# 🔹 Unity Catalog in Databricks - TheFutue of Data Governance 🔹

## What is Unity Catalog?

Unity Catalog is Databricks’ centralized governance solution for managing data assets, permissions, and lineage across multiple workspaces and cloud providers. It provides fine-grained access control at the catalog, schema, and table levels without needing to rely on external metastore configurations.

## Key Features of Unity Catalog:

✅ Centralized Data Governance – Single metastore across all workspaces.

✅ Column & Row-Level Security – Define granular access control at the column/row level.

✅ Data Lineage Tracking – Automatically tracks table lineage.

✅ Cross-Workspace Data Access – No need for workspace-specific metastore setups.

✅ Simplified Access Control – Uses RBAC (Role-Based Access Control) instead of ACLs.

## How Unity Catalog Fits into Medallion Architecture

🛠 Medallion Architecture is a layered approach for structuring a data lakehouse into three tiers:

🔹 Bronze Layer (Raw Data)

Stores raw, unprocessed data.

Typically ingested from external sources (APIs, logs, databases).

In Unity Catalog, raw data can be managed as external tables within a catalog.

🔹 Silver Layer (Cleaned & Processed)

Data is filtered, deduplicated, and enriched.

Unity Catalog provides schema enforcement and governance to maintain data integrity.

🔹 Gold Layer (Aggregated & Analytics Ready)

Data is aggregated and optimized for business insights.

Unity Catalog ensures secure access and fine-grained controls on sensitive datasets.

📌 Without Unity Catalog: You manually manage paths (dbfs:/mnt/...) for every table, creating risks of inconsistent permissions.

📌 With Unity Catalog: You manage tables using logical names, making governance easier while keeping a single source of truth.

**Unity Catalog vs. Non-Unity Catalog in Data Engineering**

| Feature | Without Unity Catalog (Legacy Metastore) | With Unity Catalog |
|---|---|---|
| Access Control | Managed via table ACLs (workspace-specific) | Centralized RBAC (catalog-level) |
| Data Governance | No native data lineage tracking | Full data lineage tracking |
| Data Discovery | No cross-workspace visibility | Unified metadata & searchability |
| Multi-Cloud Support | Separate metastore per cloud | Unified governance across clouds |
| Table References | Must use storage paths (s3://...) | Use only table names (catalog.schema.table) |

## Why Do We Remove File Paths and Use Only Table Names?

Before Unity Catalog, we had to specify file paths or metastore-dependent table paths, which made workspace migration difficult.

🔹 Without Unity Catalog:

```
SELECT * FROM delta.`s3://data-lake/bronze/table_name`
```

🔹 With Unity Catalog:

```
SELECT * FROM my_catalog.bronze.table_name
```

✅ Benefits:

✔ No Hardcoded Paths – Improves portability & eliminates path-based dependencies.

✔ Better Governance – Access is controlled at the catalog/schema level.

✔ Easier Collaboration – Teams across workspaces can reference tables consistently.

## How to Initialize Unity Catalog in Databricks Notebook?

Step : ***Enable Unity Catalog in Your Databricks Environment***

Admins must configure Unity Catalog at the account level.

Assign metastore and catalog-level permissions.

Step : **Set Up the Catalog and Schemas**

Run the following SQL to create a new catalog and a schema inside Unity Catalog:

```
-- Create a catalog (logical grouping of schemas)
CREATE CATALOG my_catalog;
-- Grant access to a user or group
GRANT USAGE ON CATALOG my_catalog TO `engineering_team`;
-- Create a schema inside the catalog
CREATE SCHEMA my_catalog.bronze;
```

Step : **Create and Query Tables**

```
-- Create a Delta table inside Unity Catalog
CREATE TABLE my_catalog.bronze.customer_data (
    customer_id INT,
    name STRING,
    purchase_amount DOUBLE
) USING DELTA;

-- Insert sample data
INSERT INTO my_catalog.bronze.customer_data VALUES (1, 'Alice', 100.50);

-- Query the table
SELECT * FROM my_catalog.bronze.customer_data;
```

Step : **Register External Tables (For data already in object storage)**

```
CREATE EXTERNAL TABLE my_catalog.bronze.sales_data

USING DELTA

LOCATION 's3://my-bucket/sales_data/';
```

## Final Thoughts

✅ Unity Catalog brings a unified governance model, eliminating the need for workspace-level metastores.

✅ Medallion Architecture integrates seamlessly, enabling better access control, schema enforcement, and data lineage.

✅ Data Engineering Pipelines become simpler, as you reference data using table names instead of file paths.

🔹