

14.8.25

Unity Catalog – Assignment

1. Introduction

Unity Catalog is Databricks' centralized data governance solution for all data and AI assets. It enables organizations to manage permissions, track data lineage, and ensure compliance across multiple Databricks workspaces.

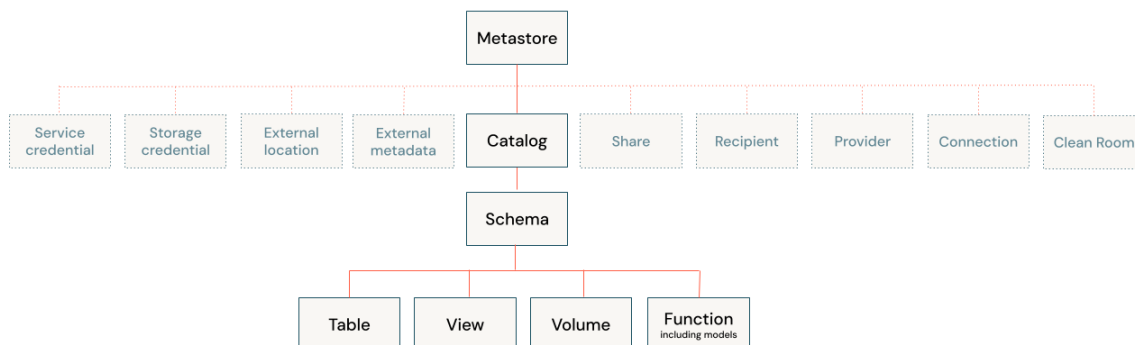
2. Objectives

- Centralize governance across data platforms.
- Provide fine-grained access control.
- Enable end-to-end data lineage tracking.
- Improve security and compliance with audit logs.

3. Architecture Overview

Unity Catalog follows a 3-level namespace:

1. Metastore – Top-level container for all catalogs.
2. Catalog – Logical grouping of schemas (databases).
3. Schema – Contains tables and views.
4. Tables/Views – The actual data objects.



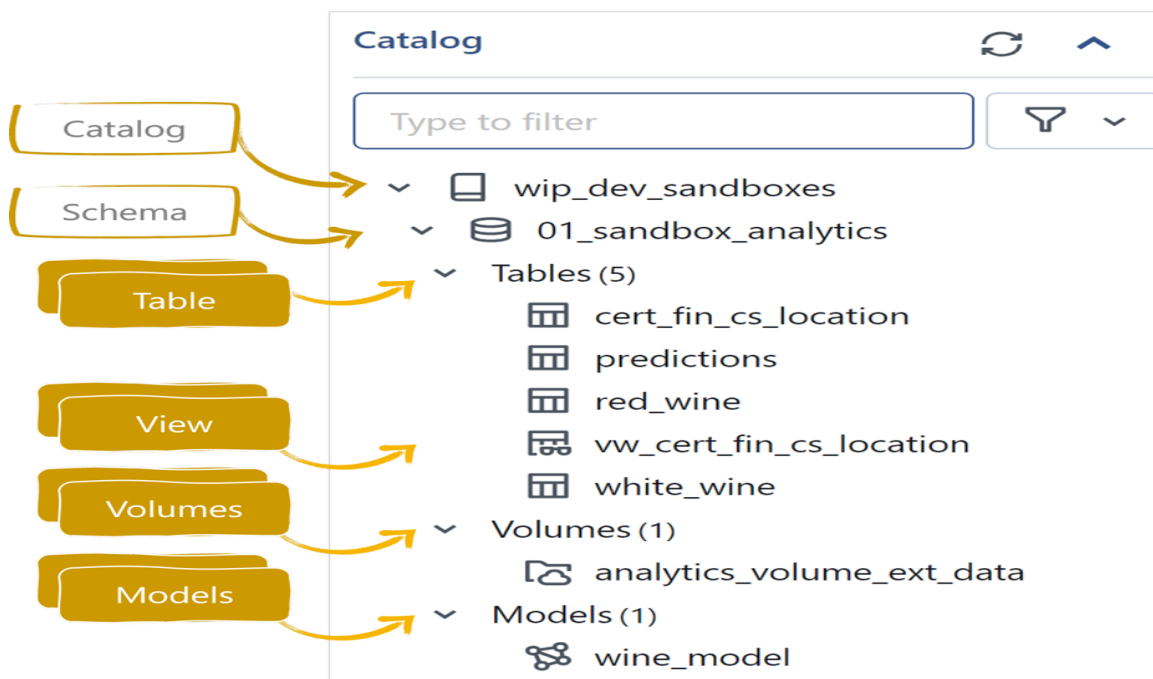
4. Key Features

- Centralized Governance – One place to manage permissions across all workspaces.
- Data Lineage – Track data usage and transformations.
- Secure Sharing – Share data securely with other accounts.

- Attribute-Based Access Control – Control access based on user attributes.
- Audit Logging – Record all access and permission changes.

5. Unity Catalog Components

Component	Description
Metastore	Holds catalogs and schemas; assigned to workspaces.
Catalog	Top-level organization for schemas.
Schema	Organizes tables and views.
Table/View	Stores actual data and queries.
External Location	Links to external storage like ADLS, S3, or GCS.
Storage Credential	Secure authentication method to access storage.



6. Workflow of Unity Catalog

1. Create a Metastore.
2. Configure Storage Credentials.
3. Define External Locations.
4. Create Catalogs → Schemas → Tables.

5. Assign permissions to users or groups.
6. Track data lineage and audit logs.

7. SQL Commands in Unity Catalog

-- Create a catalog

```
CREATE CATALOG sales_catalog;
```

-- Create a schema

```
CREATE SCHEMA sales_schema;
```

-- Create a table

```
CREATE TABLE sales_schema.orders (  
  order_id INT,  
  customer STRING,  
  amount DOUBLE  
);
```

-- Grant permissions

```
GRANT SELECT ON TABLE sales_schema.orders TO `data_analyst`;
```

8. Best Practices

- Use groups instead of individual users for permissions.
- Keep catalogs organized by department or project.
- Use service principals for automated workloads.
- Enable audit logs for security compliance.

9. Conclusion

Unity Catalog centralizes and simplifies data governance in Databricks. It ensures security, compliance, and visibility for all data and AI assets while supporting multi-cloud and multi-workspace environments.