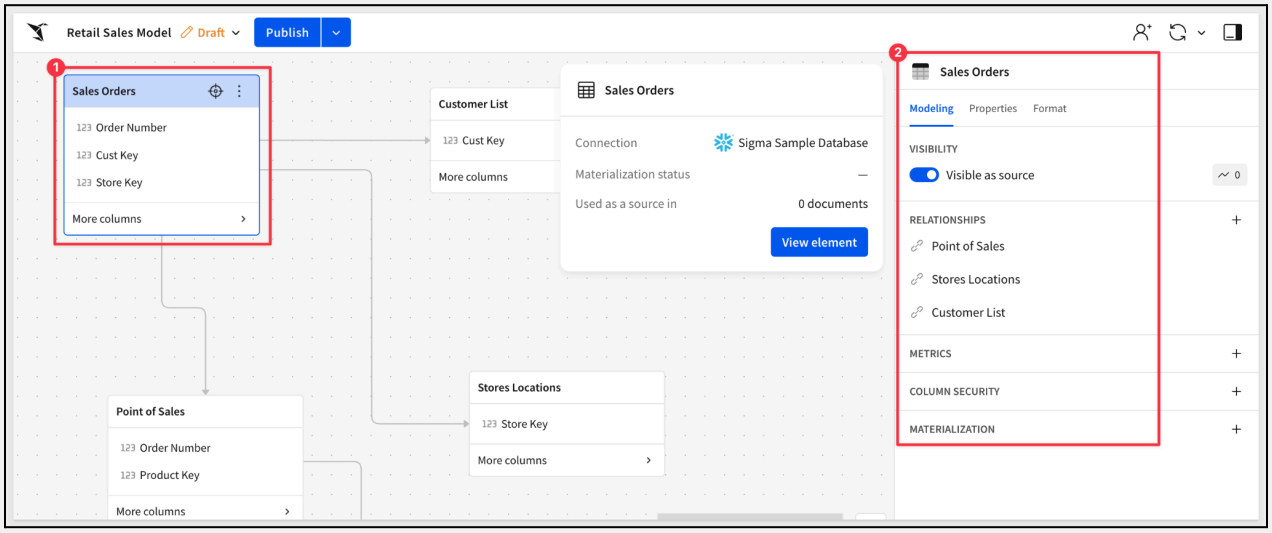
5. Data Model Entity Relationship Diagram (ERD)

By clicking on the “stack-boxes” icon in the lower left corner of the page, we access the entity relationship diagram (ERD) which provides a more visual layout that can be more familiar to some users.

All the same operations are available via the ERD as the workbook page. The tables can be dragged around to visualize the design to suit:



If we select the `Sales Orders` table, we have access to the `Element panel` and all its existing configuration:



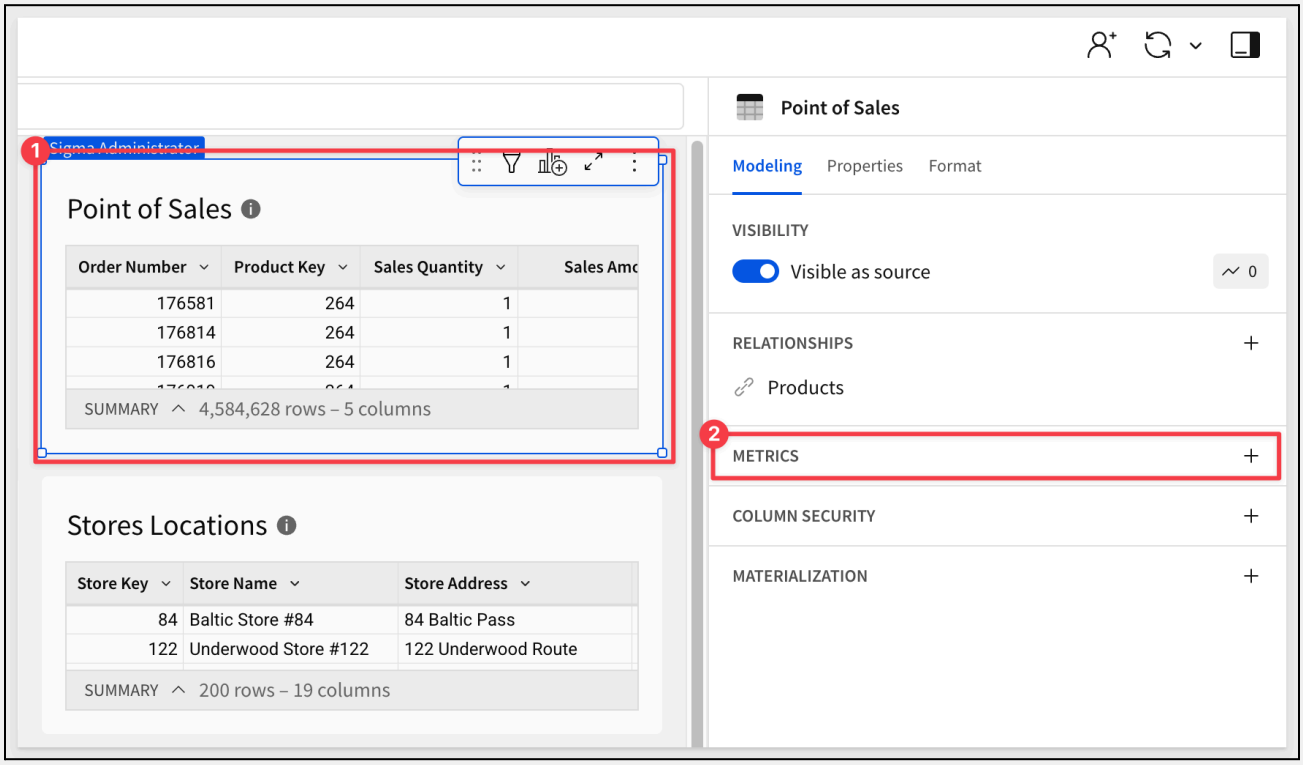
For more information, see [Guidance for modeling relationships]

Click `Publish` and then click the `ERD` icon again to exit it.

6. Metrics

Metrics in Sigma are dynamic and reusable calculations that are specific to a data source. As such, metrics usually reference specific columns. You can create a metric to provide reliable and efficient aggregate calculations, abstracting complex formulas away from business users.

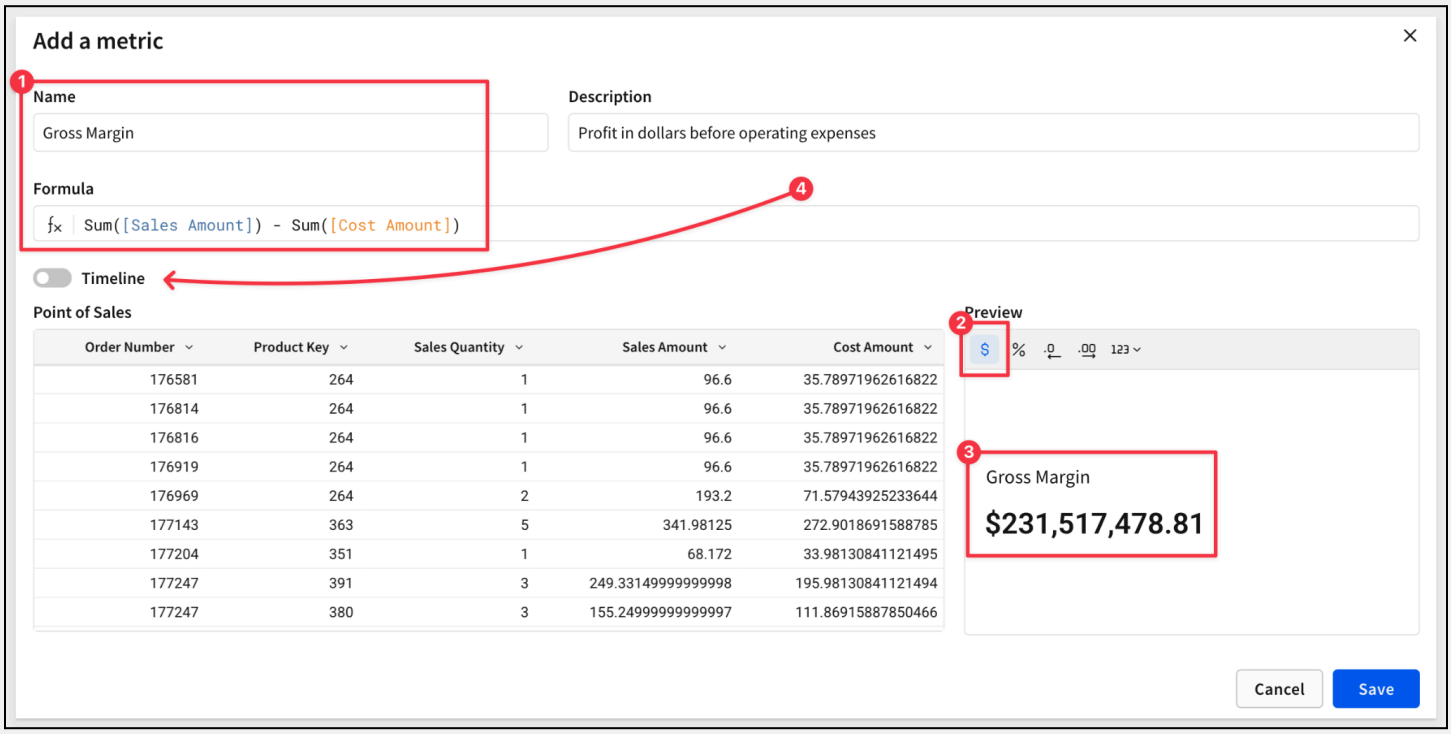
Let's add a few simple metrics by selecting the `Point of Sales` table and clicking the `+` for `Metrics`:



NOTE
The complexity of the calculation is not the focus right now. The point is that Sigma makes it easy to create anything from simple to wildly complex calculations to use as metrics. This way, users of the data model can use the result and not make mistakes by creating their own calculations for common data.

Next, configure the following metrics:

Gross Margin:

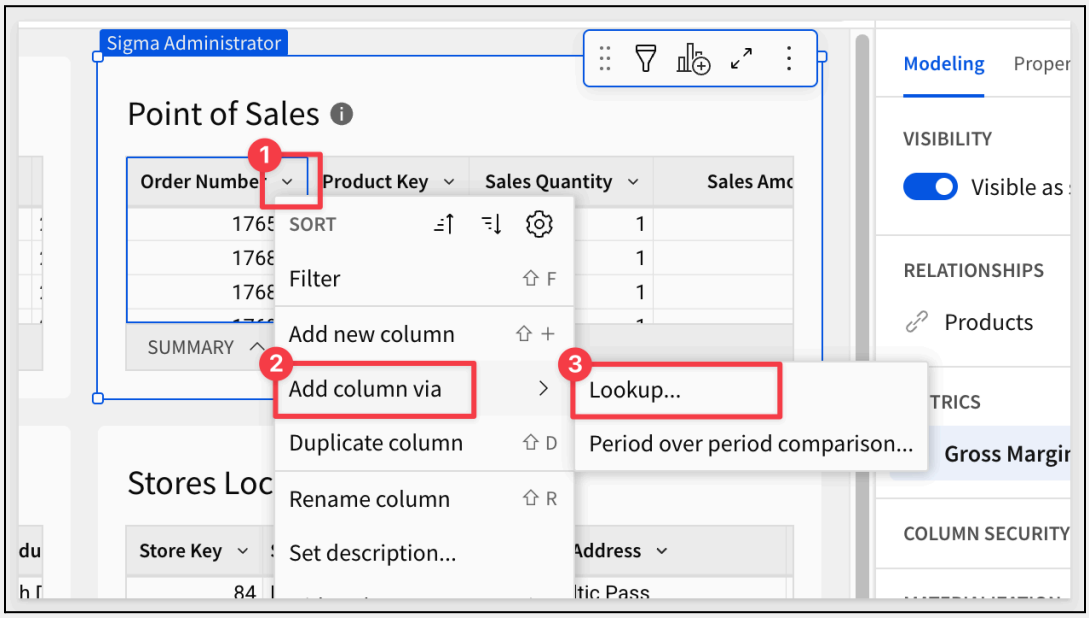


Did you notice the `Timeline` option?

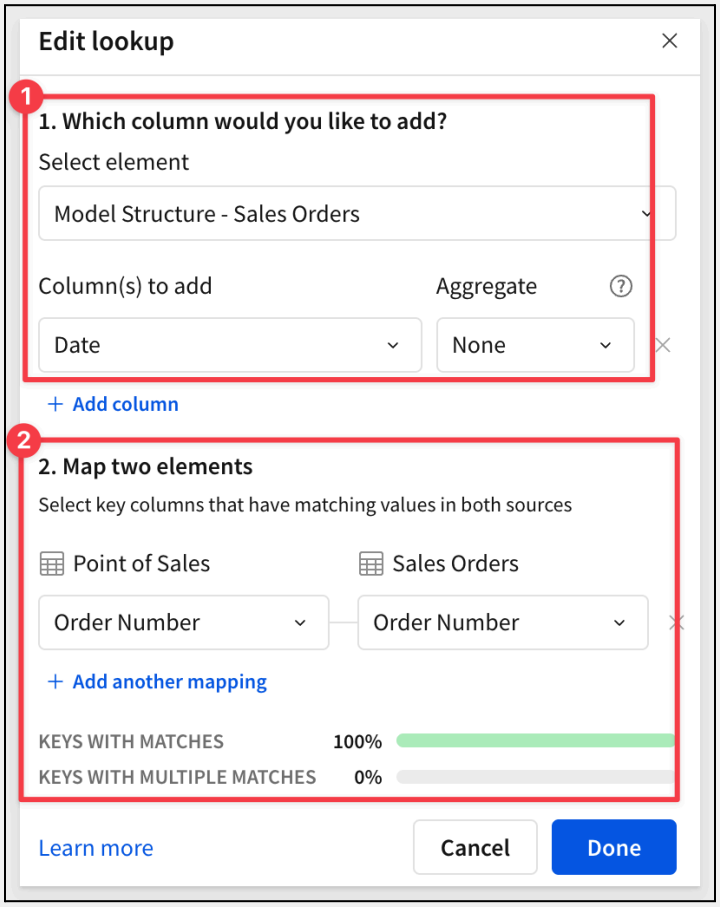
That's interesting, but notice we don't have a date column in `Point of Sales`. **No problem!** We can simply use a Lookup to pull it from the `Sales Orders` table so we can use the timeline feature.

Click `Save`.

Add a new column via lookup as shown:



Configure the lookup:

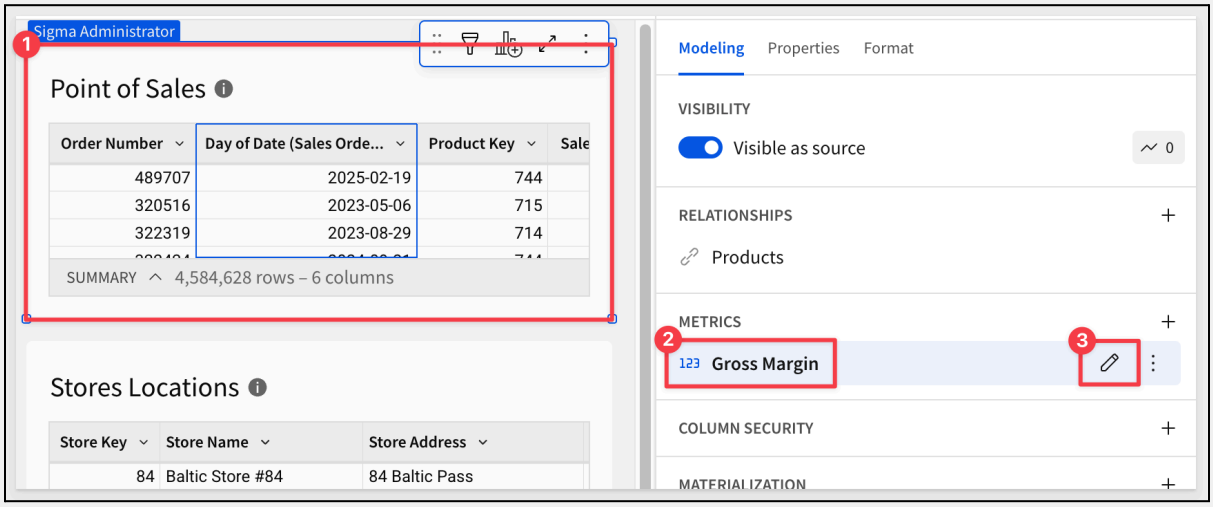


There is a 100% match as every order has to have occurred on a date.

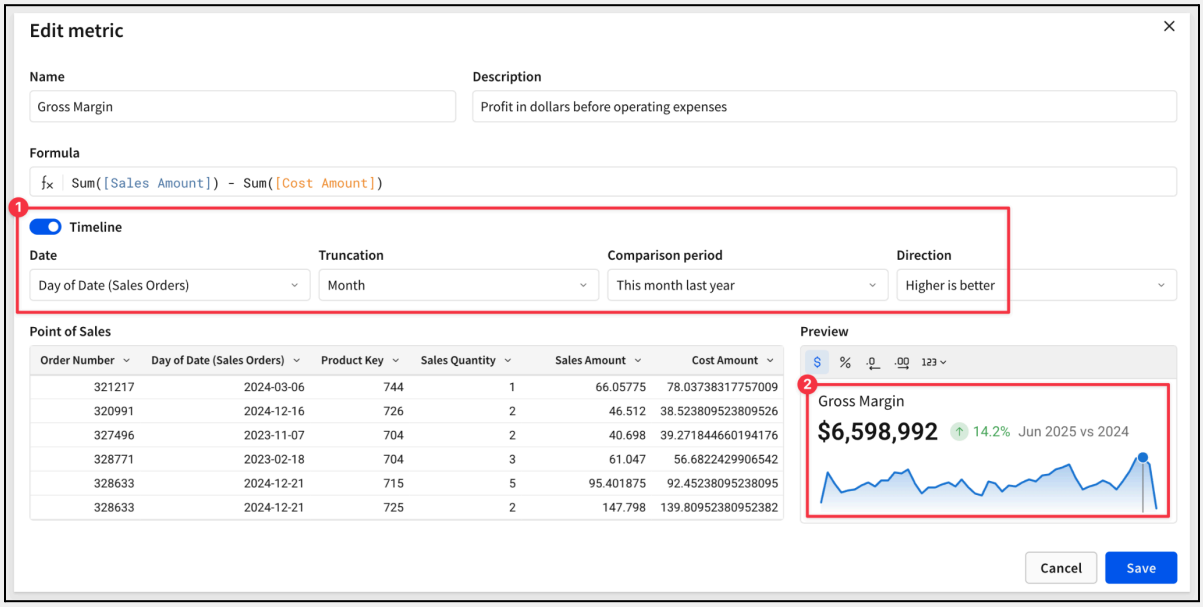
Click `Done`.

Now we have the column we need to use the `Timeline` feature.

Click the pencil icon to edit the `Gross Margin` metric:



We can configure the timeline using the date column and select from several pre-configured comparison periods. Once selected, the metric has a chart and displays data related to the comparison period:

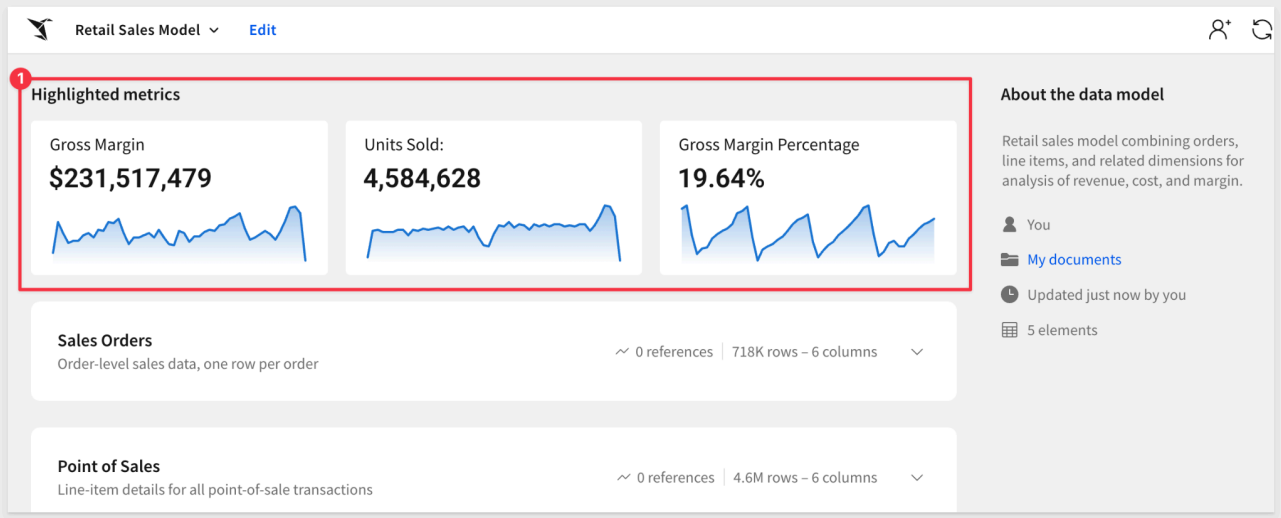


Click `Save`.

Repeat this process to add two more metrics:

Units Sold:
`COUNT([Sales Quantity])`
Total number of units sold in the period

Gross Margin Percentage:
`(SUM([Sales Amount]) - SUM([Cost Amount])) / SUM([Sales Amount])`
Profitability as a percentage of revenue
Once done, we can publish the data model and `Go to the published version` again to see how metrics are displayed:



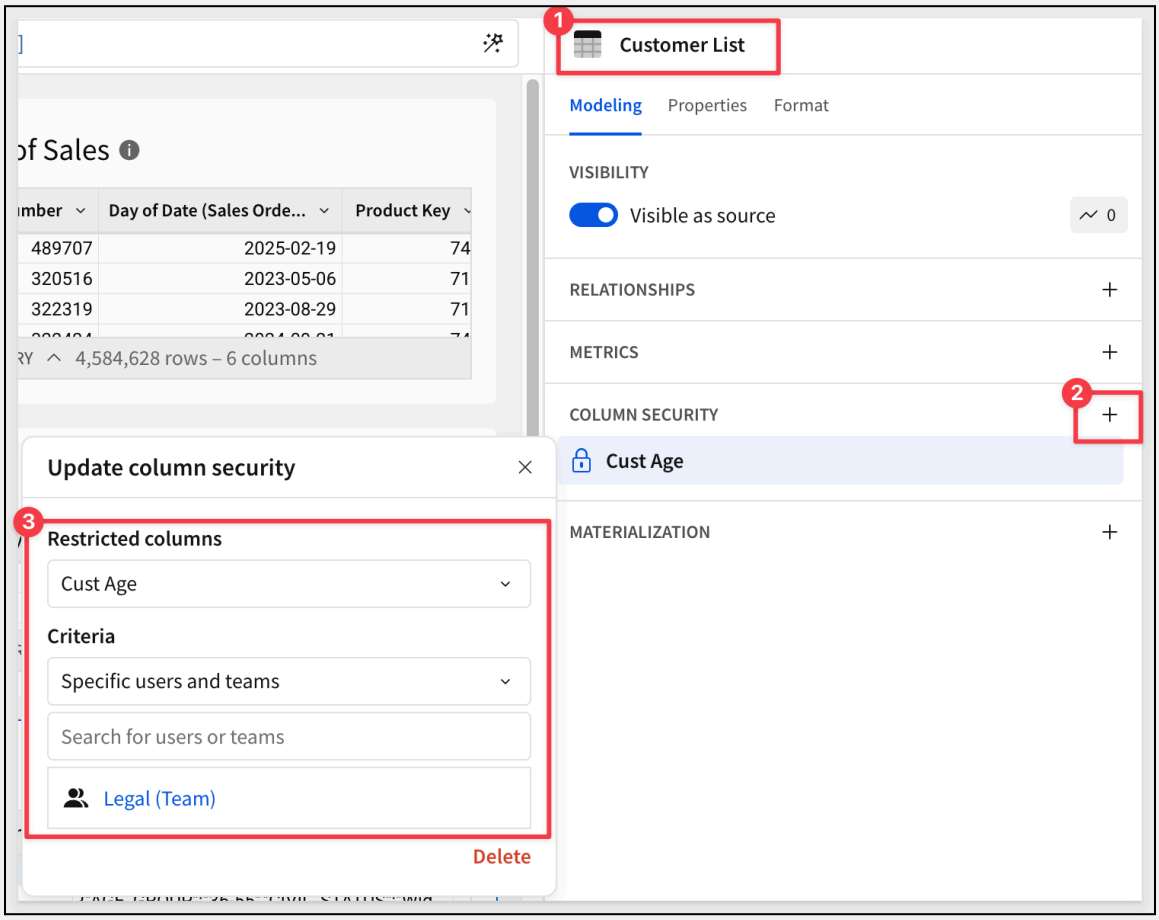
7. Protecting sensitive data - Column-Level Security (CLS)

Often the same data model will be used by users who have different needs and permission levels. In some cases, showing all columns is permissible, but in others it isn't.

For example, if we look at the `Customer List` table, we see a column `Cust Age`; we need to protect that so only select users have access to it.

In this case, we only want our "Legal" team members to access that column.

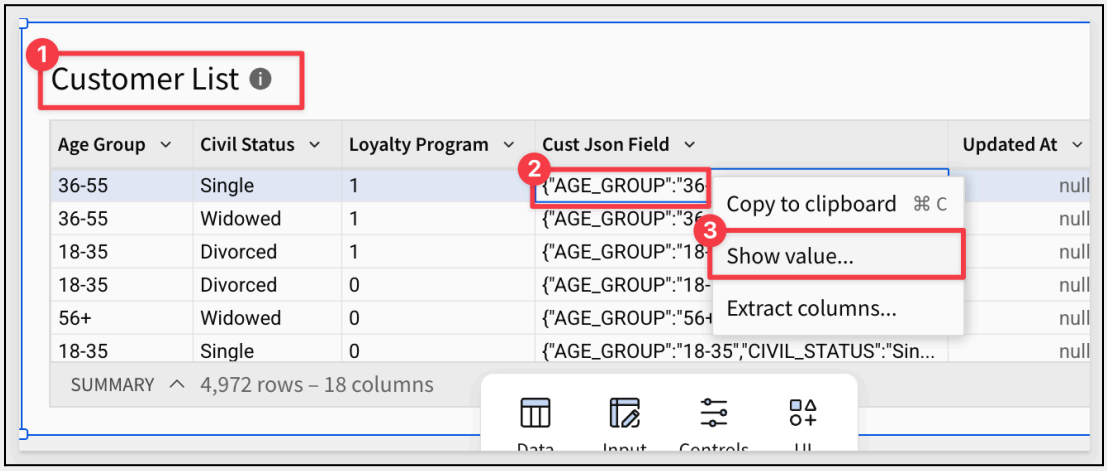
Configuring that is simple, and we can also select specific users:



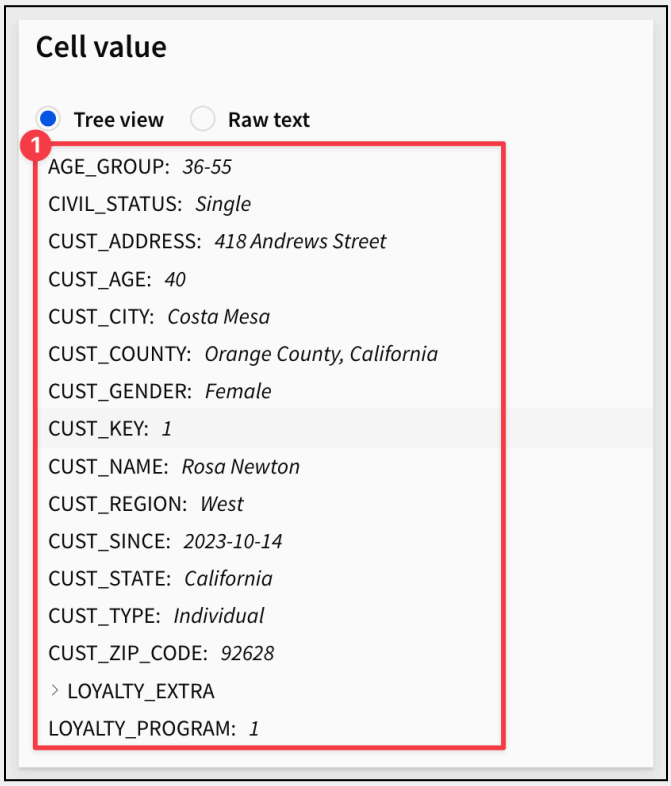
Now when users access the data model, if they are not a member of the Legal team, they will not see the column or know it even exists in the data.

There is also a column of JSON data that came through from the warehouse. We need to be sure that the data in that is permitted. Working with JSON data can be a pain but Sigma has handled the heavy-lifting for us.

First, let's see what's in the data:



This looks like data we already have in existing columns so we can safely delete the column since it does not provide additional data:



NOTE
If the JSON data did have a value, Sigma can extract it directly in a few simple "wizard-based" steps—no additional tooling required! For more information, see the QuickStart, Parsing JSON Data in Seconds

8. Materialization

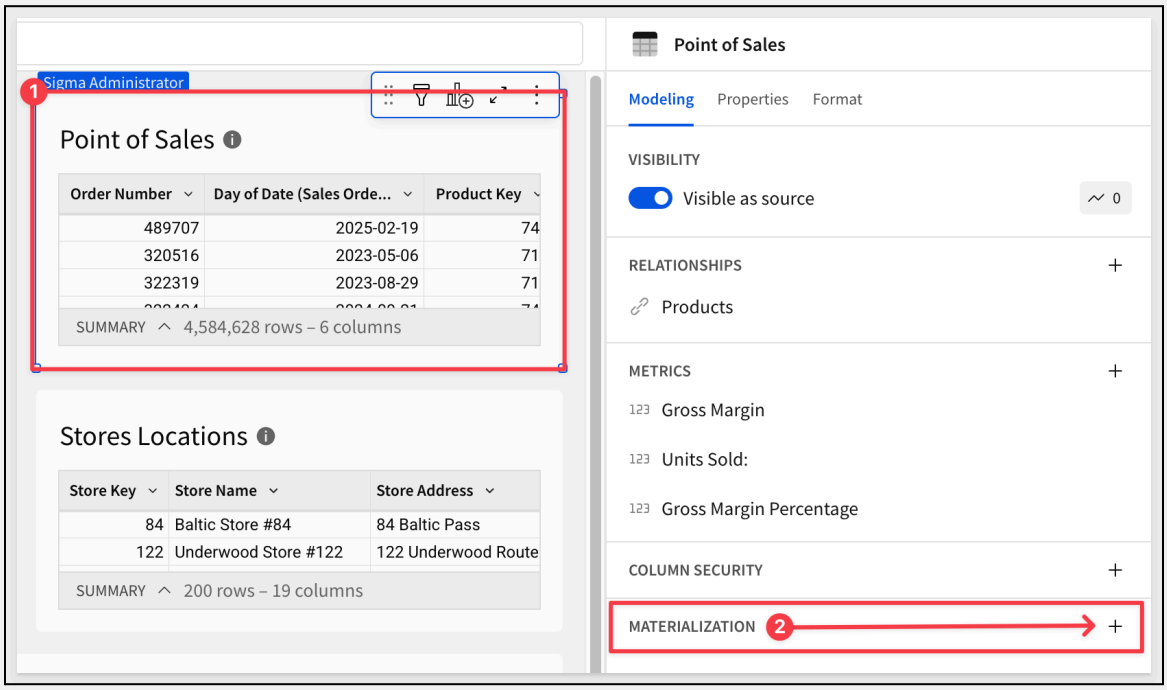
Materialization is built-in and easy to use. For those not familiar, materialization is a type of caching, where query results are written into a table in a data warehouse, and then refreshed at regular intervals (often daily).

This pre-calculated or pre-aggregated data can be accessed more efficiently and quickly than recomputing the results every time the query is executed.

In our use case, we don't have a very complex model, but when a model has a table that is very large, contains many complex calculations, or other complexities, it can be a good candidate for materialization.

For example, the `Point of Sales` table has 4.5M rows and three metrics, so let's use that to demonstrate the process.

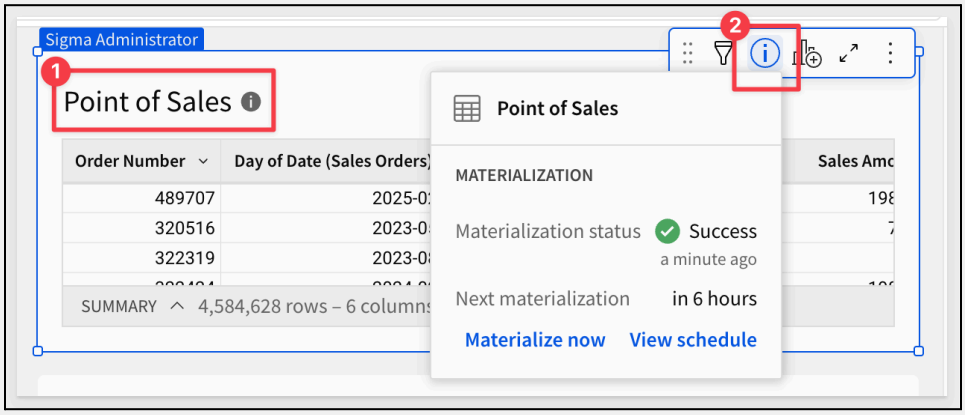
Select the table and create a materialization:



Sigma has drastically reduced the effort it takes to materialize data into the UI directly. We simply create a schedule, selecting the elements and frequency to suit our needs.

Once created, the status is `Pending` and will be run immediately, and then will run on the schedule. We can also execute it on-demand:

We can also see that materialization is configured on the `Point of Sales` table and get the status directly:



NOTE
There are "data-freshness" implications when using materialized data that need to be carefully considered. Materializing improves performance by storing a snapshot of query results, but those results will only update on the refresh schedule you define. This means users may not see the most up-to-date transactions until the next refresh. The optimal approach balances performance gains with your organization's tolerance for latency in reporting. For more information, see the QuickStart, [Materialization with Sigma]()

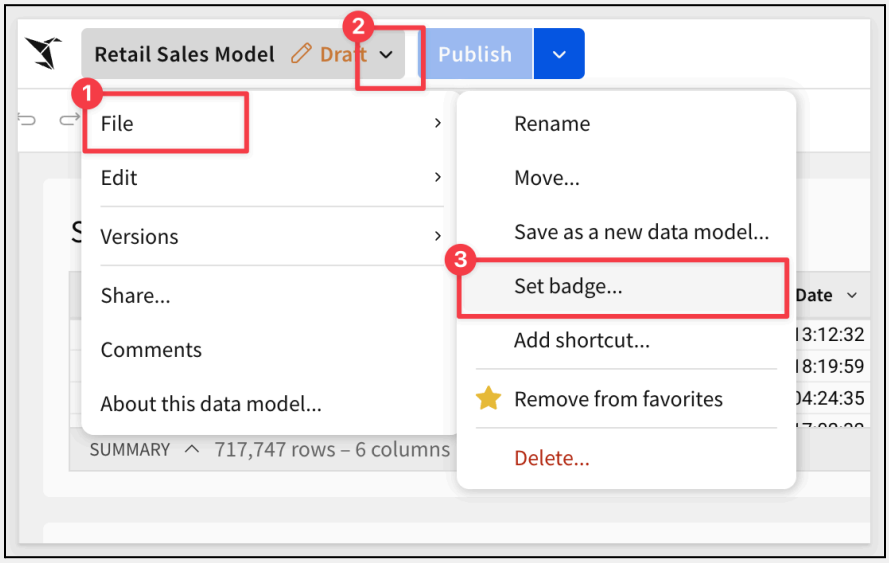
Click `Publish`

The final step is to share the data model with specific users or teams. This step is optional but useful to see the impact of column-level security.

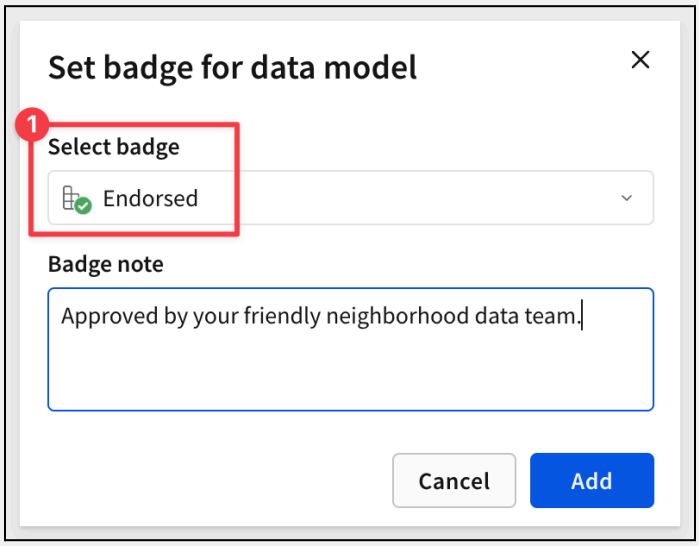
We created a few test users in our Sigma trial (using Gmail's alias feature) so that we could assign different roles for testing.

We will share the data model with one of them, permitting `View` permission:

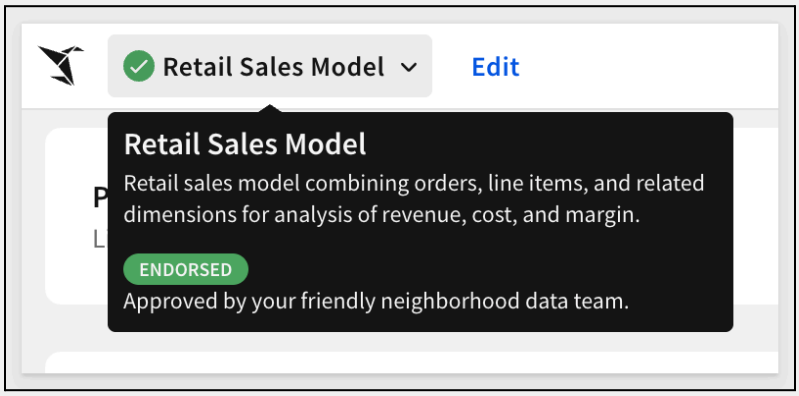
Lastly, we should `Badge` the model so that users know they are using the correct data:



Choose the `Endorsed` badge:

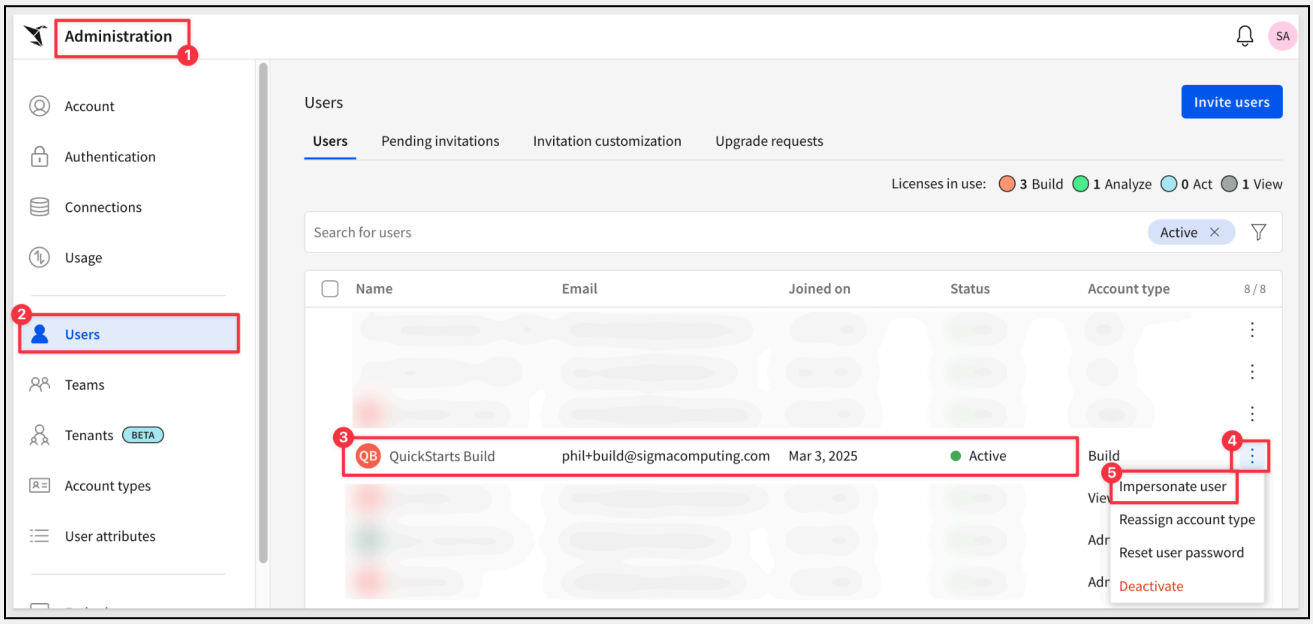


Now when users access the model, they can immediately tell it is the correct model for retails sales:

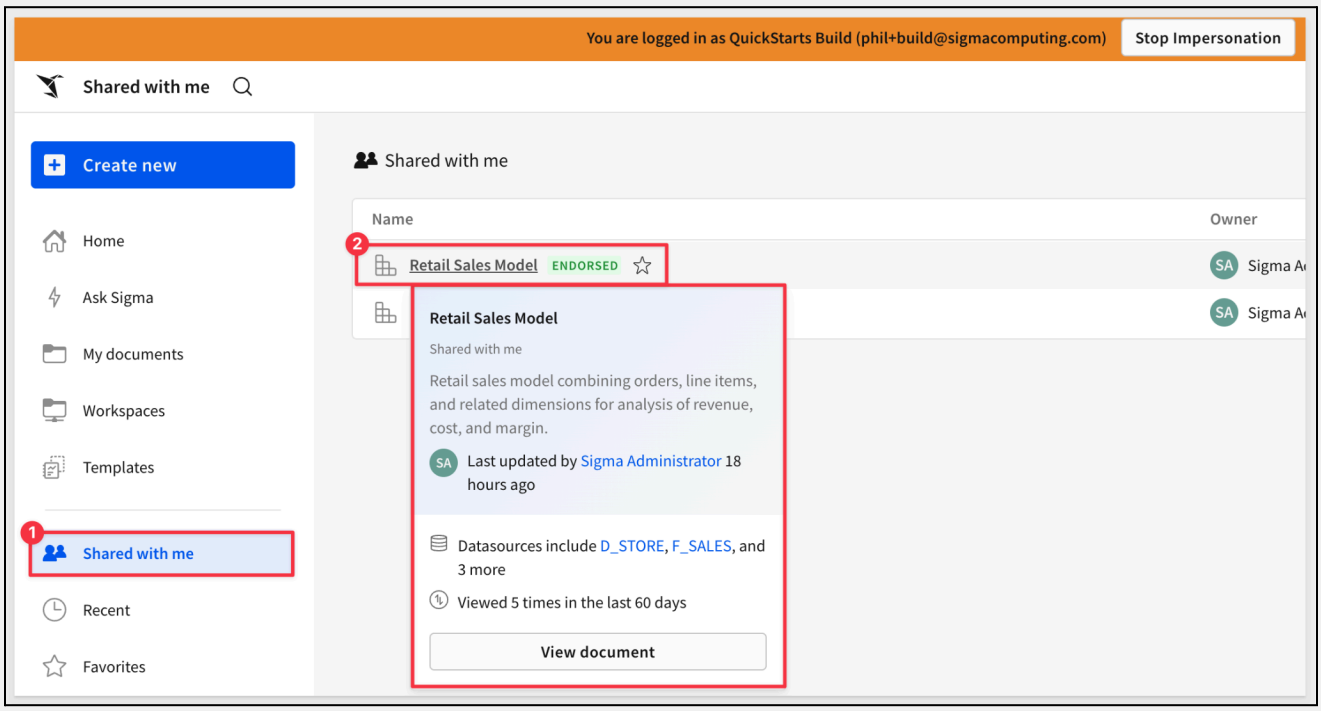


5: Using The Data Model

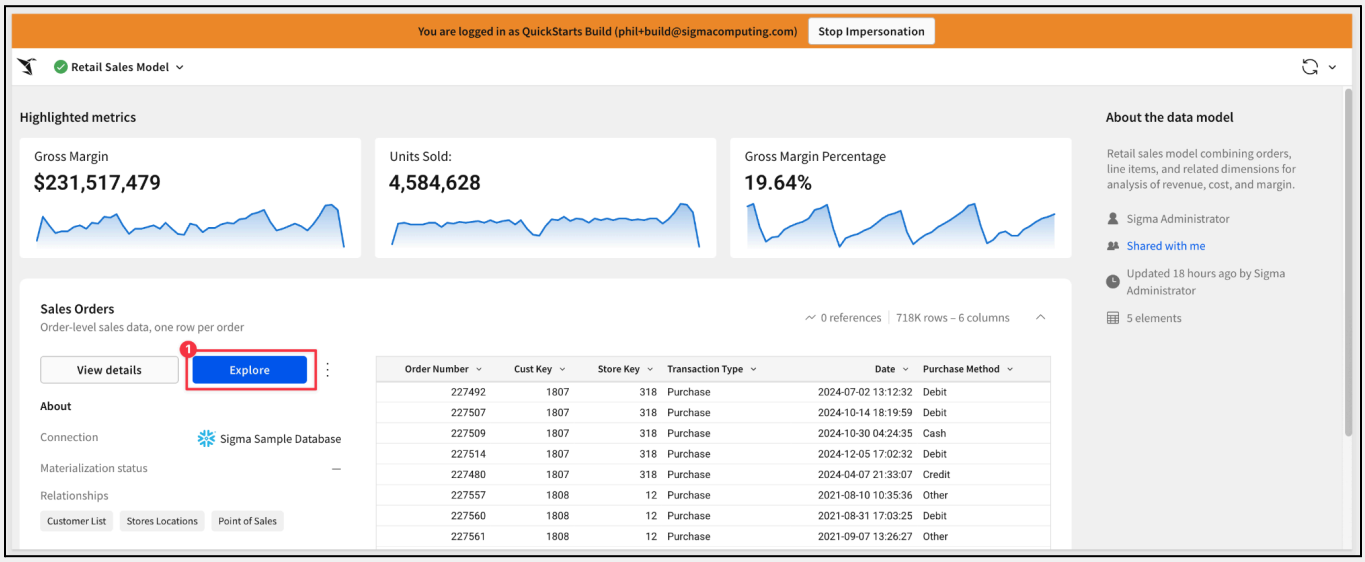
In Sigma, navigate to `Administration` > `Users` and select impersonate for the user that we shared the data model with:



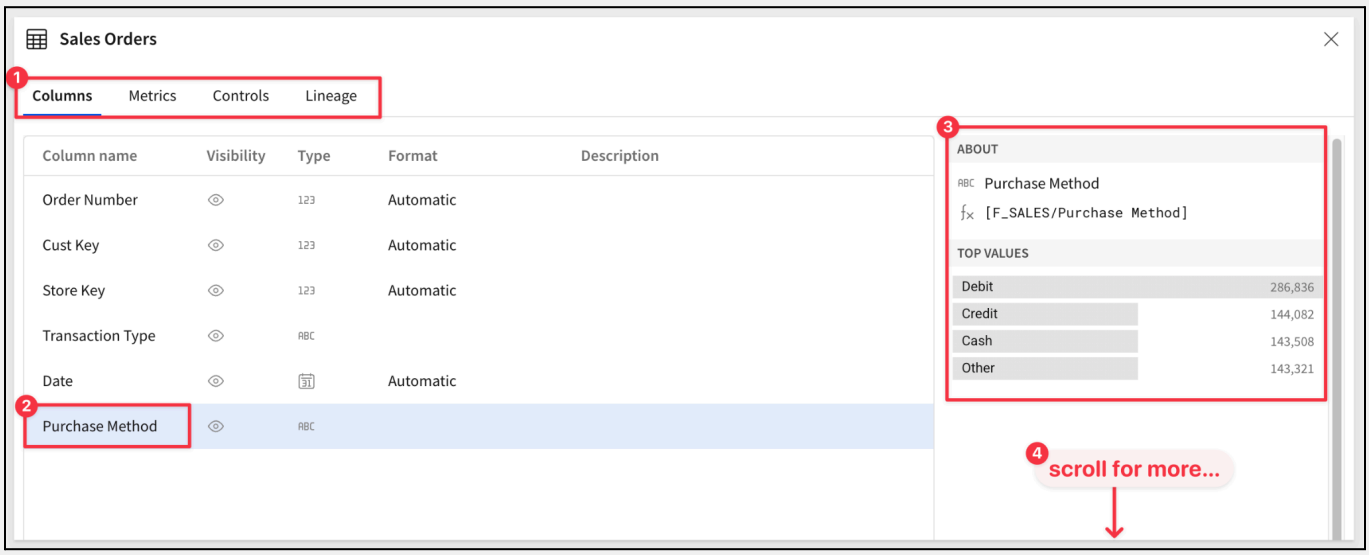
Now we are working in Sigma as if we are the selected user. On the `Shared with me` page we can see the `Retail Sales Model` listed; click on that:



Now we can see all the detail that was pre-configured for us and can select from any table to launch into an exploration:



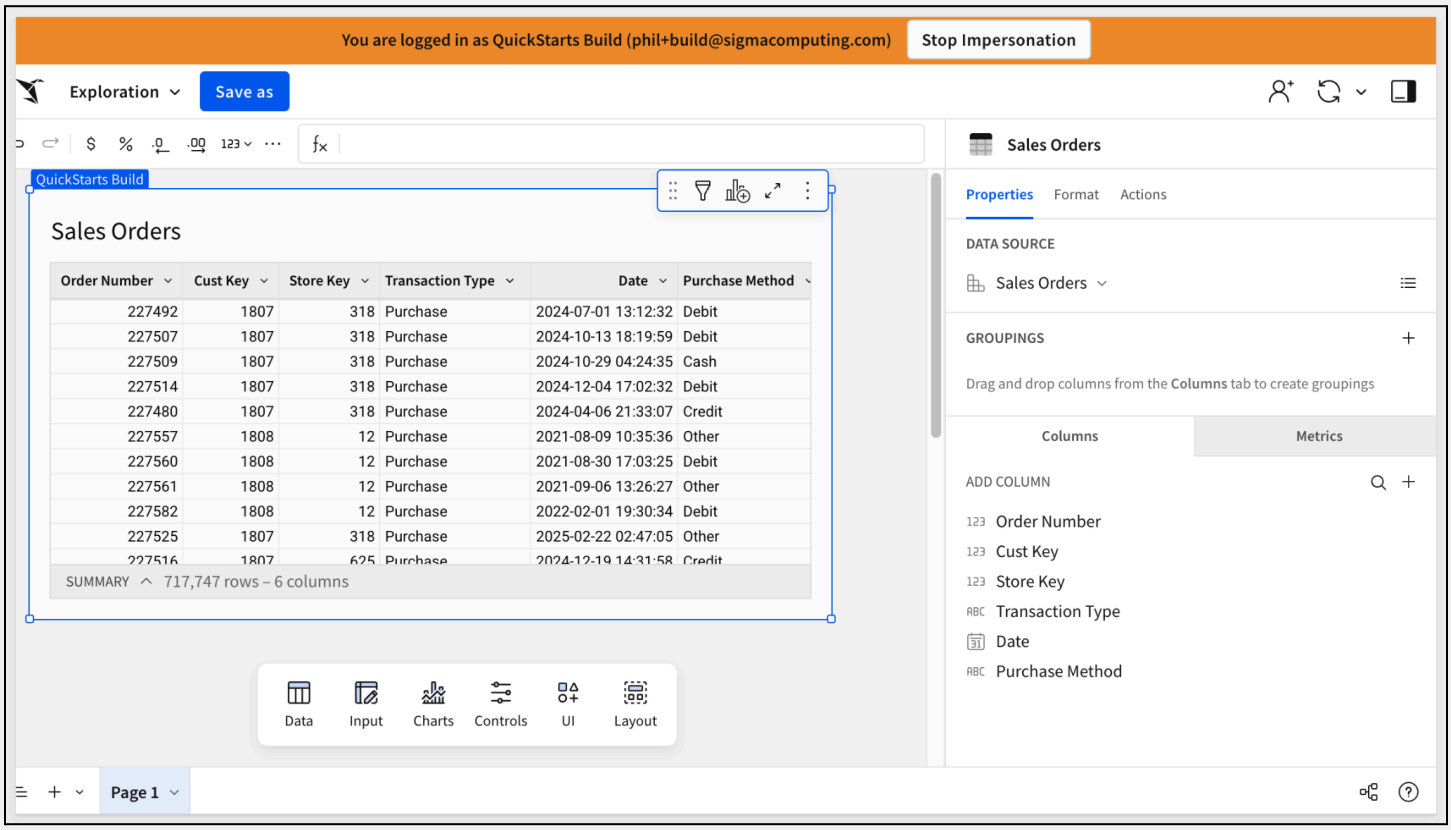
The `View Details` button provides a way to see lower-level information on columns, metrics, controls, and lineage for each table:



Note: our metrics are on the `Point of Sales` table so they won't appear in the 'Sales Orders' > `Metrics` tab.

Close the detail view and click the `Explore` button to land in a workbook.

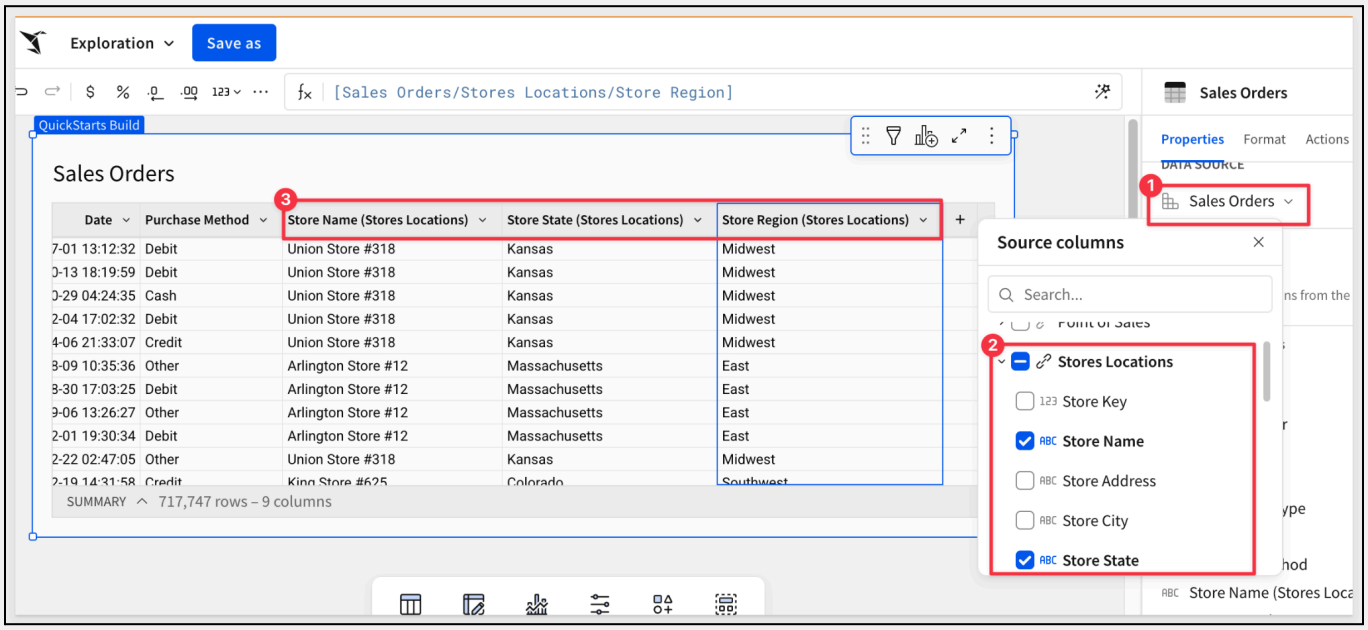
A typical Sigma workbook appears with the table pre-loaded, but there is something special about the table:



Because we configured relationships in the model, the user has access to all the other visible tables in the model and can select whichever columns they require for their analysis.

IMPORTANT
Sigma has automatically applied join pruning so that the auto-generated machine SQL used to load the data from the model—and subsequently the warehouse—only requests the data from the `Sales Orders` table. Until the user selects columns from another table they will not be requested from the warehouse. This makes the queries both efficient and cost effective. For more information, see the QuickStart, [Sigma's Query Engine]()

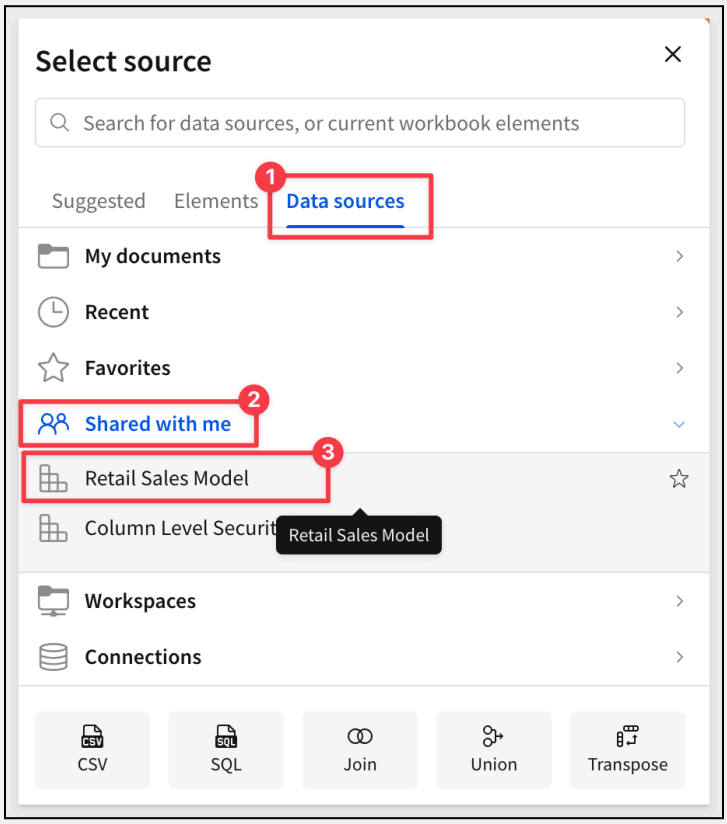
For example, we may want to add some store location related data that we can group on:



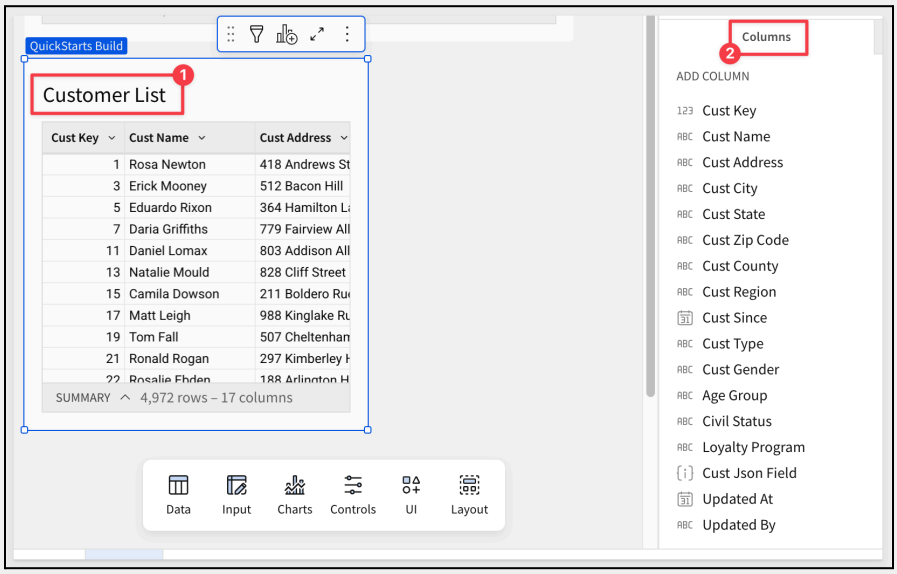
While this is interesting, we may want to add one of the tables from the model directly. That's easy to do and follows Sigma's familiar workbook workflow.

Just click on `Data` in the `Element bar` and select `Table`.

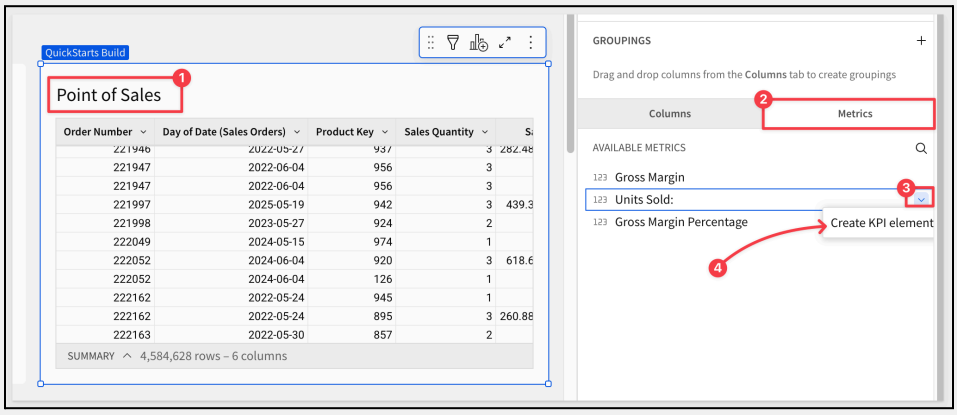
Then select the `Data sources` tab, `Shared with me` and select the `Retail Sales Model` which will act just like any other source of data:



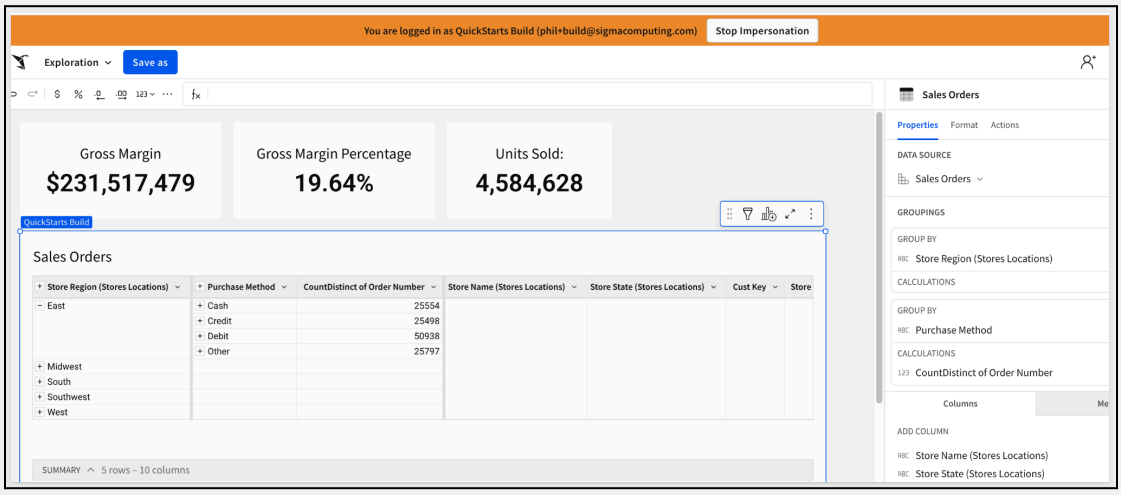
Let's select the `Customer List` table. Notice that the `Cust Age` column does not appear due to our CLS configuration:



We can also add KPI elements using our pre-built metrics:



Now that our test user has access to the data, they can build whatever they need and rest assured it is approved for use, accurate, and will produce analytics that can be trusted:



6: Best Practices

As your Sigma deployment grows, governance and reusability become critical. A well-modeled Sigma data layer reduces rework, enforces consistency, and ensures that changes don't break downstream content.

Model reusability

- Build reusable models that serve as canonical sources of truth.
- Centralize joins and relationships **inside Sigma Data Models**, not in individual workbooks.
- Define metrics at the model level so they can be reused across dashboards, Input Tables, and Ask Sigma.
- Use links to prevent logic sprawl and enforce consistent joins across datasets.
- Add navigation controls inside the model so they can be selected for use easily in workbooks.

Versioning and change management

- Use Sigma's content validation to identify downstream dependencies before pushing changes.
- Tag major versions and maintain copies if needed to support parallel development.
- Adopt a branching workflow when working with shared data models or multiple contributors.

Testable, trustworthy logic

- Test your data logic—use row counts, joins, and group-by checks before publishing.
- Avoid unnecessary nesting in calculated columns in favor of model-level calculations or metrics.
- Enable lineage to trace how each field was derived and where it's used.
- Endorse data models and provide descriptions whenever possible.
- Minimize fanout and duplicate aggregates.
- Use relationships and link keys to control join paths.

A governed Sigma data model isn't just a modeling layer—it's a **direct-to-warehouse governance platform**. By defining relationships, metrics, and permissions in Sigma while executing queries in your warehouse, you avoid the sprawl and shadow logic that plague legacy BI stacks. The result: fewer surprises, faster iteration, and a modeling layer that earns trust at every level—from dashboards to AI assistants.

7: Reference Architecture

Modern analytics stacks are converging on a shared pattern: data is transformed and secured in the warehouse, while Sigma provides the governed, reusable modeling layer on top, executing queries directly on the warehouse for real-time, secure access.

In a typical Sigma deployment:

1. **Raw warehouse tables** (often landed via tools like Fivetran or Airbyte) are transformed into subject-specific data marts using dbt or similar tools.
2. These curated marts feed **Sigma Data Models**, where relationships, metrics, and permissions are defined once and reused across dashboards, Input Tables, embedded analytics, and AI assistants.
3. Sigma connects live to the warehouse for all queries, preserving full security controls (RLS/CLS), performance tuning, and cost visibility.

This architecture enables:

- A single governed layer for business logic, owned and maintained in Sigma.
- Governed and reusable across all personas—BI, operations, finance, and embedded app developers.
- Real-time data access without extra ETL layers, cubes, or data extracts.

Sigma sits at the center of your analytics workflow—integrating governed data models, enterprise-grade platform controls, AI-assisted analysis, and a full range of workbook experiences—while remaining directly connected to your warehouse. The result is trusted, consistent analytics for every user, from spreadsheets to AI queries.

In Figure 1 below**,** the top layer represents the many ways users work with governed data models inside Sigma, from spreadsheets and reports to AI-driven queries and embedded applications.

The middle layers show Sigma's AI and enterprise platform capabilities, **which sit alongside data modeling to deliver secure, scalable analytics.**

At the bottom are the live data sources—cloud data warehouses, databases, and AI models—that Sigma connects to directly for every query.

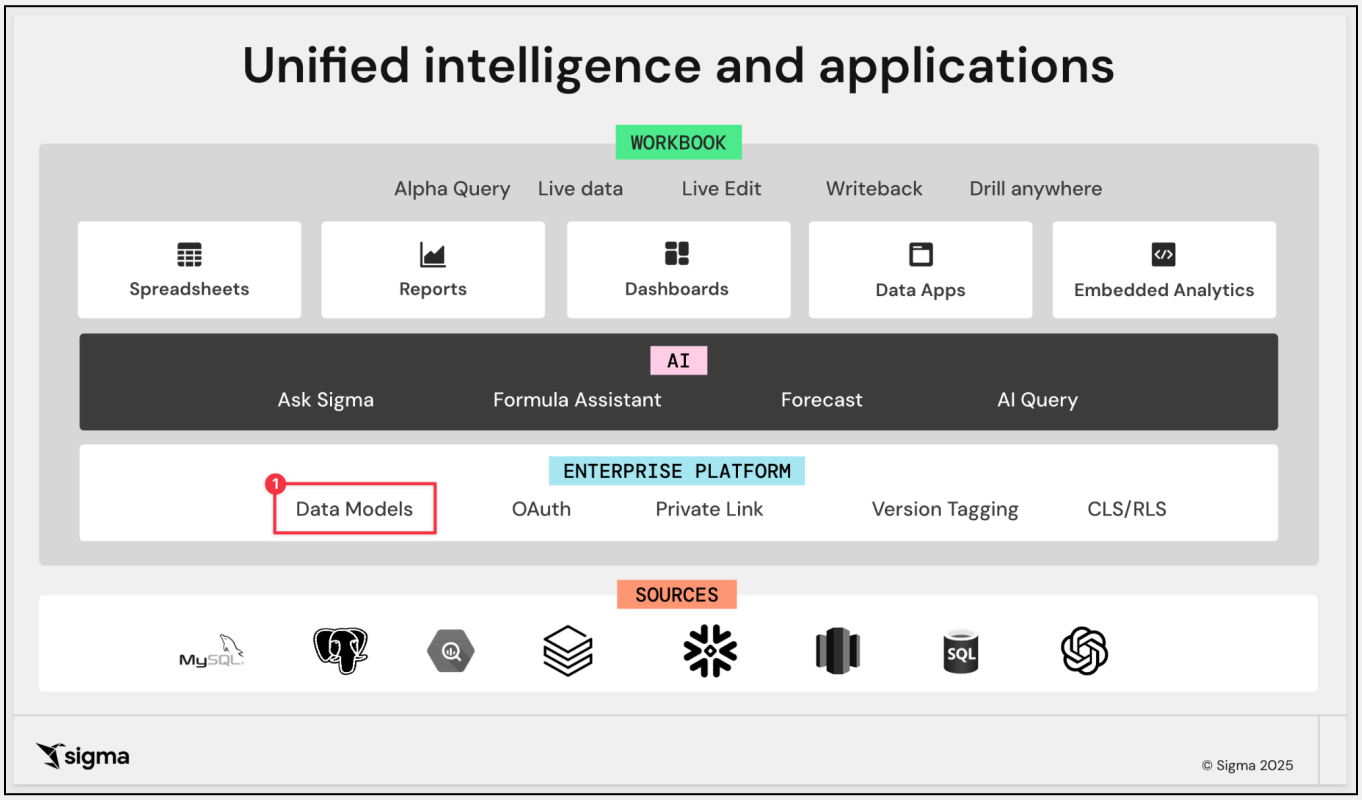


Figure 1

8: What's Next?

As data modeling matures, it's evolving beyond just defining structure—it's becoming the semantic foundation for every analytics experience. Sigma Data Models are already delivering that foundation today, providing a governed, reusable layer that **executes directly on your warehouse** and serves every type of user and workflow.

The broader analytics ecosystem is also moving toward more formalized semantics within the warehouse itself. Platforms like Snowflake, Databricks, and BigQuery are introducing capabilities to define metrics, relationships, and filters as first-class warehouse objects. Sigma is working closely with all leading cloud data warehouse providers to ensure that as these features emerge, they can integrate seamlessly with the governed models you've built in Sigma.

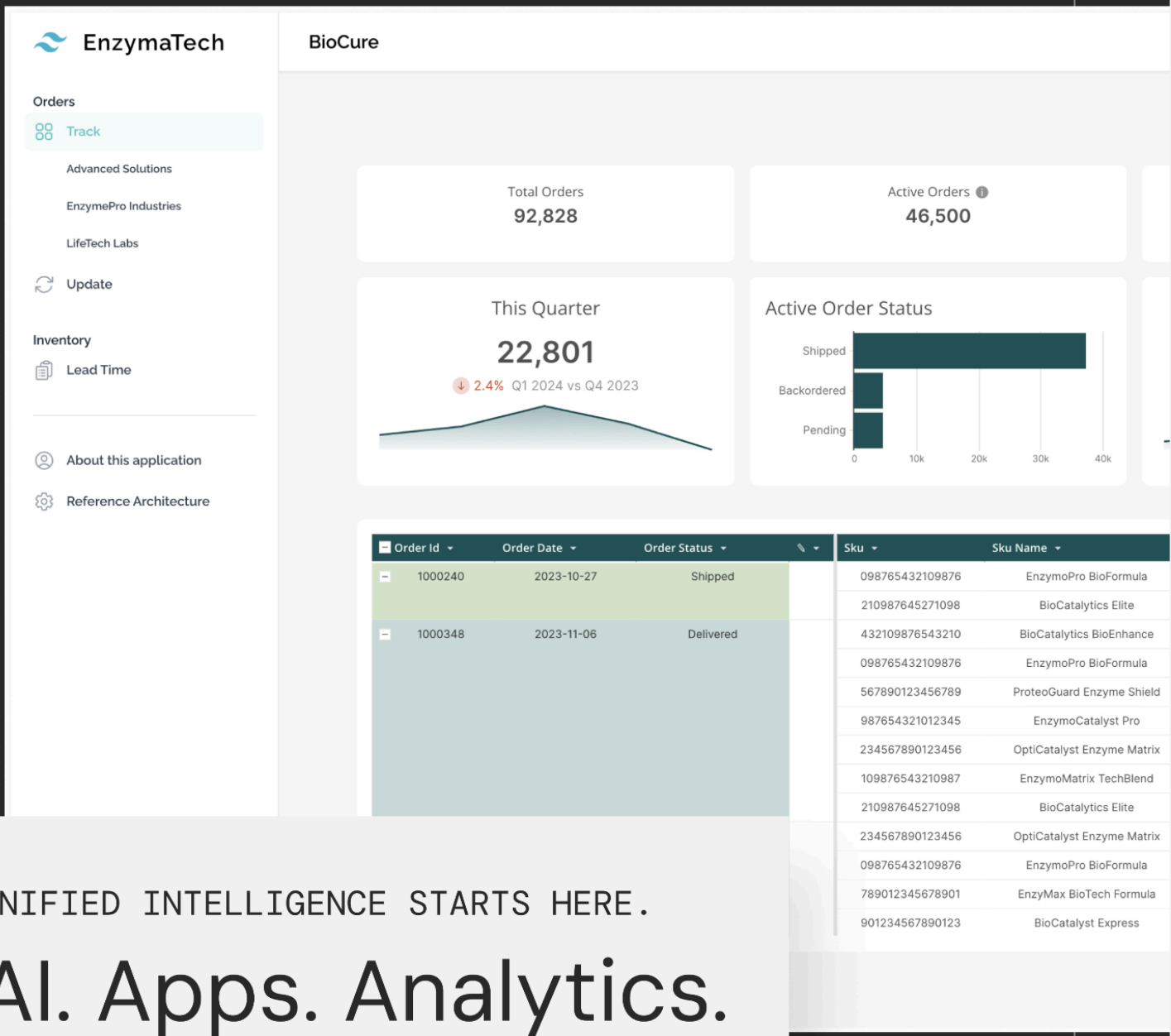
For now, the fastest path to delivering consistent, trusted, and business-friendly analytics is to model directly in Sigma. Doing so allows you to define and manage business logic in one place, reuse it everywhere, and keep it fully aligned with your warehouse's performance and security model. As warehouse-layer semantics mature, Sigma Data Models will be ready to extend and interoperate, ensuring your governance strategy remains future-proof.

BE SURE TO CHECK OUT ALL THE LATEST DEVELOPMENTS AT

Sigma's First Friday Feature page!

Help Center Home | Sigma Community | Sigma Blog

sigma



UNIFIED INTELLIGENCE STARTS HERE.

AI. Apps. Analytics.

Run granular analytics on billions of records without compromising speed or security—all from an intuitive interface anyone can use.