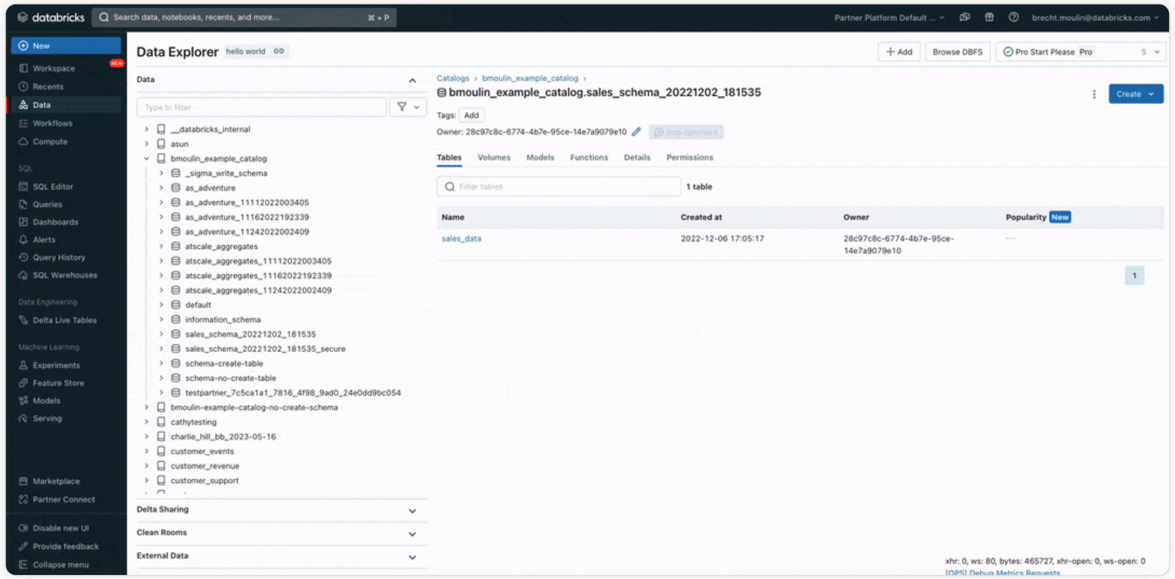
Databricks + PowerBI: Publish to PowerBI Workspaces

Publish datasets from your Databricks workspace to PowerBI Online workspace with a few clicks! No more managing odbc/jdbc connections — simply select the dataset you want to publish. Simply select the datasets or schema you want to publish and select your PBI workspace! This makes it easier for BI admins and report creators to support PowerBI workspaces without also having to use Power BI Desktop.



bmoulin_example_catalog.sales_schema_20221202_181535

Owner: 28c97c8c-6774-4b7e-95ce-14e7a9079e10

1 table

Name	Created at	Owner	Popularity New
sales_data	2022-12-06 17:05:17	28c97c8c-6774-4b7e-95ce-14e7a9079e10	---

xhr: 0, ws: 80, bytes: 465727, xhr-open: 0, ws-open: 0

Getting Started with Databricks SQL

Follow the guide (AWS | Azure | GCP) on how to setup a SQL warehouse to get started with Databricks SQL today! Databricks SQL Serverless is currently available with a 20%+ promotional discount, visit our pricing page to learn more.

You can also watch **Databricks SQL: Why the Best Serverless Data Warehouse is a Lakehouse** and **What's New in Databricks SQL — With Live Demos** for a complete overview.

SECTION 4.5

Distributed Data Governance and Isolated Environments With Unity Catalog

by Max Nienu, Zeashan Pappa, Paul Roome and Sachin Thakur

Effective data governance is essential for any organization that relies on data, analytics and AI for its operations. In many organizations, there is a growing recognition of the value proposition of centralized data governance. However, even with the best intentions, implementing centralized governance can be challenging without the proper organizational processes and resources. The role of Chief Data Officer (CDO) is still emerging in many organizations, leaving questions about who will define and execute data governance policies across the organization.

As a result, the responsibility for defining and executing data governance policies across the organization is often not centralized, leading to policy variations or governing bodies across lines of business, sub-units, and other divisions within an organization. For simplicity, we can call this pattern distributed governance, where there is a general agreement on the distinctions between these governing units but not necessarily a central data governance function.

In this blog, we'll explore implementing a distributed governance model using Databricks Unity Catalog, which provides a unified governance solution for data, analytics, and AI in the lakehouse.

Evolution of Data Governance in Databricks

Before the introduction of Unity Catalog, the concept of a workspace was monolithic, with each workspace having its own metastore, user management, and Table ACL store. This led to intrinsic data and governance isolation boundaries between workspaces and duplication of effort to address consistency across them.

To handle this, some customers resorted to running pipelines or code to synchronize their metastores and ACLs, while others set up their own self-managed metastores to use across workspaces. However, these solutions added more overhead and maintenance costs forcing upfront architecture decisions on how to partition data across the organization, creating data silos.

Data Governance with Unity Catalog

To overcome these limitations, Databricks developed Unity Catalog, which aims to make it easy to implement data governance while maximizing the ability to collaborate on and share data. The first step in achieving this was implementing a common namespace that permits access to any data within an organization.

This approach may seem like a challenge to the distributed governance pattern mentioned earlier but Unity Catalog offers new isolation mechanisms within the namespace that organizations have traditionally addressed using multiple Hive metastores. These isolation mechanisms enable groups to operate independently with minimal or no interaction and also allow them to achieve isolation in other scenarios, such as production vs development environments.

Hive Metastore versus Unity Catalog in Databricks

With Hive, a metastore was a service boundary, meaning that having different metastores meant different hosted underlying Hive services and different underlying databases. Unity Catalog is a platform service within the Databricks Data Intelligence Platform, so there are no service boundaries to consider.

Unity Catalog provides a common namespace that allows you to govern and audit your data in one place.

When using Hive, it was common to use multiple metastores, each with its own namespace, to achieve isolation between development and production environments, or to allow for the separation of data between operating units.

In Unity Catalog, these requirements are solved through dynamic isolation mechanisms on namespaces that don't compromise the ability to share and collaborate on data and don't require hard one-way upfront architecture decisions.

Working across different teams and environments

When using a data platform, there is often a strong need to have isolation boundaries between environments like dev/prod and between business groups, teams, or operating units of your organization.

Let's begin by defining isolation boundaries in a data platform such as Databricks:

- Users should only gain access to data based on agreed access rules
- Data can be managed by designated people or teams
- Data should be physically separated in storage
- Data should only be accessed in designated environments

Users should only gain access to data based on agreed access rules

Organizations usually have strict requirements around data access based on some organizational/regulatory requirements which is fundamental to keeping data secure. Typical examples include employee salary information or credit card payment information.

Access to this type of information is typically tightly controlled and audited periodically. Unity Catalog provides organizations granular control over data assets within the catalog to meet these industry standards. With the controls, Unity Catalog provides users will only see and query the data they are entitled to see and query.

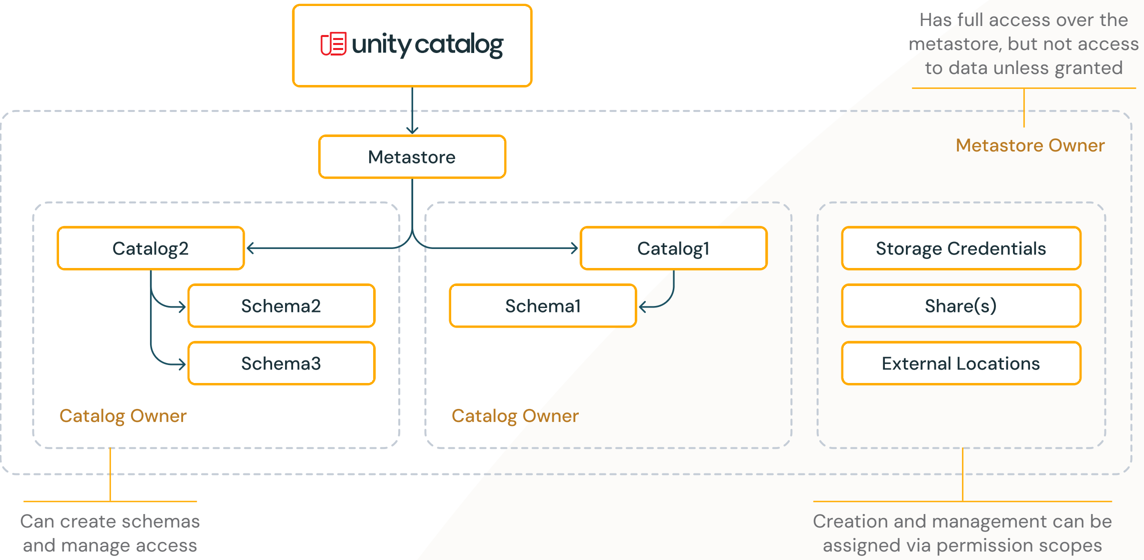
Data can be managed by designated people or teams

Unity Catalog gives you the ability to choose from centralized governance or distributed governance models.

In the centralized governance model, your governance administrators are owners of the metastore and can take ownership of any object and set ACLs and policy.

In a distributed governance model, you would consider a catalog or set of catalogs to be a data domain. The owner of that catalog can create and own all assets and manage governance within that domain. Therefore the owners of domains can operate independently of other owners in other domains.

We strongly recommend setting a group to be the owner or service principal for both of these options if management is done through tooling.


unity catalog
Has full access over the metastore, but not access to data unless granted
Metastore
Metastore Owner
Catalog2
Schema2
Schema3
Catalog Owner
Catalog1
Schema1
Catalog Owner
Storage Credentials
Share(s)
External Locations
Can create schemas and manage access
Creation and management can be assigned via permission scopes

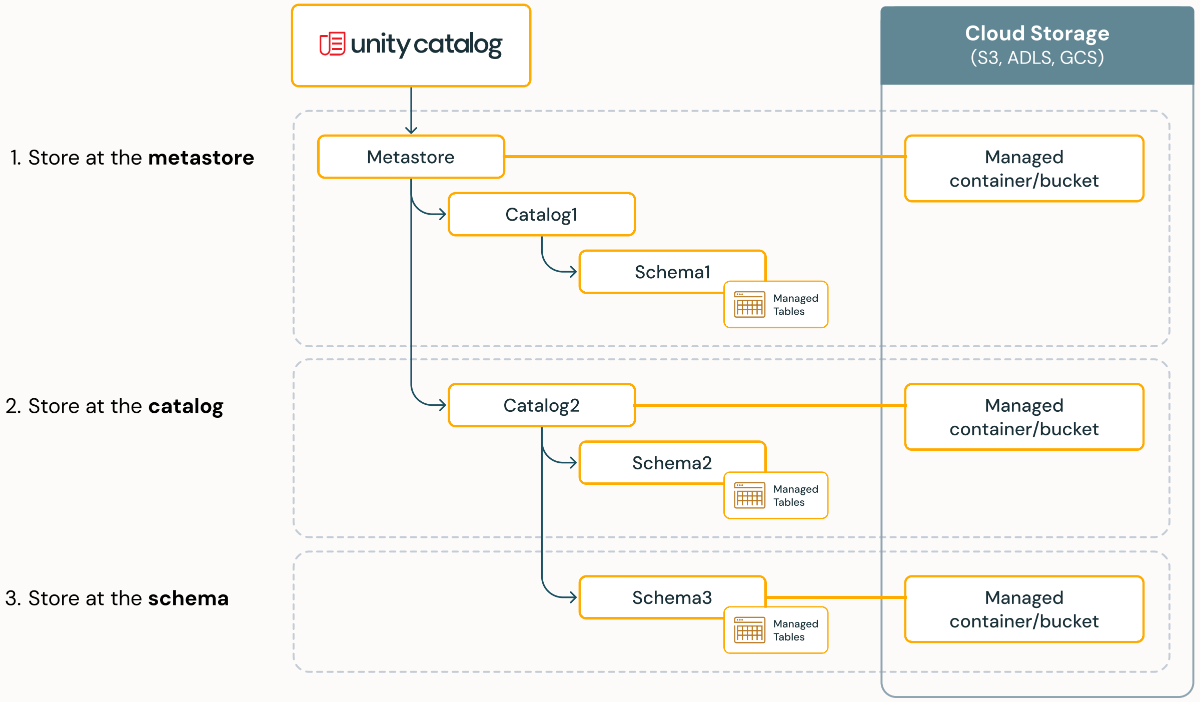
Data should be physically separated in storage

By default, when creating a UC metastore, the Databricks Account Admin provides a single cloud storage location and credential as the default location for managed tables.

Organizations that require physical isolation of data, for regulatory reasons, or for example across SDLC scopes, between business units, or even for cost allocation purposes, should consider managed data source features at the catalog and schema level.

Unity Catalog allows you to choose the defaults for how data is separated in storage. By default, all data is stored at the metastore. With feature support for managed data sources on catalogs and schemas, you can physically isolate data storage and access, helping your organization achieve their governance and data management requirements.

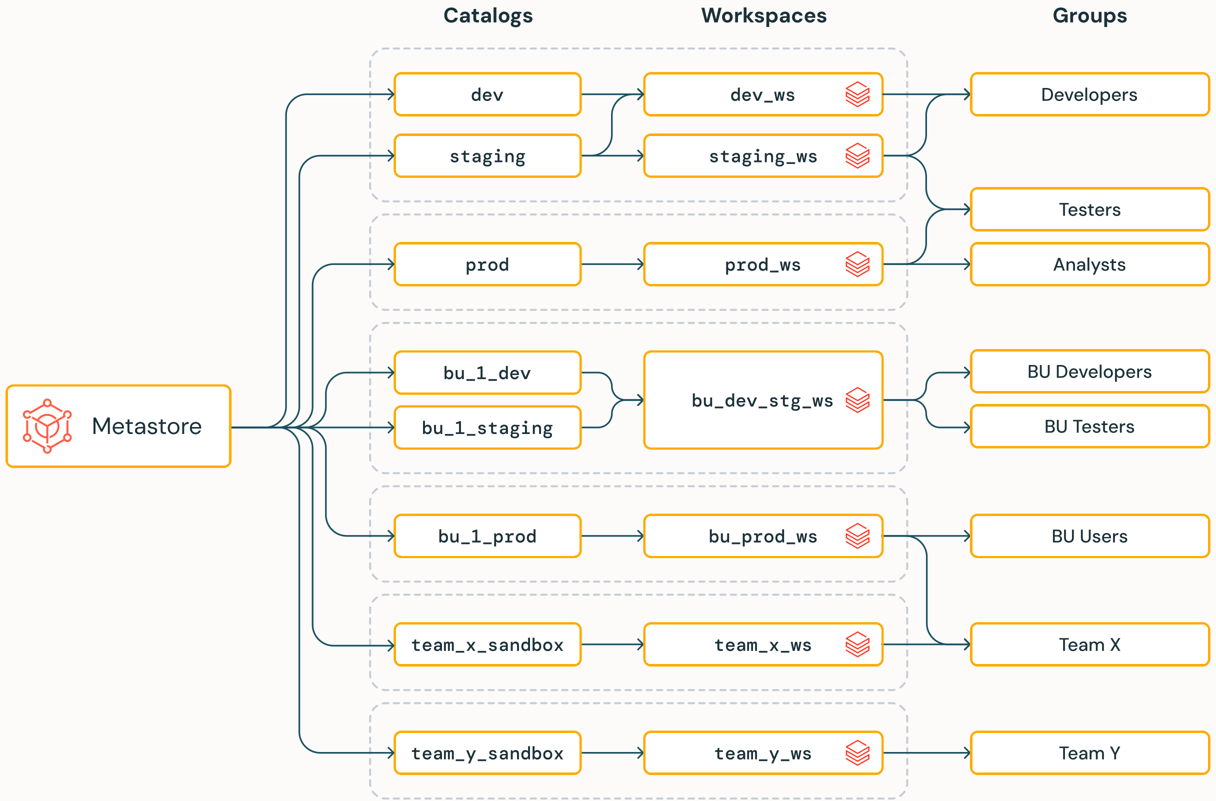
When creating managed tables, the data will then be stored using the schema location (if present) followed by the catalog location (if present), and will only use the metastore location if the prior two locations have not been set.



Data should only be accessed in designated environments, based on the purpose of that data

Oftentimes, organizational and compliance requirements maintain that you need to keep certain data accessible only in certain environments versus others. An example of this could be dev and production, or HIPAA or PII environments that contain PII data for analysis and have special access rules around who can access the data and the environments that allow access to that data. Sometimes requirements dictate that certain data sets or domains cannot be crossed or combined together.

In Databricks, we consider a workspace to be an environment. Unity Catalog has a feature that allows you to 'bind' catalogs to workspace. These environment-aware ACLs give you the ability to ensure that only certain catalogs are available within a workspace, regardless of a user's individual ACLs. This means that the metastore admin, or the catalog owner can define the workspaces that a data catalog can be accessed from. This can be controlled via our UI or via API/terraform for easy integrations. We even recently published a blog on how to control Unity Catalog via terraform to help fit your specific governance model.



Access to data and availability of data can be isolated across workspaces and groups. Users can only access certain catalogs in certain environments.

Conclusion

With Unity Catalog at the center of your lakehouse architecture, you can achieve a flexible and scalable governance implementation without sacrificing your ability to manage and share data effectively. With Unity Catalog, you can overcome the limitations and constraints of your existing Hive metastore, enabling you to more easily isolate and collaborate on data according to your specific business needs. Follow the Unity Catalog guides (AWS, Azure) to get started. Download this free ebook on Data, analytics and AI governance to learn more about best practices to build an effective governance strategy for your data lakehouse.

SECTION 4.6

The Hitchhiker's Guide to Data Privilege Model and Access Control in Unity Catalog

Distill concepts within the Unity Catalog privilege model in a simple, digestible way, to support different access needs and patterns

by Som Natarajan and Vuong Nguyen

As the volume, velocity and variety of data grows, organizations are increasingly relying on staunch data governance practices to ensure their core business outcomes are adequately met. Unity Catalog is a fine-grained governance solution for data and AI powering the Databricks Data Intelligence Platform. It helps simplify the security and governance of your enterprise data assets by providing a centralized mechanism to administer and audit data access.

Taking a journey down memory lane, before Unity Catalog unified the permission model for files, tables and added support for all languages, customers were implementing fine-grained data access control on Databricks using the legacy workspace-level Table ACL (TACL), which were essentially restricted to certain cluster configurations and worked only for Python and SQL. Both Unity Catalog and TACL let you control access to securable objects like catalogs, schemas (databases), tables, views, but there are some nuances in how each access model works.

A good understanding of the object access model is essential for implementing data governance at scale using Unity Catalog. Even more so, if you have already implemented the Table ACL model and are looking to upgrade to Unity Catalog to take advantage of all the newest features, such as multi-language support, centralized access control and data lineage.

The Axioms of Unity Catalog access model

- Unity Catalog privileges are defined at metastore — Unity Catalog permissions always refer to account-level identities, while TACL permissions defined within the hive_metastore catalog always refer to the local identities in the workspace
- Privilege inheritance — Objects in Unity Catalog are hierarchical and privileges are inherited downward. The highest level object that privileges are inherited from is the catalog
- Object ownership is important — Privileges can only be granted by a metastore admin, the owner of an object, or the owner of the catalog or schema that contains the object. Only the owner of an object, or the owner of the catalog or schema that contains it can drop the object
- USE privileges for boundaries — USE CATALOG/SCHEMA is required to interact with objects within a catalog/schema. However, USE privilege does not allow one to browse the object metadata that is housed within the catalog/schema
- Permissions on derived objects are simplified — Unity Catalog only requires the owner of a view to have SELECT privilege, along with USE SCHEMA on the views' parent schema and USE CATALOG on the parent catalog. In contrast with TACL, a view's owner needs to be an owner of all referenced tables and views

Some more complex axioms

- Secure by default — only clusters with Unity-Catalog specific access modes (shared or single-user) can access Unity Catalog data. With TACL, all users have access to all data on non-shared clusters
- Limitation of single-user clusters — Single users clusters do not support dynamic views. Users must have SELECT on all referenced tables and views to read from a view
- No support for ANY FILE or ANONYMOUS FUNCTIONs: Unity Catalog does not support these permissions, as they could be used to circumvent access control restrictions by allowing an unprivileged user to run privileged code

Interesting patterns

There are many governance patterns that can be achieved using the Unity Catalog access model.

Example 1 – Consistent permissions across workspaces

Axiom 1 allows product team to define permissions for their data product within their own workspace, and having those reflected and enforced across all other workspaces, no matter where their consumers are coming from

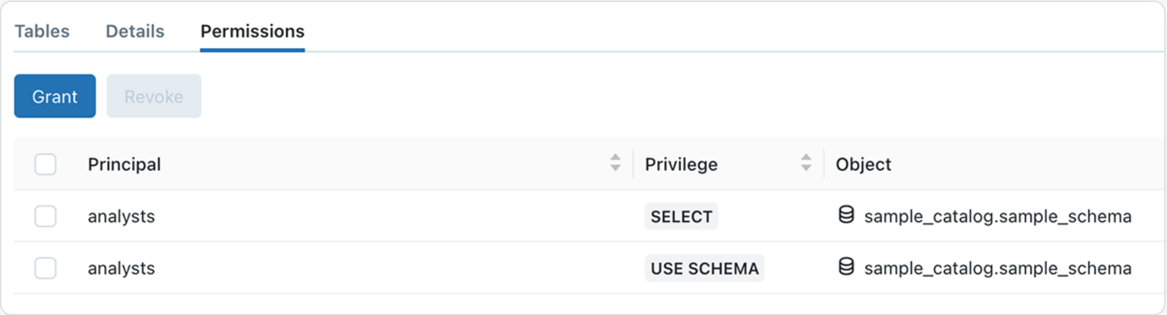
Example 2 – Setting boundary for data sharing

Axiom 2 allows catalog/schema owners to set up default access rules for their data. For example the following commands enable the machine learning team to create tables within a schema and read each other's tables:

```sql
CREATE CATALOG ml;
CREATE SCHEMA ml.sandbox;
GRANT USE_CATALOG ON CATALOG ml TO ml_users;
GRANT USE_SCHEMA ON SCHEMA ml.sandbox TO ml_users;
GRANT CREATE TABLE ON SCHEMA ml.sandbox TO ml_users;
GRANT SELECT ON SCHEMA ml.sandbox TO ml_users;
```

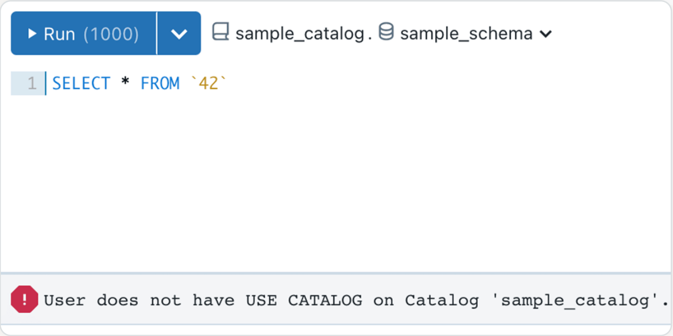
More interestingly, axiom 4 now allows catalog/schema owners to limit how far individual schema and table owners can share data they produce. A table owner granting SELECT to another user does not allow that user read access to the table unless they also have been granted USE CATALOG privileges on its parent catalog as well as USE SCHEMA privileges on its parent schema.

In the below example, sample_catalog is owned by user A, user B created a sample_schema schema, and table 42. Even though USE SCHEMA and SELECT permission is granted to the analysts team, they still cannot query the table, due to permission boundary set by user A



☐	Principal	Privilege	Object
☐	analysts	SELECT	sample_catalog.sample_schema
☐	analysts	USE SCHEMA	sample_catalog.sample_schema

Permission page showing analysts group having SELECT and USE SCHEMA permission on sample_catalog.sample_schema



```
SELECT * FROM `42`
```

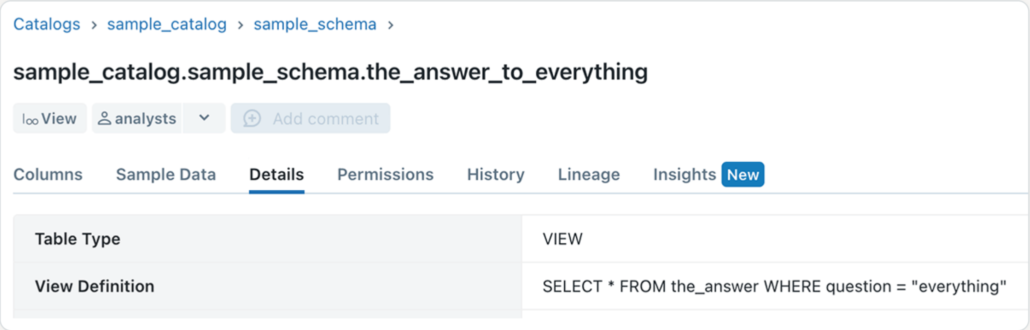
```
User does not have USE CATALOG on Catalog 'sample_catalog'.
```

Query page with an error message

Example 3 – Easier sharing of business logic

Data consumers have a need to share their workings and transformation logic, and a reusable way of doing it is by creating and sharing views to other consumers.

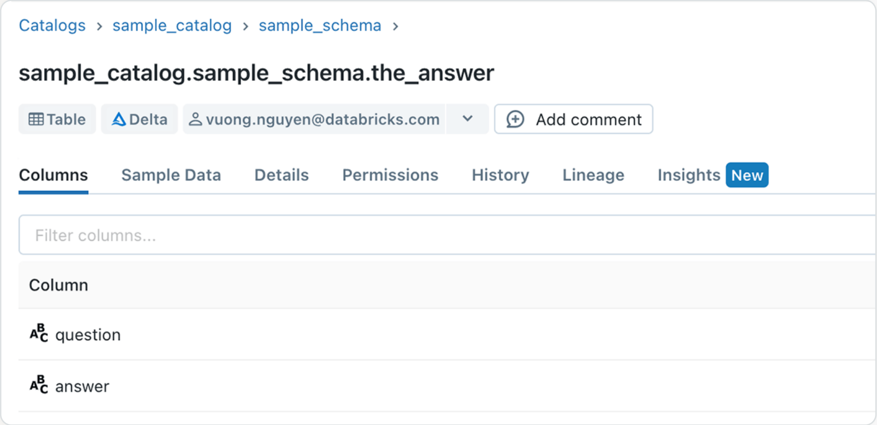
Axiom 5 unlocks the ability for data consumers to do this seamlessly, without requiring manual back and forth with the table owners.



sample_catalog.sample_schema.the_answer_to_everything

Table Type	VIEW
View Definition	SELECT * FROM the_answer WHERE question = "everything"

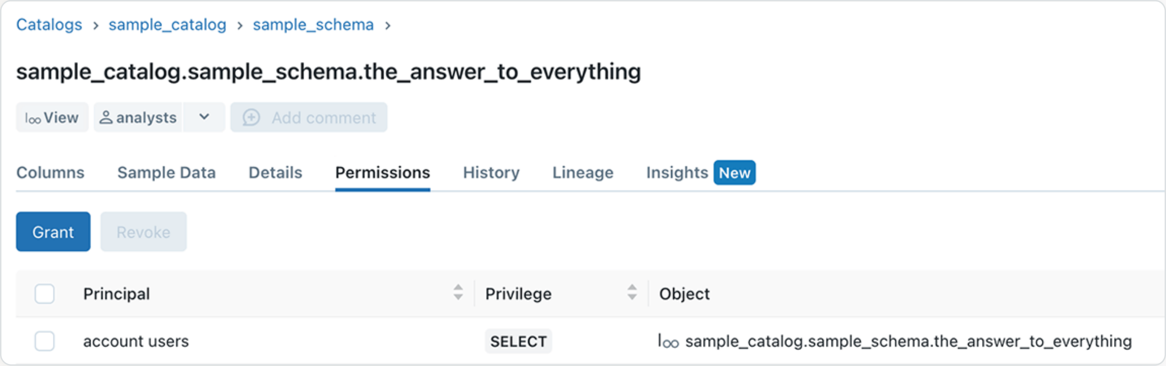
Definition of the view



sample_catalog.sample_schema.the_answer

Column
question
answer

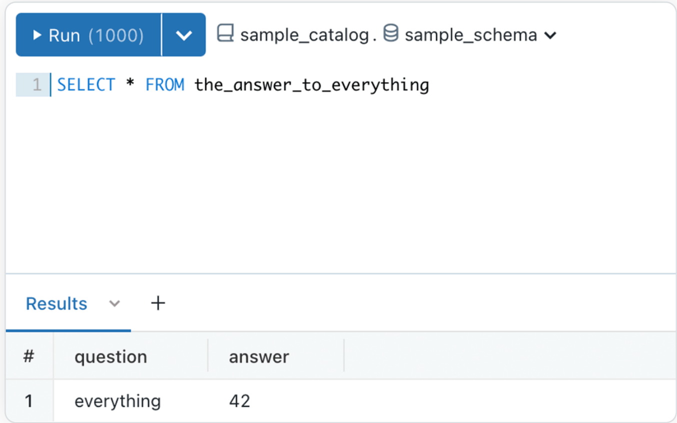
Ownership of the table



sample_catalog.sample_schema.the_answer_to_everything

☐	Principal	Privilege	Object
☐	account users	SELECT	sample_catalog.sample_schema.the_answer_to_everything

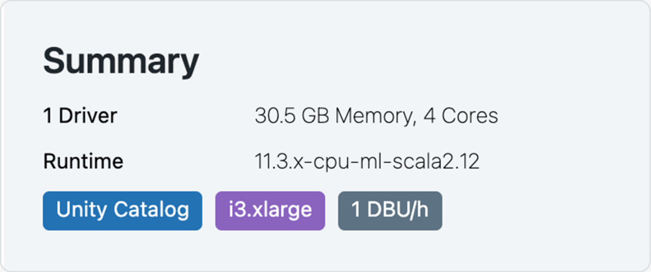
Permission page showing a view owned by analysts group, and account users group having SELECT permission



```
SELECT * FROM the_answer_to_everything
```

#	question	answer
1	everything	42

Query page showing the result of the answer to everything



Summary

1 Driver	30.5 GB Memory, 4 Cores
Runtime	11.3.x-cpu-ml-scala2.12

Cluster summary indicating Unity Catalog support

Example 4 – No more data leakage

Thanks to axiom 6, data owners can be certain that there will be no unauthorized access to their data due to cluster misconfiguration. Any cluster that is not configured with the correct access mode will not be able to access data in Unity Catalog.

Users can check that their clusters can access Unity Catalog data thanks to this handy tooltip on the Create Clusters page

Now that data owners can understand the data privilege model and access control, they can leverage Unity Catalog to simplify access policy management at scale.

There are upcoming features that will further empower data administrators land owners to author even more complex access policy:

- **Row filtering and column masking:** Use standard SQL functions to define row filters and column masks, allowing fine-grained access controls on rows and columns.
- **Attribute Based Access Controls:** Define access policies based on tags (attributes) of your data assets.

SECTION 4.7

Onboarding Your New AI/BI Genie

by Chao Cai and Richard Tomlinson

At Databricks, we want to make data and AI accessible to everyone on the planet. This is why we're building solutions like AI/BI that make it possible for business users, even those who don't speak SQL or write code, to get trusted insights quickly and accurately.

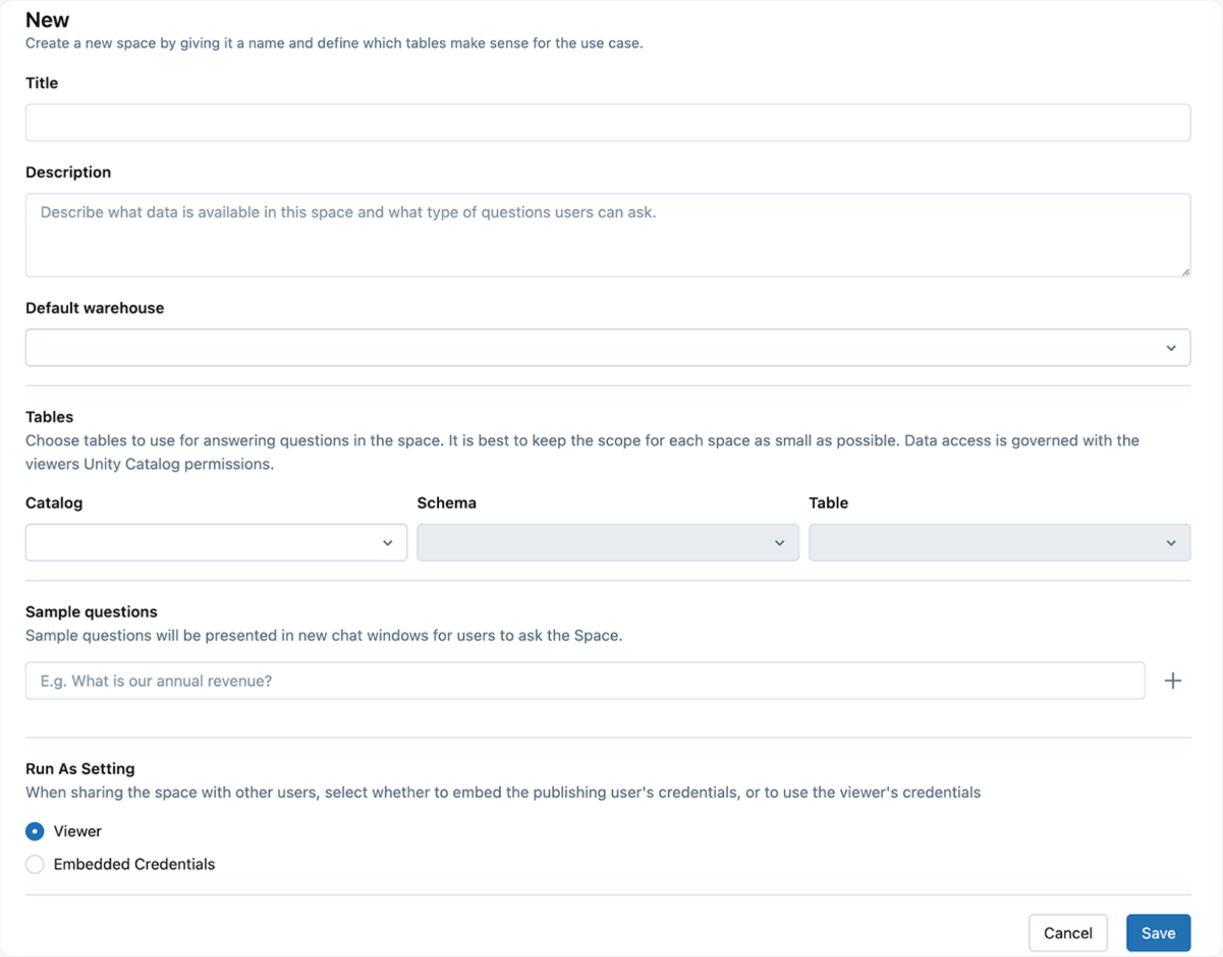
At this point, dashboards are part of most people's daily lives. On the other hand, AI/BI Genie is a newer experience, one that many people might not be familiar with yet. With Genie, users can self-serve and obtain answers to questions not addressed in their dashboards without having to learn how to use BI tools or rely on expert practitioners to create insights for them. With this backdrop, we figured it might be worthwhile to take a look at a day in the life of a Genie so that you can get to know how to use it and what to expect.

New hire orientation

Some folks have made the analogy that the process of creating a Genie is akin to hiring a new analyst for your data team. This new hire is smart and hungry to learn but is a tad robotic and has zero experience with any of your internal terminology, metrics, definitions and assumptions. Your new Genie is well-read in public knowledge but very ignorant of company-internal and team-internal knowledge.

The first step is for someone to give your new hire some much-needed orientation. This task likely falls to you, the experienced analyst who has been fielding all of the team's questions, perhaps drowning in them and looking for some help.

The Genie setup page is your guide to what this new hire needs to know:



New

Create a new space by giving it a name and define which tables make sense for the use case.

Title

Description

Describe what data is available in this space and what type of questions users can ask.

Default warehouse

Tables

Choose tables to use for answering questions in the space. It is best to keep the scope for each space as small as possible. Data access is governed with the viewers Unity Catalog permissions.

Catalog **Schema** **Table**

Sample questions

Sample questions will be presented in new chat windows for users to ask the Space.

E.g. What is our annual revenue?

Run As Setting

When sharing the space with other users, select whether to embed the publishing user's credentials, or to use the viewer's credentials

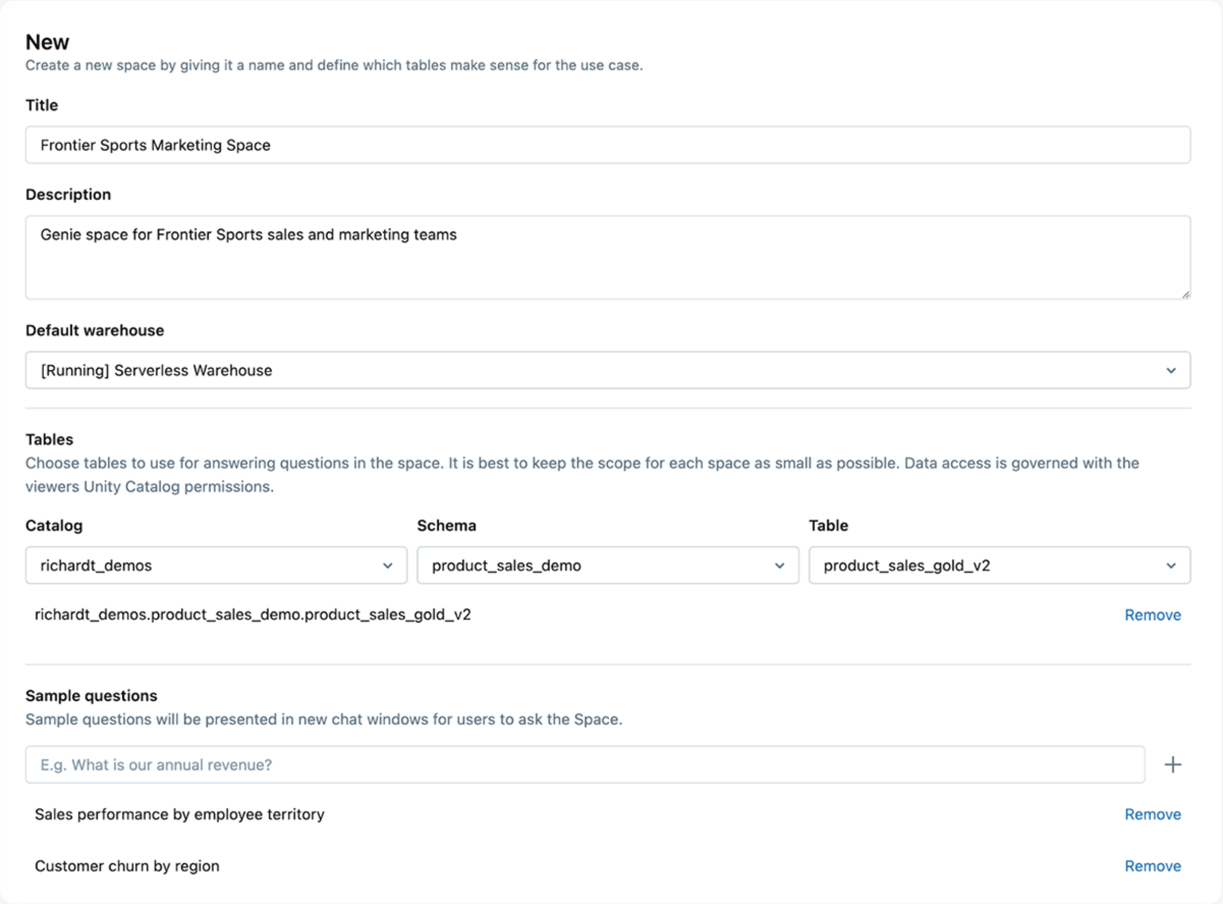
● Viewer

○ Embedded Credentials

For this screen, the goal is to provide the basic information needed to get Genie going:

- **Title and description:** What is the scope and domain of questions you expect Genie to know how to answer? This is visible to anyone browsing and searching across all of the Genie spaces they can access, so it's important to make this clear and succinct.
- **Default warehouse:** What compute will this Genie use to run SQL queries for you?
- **Tables:** What tables are in scope for this Genie to use to answer questions? Genie will read up on all of the comments in each table and in each of its columns and use these to inform its answers.
- **Sample questions:** Optionally, you can specify some example questions you want to present to people who are talking to this Genie if they're not sure what to ask.

Once you've chosen sensible values for these fields, click Save and your new Genie is ready for action.



New

Create a new space by giving it a name and define which tables make sense for the use case.

Title

Frontier Sports Marketing Space

Description

Genie space for Frontier Sports sales and marketing teams

Default warehouse

[Running] Serverless Warehouse

Tables

Choose tables to use for answering questions in the space. It is best to keep the scope for each space as small as possible. Data access is governed with the viewers Unity Catalog permissions.

Catalog	Schema	Table
richardt_demos	product_sales_demo	product_sales_gold_v2

richardt_demos.product_sales_demo.product_sales_gold_v2

Sample questions

Sample questions will be presented in new chat windows for users to ask the Space.

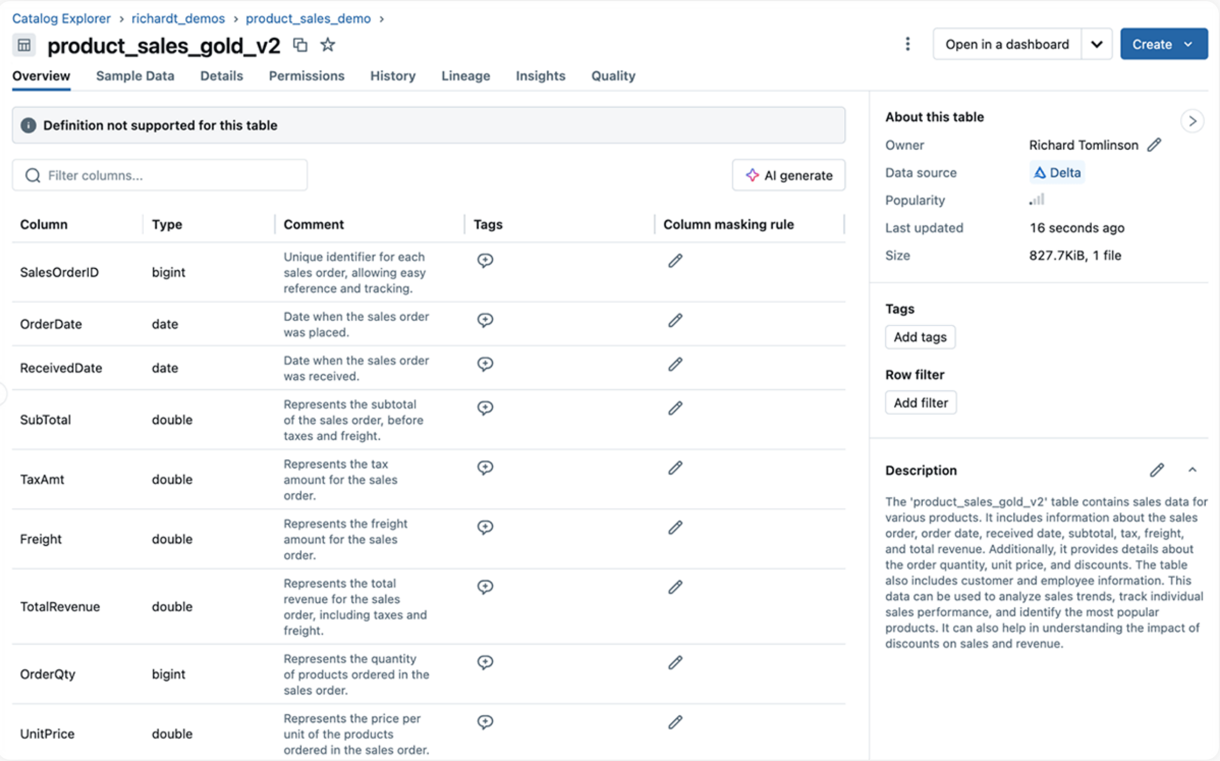
E.g. What is our annual revenue?

Sales performance by employee territory

Customer churn by region

Well, not quite. If you were training a new hire, you'd probably want to check their knowledge first before letting them go off to answer your team's hardest questions.

To ensure your new hire has the best possible start in their new career you'll want to bootstrap their knowledge with some basic information. If you've invested in some basic data governance best practices you'll ensure they get a flying start. This is where Unity Catalog can help. Make sure you've created some useful comments in your tables and columns, and take the time to create primary/foreign key relationships between the tables you want to give to Genie. This basic information will ensure Genie has a lot of basic knowledge on its very first day.



Definition not supported for this table

Column	Type	Comment	Tags	Column masking rule
SalesOrderID	bigint	Unique identifier for each sales order, allowing easy reference and tracking.		
OrderDate	date	Date when the sales order was placed.		
ReceivedDate	date	Date when the sales order was received.		
SubTotal	double	Represents the subtotal of the sales order, before taxes and freight.		
TaxAmt	double	Represents the tax amount for the sales order.		
Freight	double	Represents the freight amount for the sales order.		
TotalRevenue	double	Represents the total revenue for the sales order, including taxes and freight.		
OrderQty	bigint	Represents the quantity of products ordered in the sales order.		
UnitPrice	double	Represents the price per unit of the products ordered in the sales order.		

About this table

Owner	Richard Tomlinson
Data source	Delta
Popularity	
Last updated	16 seconds ago
Size	827.7KiB, 1 file

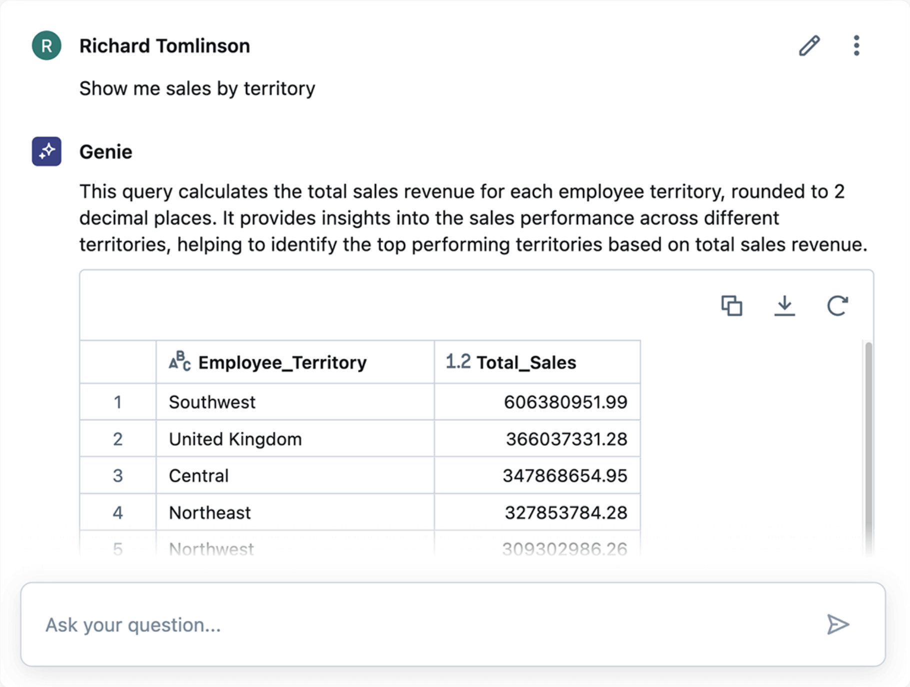
Tags: Add tags

Row filter: Add filter

Description

The 'product_sales_gold_v2' table contains sales data for various products. It includes information about the sales order, order date, received date, subtotal, tax, freight, and total revenue. Additionally, it provides details about the order quantity, unit price, and discounts. The table also includes customer and employee information. This data can be used to analyze sales trends, track individual sales performance, and identify the most popular products. It can also help in understanding the impact of discounts on sales and revenue.

The next step is to ask Genie a few things that you'd expect your team to ask in daily practice.



Richard Tomlinson

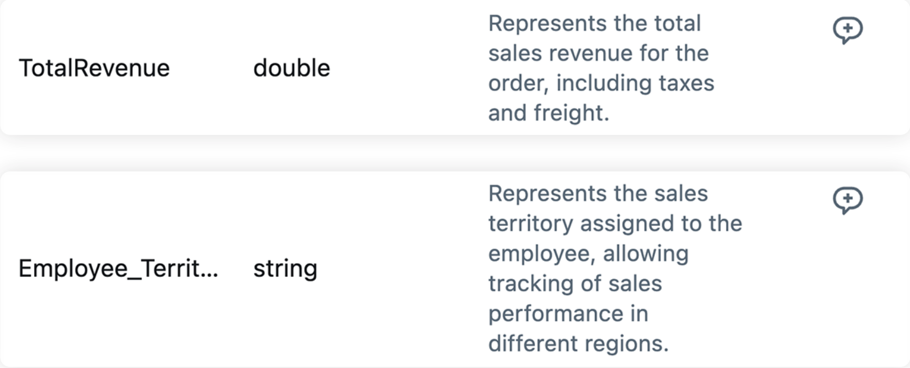
Show me sales by territory

Genie

This query calculates the total sales revenue for each employee territory, rounded to 2 decimal places. It provides insights into the sales performance across different territories, helping to identify the top performing territories based on total sales revenue.

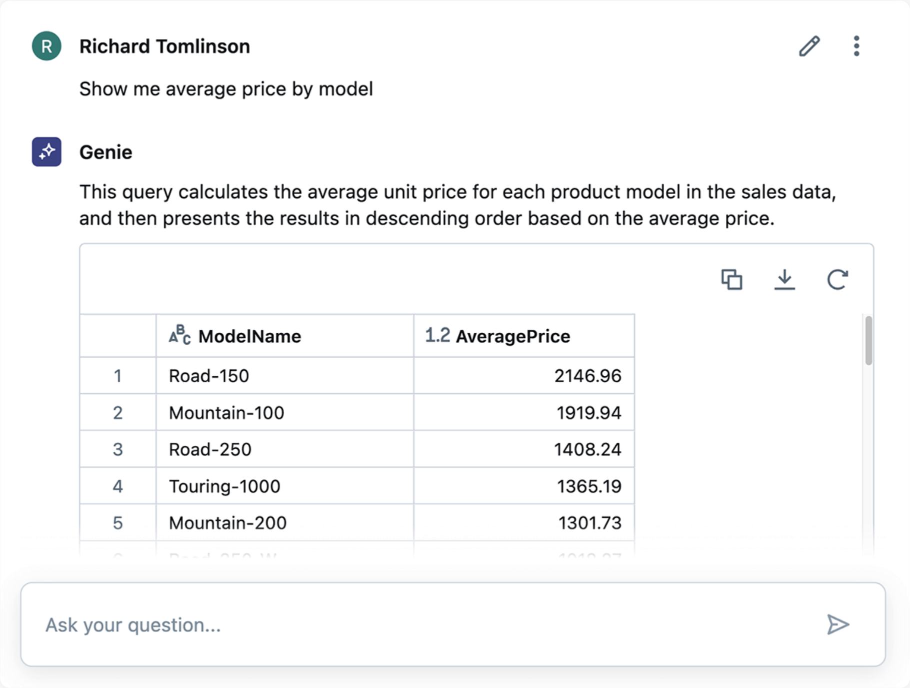
	Employee_Territory	Total_Sales
1	Southwest	606380951.99
2	United Kingdom	366037331.28
3	Central	347868654.95
4	Northeast	327853784.28
5	Northwest	309302986.26

In this first example, Genie easily gets the question correct.



TotalRevenue	double	Represents the total sales revenue for the order, including taxes and freight.
Employee_Territ...	string	Represents the sales territory assigned to the employee, allowing tracking of sales performance in different regions.

In Catalog Explorer (above), if we look at the table metadata, we see the column comments provide enough information for Genie to get the answer right. Yay!



Richard Tomlinson

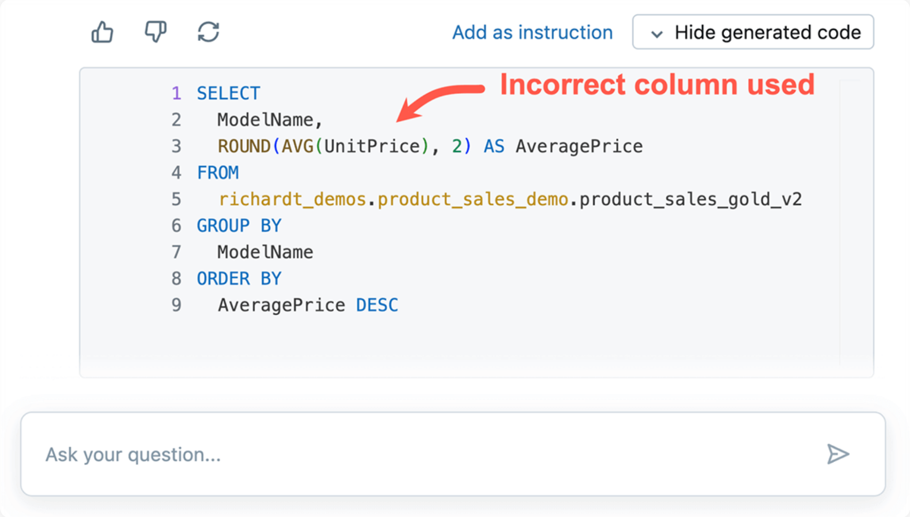
Show me average price by model

Genie

This query calculates the average unit price for each product model in the sales data, and then presents the results in descending order based on the average price.

	ModelName	AveragePrice
1	Road-150	2146.96
2	Mountain-100	1919.94
3	Road-250	1408.24
4	Touring-1000	1365.19
5	Mountain-200	1301.73

In this second example, Genie guessed wrong because the most obvious definition it infers from the column name for average price turns out to be wrong.

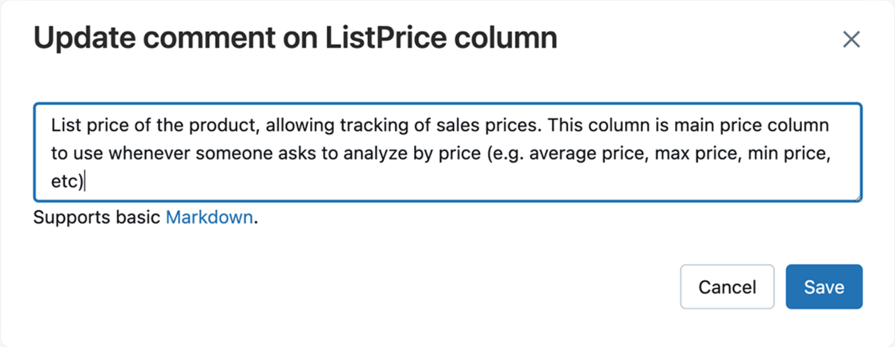


```sql
SELECT
  ModelName,
  ROUND(AVG(UnitPrice), 2) AS AveragePrice
FROM
  richardt_demos.product_sales_demo.product_sales_gold_v2
GROUP BY
  ModelName
ORDER BY
  AveragePrice DESC
```

Incorrect column used

When we view the underlying SQL we can see that Genie picked the wrong column for price. Genie should've picked the ListPrice column, not UnitPrice.

We can fix this by giving the desired column a useful clarifying comment in Catalog Explorer:

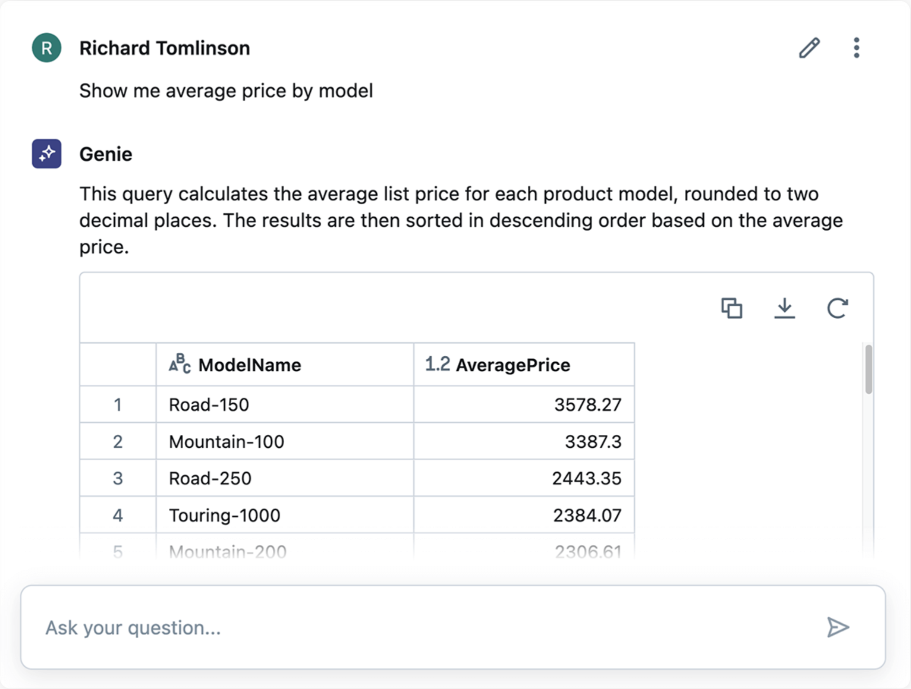


Update comment on ListPrice column

List price of the product, allowing tracking of sales prices. This column is main price column to use whenever someone asks to analyze by price (e.g. average price, max price, min price, etc)

Supports basic Markdown.

With this update, Genie will now produce the right answer.



Richard Tomlinson

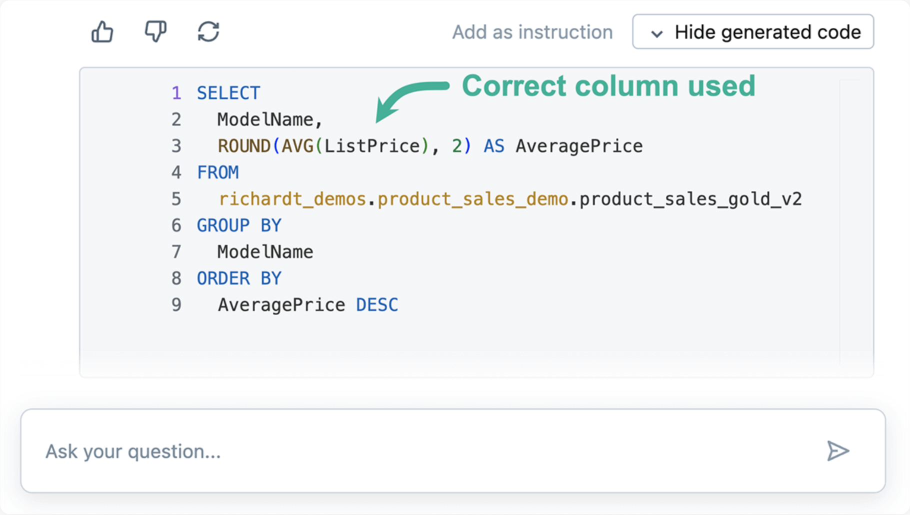
Show me average price by model

Genie

This query calculates the average list price for each product model, rounded to two decimal places. The results are then sorted in descending order based on the average price.

	ModelName	AveragePrice
1	Road-150	3578.27
2	Mountain-100	3387.3
3	Road-250	2443.35
4	Touring-1000	2384.07
5	Mountain-200	2306.61

The user asks Genie the exact same question as before.

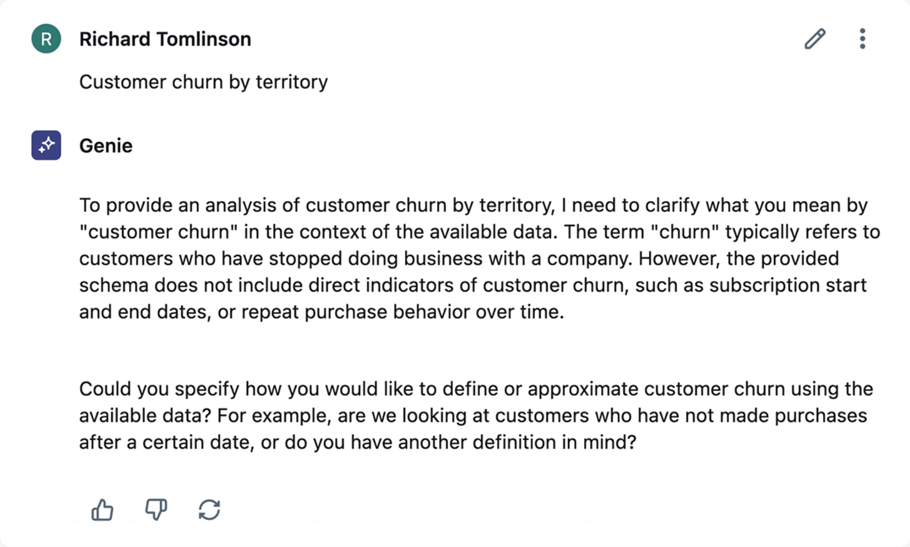


Correct column used

```sql
SELECT
  ModelName,
  ROUND(AVG(ListPrice), 2) AS AveragePrice
FROM
  richardt_demos.product_sales_demo.product_sales_gold_v2
GROUP BY
  ModelName
ORDER BY
  AveragePrice DESC
```

This time Genie uses the correct column, ListPrice, in its query.

In the next example, Genie won't even try to guess because it knows that it lacks an important context: how a particular team-specific metric is defined. Without this context, Genie has no way of deducing the right answer.



Richard Tomlinson

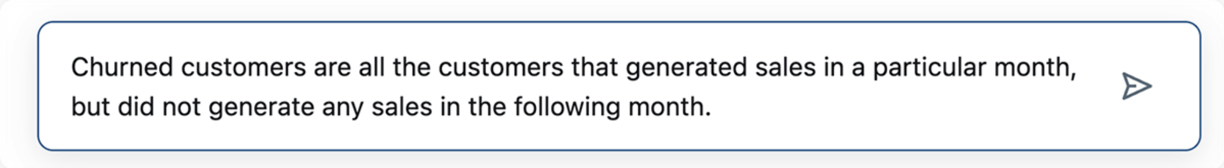
Customer churn by territory

Genie

To provide an analysis of customer churn by territory, I need to clarify what you mean by "customer churn" in the context of the available data. The term "churn" typically refers to customers who have stopped doing business with a company. However, the provided schema does not include direct indicators of customer churn, such as subscription start and end dates, or repeat purchase behavior over time.

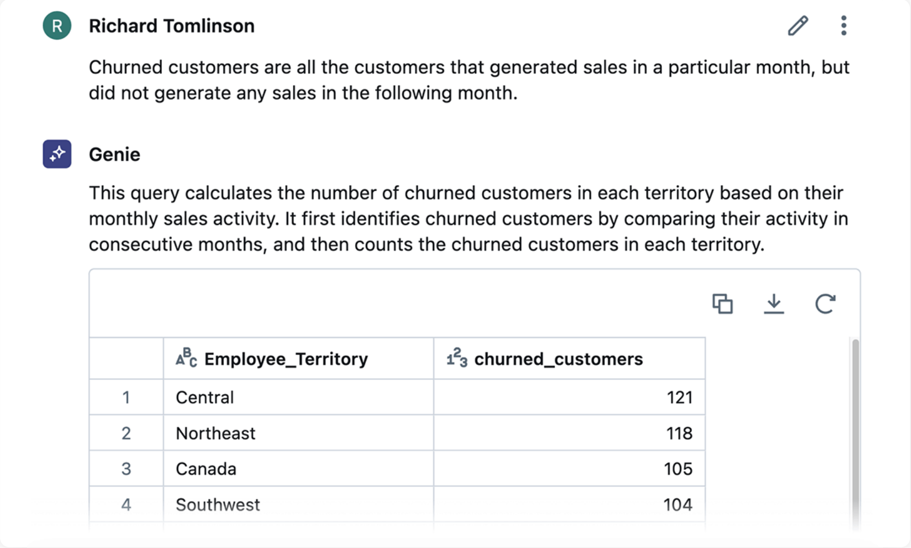
Could you specify how you would like to define or approximate customer churn using the available data? For example, are we looking at customers who have not made purchases after a certain date, or do you have another definition in mind?

So let's teach Genie this metric. We can provide the metric directly to Genie and see what happens.



Churned customers are all the customers that generated sales in a particular month, but did not generate any sales in the following month.

We simply tell Genie the definition of customer churn directly in the UI and run the prompt.



Richard Tomlinson

Churned customers are all the customers that generated sales in a particular month, but did not generate any sales in the following month.

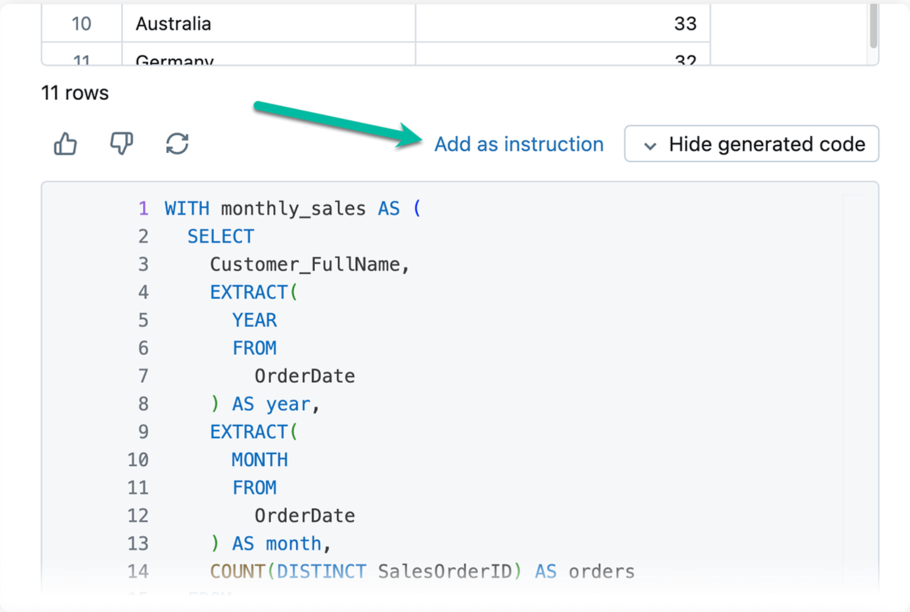
Genie

This query calculates the number of churned customers in each territory based on their monthly sales activity. It first identifies churned customers by comparing their activity in consecutive months, and then counts the churned customers in each territory.

	Employee_Territory	churned_customers
1	Central	121
2	Northeast	118
3	Canada	105
4	Southwest	104

We can see that Genie has understood the prompt and is able to query the data correctly.

Now that we've confirmed that Genie learned how to calculate this new metric correctly, we can tell it to remember this going forward by clicking "Add as instruction."

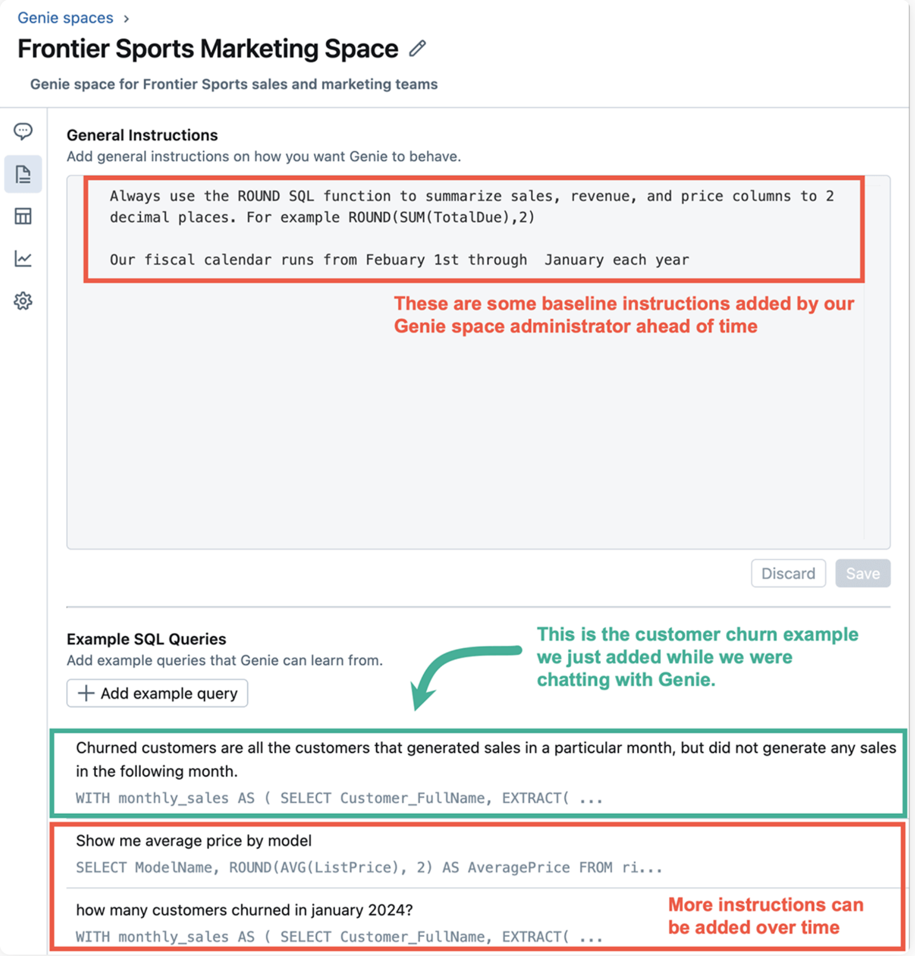


10	Australia	33
11	Germany	32

11 rows

```sql
WITH monthly_sales AS (
  SELECT
    Customer_FullName,
    EXTRACT(
      YEAR
      FROM
        OrderDate
    ) AS year,
    EXTRACT(
      MONTH
      FROM
        OrderDate
    ) AS month,
    COUNT(DISTINCT SalesOrderID) AS orders
```

This action saves the instruction into the Instructions section, which contains all the things you want Genie to consider as it does its job. This instruction can be found in Example SQL Queries, but you can also manually update any of the instructions.



Frontier Sports Marketing Space

Genie space for Frontier Sports sales and marketing teams

General Instructions

Add general instructions on how you want Genie to behave.

```
Always use the ROUND SQL function to summarize sales, revenue, and price columns to 2
decimal places. For example ROUND(SUM(TotalDue),2)

Our fiscal calendar runs from Febuary 1st through  January each year
```

These are some baseline instructions added by our Genie space administrator ahead of time

Example SQL Queries

Add example queries that Genie can learn from.

This is the customer churn example we just added while we were chatting with Genie.

Churned customers are all the customers that generated sales in a particular month, but did not generate any sales in the following month.

```
WITH monthly_sales AS ( SELECT Customer_FullName, EXTRACT( ...
```

Show me average price by model

```
SELECT ModelName, ROUND(AVG(ListPrice), 2) AS AveragePrice FROM ri...
```

how many customers churned in january 2024?

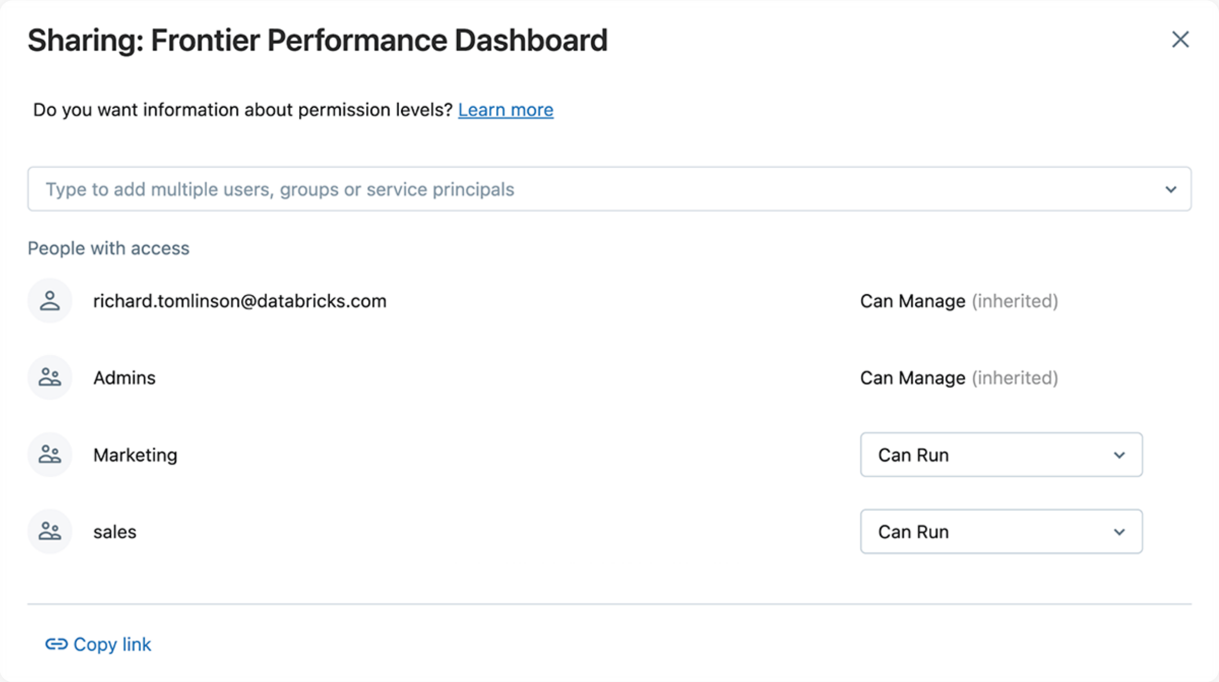
```
WITH monthly_sales AS ( SELECT Customer_FullName, EXTRACT( ...
```

More instructions can be added over time

Before releasing your Genie space into the wild, we recommend you add as many of these types of instructions as possible for the known/anticipated questions and business semantics your users will need. You can do this by saving instructions as you chat or by adding them directly to the Instructions section in your space. The more you train your Genie, the smarter it gets.

Once you've tested Genie enough to be confident that it's providing useful and correct answers to your team's anticipated questions, it's time to let your team talk to the new hire.

Click Share to tell Genie who can talk to it. Depending on who this Genie is supposed to answer questions for, you can choose individual people, a group representing your whole team or the entire company.



Sharing: Frontier Performance Dashboard

Do you want information about permission levels? Learn more

Type to add multiple users, groups or service principals

People with access

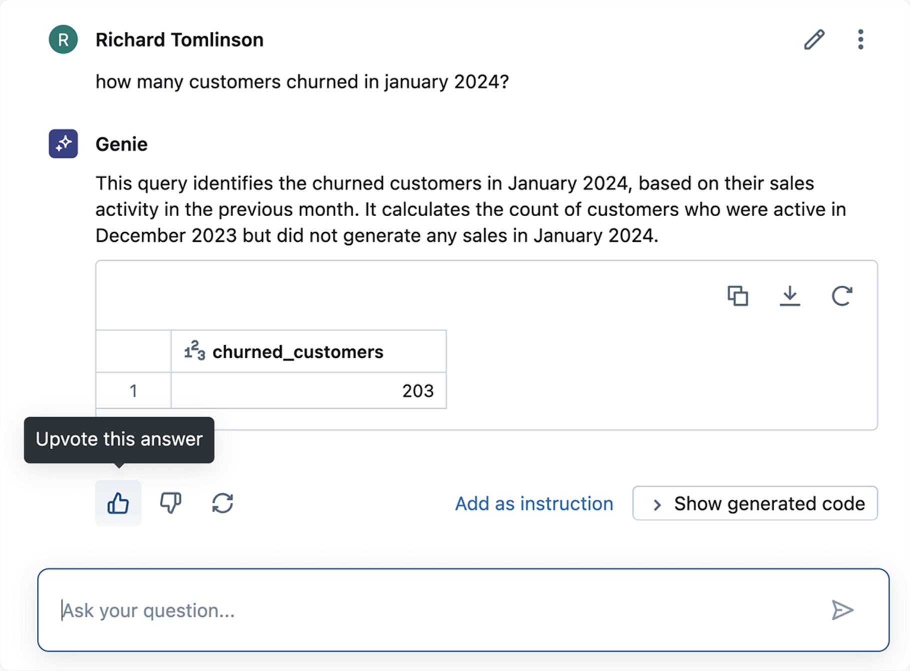
richard.tomlinson@databricks.com	Can Manage (inherited)
Admins	Can Manage (inherited)
Marketing	Can Run
sales	Can Run

You should also tell your team about what to expect from Genie, since they've probably never worked with such a robotic teammate before. Remember that they probably can't speak SQL, so they might not be able to tell whether Genie is actually giving them the right answer or just making a guess that's actually wrong. You can tell your team about what questions you expect Genie to answer correctly. Be sure to let them know that if they have a mission-critical question but aren't sure whether Genie is answering correctly, they should check with you before using it.

Getting to work

So now that you've unleashed your new Genie teammate, let's see how a member of your team interacts with it.

Similar to when you tested Genie, anyone on the team can start asking questions and getting answers back.



Richard Tomlinson

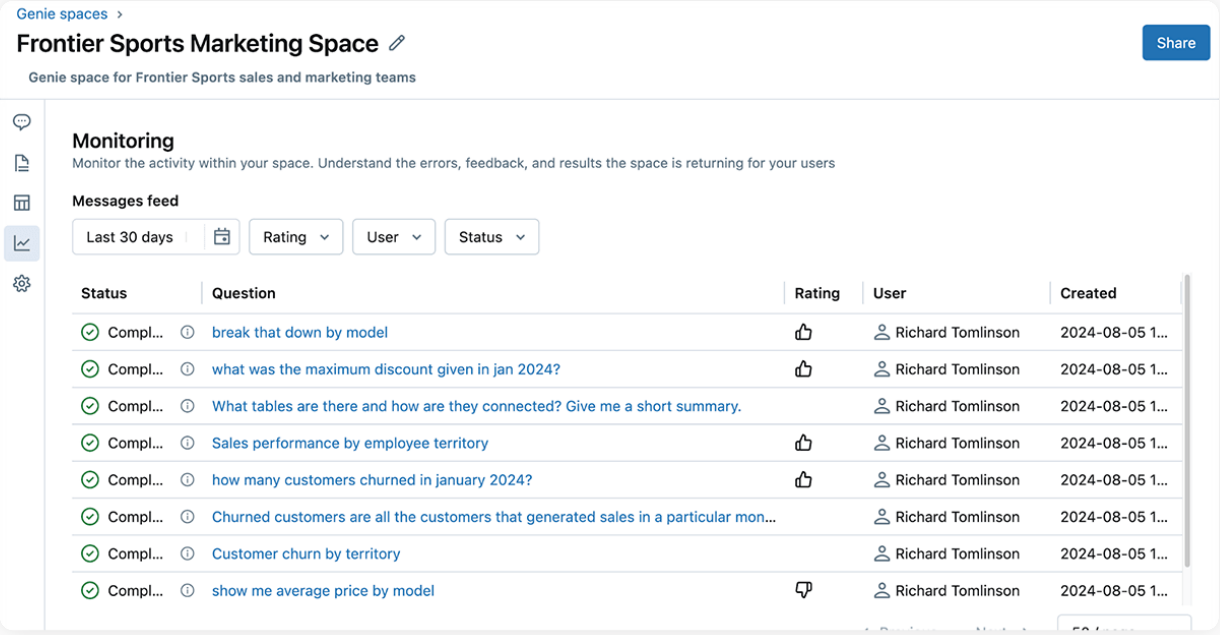
how many customers churned in january 2024?

Genie

This query identifies the churned customers in January 2024, based on their sales activity in the previous month. It calculates the count of customers who were active in December 2023 but did not generate any sales in January 2024.

	churned_customers
1	203

Your users can give feedback by clicking the thumbs-up/thumbs-down buttons after seeing Genie's response. These votes show up in the Monitoring section, where you can see the questions your users are asking and identify new things you need to teach Genie to improve it.



Frontier Sports Marketing Space

Genie space for Frontier Sports sales and marketing teams

Monitoring

Monitor the activity within your space. Understand the errors, feedback, and results the space is returning for your users

Messages feed

Status	Question	Rating	User	Created
Compl...	break that down by model	👍	Richard Tomlinson	2024-08-05 1...
Compl...	what was the maximum discount given in jan 2024?	👍	Richard Tomlinson	2024-08-05 1...
Compl...	What tables are there and how are they connected? Give me a short summary.		Richard Tomlinson	2024-08-05 1...
Compl...	Sales performance by employee territory	👍	Richard Tomlinson	2024-08-05 1...
Compl...	how many customers churned in january 2024?	👍	Richard Tomlinson	2024-08-05 1...
Compl...	Churned customers are all the customers that generated sales in a particular mon...		Richard Tomlinson	2024-08-05 1...
Compl...	Customer churn by territory		Richard Tomlinson	2024-08-05 1...
Compl...	show me average price by model	👎	Richard Tomlinson	2024-08-05 1...

Here you can view all of the questions that have been asked and any thumbs-up/thumbs-down feedback from your team.

Look for questions that you hadn't anticipated earlier and aren't sure whether Genie might answer correctly. For each question, you can look at how Genie answered to check whether it's correct. If it isn't correct, this is an opportunity to hop into a chat session of your own to teach Genie the right answer.

You can also filter for "thumbs down" interactions to check for two different scenarios. First, Genie might've answered incorrectly, so you'll need to teach Genie something it doesn't know. Second, you might think that Genie answered correctly, but your teammate talking to Genie might have different assumptions than you do, leading them to conclude that Genie isn't answering correctly or usefully. It'll be helpful to clear up those differences in assumptions.

Learn more about Genie

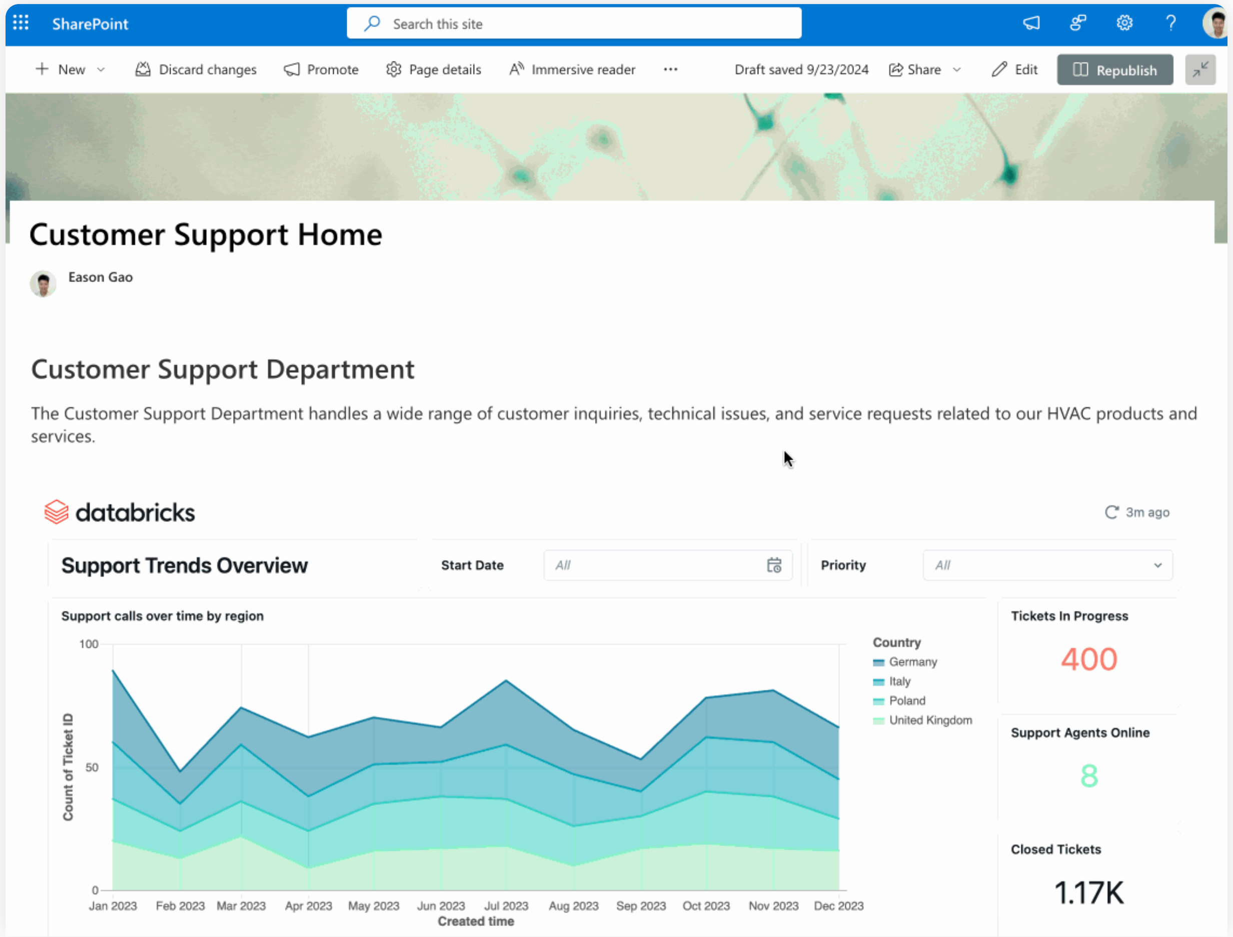
As you can see, a main goal of Genie is to make it possible for anyone on any team to get the insights they're looking for without overtaxing or waiting for the more technical members of their team to provide answers. We hope you've found these examples helpful and hope that you'll give Genie a try. For a deeper look at Genie, check out the Databricks documentation around Genie, especially the best practices we've summarized. Please let us know if you have feedback. We look forward to learning about your experiences with Genie.

SECTION 4.8

How to Embed AI/BI Dashboards Into Your Websites and Applications

by Eason Gao

We're thrilled to announce that embedding for AI/BI Dashboards is now available. Embedding enables you to seamlessly integrate Databricks AI/BI Dashboards into third-party tools such as your internal websites and applications. This integration simplifies access to data insights for your team, allowing them to interact with the information without disrupting their workflow.



Customer Support Home

Eason Gao

Customer Support Department

The Customer Support Department handles a wide range of customer inquiries, technical issues, and service requests related to our HVAC products and services.

What is dashboard embedding?

Dashboard embedding enables Databricks AI/BI Dashboards to be displayed within other platforms using iframes. This means that users can view dashboards without having to leave the applications they're already using, streamlining workflows and enhancing productivity.

Key features

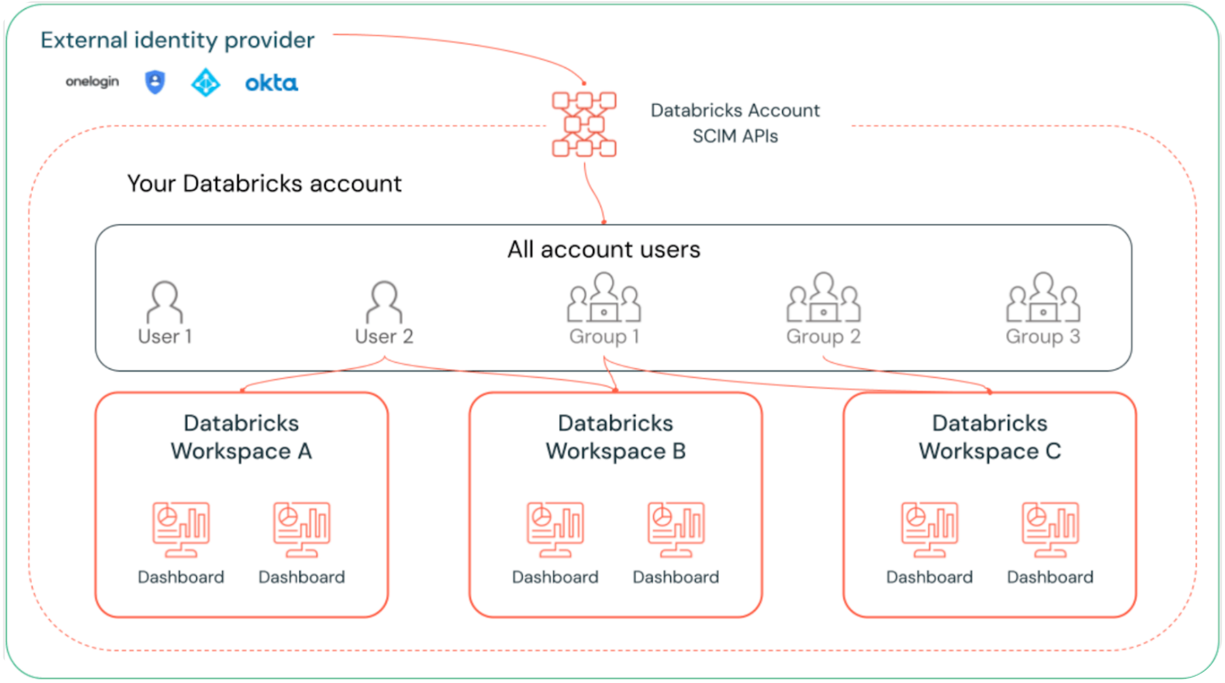
- **Integration Flexibility:** You can embed dashboards anywhere iframes are welcome. This list encompasses common productivity tools like Confluence, Salesforce and SharePoint. This flexibility ensures that data insights are accessible in the tools your teams use regularly.
- **Secure Access:** To maintain security, only authorized users can access the dashboard in the embedded surface. To authorize a user, make sure they're part of your Databricks account and have been shared the dashboard.
- **Optimized Performance:** Embedded dashboards benefit from the same client-side filtering and caching techniques as in-product dashboards, ensuring snappy performance

How to use dashboard embedding

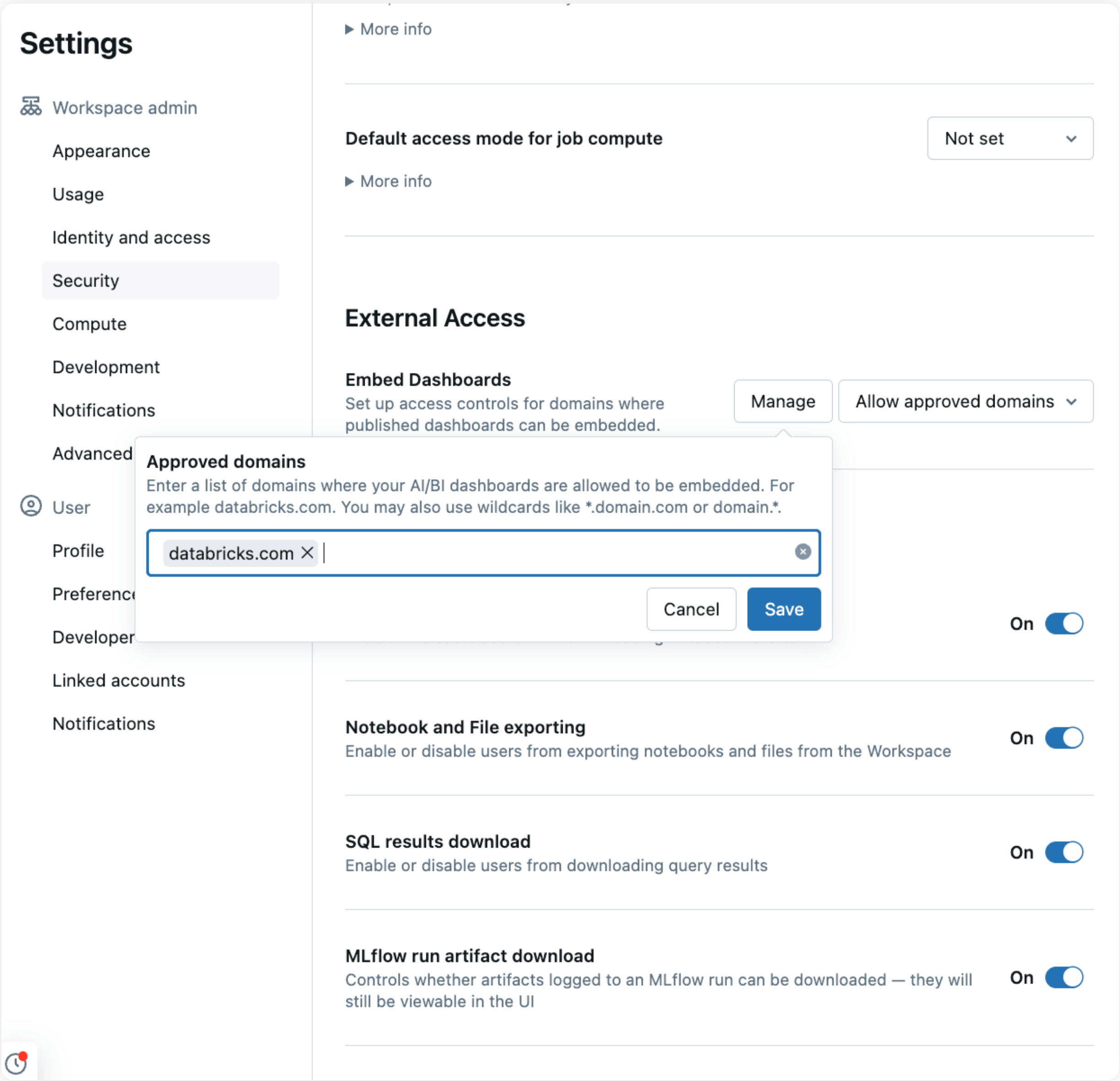
To start embedding dashboards, follow these steps:

1. **User Provisioning:** Ensure that all users who need to view embedded dashboards are first registered with Databricks. Do this by adding users to the Databricks account. They don't need to be granted access to any workspaces. Users without workspace access can view shared AI/BI Dashboards but they can't edit them or access other workspace content like Workflows, ML experiments, etc.

   Account-level SCIM provisioning can simplify registering authorized dashboard viewers with Databricks. See this blog post to learn more about sharing AI/BI Dashboards with everyone in your organization.



2. **Define Allowed Domains:** In workspace settings, ensure that a list of allowed domains where dashboards may be embedded is specified. This allow list is empty by default so this feature is effectively disabled until a workspace admin enables it by specifying allowed domains.



Default access mode for job compute: Not set

External Access

Embed Dashboards
Set up access controls for domains where published dashboards can be embedded.
Allow approved domains

Approved domains
Enter a list of domains where your AI/BI dashboards are allowed to be embedded. For example databricks.com. You may also use wildcards like *.domain.com or domain.*.
databricks.com ✕

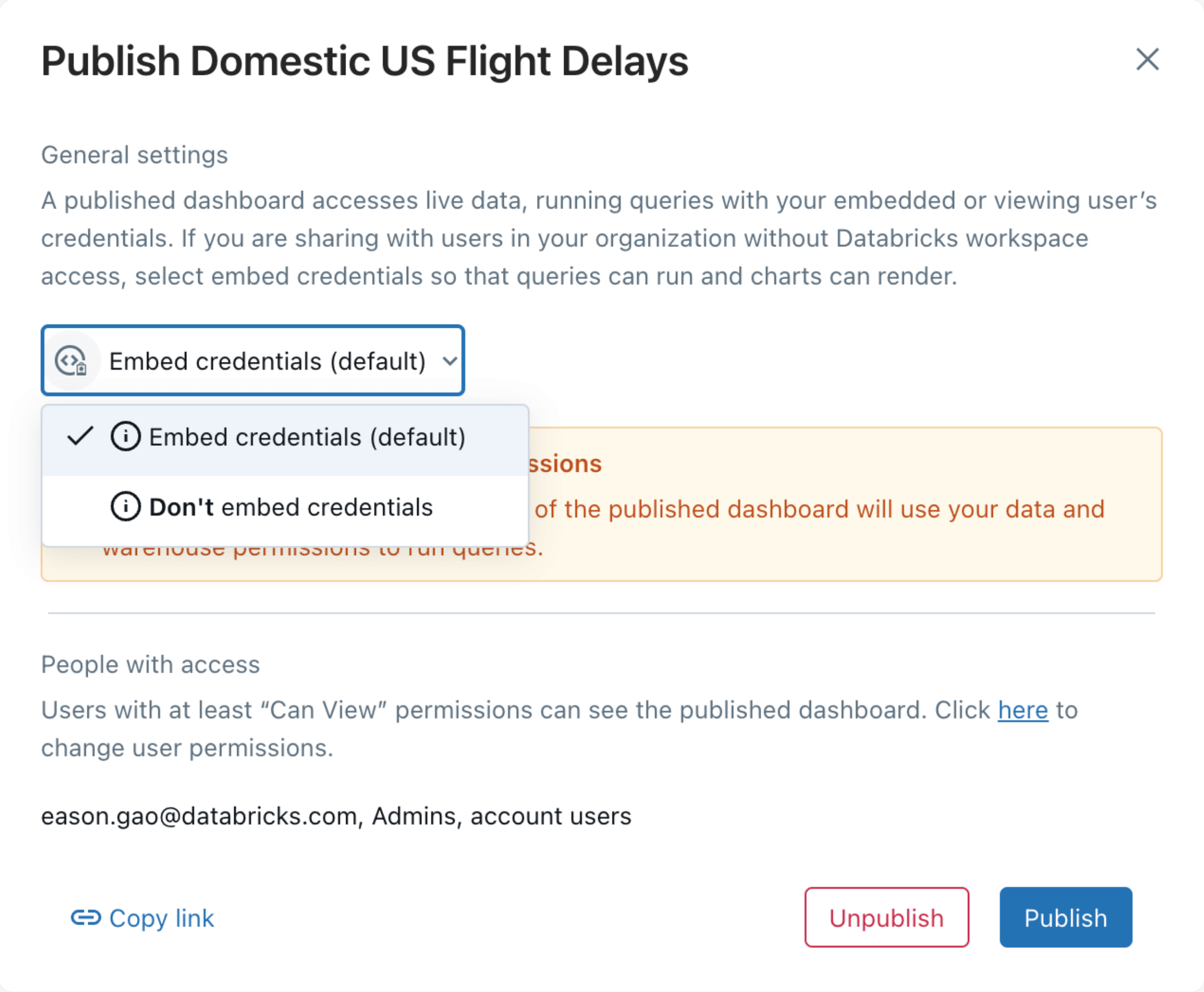
On

Notebook and File exporting
Enable or disable users from exporting notebooks and files from the Workspace
On

SQL results download
Enable or disable users from downloading query results
On

MLflow run artifact download
Controls whether artifacts logged to an MLflow run can be downloaded — they will still be viewable in the UI
On

3. **Dashboard Creation:** Design and publish your dashboards within Databricks. Share your dashboard with all users and/or groups who should have access. If you're sharing to account-level users, publish with embedded credentials to ensure that they can access the dashboard.



Publish Domestic US Flight Delays

General settings

A published dashboard accesses live data, running queries with your embedded or viewing user's credentials. If you are sharing with users in your organization without Databricks workspace access, select embed credentials so that queries can run and charts can render.

Embed credentials (default)

- ✓ Embed credentials (default)
- Don't embed credentials

ssions

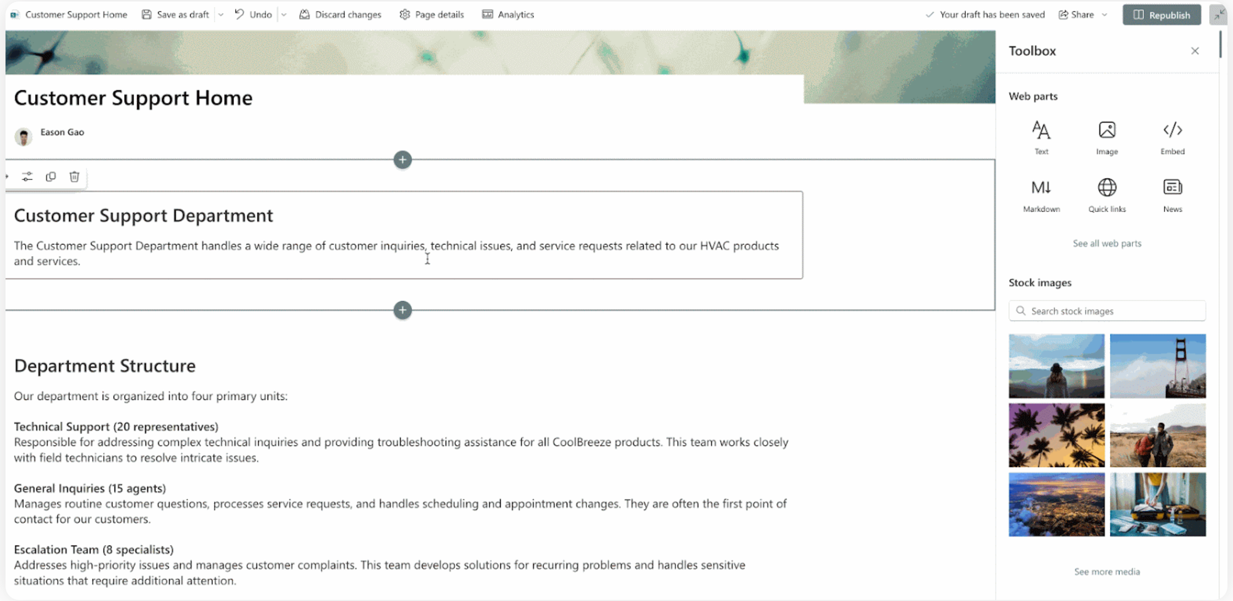
of the published dashboard will use your data and warehouse permissions to run queries.

People with access

Users with at least "Can View" permissions can see the published dashboard. Click here to change user permissions.

eason.gao@databricks.com, Admins, account users

4. **Embedding:** Copy the iframe embed code from the dashboard share dialog and insert it into your desired platform.



Customer Support Home

Eason Gao

Customer Support Department

The Customer Support Department handles a wide range of customer inquiries, technical issues, and service requests related to our HVAC products and services.

Department Structure

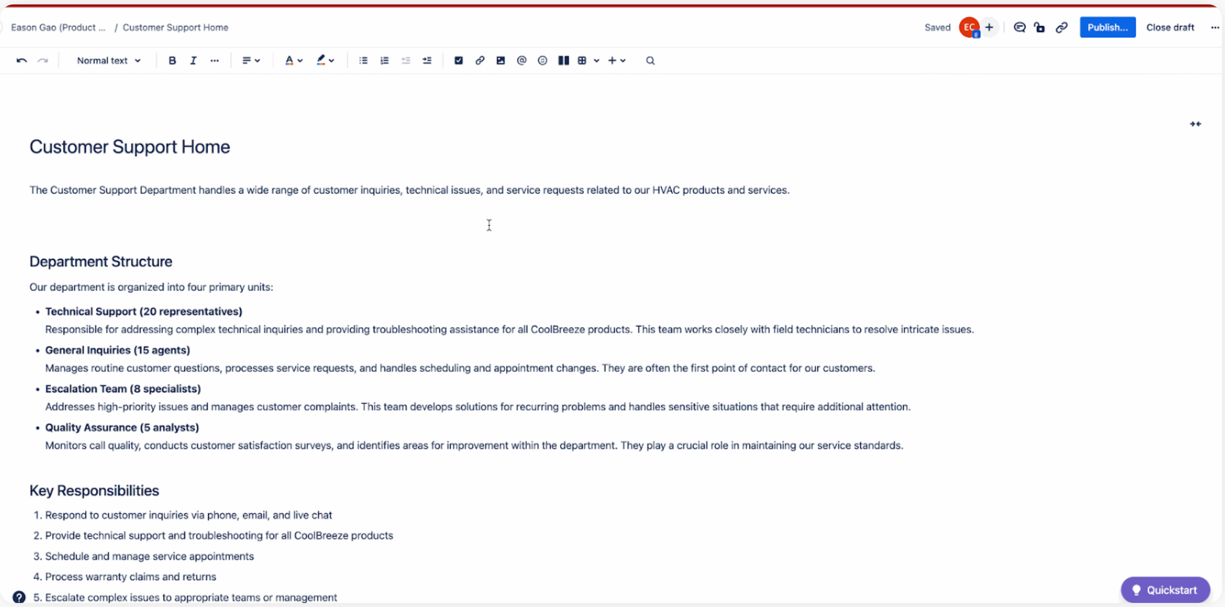
Our department is organized into four primary units:

Technical Support (20 representatives)
Responsible for addressing complex technical inquiries and providing troubleshooting assistance for all CoolBreeze products. This team works closely with field technicians to resolve intricate issues.

General Inquiries (15 agents)
Manages routine customer questions, processes service requests, and handles scheduling and appointment changes. They are often the first point of contact for our customers.

Escalation Team (8 specialists)
Addresses high-priority issues and manages customer complaints. This team develops solutions for recurring problems and handles sensitive situations that require additional attention.

Example for embedding in SharePoint



Customer Support Home

The Customer Support Department handles a wide range of customer inquiries, technical issues, and service requests related to our HVAC products and services.

Department Structure

Our department is organized into four primary units:

- **Technical Support (20 representatives)**
  Responsible for addressing complex technical inquiries and providing troubleshooting assistance for all CoolBreeze products. This team works closely with field technicians to resolve intricate issues.
- **General Inquiries (15 agents)**
  Manages routine customer questions, processes service requests, and handles scheduling and appointment changes. They are often the first point of contact for our customers.
- **Escalation Team (8 specialists)**
  Addresses high-priority issues and manages customer complaints. This team develops solutions for recurring problems and handles sensitive situations that require additional attention.
- **Quality Assurance (5 analysts)**
  Monitors call quality, conducts customer satisfaction surveys, and identifies areas for improvement within the department. They play a crucial role in maintaining our service standards.

Key Responsibilities

1. Respond to customer inquiries via phone, email, and live chat
2. Provide technical support and troubleshooting for all CoolBreeze products
3. Schedule and manage service appointments
4. Process warranty claims and returns
5. Escalate complex issues to appropriate teams or management

Example for embedding in Confluence

Conclusion

Embedding provides an opportunity for you to integrate data-driven insights seamlessly into your organization's existing tools. We're excited to see how you and your teams will leverage this alongside AI/BI's many other capabilities to transform your business processes. We encourage you to dive in and start creating your own dashboards today. To get started, explore our AI/BI Dashboards documentation and follow our tutorial to build your first dashboard. To see AI/BI Dashboards and Genie in action, check out our demo and take the product tour.

Your feedback is invaluable as we continue to refine and expand our AI/BI Dashboards. We look forward to hearing your thoughts and suggestions.
Happy dashboarding!