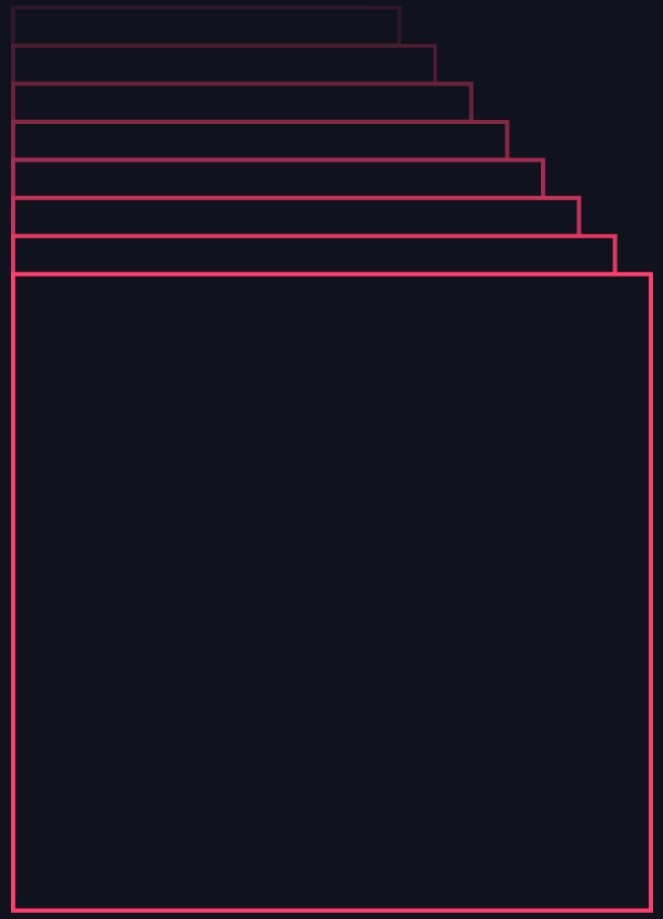
DATA+AI SUMMIT
BY databricks

DELIVERING PORTABILITY TO OPEN DATA LAKES WITH DELTA LAKE UNIFORM

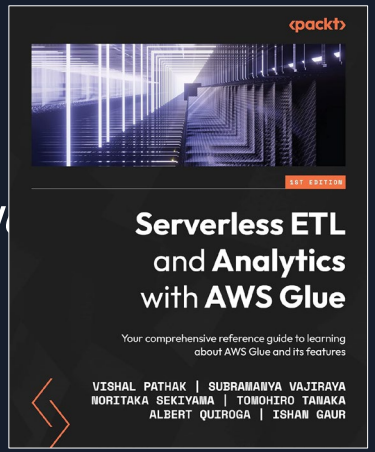


Tomohiro Tanaka
Senior Cloud Support Engineer, Amazon Web Services
2024 Jun. 11

Tomohiro Tanaka

Senior Cloud Support Engineer, AWS Support Engineering

- Working for customers to build data lake with Open Table Formats
- Contributions to Apache Iceberg
- Co-author of *Serverless ETL and Analytics with AWS Glu*



tomtongue

in/ttomtan

Agenda

Rise of Open Table Format

Delta Lake Universal Format (UniForm)

UniForm Under-the-hood

Demo: Extend Delta Lake capabilities to other OTF clients

Rise of Open Table Format

What are "Open Table Formats (OTFs)"?

Linux Foundation
Delta Lake

Apache Iceberg

Apache Hudi

An abstraction layer providing enhancement and new capabilities for traditional data lakes

ACID transaction

New features such as Time-travel, UPSERT, Compaction etc.

Challenges of Open Table Format

As scaling your OTF analytic environments along with business growth,

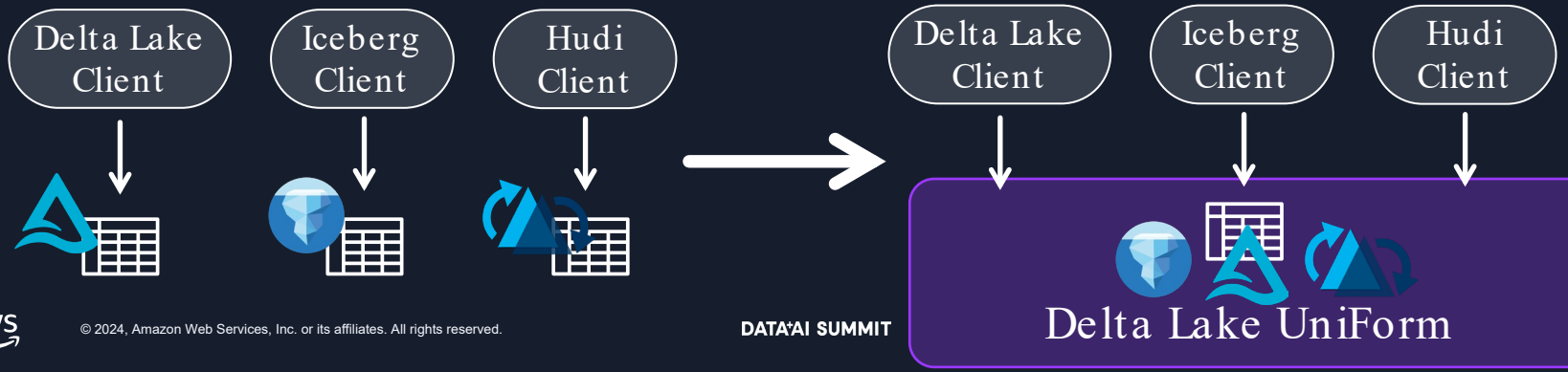
- Difficulty in managing multiple OTF environments across organizations and teams
- Limited access to some OTFs depending on your client or service specification

Delta Lake Universal Format (UniForm)

What is Delta Lake Universal Format (UniForm)?

Delta Lake feature that provides the Delta Lake table access from Iceberg or Hudi client.

- Create a new Delta Lake table with UniForm, or
  Enable UniForm on an existing Delta Lake table
- Delta Lake on Databricks runtime and OSS Delta Lake



Use Delta Lake table with UniForm

1. Select **Catalog** where you register a Delta Lake table with UniForm
2. Based on your environment,
   - Create a Delta Lake table with UniForm, or
   - Enable UniForm on your existing Delta Lake table

1. Select a catalog for Delta Lake UniForm table

Catalog options:

Databricks Unity Catalog

Hive Metastore

AWS Glue Data Catalog

1. Select a catalog for Delta Lake UniForm table

	Databricks Unity Catalog	Hive Metastore	AWS Glue Data Catalog
Runtime	*Databricks Runtime 15.0+/14.3 LTS*	*OSS Delta Lake 3.1.0+*	*OSS Delta Lake 3.2.0*
Supported OTF	Iceberg	Iceberg Hudi*2	Iceberg*1 Hudi*2

*1 To use Iceberg, need to apply this patch https://github.com/delta-io/delta/pull/2922

*2 UniForm for Hudi is currently supported as preview in OSSDelta Lake 3.2.0.

2. Create a Delta Lake table with UniForm

```
CREATE TABLE catalog.db.tbl (col_0 …) USING delta
TBLPROPERTIES('delta.enableIcebergCompatV2'='true',
'delta.universalFormat.enabledFormats'='iceberg')
```

```
CREATE TABLE catalog.db.tbl (col_0 …) USING delta
TBLPROPERTIES(
'delta.universalFormat.enabledFormats'='hudi')
```

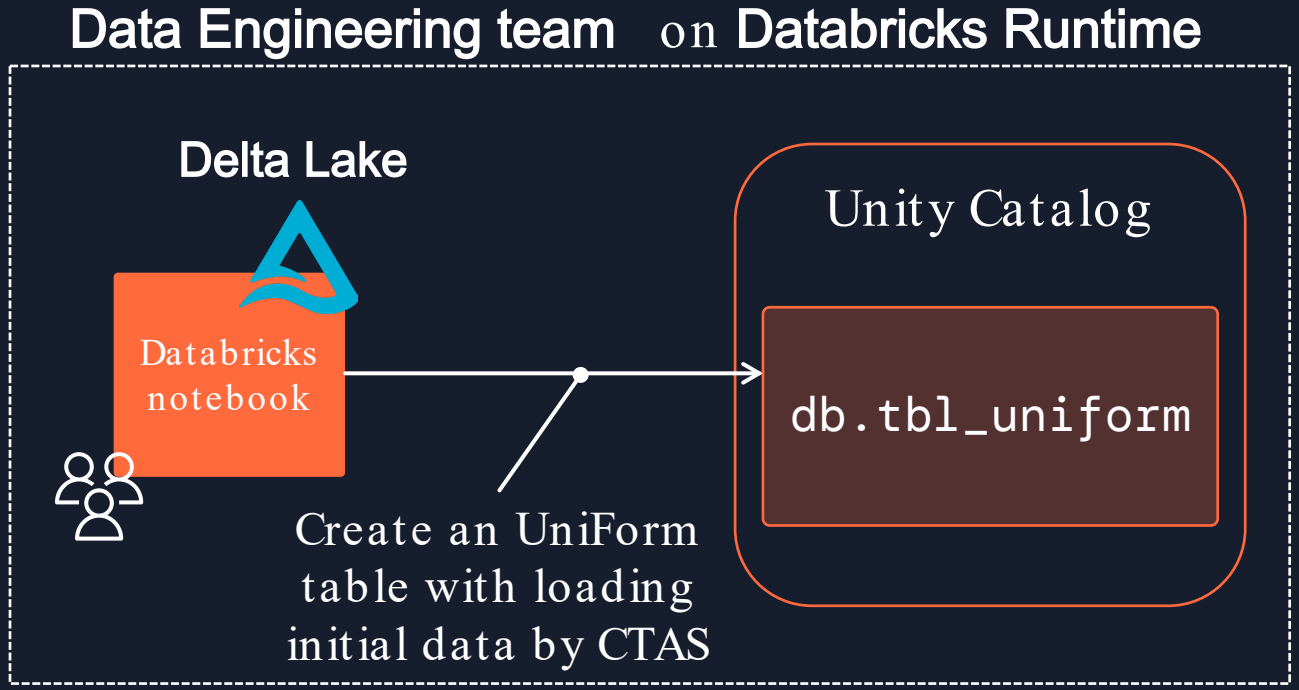
```
CREATE TABLE catalog.db.tbl (col_0 …) USING delta
TBLPROPERTIES('delta.enableIcebergCompatV2'='true',
'delta.universalFormat.enabledFormats'='iceberg,hudi')
```

2. Enable UniForm on an existing Delta Lake table

```
REORG TABLE catalog.db.tbl APPLY (
UPGRADE UNIFORM(ICEBERG_COMPAT_VERSION=2))
```

```
ALTER TABLE catalog.db.tbl SET TBLPROPERTIES (
'delta.universalFormat.enabledFormats'='hudi')
```

Demo



dais2024-demo Python

File Edit View Run Help Last edit was 3 days ago Provide feedback

Run all DAIS2024 Demo Schedule Share

Demo

Part I

1. Create an UniForm table for Iceberg with loading initial data by CREATE TABLE AS SELECT (CTAS)
2. Check files in Amazon S3

11 SQL

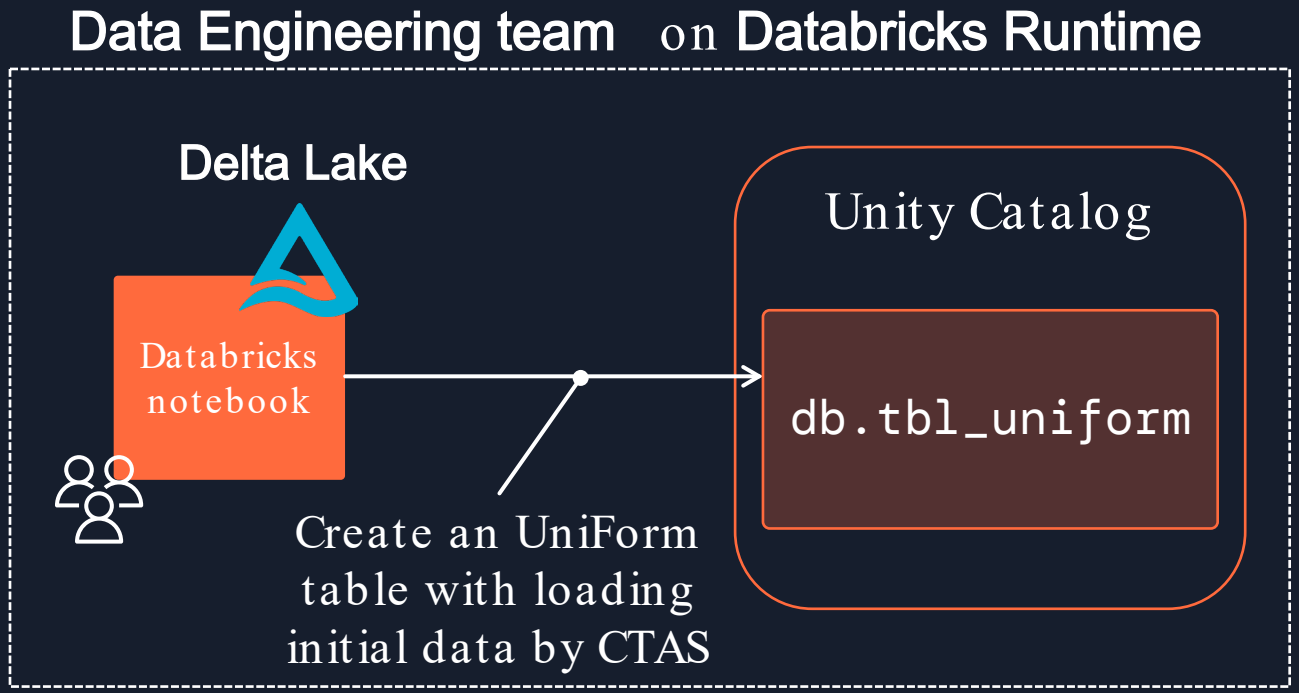
```
%sql
USE CATALOG dais2024
```

12

```
from pyspark.sql import Row

data = [
    {"product_name":"americano","price":1.50,"customer_id":1312,"order_id":"DRBY2OLZ9HBY2OLZ9H",
    "ts":"2024-02-28T23:42:09Z","year":2024},
    {"product_name":"milk","price":1.00,"customer_id":1137,"order_id":"DRLDVH8Y8AYLDVH8Y8AY",
```

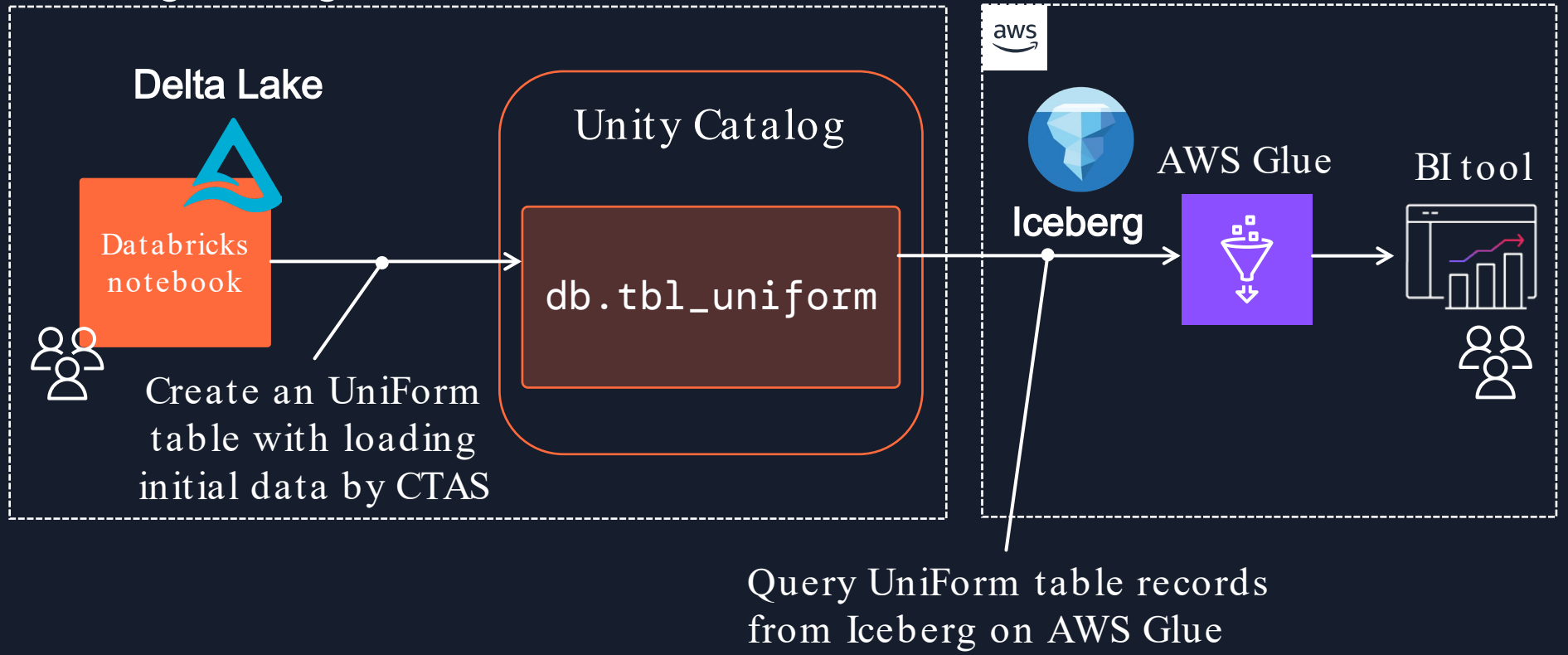
Demo



Demo

Data Engineering team on Databricks Runtime

Data Analytics team on AWS



AWS Glue

Getting started
ETL jobs
Visual ETL
Notebooks
Job run monitoring
Data Catalog tables
Data connections
Workflows (orchestration)
Data Catalog
Data Integration and ETL
Legacy pages

What's New
Documentation
AWS Marketplace

Enable compact mode
Enable new navigation

Welcome to AWS Glue

Get started by setting up your account and users, cataloging your data, and building ETL jobs to prepare data for analytics.

Prepare your account for AWS Glue

Admins: Grant access to AWS Glue and **set a default IAM role.**

Set up roles and users

Catalog and search for datasets

View your databases & tables and catalog data using Crawlers.

Go to the Data Catalog

Move and transform data

Transform data using a visual, notebook, or code interface.

Author and edit ETL jobs

Resources and tutorials

Getting started with AWS Glue: Documentation | AWS Training

Video on working with AWS Glue Studio: Part 1 | Part 2 | Part 3

Using connectors and connections

AWS Glue Documentation home

Examples: AWS Glue blog posts | AWS Glue on GitHub

What's new in Glue

Data integration and management

Monitor & debug ETL jobs and track usage

Go to job run monitoring

Connect to your data stores

Go to connections

Orchestrate jobs to build data pipelines

Go to workflows

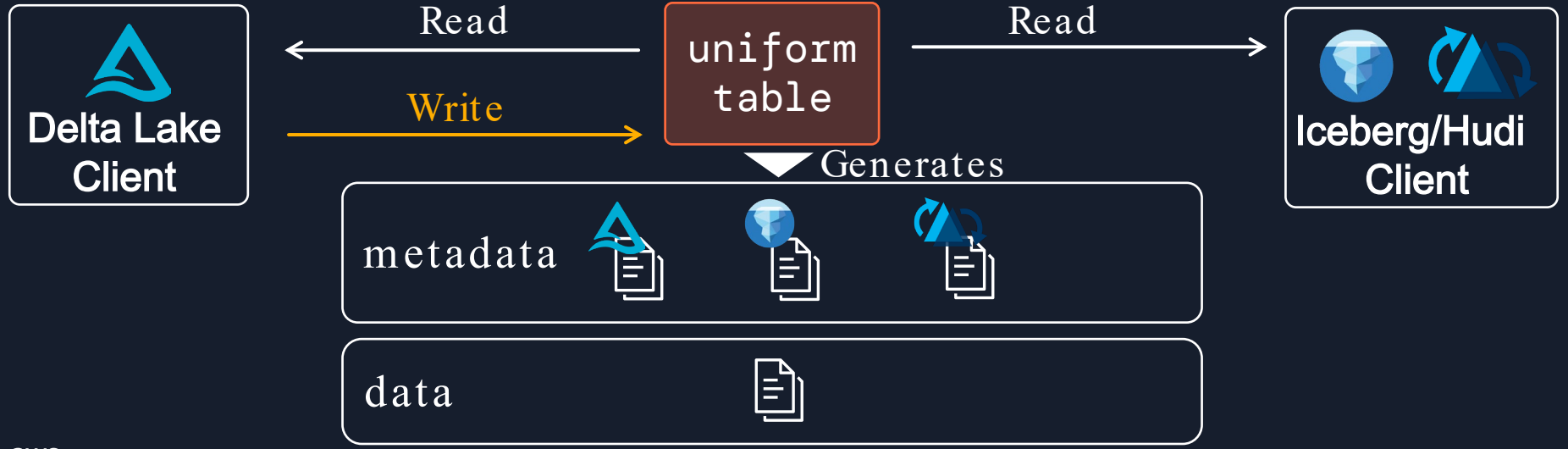
UniForm under -the-hood

How UniForm enables Iceberg/Hudi to access Delta Lake table

Writing data is only allowed from Delta Lake.

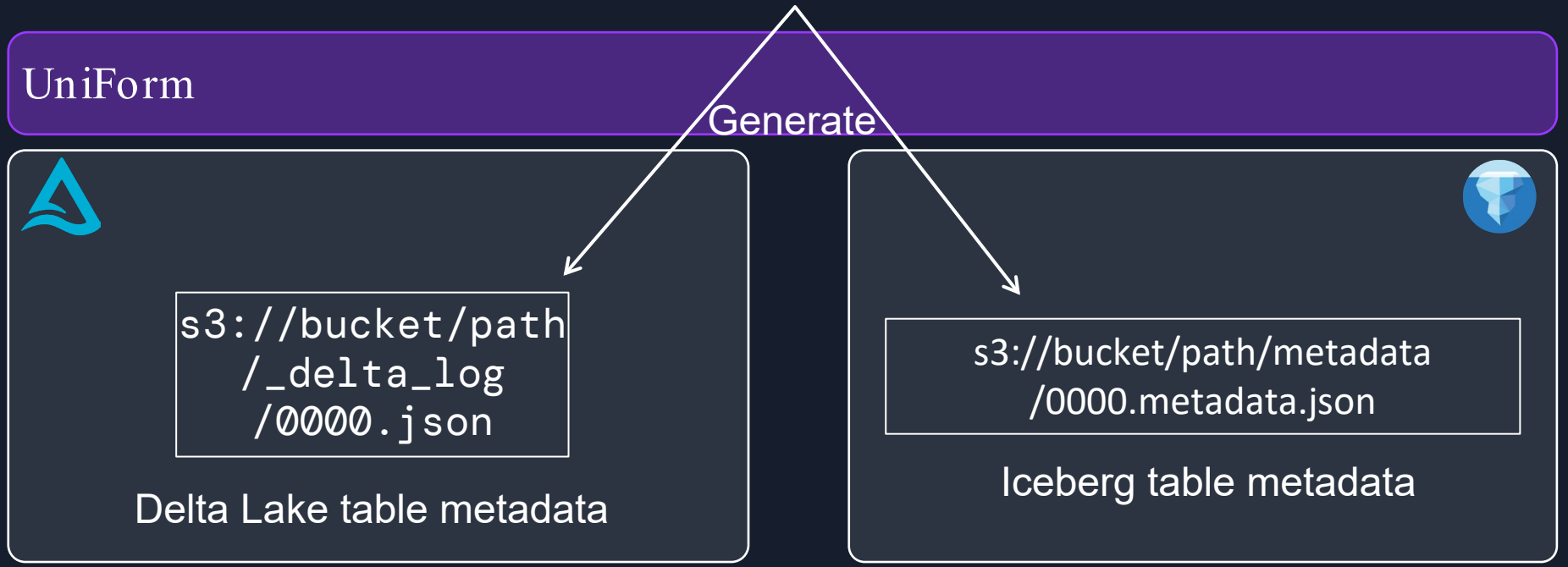
Delta Lake generates multiple OTFs metadata.

Single copy-of-data

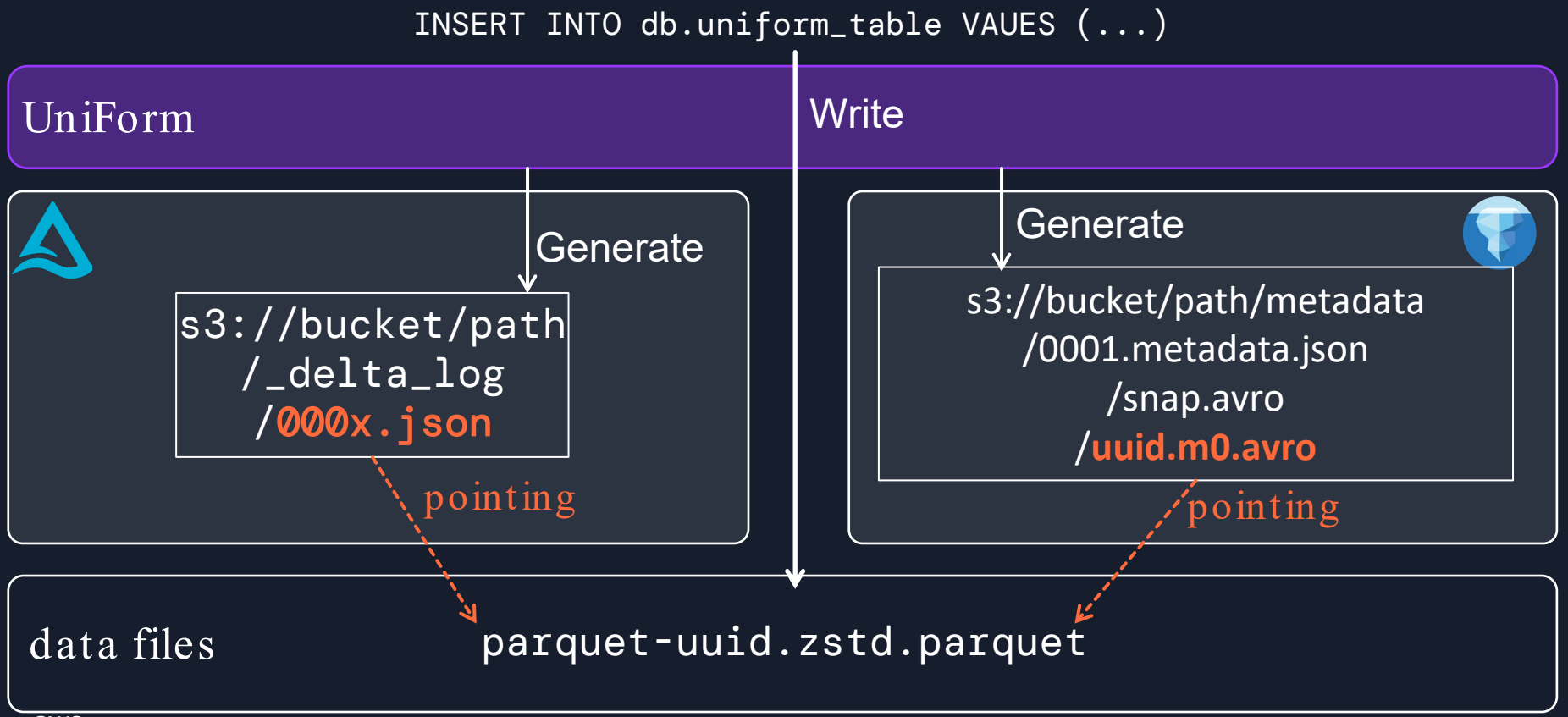


UniForm generates multiple metadata

```
CREATE TABLE ...  USING delta
TBLPROPERTIES ('delta.enableIcebergCompatV2'='true'
'delta.universalFormat.enabledFormats'='iceberg')
```



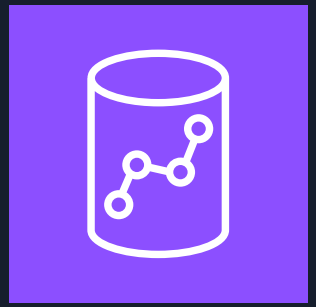
Writes: Single copy of data



Demo

Extend Delta Lake access to other OTF clients

Query Delta Lake table from Amazon Redshift



Amazon Redshift

Directly query to data in Amazon S3 including Open Table Formats

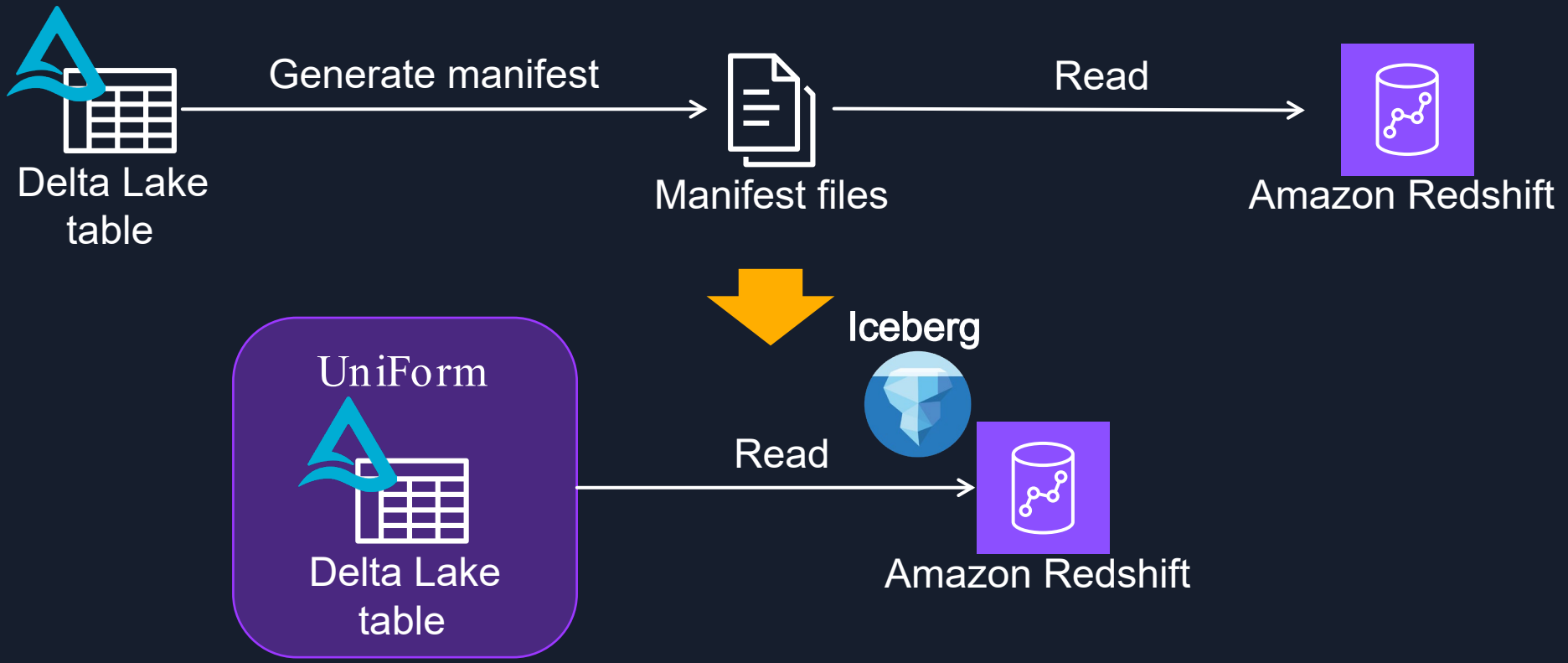
Needs **Manifest files** to read Delta Lake tables*1

- List of Parquet files for querying a Delta Lake table
- Created by a `GENERATE` query

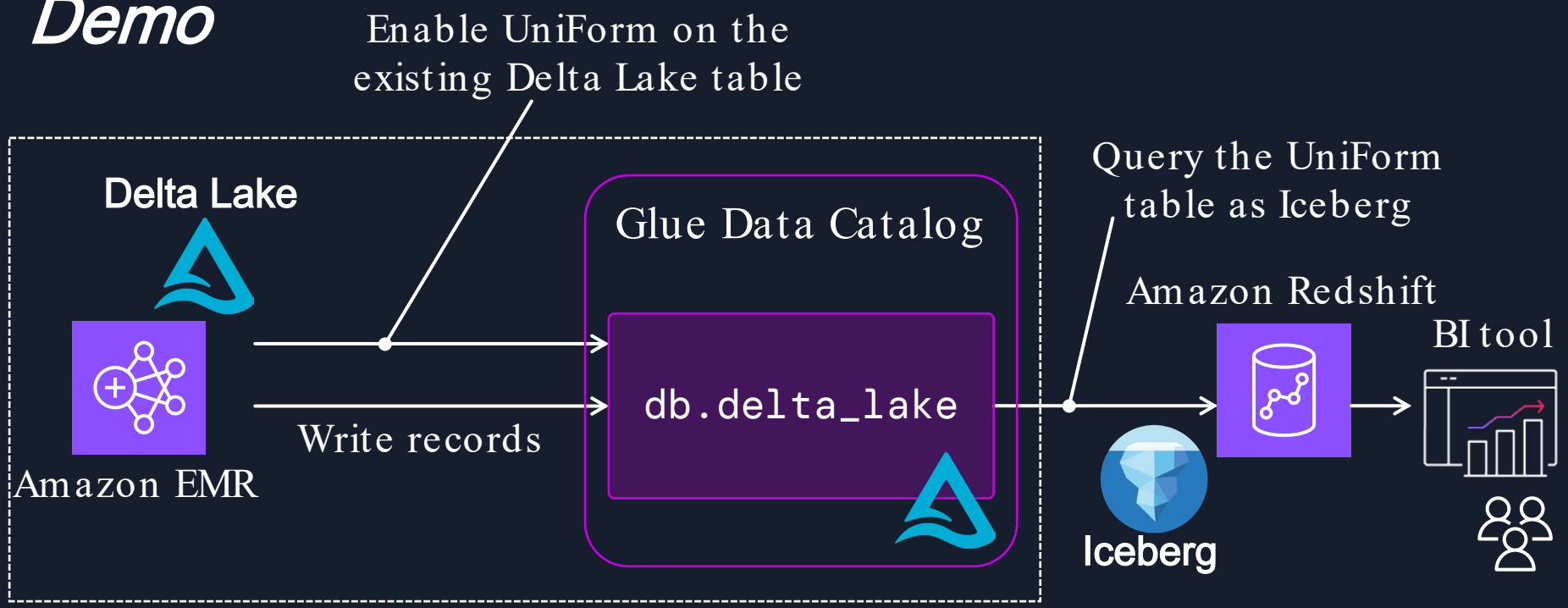
*1: Redshift Spectrum to Delta Lake Integration, https://docs.delta.io/latest/redshift-spectrum-integration.html

Query Delta Lake table from Amazon Redshift

UniForm enables Amazon Redshift to read Delta Lake table



Demo



dais2024.ipynb

File Edit View Run Kernel Git Tabs Settings Help

Code Cluster attached. … PySpark

Demo: Enable UniForm on an Existing Delta Lake table

Setup Delta Lake UniForm for Iceberg on Amazon EMR

```
%%configure -f
{
    "conf": {
        "spark.jars.packages": "io.delta:delta-spark_2.12:3.2.0",
        "spark.jars": "s3://dais2024-uniform-tmp/jars/delta-iceberg_2.12-3.3.0-SNAPSHOT.jar,s3://dais2024-uniform-tmp/jars/url-connection-client-2.20.162.jar",
        "spark.sql.extensions": "io.delta.sql.DeltaSparkSessionExtension"
    }
}
```

Create a SparkSession

```
from pyspark.sql import SparkSession


spark = SparkSession.builder \
    .master("yarn") \
    .config("spark.sql.catalog.spark_catalog", "org.apache.spark.sql.delta.catalog.DeltaCatalog") \
    .config("spark.sql.extensions","io.delta.sql.DeltaSparkSessionExtension") \
    .config("spark.sql.catalog.spark_catalog.catalog-impl", "org.apache.iceberg.aws.glue.GlueCatalog") \
    .config("spark.sql.catalog.spark_catalog.warehouse", "s3://dais2024-uniform-tmp/products/product_reviews") \
    .getOrCreate()
```

Review the existing `product_reviews` Delta Lake table

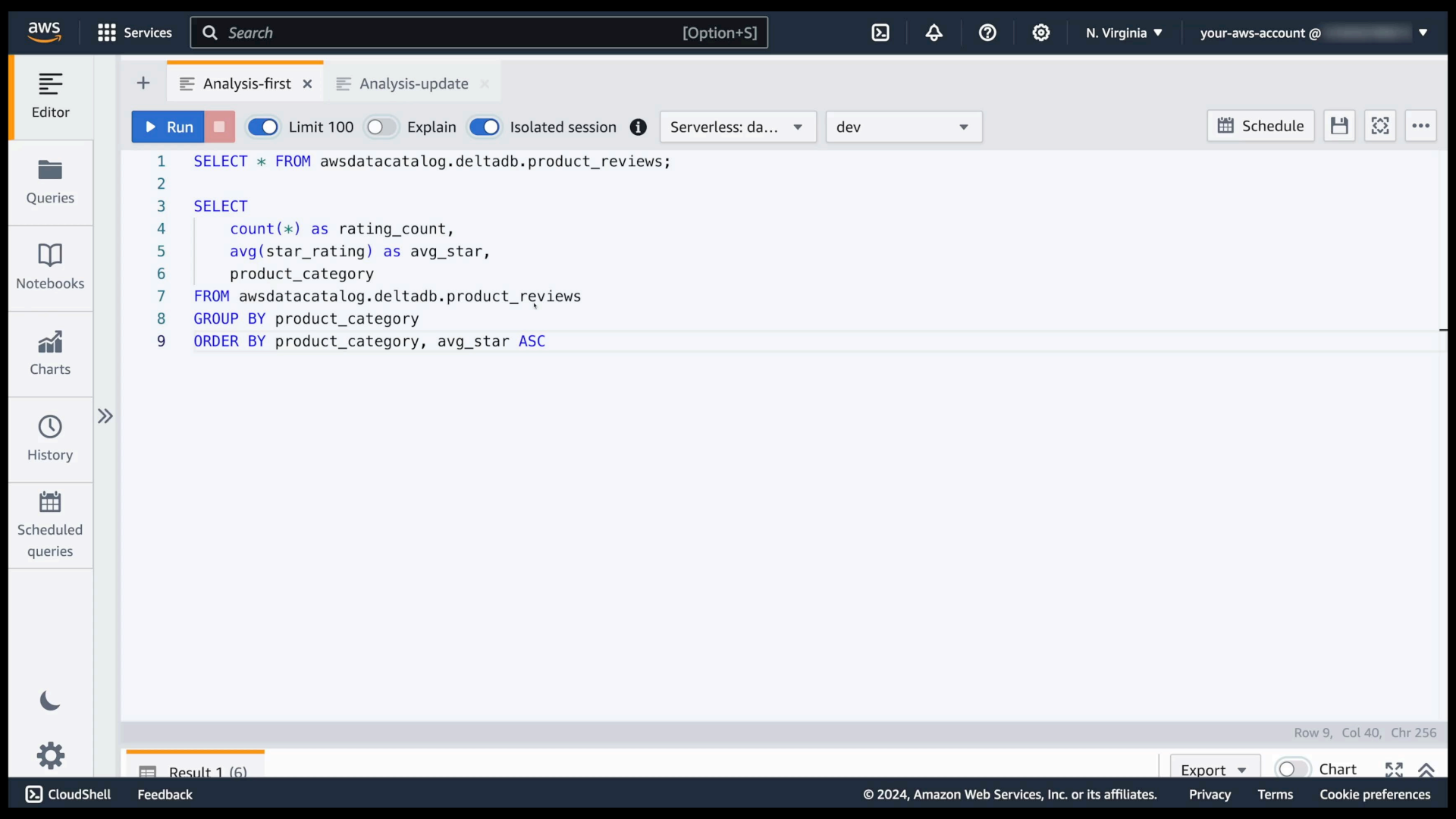
```
%%pretty
spark.sql("DESCRIBE EXTENDED deltadb.product_reviews").show(truncate=False)
```

```
%%pretty
spark.sql("SELECT * FROM deltadb.product_reviews").show(truncate=False)
```

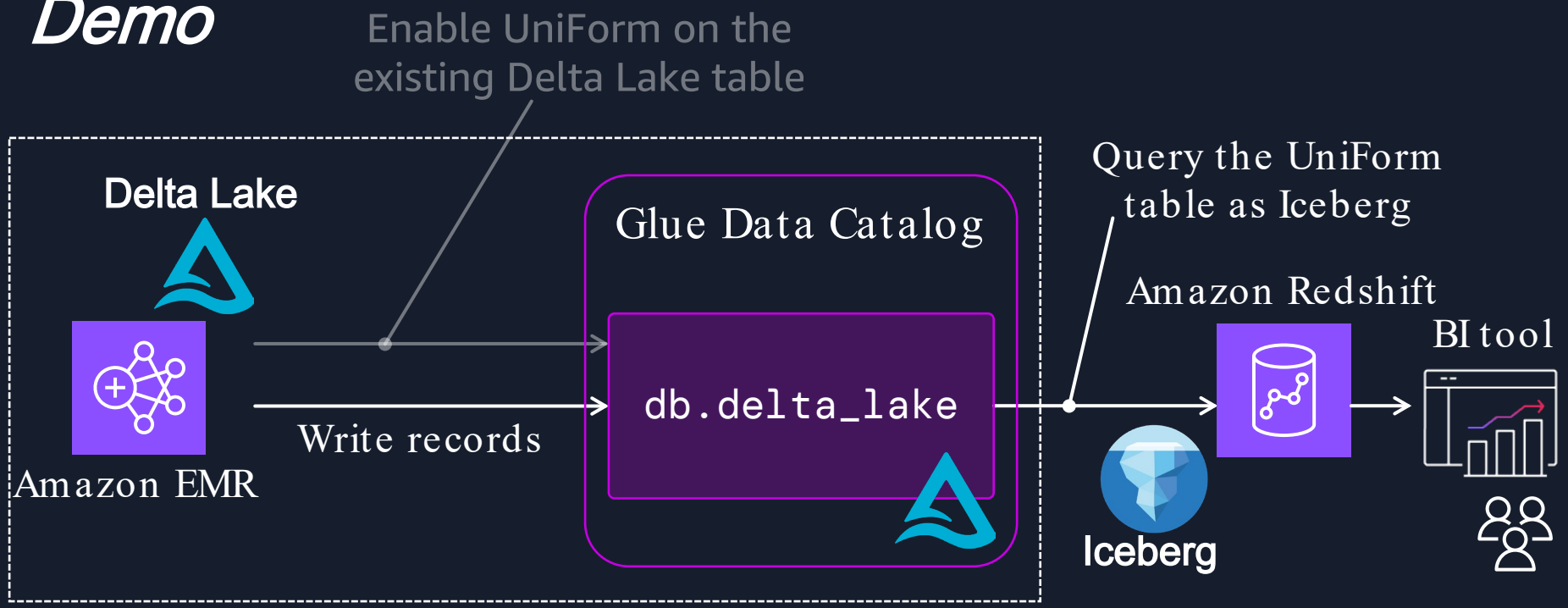
Run REORG query to enable UniForm for Iceberg

```
%%pretty
spark.sql("REORG TABLE deltadb.product_reviews APPLY (UPGRADE UNIFORM(ICEBERG_COMPAT_VERSION=2))")
```

Simple 0 1 PySpark | Idle CodeWhisperer Saving completed Mode: Command Ln 6, Col 23 dais2024.ipynb


aws
Services
Search
[Option+S]
N. Virginia
your-aws-account @
Editor
Queries
Notebooks
Charts
History
Scheduled queries
Analysis-first
Analysis-update
Run
Limit 100
Explain
Isolated session
Serverless: da...
dev
Schedule
1 SELECT * FROM awsdatacatalog.deltadb.product_reviews;
2
3 SELECT
4 count(*) as rating_count,
5 avg(star_rating) as avg_star,
6 product_category
7 FROM awsdatacatalog.deltadb.product_reviews
8 GROUP BY product_category
9 ORDER BY product_category, avg_star ASC
Row 9, Col 40, Chr 256
Result 1 (6)
Export
Chart
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Demo



dais2024.ipynb

File Edit View Run Kernel Git Tabs Settings Help

Code

Cluster attached. … PySpark

Update Delta Lake table data for Data Analytics team

New reviews:

```
[ ]: df_new_reviews = spark.read.parquet("s3://dais2024-uniform-tmp/src/new-reviews/")
df_new_reviews.createOrReplaceTempView("new_reviews")
```

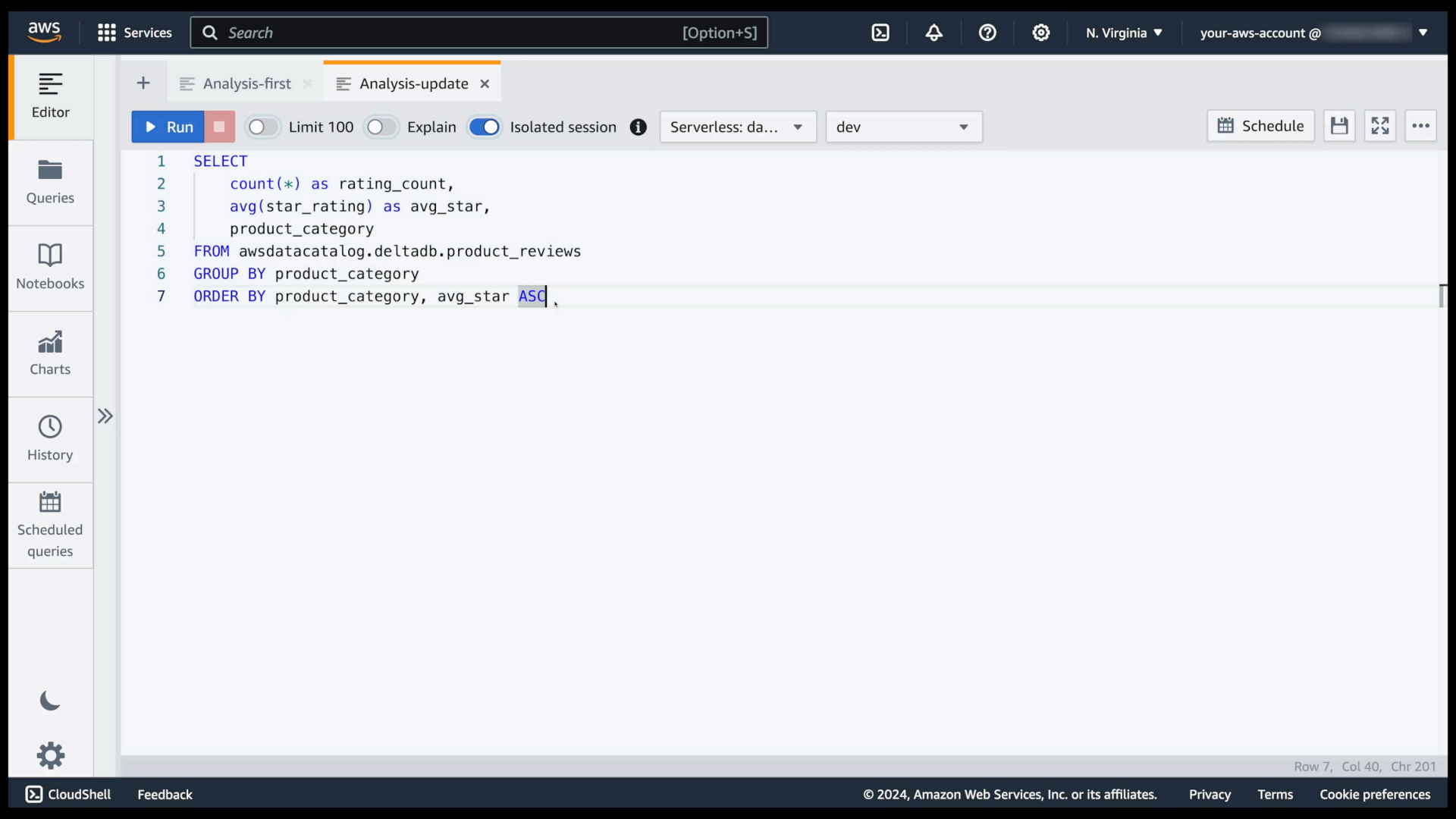
```
[ ]: %%display
df_new_reviews
```

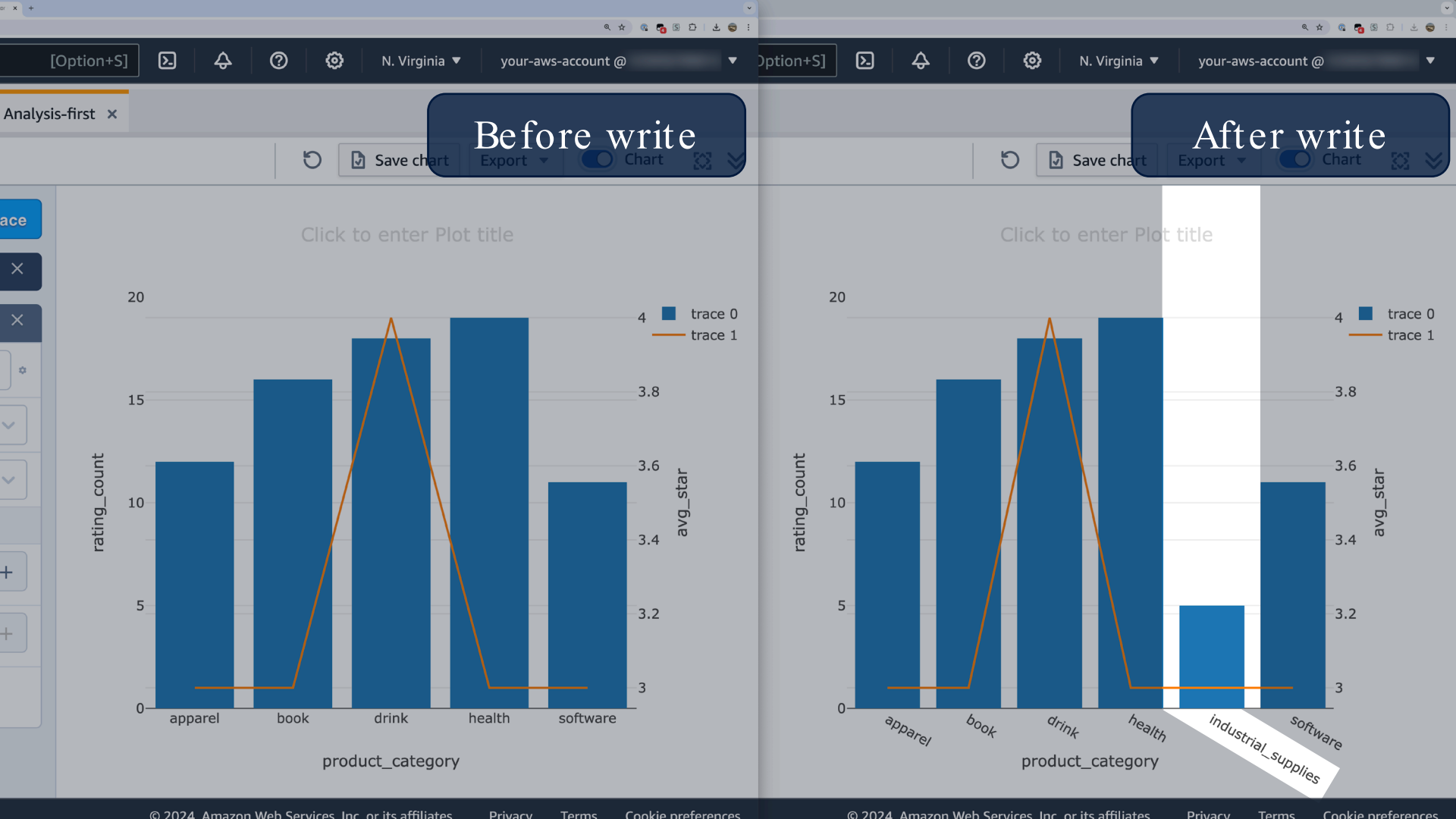
Write new review into the Delta Lake UniForm table:

```
[ ]: %%sql
INSERT INTO deltadb.product_reviews SELECT * FROM new_reviews
```

```
[ ]:
```

Simple 0 1 PySpark | Idle ✓ CodeWhisperer Saving completed Mode: Command Ln 1, Col 11 dais2024.ipynb


aws
Services
Search
[Option+S]
N. Virginia
your-aws-account @
Editor
Queries
Notebooks
Charts
History
Scheduled queries
Analysis-first
Analysis-update
Run
Limit 100
Explain
Isolated session
Serverless: da...
dev
Schedule
SELECT
count(*) as rating_count,
avg(star_rating) as avg_star,
product_category
FROM awsdatacatalog.deltadb.product_reviews
GROUP BY product_category
ORDER BY product_category, avg_star ASC
Row 7, Col 40, Chr 201
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Before write
After write
Analysis-first
Save chart
Export
Chart
Click to enter Plot title
rating_count
avg_star
trace 0
trace 1
apparel
book
drink
health
industrial_supplies
software
product_category
© 2024, Amazon Web Services, Inc. or its affiliates
Privacy
Terms
Cookie preferences

Key takeaways

Delta Lake UniForm enables to access Delta Lake tables from Iceberg and Hudi clients.

UniForm internally generates multiple open table format metadata, against single copy of the data.

UniForm enables clients with limited access to Delta Lake to query its tables as an Iceberg or Hudi client.

Thank you!

Tomohiro Tanaka

tomtongue

in/ttomtan

Appendix

Requirements to use UniForm

Delta Lake 3.1 or later

Delta table protocol version must have:

`minReaderVersion >= 2` and `minWriterVersion >= 7`

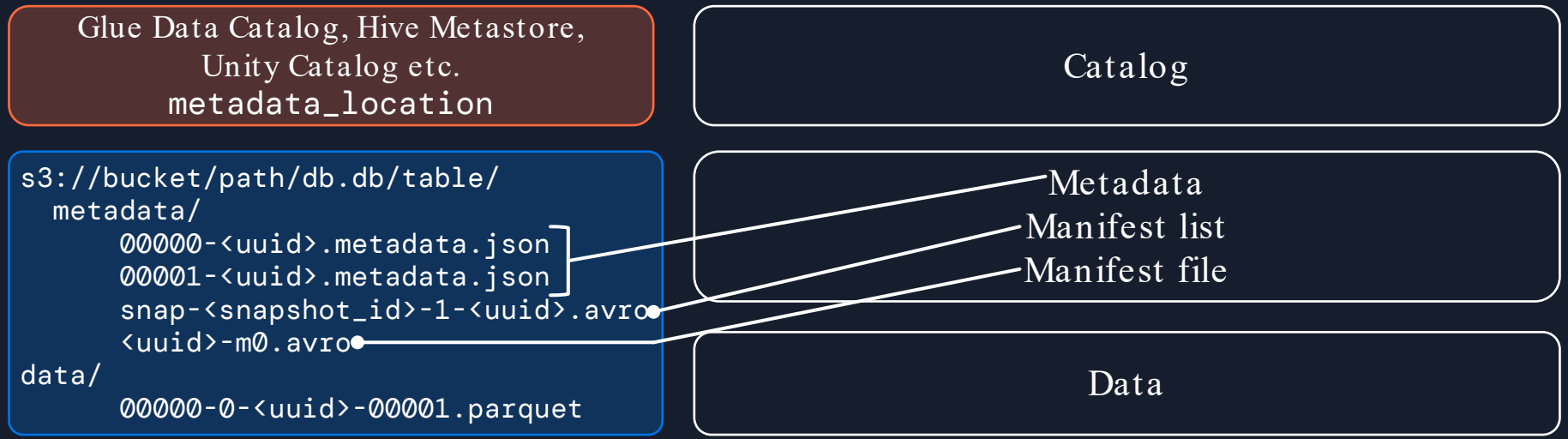
Delta column mapping need to be set to `true`

For Iceberg, **Unity Catalog on Databricks** or **Hive Metastore** must be configured as its catalog

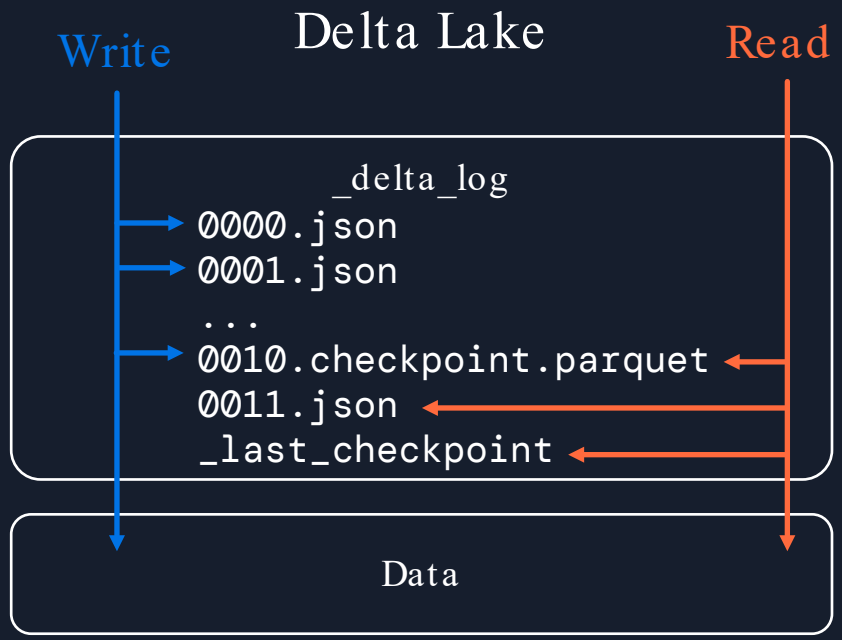
For Hudi, **OSS Delta Lake 3.2.0** is needed and it's in preview (as of 2024 May 28th)

Iceberg table architecture

Iceberg table has 3 layers; **Catalog**, **Metadata** and **Data** layers.



Delta and Iceberg table architecture



UniForm on Glue Data Catalog

UniForm for Hudi

Current UniForm for Iceberg uses Hive Catalog for Hive Metastore

UniForm for Iceberg with Glue Catalog enables UniForm on Glue Data Catalog

[Spark] Support Glue catalog for iceberg tables using UniForm

https://github.com/delta-io/delta/pull/2922