INDIAN DATA CLUB

CODE BASICS

databricks

14 DAYS

AI CHALLENGE

DAY 08

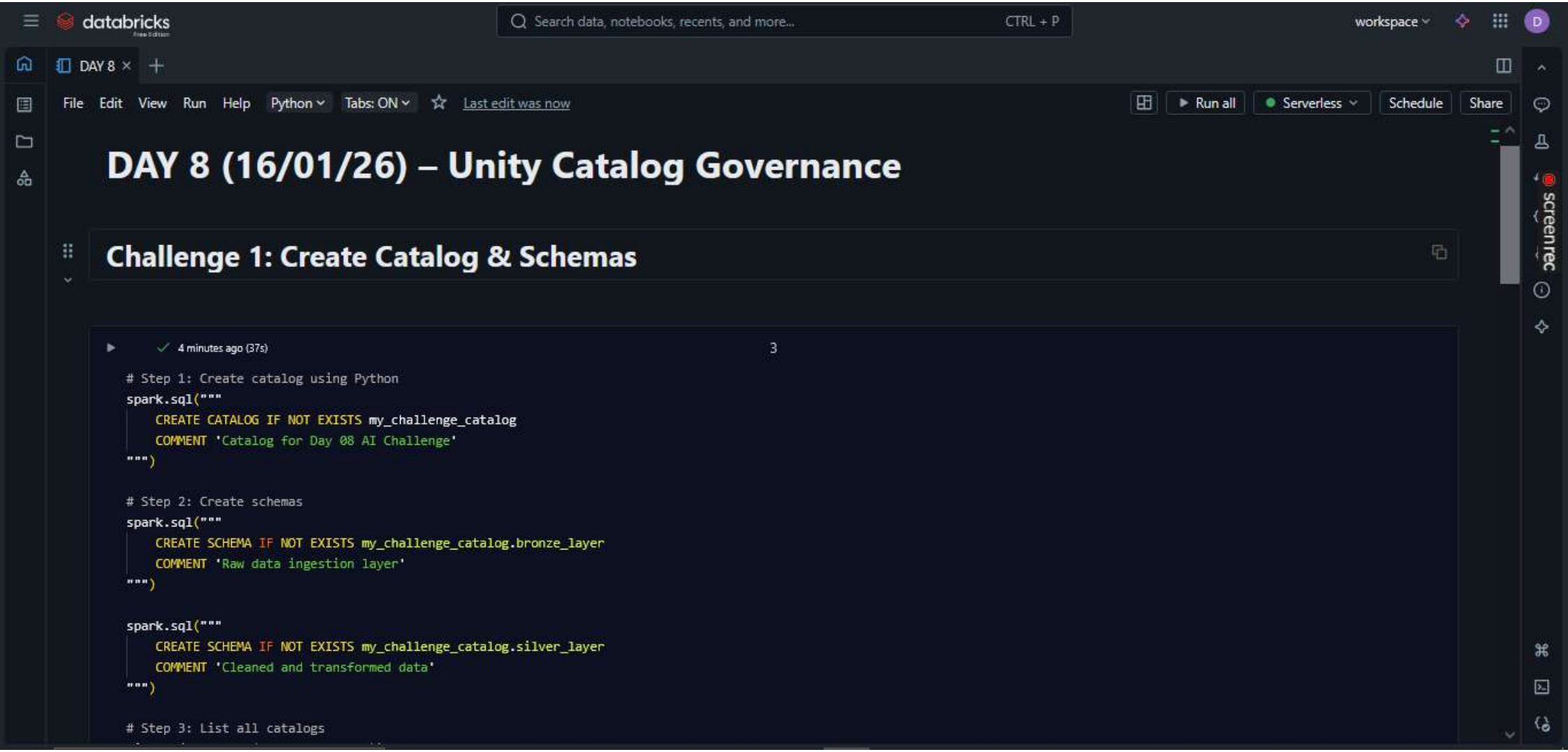
Topic:

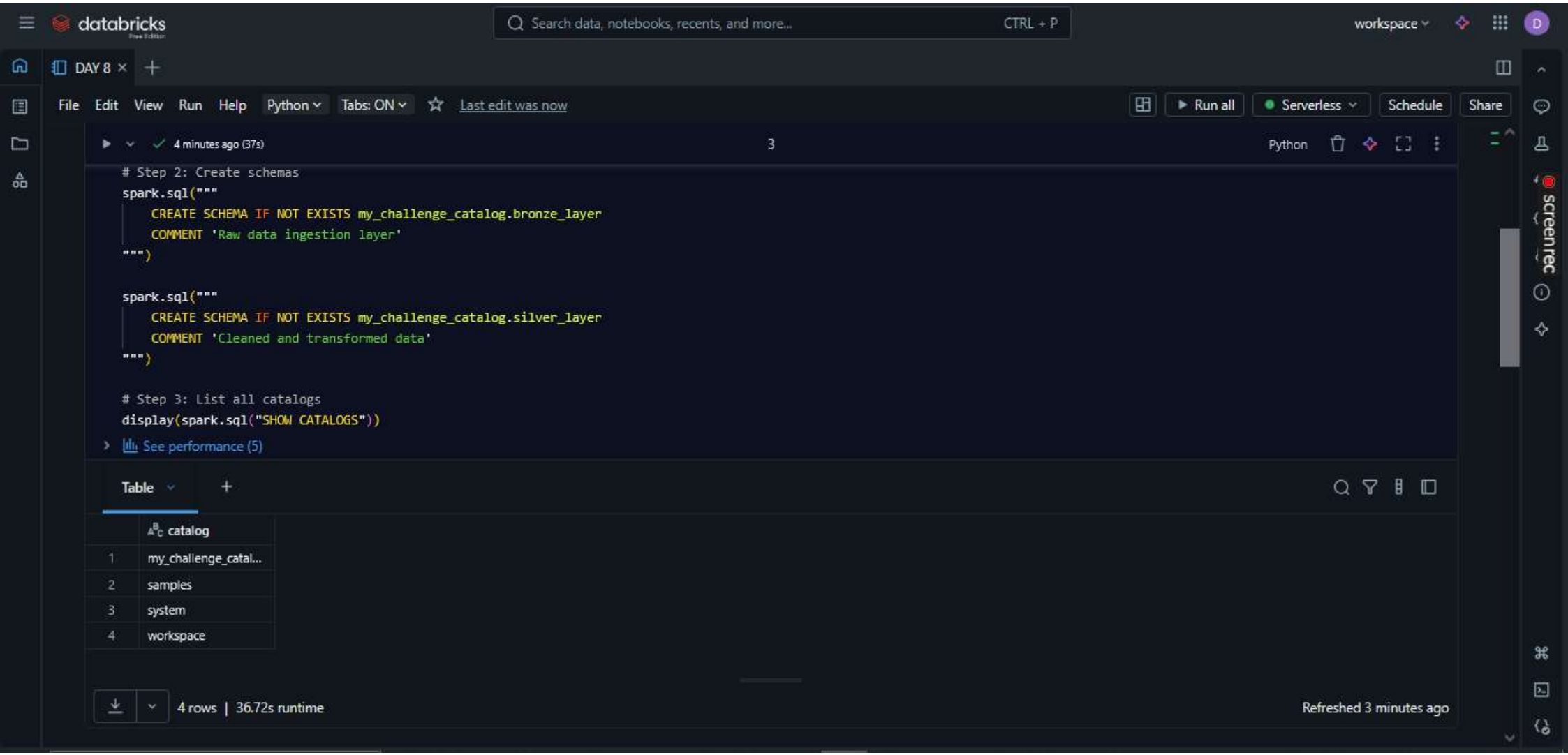
Unity Catalog Governance

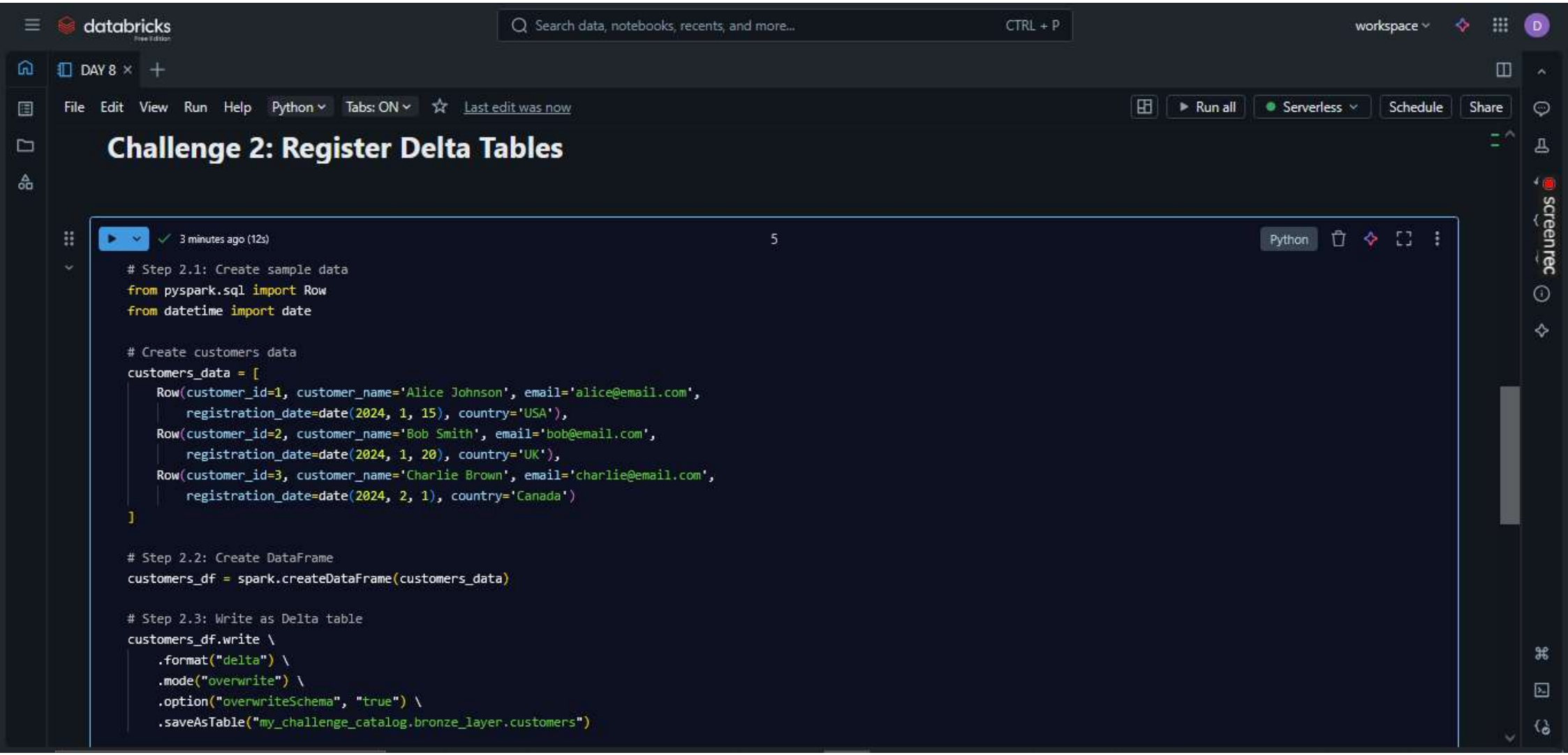
Challenge:

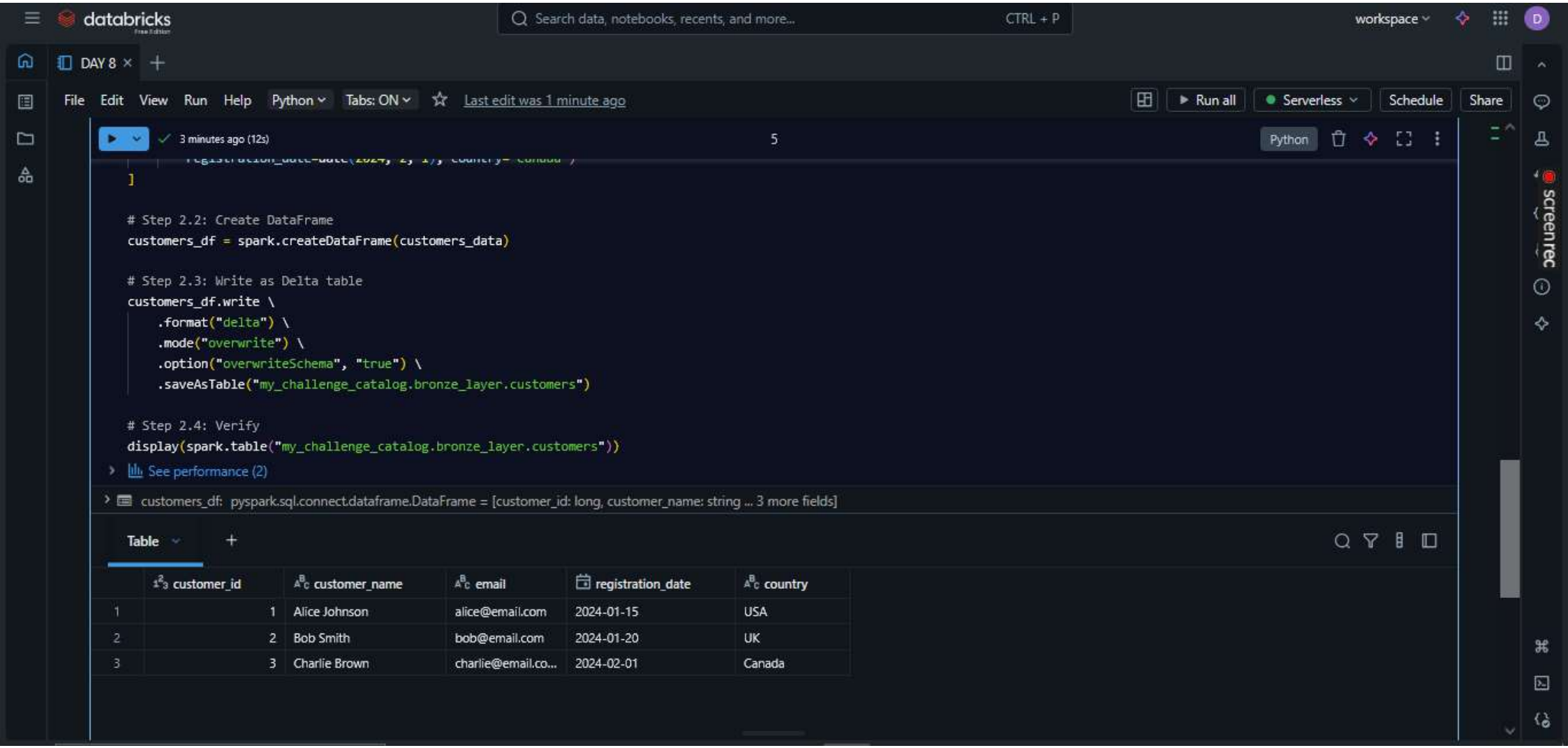
1. Create catalog & schemas
2. Register Delta tables
3. Set up permissions
4. Create views for controlled access

#DatabricksWithIDC


databricks
Search data, notebooks, recents, and more...
CTRL + P
workspace
DAY 8
File Edit View Run Help Python Tabs: ON Last edit was now
Run all
Serverless
Schedule
Share
DAY 8 (16/01/26) – Unity Catalog Governance
Challenge 1: Create Catalog & Schemas
4 minutes ago (37s)
3
Step 1: Create catalog using Python
spark.sql("""
CREATE CATALOG IF NOT EXISTS my_challenge_catalog
COMMENT 'Catalog for Day 08 AI Challenge'
""")
Step 2: Create schemas
spark.sql("""
CREATE SCHEMA IF NOT EXISTS my_challenge_catalog.bronze_layer
COMMENT 'Raw data ingestion layer'
""")
spark.sql("""
CREATE SCHEMA IF NOT EXISTS my_challenge_catalog.silver_layer
COMMENT 'Cleaned and transformed data'
""")
Step 3: List all catalogs

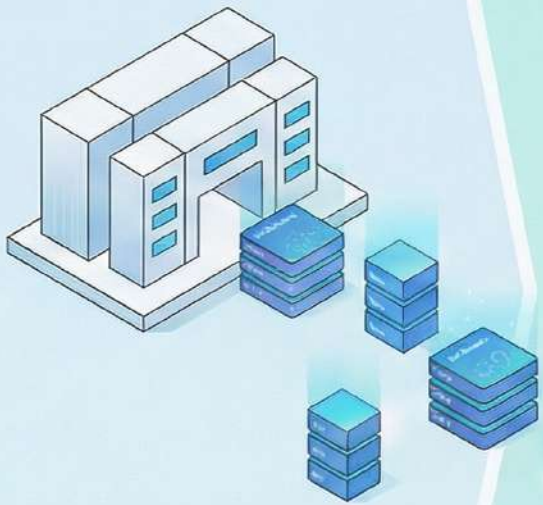

databricks
Search data, notebooks, recents, and more... CTRL + P
workspace
DAY 8
File Edit View Run Help Python Tabs: ON Last edit was now
Run all Serverless Schedule Share
4 minutes ago (37s)
3
Python
Step 2: Create schemas
spark.sql("""
CREATE SCHEMA IF NOT EXISTS my_challenge_catalog.bronze_layer
COMMENT 'Raw data ingestion layer'
""")
spark.sql("""
CREATE SCHEMA IF NOT EXISTS my_challenge_catalog.silver_layer
COMMENT 'Cleaned and transformed data'
""")
# Step 3: List all catalogs
display(spark.sql("SHOW CATALOGS"))
See performance (5)
Table
catalog
1 my_challenge_catal...
2 samples
3 system
4 workspace
4 rows | 36.72s runtime
Refreshed 3 minutes ago
screenrec


databricks
Search data, notebooks, recents, and more... CTRL + P
workspace
DAY 8
File Edit View Run Help Python Tabs: ON Last edit was now
Run all Serverless Schedule Share
Challenge 2: Register Delta Tables
3 minutes ago (12s)
5
Python
# Step 2.1: Create sample data
from pyspark.sql import Row
from datetime import date
# Create customers data
customers_data = [
Row(customer_id=1, customer_name='Alice Johnson', email='alice@email.com',
registration_date=date(2024, 1, 15), country='USA'),
Row(customer_id=2, customer_name='Bob Smith', email='bob@email.com',
registration_date=date(2024, 1, 20), country='UK'),
Row(customer_id=3, customer_name='Charlie Brown', email='charlie@email.com',
registration_date=date(2024, 2, 1), country='Canada')
]
# Step 2.2: Create DataFrame
customers_df = spark.createDataFrame(customers_data)
# Step 2.3: Write as Delta table
customers_df.write \
.format("delta") \
.mode("overwrite") \
.option("overwriteSchema", "true") \
.saveAsTable("my_challenge_catalog.bronze_layer.customers")


databricks
Search data, notebooks, recents, and more... CTRL + P
workspace
DAY 8
File Edit View Run Help Python Tabs: ON Last edit was 1 minute ago
Run all Serverless Schedule Share
3 minutes ago (12s)
5
Python
]
# Step 2.2: Create DataFrame
customers_df = spark.createDataFrame(customers_data)
# Step 2.3: Write as Delta table
customers_df.write \
.format("delta") \
.mode("overwrite") \
.option("overwriteSchema", "true") \
.saveAsTable("my_challenge_catalog.bronze_layer.customers")
# Step 2.4: Verify
display(spark.table("my_challenge_catalog.bronze_layer.customers"))
See performance (2)
customers_df: pyspark.sql.connect.dataframe.DataFrame = [customer_id: long, customer_name: string ... 3 more fields]
Table
customer_id
customer_name
email
registration_date
country
1 1 Alice Johnson alice@email.com 2024-01-15 USA
2 2 Bob Smith bob@email.com 2024-01-20 UK
3 3 Charlie Brown charlie@email.co... 2024-02-01 Canada
Screenrec

Databricks 14-Day AI Challenge: Day 8

Unity Catalog Governance



1. Create Catalog & Schemas

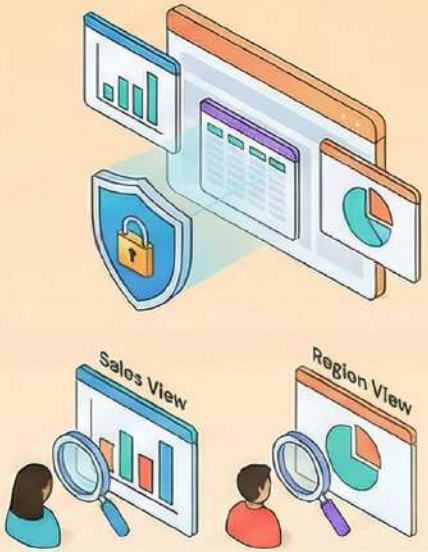
Establish the foundational structure for organizing your data assets.

2. Register Delta Tables

Make your existing Delta Lake tables discoverable and governable.

3. Set Up Permissions

Define and apply access control rules for users and groups.



4. Create Views for Controlled Access

Build secure views to expose specific subsets of data to users.

NotebookLM

INDIAN DATA CLUB databricks ODE BASICS

14 DAYS AI CHALLENGE

DAY 08

Unity Catalog Governance: Building Your Secure Data Fortress

#DatabricksWithIDC

NotebookLM

Without Governance, Data Becomes a Liability

Before we build, let's understand the landscape. An ungoverned data lake often becomes a "data swamp"—a place where valuable information is hard to find, trust, and secure.

Security Risks

Sensitive data is exposed and unmonitored.

Poor Discovery

Teams can't find the data they need.

Low Trust

Inconsistent and duplicated data erodes confidence.

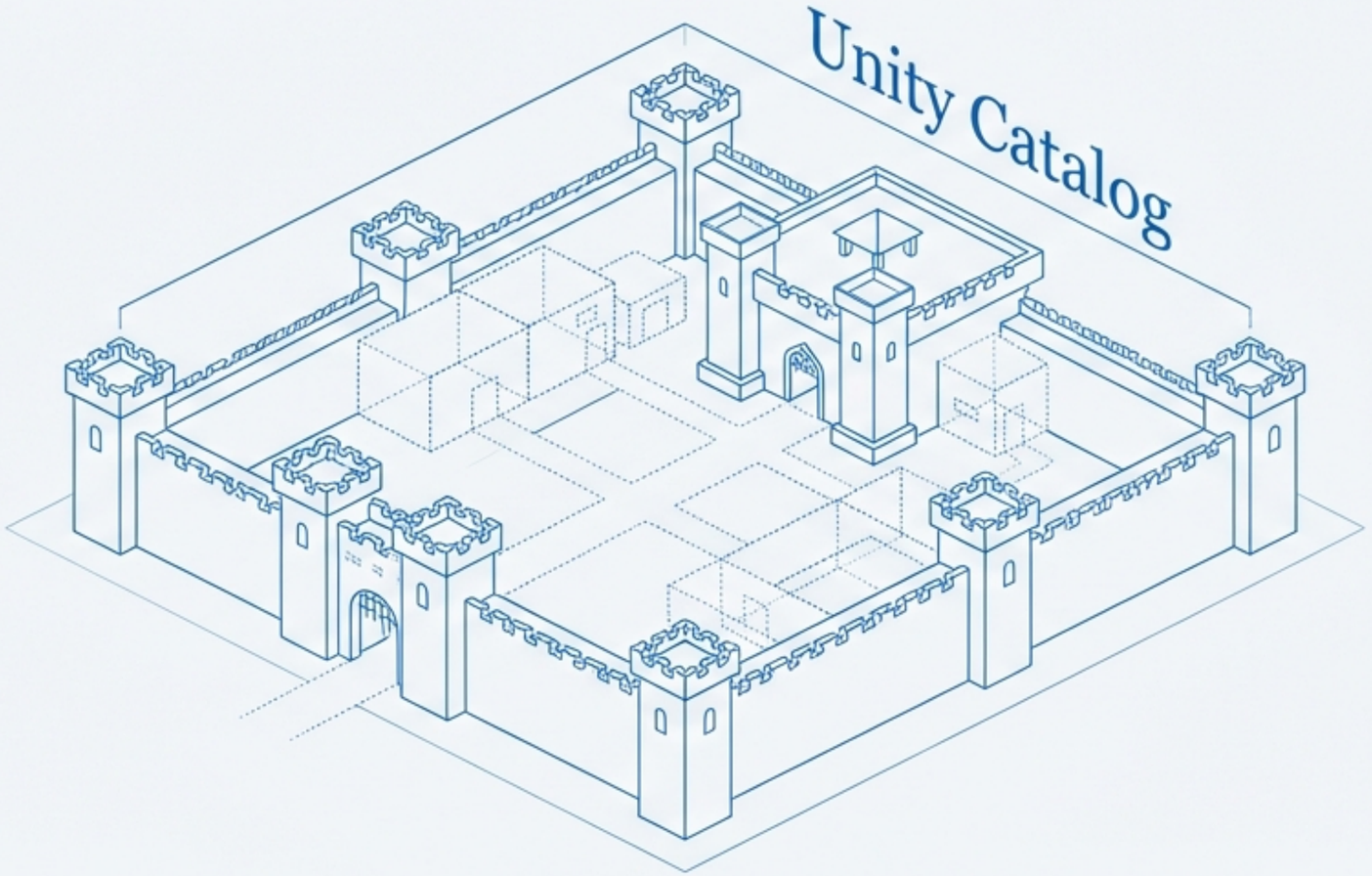
Compliance Hurdles

Auditing access and proving compliance is nearly impossible.

NotebookLM

Our Goal: Constructing a Secure Data Fortress

Our mission is to transform the data swamp into a fortress. A governed environment where data is not just stored, but organized, protected, and made accessible to the right people for the right reasons. This is the foundation for reliable analytics and AI.



NotebookLM

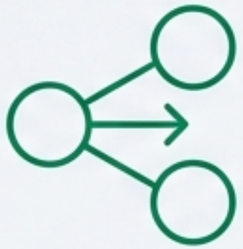
The Blueprint: Databricks Unity Catalog

Unity Catalog is the fine-grained governance solution for data and AI on the Databricks Lakehouse Platform. It provides a single, centralized place to manage all of your data assets.

Unified Governance: One place to govern files, tables, and machine learning models.

Centralized Access Control: Consistent permissions across all workspaces.

Built-in Auditing & Lineage: Automatically capture user actions and track how data is created and used.



Secure Data Sharing: Share live data with other organizations using Delta Sharing.

NotebookLM

The 4-Step Plan to Build Your Fortress

Today’s challenge is a 4-step construction plan. Each step is a critical component in building a robust and secure data governance foundation.

1. Lay the Foundation

Create catalogs & schemas.

2. Secure the Assets

Register Delta tables.

3. Distribute the Keys

Set up permissions.

4. Install Secure Windows

Create views for controlled access.

NotebookLM

Step 1: Lay the Foundation with Catalogs & Schemas

- **Catalogs** are the main walls of your fortress.

- **Schemas** (or databases) are the individual rooms within those walls.

```
-- Create a new catalog to house our assets
CREATE CATALOG IF NOT EXISTS idc_challenge;

-- Create a schema within the catalog for curated data
CREATE SCHEMA IF NOT EXISTS idc_challenge.governed_data;
```

Why It Matters

The 3-level namespace (`catalog.schema.table`) provides a logical, unambiguous hierarchy for all your data. This prevents name collisions and makes data discovery intuitive and scalable across the entire organization.

NotebookLM

Step 2: Secure the Assets by Registering Delta Tables

With the structure in place, it's time to bring your valuable assets—your data—inside the protected walls of the fortress. Registering a table places it under the governance of Unity Catalog.

```
-- Create a managed table inside our new schema
-- Unity Catalog will now manage its lifecycle and data files
CREATE TABLE idc_challenge.governed_data.customer_profiles (
  customer_id STRING,
  email STRING,
  country STRING
);
```

Why It Matters

Once a table is registered in Unity Catalog, it's no longer just a collection of files in cloud storage. It becomes a first-class, securable object that can be audited, tracked with lineage, and have its access controlled centrally.

NotebookLM

Step 3: Distribute the Keys by Setting Permissions

A fortress isn’t secure if everyone has the master key. Unity Catalog allows you to grant specific permissions—the keys and access badges—to the right users and groups for the right assets.

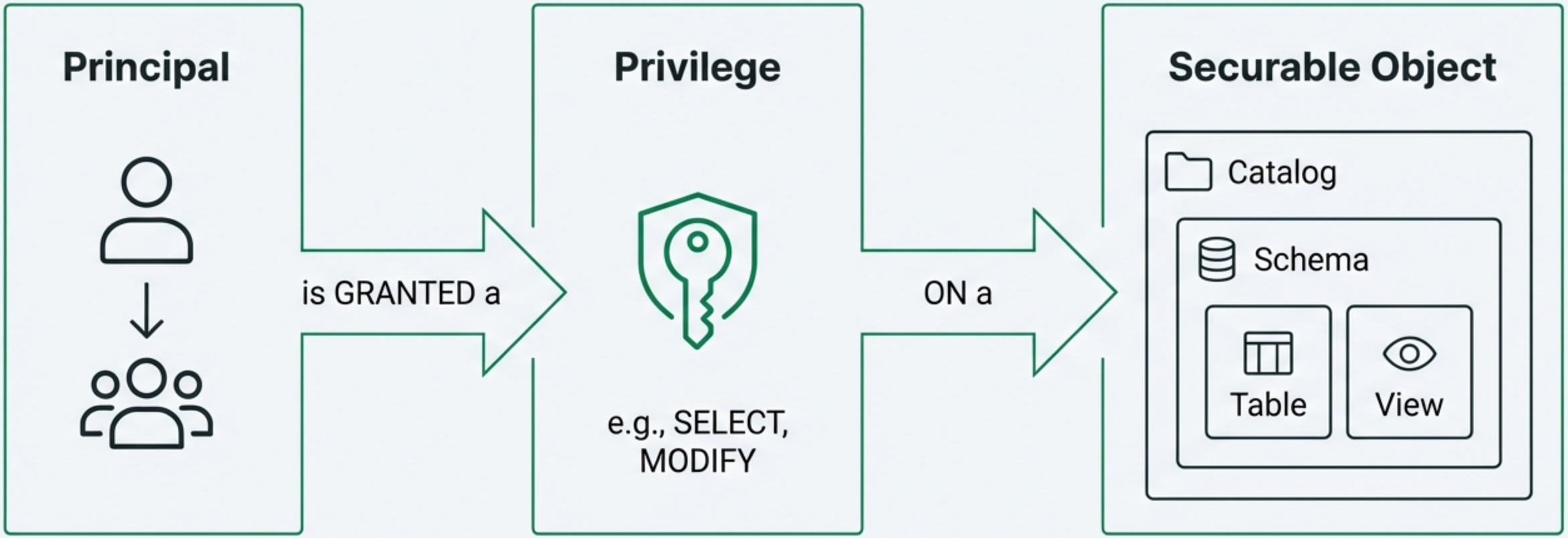
```
-- Grant read-only access to the analytics team
GRANT SELECT ON TABLE idc_challenge.governed_data.customer_profiles
TO `analytics_team`;

-- Grant a specific user rights to modify the table
GRANT MODIFY ON TABLE idc_challenge.governed_data.customer_profiles
TO `data_steward@company.com`;
```

Why It Matters

This enforces the principle of least privilege, a cornerstone of data security. It ensures users can only perform actions they are explicitly authorized to do, drastically reducing the risk of accidental data deletion or unauthorized access.

NotebookLM

The Unity Catalog Access Control Model



Permissions are inherited down the hierarchy. Granting a privilege on a catalog grants it on all schemas and tables within it. This makes managing permissions at scale simple and powerful.

NotebookLM

Step 4: Install Secure Windows with Views

Sometimes, you don’t want to give someone the key to a whole room, just a secure window to see specific items inside. Views allow you to create a pre-defined, filtered, and often simplified look at your data.

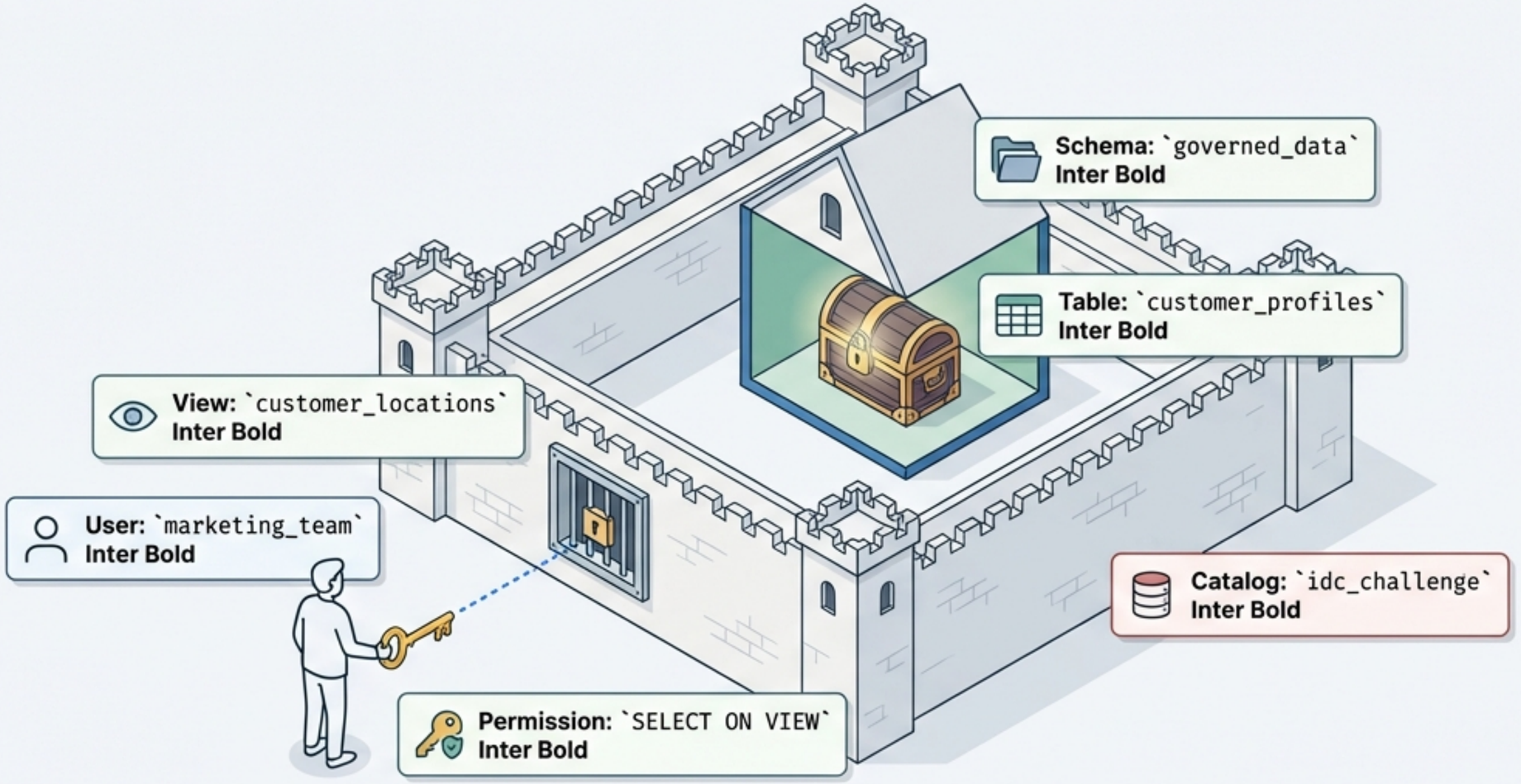
```
-- Create a view that hides Personally Identifiable Information (PII)
CREATE VIEW idc_challenge.governed_data.customer_locations AS
  SELECT customer_id,eoge.governed_data.customer_locations AS
  FROM idc_challenge.governed_data.customer_profiles;

-- Now, grant access to the view, not the underlying table
GRANT SELECT ON VIEW idc_challenge.governed_data.customer_locations
TO `marketing_team`;
```

Why It Matters

Views are a critical tool for row-level and column-level security. They allow you to abstract away complexity and protect sensitive columns (like PII) while still empowering users with the data they need to do their jobs.

NotebookLM

Your Data Fortress is Complete

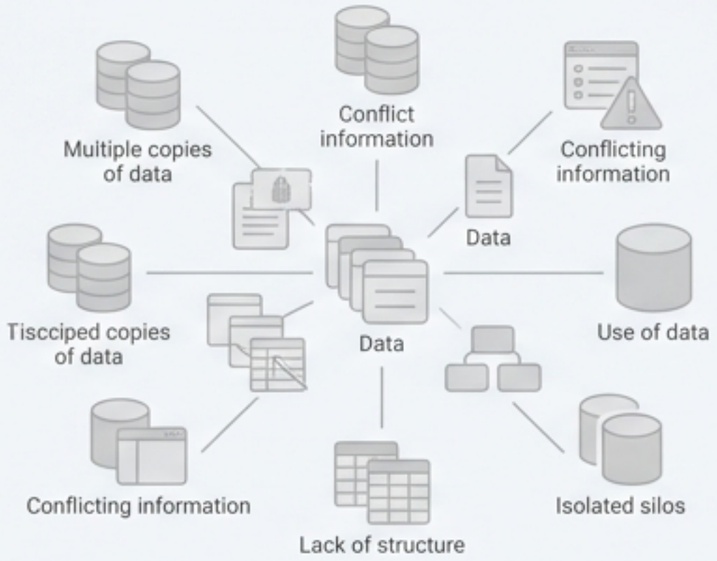


You have successfully used the four core components of Unity Catalog to build a secure, organized, and governed data asset.

NotebookLM

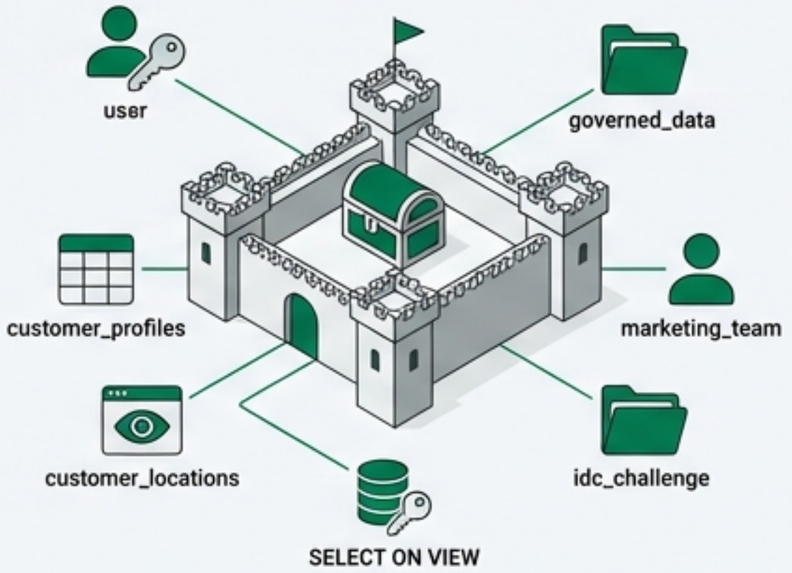
The Transformation: From Data Chaos to Governed Clarity

The Data Swamp



Risky Disorganized Untrusted Siloed

The Governed Fortress



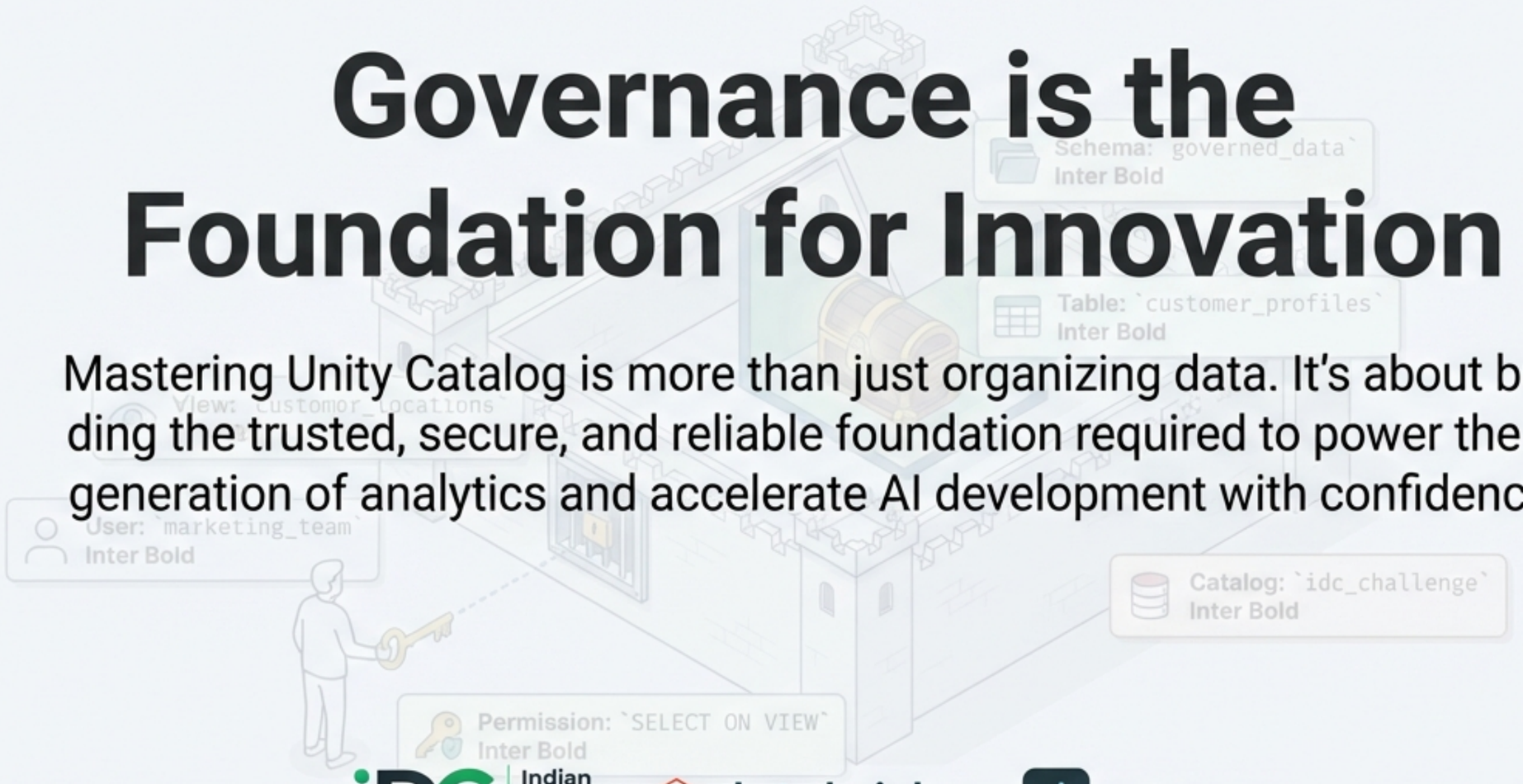
Secure Discoverable Reliable Unified

By implementing governance, you've transformed a liability into a strategic asset ready for high-impact analysis and AI.

NotebookLM

Governance is the Foundation for Innovation

Mastering Unity Catalog is more than just organizing data. It's about building the trusted, secure, and reliable foundation required to power the next generation of analytics and accelerate AI development with confidence.



Indian Data Club

databricks

Code Basics

#DatabricksWithIDC

NotebookLM