



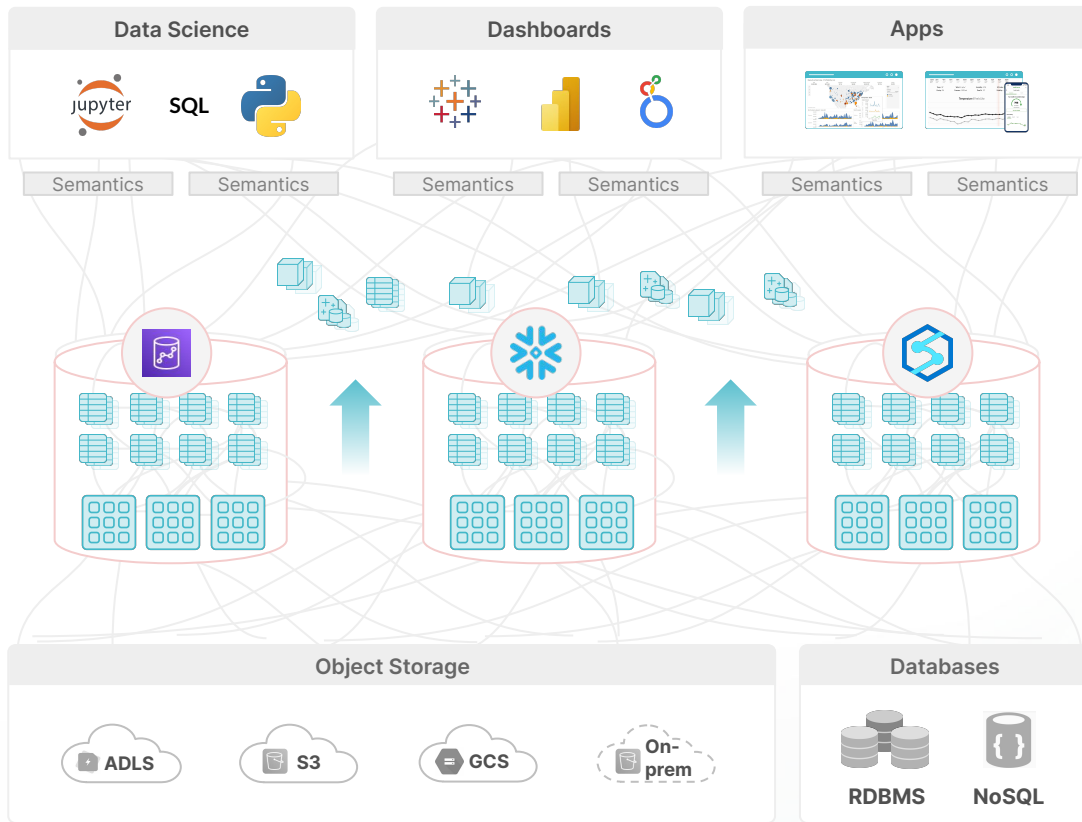
Your Lakehouse Just Got Gnarlier

What's New in Dremio

Agenda

- Dremio Overview
- What's new in Dremio?
- Arctic Intelligent Catalog
- How can you get started?

Data is Everywhere



- × Complex
- × Expensive
- × Lock-in
- × Impossible to secure
- × No self-service
- × Limited data exploration
- × Inconsistent data

Dremio Data Lakehouse: Easy, Open, 1/3 the Cost

Data Science

jupyter SQL Python

Dashboards

Tableau Power BI Google Data Studio

Apps

Mobile App Web App

↕ ODBC | JDBC | REST | Arrow Flight ↕



- ✓ No data copies
- ✓ No complex ETL pipelines
- ✓ Inexpensive
- ✓ Self-service analytics
- ✓ Data engineering productivity
- ✓ Security
- ✓ Governance
- ✓ Consistent data

Object Storage

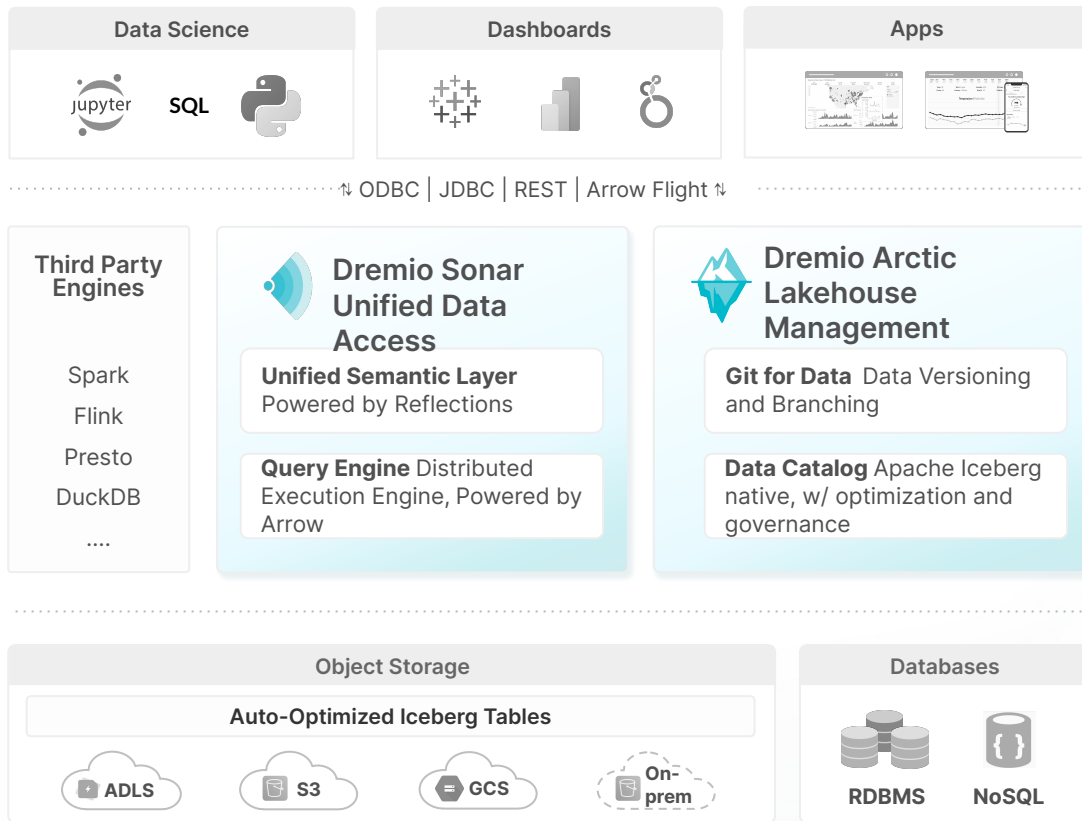
Auto-Optimized Iceberg Tables

ADLS S3 GCS On-prem

Databases

RDBMS NoSQL

Dremio is a Modern and Open Enterprise-Grade Lakehouse



- ✓ No data copies
- ✓ No complex ETL pipelines
- ✓ Inexpensive
- ✓ Self-service analytics
- ✓ Data engineering productivity
- ✓ Security
- ✓ Governance
- ✓ Consistent data



Dremio Generative AI

GenAI for Data Engineers and Analysts

The screenshot displays the Dremio SQL Runner interface. At the top, there's a navigation bar with a dropdown menu and a 'SQL Runner' tab. Below this, a toolbar contains buttons for 'Run', 'Preview', 'Discard', and an 'Engine' dropdown set to 'Automatic'. A 'Hide SQL pane' button and a 'Save Script As' dropdown are also present. The main workspace is divided into two sections. The left section, titled 'Context: @username@dremio.com', contains a SQL editor with the following query:

```
1 SELECT AVG(tip_amount) AS avg_tip
2 FROM Samples."samples.dremio.com"."NYC-taxi-trips-iceberg"
3 WHERE passenger_count = 2;
```

 The right section, titled 'Datasets to analyze', shows a list with 'NYC-taxi-trips-iceberg'. Below this is a text input field with the prompt 'What's the average tip for a two-passenger ride?' and a 'Generate' button. A green checkmark and the text 'Query generated!' are visible below the input. At the bottom, a results pane shows a table with one column, 'avg_tip', and one row with the value '1.420523461214428'.

TEXT-TO-SQL

- Generate SQL from natural language (Available now!)
- Refine generated SQL using Generative AI (Q3)
- Ask questions on entire sources and catalogs (Q3)

Semantic Layer Enables Easy Data Exploration with GenAI

Automatically Generate Wiki (Q4)

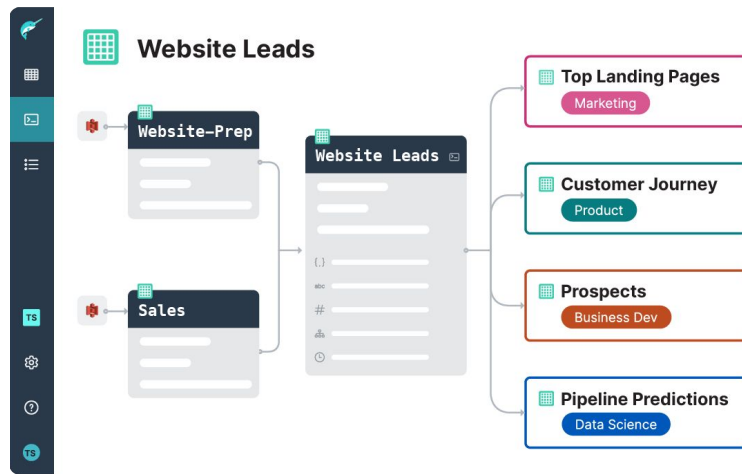
Automatic generation of dataset descriptions, SQL examples, and more.

Automatically Generate Labels (Q4)

Automatic generation of data tags for tables and views.

Autonomous Semantic Layer

Dremio enables easy data exploration for analysts and data scientists without the need to enrich and catalog manually.





Dremio Next Gen Reflections

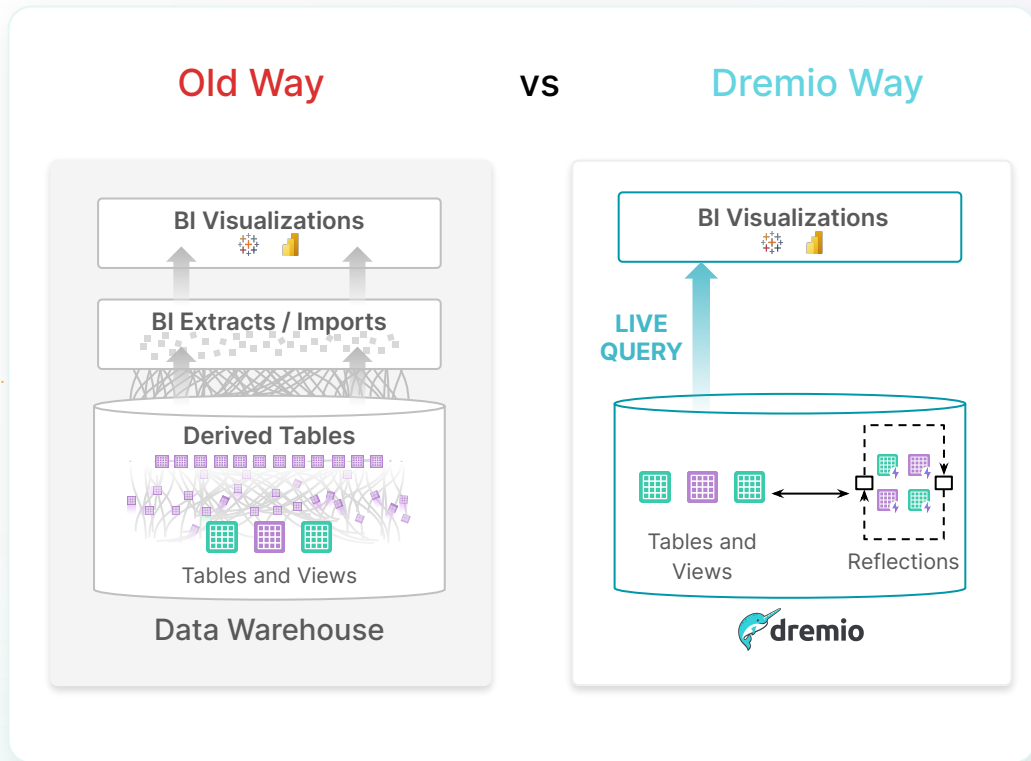
Reflections Accelerate Enterprise Analytics

Why Reflections?

- ✓ Query Acceleration
- ✓ Transparent to end users
- ✓ Reusable
- ✓ Persisted on (S3, ADLS, GCS, HDFS or your data lake) as Parquet/Iceberg

What this means

- ✓ Reduced TCO for Data & Analytics Management
- ✓ More efficient data delivery
- ✓ Deliver faster business results with less data requests: Self Service Democratization



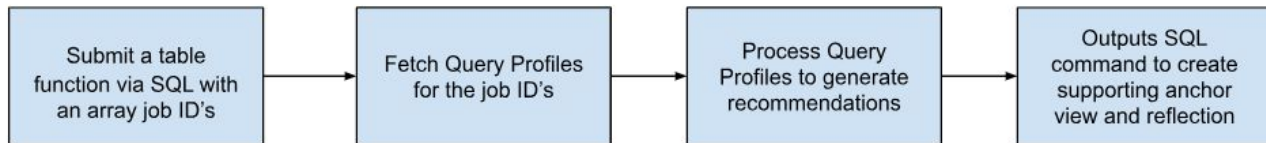
Real World Impact

"Dremio slashed our query times by over 99%, massively reducing our analytics costs. We had a production query that took 70 minutes to return. We deployed Dremio on top of Amazon S3, and query time dropped to 33 seconds. With Dremio Reflections, the query time was accelerated even further to just three seconds. We're all blown away by the shocking performance we get from Dremio and Reflections."

Andy Kenna, senior vice president and head of data
RenaissanceRe

New: Reflection Recommender

- Evaluate selected SQL queries & automatically generate script for new Reflection
- Significant time-and-cost reduction
 - No analyzing existing queries
 - No expertise in data sources required
 - No analyzing existing workloads
 - Create optimal - and not unnecessary - Reflections
 - Simple to manage - same as any other Reflection



The following command generates recommendations for input BI workloads in seconds:

```
select * from TABLE(sys.recommend_reflections(ARRAY['<job_id>', ...] ));
```


New: Intelligent Reflections Refresh for Iceberg Tables

- Optimized Apache Iceberg Table refresh
 - Snapshot-based (append only changes)
 - Partition-scoped (DML changes)
- Uses Iceberg manifests to track data changes and incrementally update Reflection caches
- Ensures fastest and most economical access to freshest data
- Increases analytics reliability and accuracy



Query Acceleration Enables Low-Latency BI

Observability (Available now!)

Real-time observability for Reflections, including refresh and usage information



The screenshot shows the Dremio 'Reflections' settings page. On the left is a sidebar with navigation options: Settings, Node Activity, Engines, Queues, Engine Routing, Reflections (selected), BI Applications, Users, Roles, Support, and SQL. The main panel is titled 'Reflections' and contains a search bar with 'dataset960', filters for 'Acceleration Status: All', 'Refresh Status: All', and 'Type: All', and a 'Manage Columns' button. Below is a table listing reflections.

Name	Type	Dataset	Current Footprint	Last Refresh Duration	Accelerated Count	Refresh Job History
dataset960_agg960	Aggregation	dataset960	9.05 KB	00:00:46	5	History
dataset960_raw960	Raw	dataset960	799.60 KB	00:00:44	15	History
dataset960_2_agg960	Aggregation	dataset960.2	8.19 KB	00:00:34	0	History
dataset960_2_raw960	Raw	dataset960.2	8.23 KB	00:00:44	0	History

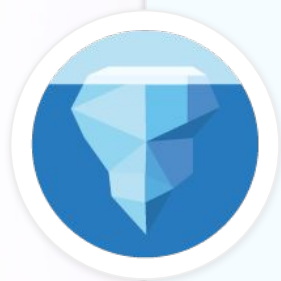
From Data Lake to Lakehouse: Table Format Enhancements

Open Table Formats are the Foundation of an Open Lakehouse



- 1 The future is open data architectures
- 2 Data lake → Data lakehouse
- 3 Apache Iceberg: An open table format

Apache Iceberg is the Format of Choice in Big Tech



NETFLIX



stripe

 **Expedia**

 **airbnb**

 **twilio**

Tencent

Linked in

 **Adobe**

Our Approach on Table Formats: Iceberg and Delta Lake



ICEBERG

- Dremio delivers high-performance **read/write** access (concurrently with other engines)
- Dremio performs automatic data optimization and garbage collecting
- Variety of open source and commercial catalogs

DELTA LAKE

- Dremio delivers high-performance read-only access (SELECT)
- Databricks manages data optimization and garbage collection

Table Formats: New Delta Lake Time-Travel

- New **time-travel capabilities** for Delta Lake for TIMESTAMP and SNAPSHOT
- Compare historical point-in-time analysis or perform time-series analytics
- Time-travel (and DML) already available for Apache Iceberg (and uses identical syntax)

Time travel

```
SELECT * FROM t1 AT/BEFORE <timestamp>
```





Expanding Connector Ecosystem

Unified Analytics in Dremio: New Apache Druid Connector

Across Sources



New Connectors

- **Apache Druid**
- Snowflake
- Db2
- OpenSearch SQL
- MongoDB Connector for BI
- BigQuery
- Elasticsearch SQL



Across Regions & Clouds



Dremio-to-Dremio

- Cross-cluster queries
- Multi-region, multi-cloud, and hybrid cloud

Easy to Extend



Third-Party SDK

- Easier way to build connectors
- Allows companies to easily create new connectors
- Auto-detection of database capabilities

Improved AWS Lake Formation Integration

- **AWS Lake Formation integration improvements** - inherits row, column, and cell permissions set in Lake Formation



Expanded SQL Functions

The background is a solid teal color. Overlaid on this is a complex, abstract pattern of thin, light-colored lines and small dots. These lines and dots form a network of interconnected geometric shapes, including hexagons and other polygons, creating a sense of depth and complexity. The pattern is more dense on the right side of the image and fades slightly towards the left.

New SQL Functions in 24.2

Signature	Description
<code>array_avg(A)</code>	Returns the average of all non-null elements in A.
<code>array_contains(A, V)</code>	Returns whether A contains V.
<code>array_max(A)</code>	Returns the maximum value in A.
<code>array_min(A)</code>	Returns the minimum value in A.
<code>array_remove(A, V)</code>	Removes all elements that equal V in A.
<code>array_sum(A)</code>	Returns the sum of all non-null elements in A.
<code>cardinality(A)</code>	Returns the number of elements in A.
<code>unnest(A)</code>	Converts elements in A into rows.

New SQL Function: Create Array Literals

```
SELECT ARRAY['apple', 'strawberry', 'banana']  
-- ["apple", "strawberry", "banana"]
```

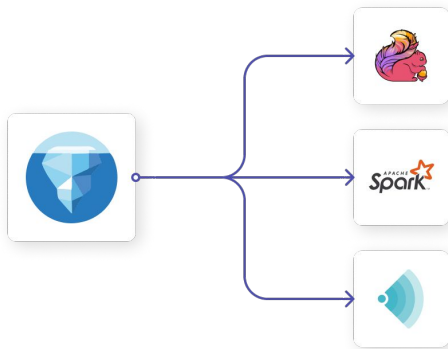
What's next for SQL functions?

Signature	Description
<code>array_agg(expr)</code>	Returns an array consisting of all values in expr.
<code>array_append(A, E)</code>	Returns a new array with E at the end of A.
<code>array_distinct(A)</code>	Returns a new array with only the distinct elements from A.
<code>array_frequency(A)</code>	Returns a map where the keys are the unique elements in A, and the values are how many times the key appears.
<code>array_prepend(A, E)</code>	Returns a new array with E at the beginning of A.
<code>arrays_overlap(X, Y)</code>	Returns whether X and Y have any elements in common.
<code>set_union(X, Y, ...)</code>	Returns an array of all the distinct values contained in each array of the input.

Dremio Arctic

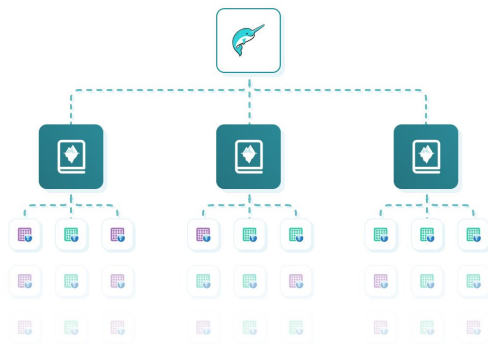


Dremio Arctic is a Modern Lakehouse Catalog



ICEBERG-NATIVE

- Nessie (the Arctic catalog) is built into the open source Apache Iceberg project
- Use a variety of Iceberg-compatible engines including Dremio Sonar, Spark and Flink



MULTIPLE DOMAINS

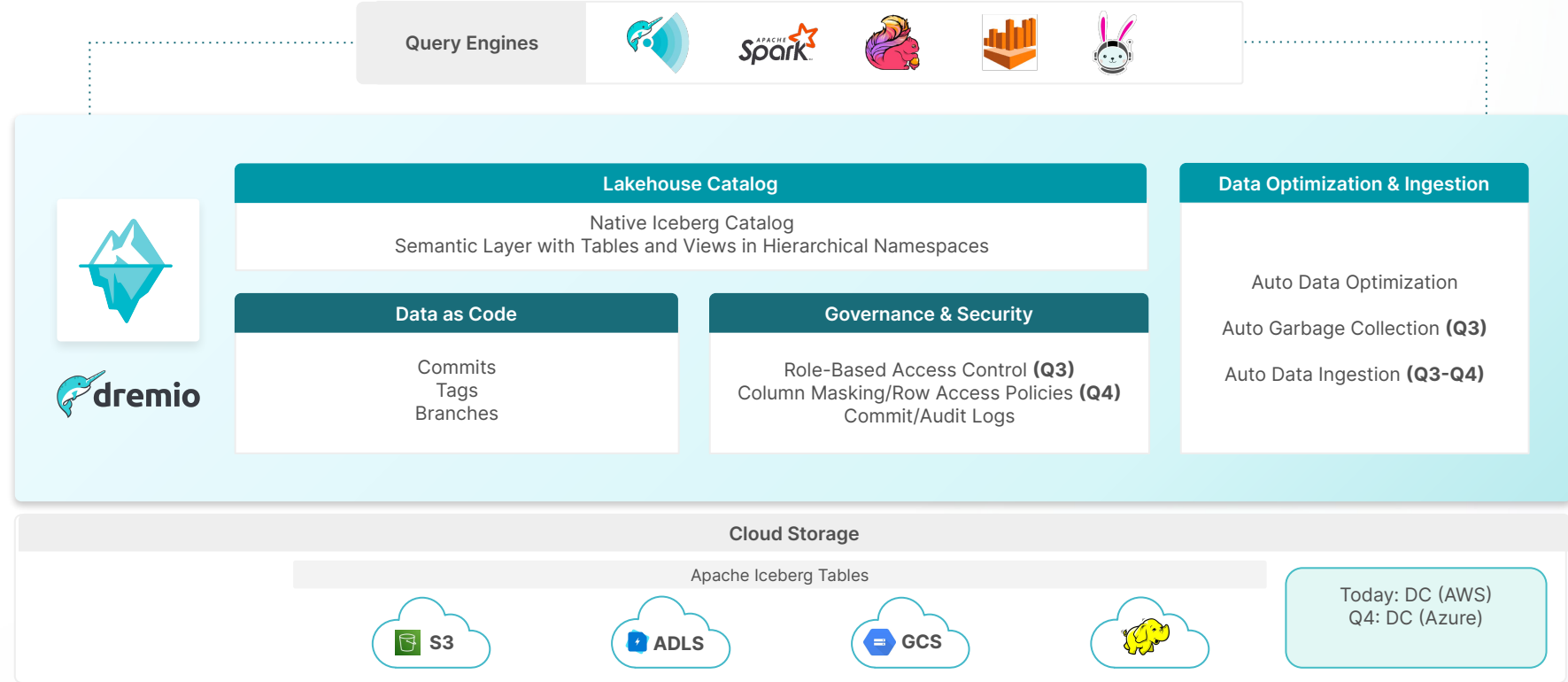
- Multiple isolated domains/catalogs in an organization, each containing a folder hierarchy of tables and views
- Designed to enable data mesh (including federated ownership and data sharing)



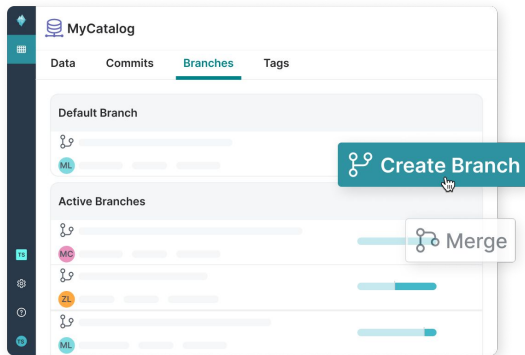
ACCESS CONTROL

- Table, column- and row-based access control
- Custom roles and integration with existing user/group directories (AAD, Okta, etc.)

Dremio Arctic is a Data Lakehouse Management Service

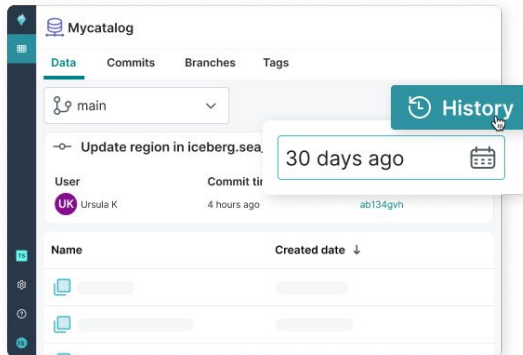


Simplify Data Engineering with Data-as-Code



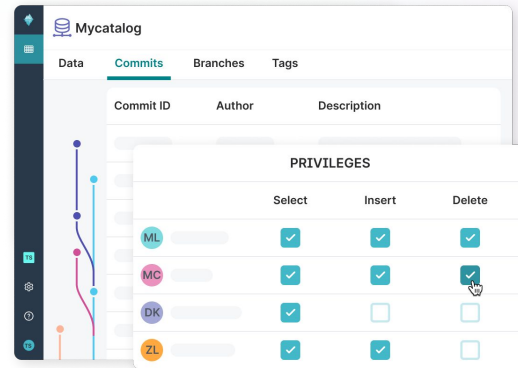
BRANCHING

- **Ingest, transform, and validate data** on isolated branches and **instantly promote changes** to production
- **Experiment with data immediately** using branches instead of creating data copies



VERSION CONTROL

- **Reproduce models and dashboards** from historical data based on time or tags
- **Roll back changes instantly** and undo mistakes without downtime



GOVERNANCE

- **Track all changes** to data and metadata: who accessed what data and when
- **Control access** using role-based access control (Q3) and fine-grained privileges (Q4)

Ingest and Optimize Data Automatically

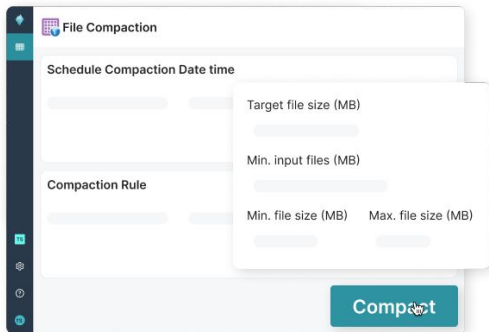
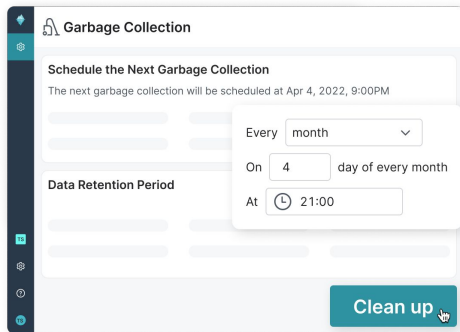


TABLE OPTIMIZATION

- Automatically **compact small files** and **group similar rows** together
- Table optimization significantly accelerates query performance



GARBAGE COLLECTION

- Automatically **remove unused data files, manifest files, and manifest lists** (Q4)
- Background cleanup ensures efficient use of data lake storage



INGESTION

- **Event-driven pipelines:** Automatically ingest from Amazon S3, ADLS, GCS (Q3)
- **Continuous ingestion:** Automatically write from Kafka topics into Arctic (Q4)



Git for Data Key Use Cases

1: Ensure data quality with ETL branches

Create an ETL branch and ingest the data with COPY INTO, CTAS or Spark:

```
CREATE BRANCH events_etl_9_28_22
USE BRANCH events_etl_9_28_22
COPY INTO web.events ...
```

Run queries to test data quality:

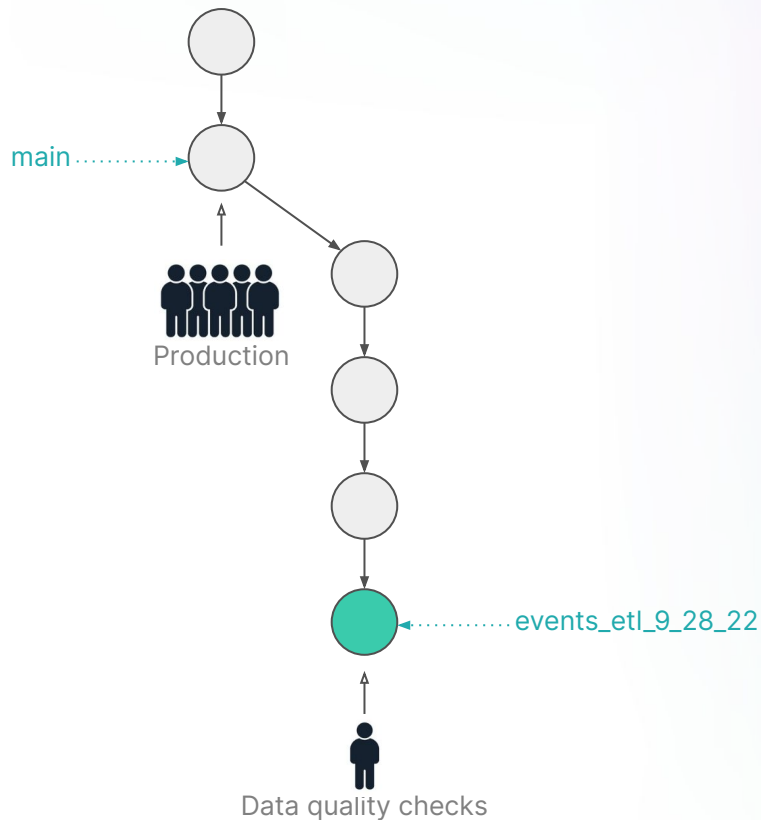
```
SELECT COUNT(*) FROM web.events WHERE
length(ip_address) >= 7
```

Test the dashboard to see that it looks ok:



Fix the problems and merge into main:

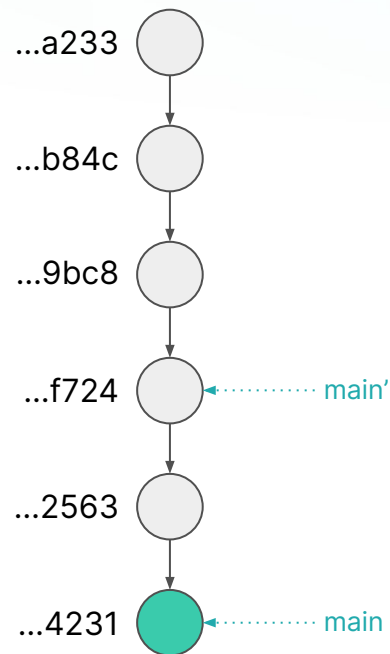
```
DELETE FROM web.events WHERE length(ip_address) >= 7
USE BRANCH main
MERGE BRANCH events_etl_9_28_22
```



2: Recover from mistakes immediately

Move the branch head to a historical commit:

```
ALTER BRANCH main ASSIGN COMMIT ...f724
```



3: Experiment with data in transient branches

Create a transient branch and perform data explorations and transformations in it:

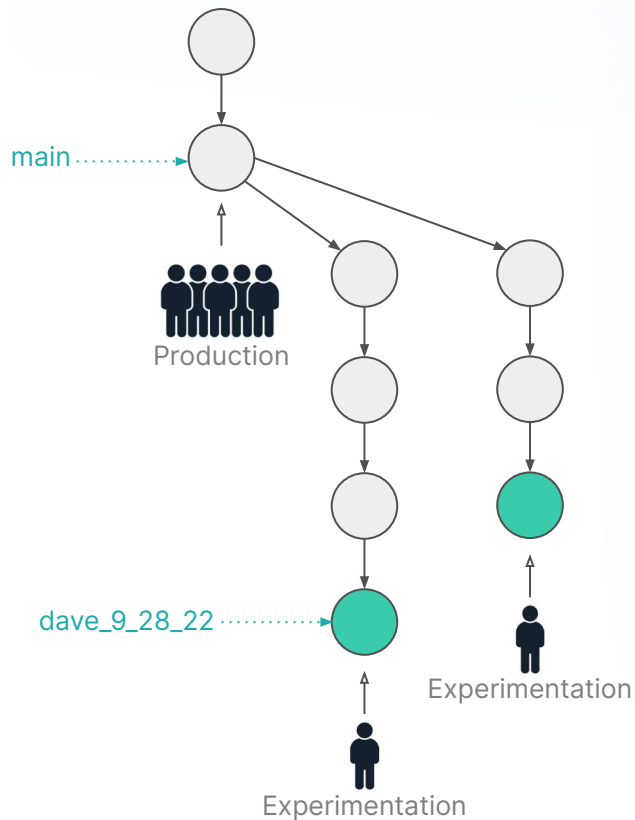
```
CREATE BRANCH dave_9_28_22  
USE BRANCH dave_9_28_22  
CREATE TABLE t AS SELECT ...  
UPDATE t ... SET ...
```

Create ad-hoc visualizations on the branch via a Notebook:



Delete the branch or merge it when experimentation is complete:

```
DROP BRANCH dave_9_28_22
```



4: Reproduce models and analyses

Change context to a named tag:

```
spark.sql("USE REFERENCE modelA in arctic;")
```

Create ML model based on historic data:

```
val trainingData = spark.read.table("arctic.t")  
val lr = new LogisticRegression()  
// configure logistic regression...  
val paramMap = ParamMap(...)  
val model = lr.fit(trainingData, paramMap)
```

Select a tag, commit or branch to query in SQL Runner:

The screenshot shows the Dremio SQL Runner interface. On the left, a sidebar displays a file tree with folders like 'Taxi', 'Asia', 'GCS', and 'Level 1'. The main area shows a query with four lines:
1. INSERT INTO new-catalog.table-sonar
2. SELECT * FROM samples."samples.dremio.com"."NYC-taxi-trips"
3. WHERE pickup_datetime = '2013-02-01'
4. SELECT COUNT(pickup_datetime) as num_rows, MAX(pickup_datetime) as
A 'Set Nessie references' dialog is open, showing 'new-catalog' and 'etl1'. Below the query, the execution plan shows 'INSERT' and 'SELECT' steps. At the bottom, a table with two columns, 'num_rows' and 'max_time', displays the results of the query.

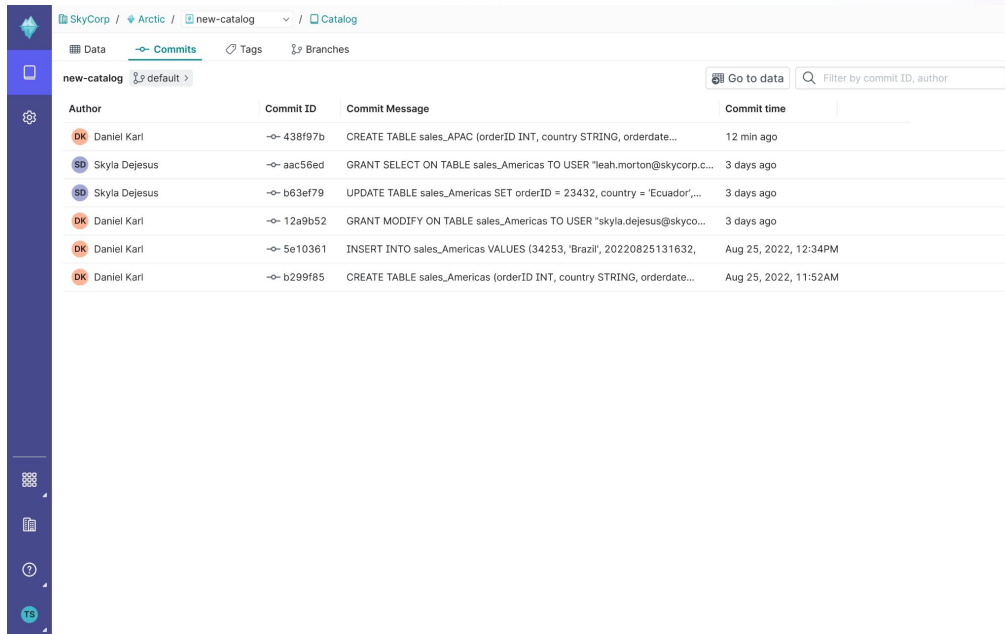
5: Troubleshoot changes (see who changed the data)

Get the commit history for a branch:

```
SHOW LOGS AT REFERENCE etl;
```

Get the commit history for a specific table:

```
curl -X GET -H 'Authorization: Bearer  
<PAT>' <Catalog API  
Endpoint>/trees/tree/<reference  
name>/log\?filter="operations.exists(op,op.  
key='<table name>')"
```



Author	Commit ID	Commit Message	Commit time
DK Daniel Karl	-o- 438f97b	CREATE TABLE sales_APAC (orderId INT, country STRING, orderdate...	12 min ago
SD Skyla Dejesus	-o- aac56ed	GRANT SELECT ON TABLE sales_Americas TO USER "leah.morton@skycorp.c...	3 days ago
SD Skyla Dejesus	-o- b63ef79	UPDATE TABLE sales_Americas SET orderId = 23432, country = 'Ecuador',...	3 days ago
DK Daniel Karl	-o- 12a9b52	GRANT MODIFY ON TABLE sales_Americas TO USER "skyla.dejesus@skycor...	3 days ago
DK Daniel Karl	-o- 5e10361	INSERT INTO sales_Americas VALUES (34253, 'Brazil', 20220825131632,	Aug 25, 2022, 12:34PM
DK Daniel Karl	-o- b299f85	CREATE TABLE sales_Americas (orderId INT, country STRING, orderdate...	Aug 25, 2022, 11:52AM



How to get started

Ready to get started?

- Current Dremio Software customer? Visit Dremio Support Portal to download.
- Current Dremio Cloud customer? It's live!
- New to Dremio? Try [it for free](#) using Dremio Cloud or the Self-Managed Community Edition

The background is a solid teal color. Overlaid on this is a white geometric line art pattern. The pattern consists of various interconnected lines, some straight and some forming hexagonal shapes, with small circles at the vertices and intersections, creating a network-like or circuit-like appearance.

Thank you!