**Unity Catalog:**

Unity Catalog is a unified governance solution provided by Databricks for managing and securing access to all data and AI assets across multiple workspaces and environments.  
It offers a **centralized metadata management system** that integrates with data lakes, data warehouses, and AI models, enabling organizations to enforce consistent security, auditing, and lineage tracking across their data estate.

Unity Catalog is especially beneficial in multi-cloud or large-scale deployments because it centralizes:

* Access control
* Data discovery
* Data lineage
* Compliance management

It replaces the traditional workspace-specific Hive metastore with a **single account-level metastore**, which can be shared across multiple Databricks workspaces.

**Key Features of Unity Catalog:**

1. **Centralized Metadata Management**  
   Stores and manages metadata for all catalogs, schemas, and tables in a central location.
2. **Fine-Grained Access Control**  
   Supports permissions at the catalog, schema, table, view, and column levels.
3. **Data Lineage**  
   Automatically tracks how data flows and transforms across queries, jobs, and dashboards.
4. **Audit Logging**  
   Provides audit logs for compliance and security monitoring.
5. **Multi-Workspace Sharing**  
   A single metastore can serve multiple Databricks workspaces.
6. **Support for All Data Assets**  
   Manages not only tables but also files, AI models, functions, and dashboards.

**Creating a Unity Catalog Metastore:**

The **metastore** in Unity Catalog is the top-level container for all data assets and metadata.  
Unlike the default Hive metastore (workspace-specific), the Unity Catalog metastore is **account-level** and can be shared by multiple workspaces.

**1. Prerequisites**

* Databricks account with **Account Admin** privileges.
* Workspace in the **Premium** tier or higher.
* Cloud storage configured:
  + **AWS:** S3 bucket
  + **Azure:** ADLS Gen2
  + **GCP:** Google Cloud Storage bucket
* Storage credentials and access policies configured for Databricks.

**2. Steps to Create a Metastore**

1. **Log in to Databricks Account Console**
2. **Navigate to Unity Catalog → Metastores**  
   In the left panel, click on **Data** and select **Metastores**.
3. **Click "Create Metastore"** and provide:
   * **Name**: e.g., main\_metastore
   * **Storage Root**: Cloud storage path for managed tables  
     Example:
     + AWS: s3://my-databricks-bucket/unitycatalog
     + Azure: abfss://mycontainer@mystorageaccount.dfs.core.windows.net/unitycatalog
     + GCP: gs://my-databricks-bucket/unitycatalog
   * **Region**: Must match the region of your workspace.
4. **Click "Create"** to save the metastore.

**Enabling Unity Catalog for a Workspace:**

Once the metastore is created, it must be assigned to a Databricks workspace.

**1. Assign Workspace to the Metastore**

1. In the **Account Console**, go to **Workspaces**.
2. Select the desired workspace.
3. Click **Assign Metastore**.
4. Select the created metastore (e.g., main\_metastore).
5. Choose a **Default Catalog** (usually main).
6. Save the changes.

**2. Enable Unity Catalog**

1. Go to **Workspace Admin Settings** in the Databricks workspace.
2. Navigate to **Unity Catalog** settings.
3. Ensure Unity Catalog is enabled and users have **USE CATALOG**, **USE SCHEMA**, and **CREATE** permissions where required.

**Unity Catalog 3-Level Namespace Structure:**

catalog.schema.table

Components:

* **Catalog**:  
  The top-level container, typically representing an organization, environment (e.g., dev/test/prod), or data domain.
* **Schema**:  
  Logical grouping of tables and views within a catalog. Similar to a database in traditional systems.
* **Table/View**:  
  The actual dataset or query result.

EXAMPLE:

SELECT \* FROM sales\_catalog.marketing\_data.customer\_info;

**Creating Unity Catalog Objects:**

**Create a Catalog -** CREATE CATALOG sales\_catalog;

**Create a Schema -** CREATE SCHEMA sales\_catalog.marketing\_data;

**Create a Table -** CREATE TABLE sales\_catalog.marketing\_data.customer\_info ( customer\_id INT,name STRING,email STRING);

**Grant Permissions -** GRANT SELECT ON TABLE sales\_catalog.marketing\_data.customer\_info TO `analyst\_group`;

**Advantages of Unity Catalog:**

Central Governance — Manage all data assets from one place.

Improved Security — Granular access control down to the column level.

Scalability — Works across multiple workspaces and regions.

Auditability — Built-in audit logs for compliance.

Data Discovery — Easier to search and understand available datasets.

Unity Catalog is a powerful governance solution that simplifies and secures data management within Databricks. By centralizing metadata, permissions, and lineage tracking, it enables organizations to have consistent data governance across multiple workspaces.  
Its 3-level namespace provides a clear and structured way to organize datasets, making it easier for teams to manage large-scale analytics environments.