Databricks Unity Catalog — Architecture, Data Governance & Access Control Guide

A Light■Theme Illustrated PDF covering architecture, ABAC policies, fine■grained access control (row/column), SQL examples, and interview Q&A; for data governance professionals.

■ Unity Catalog in Data Governance Framework

```
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
■         Data Governance         ■
■  ■■■■■■■■■■■■■■■■■■■■■■■■■■■  ■
■  ■ Unity Catalog (Control) ■  ■
■  ■ ■■ Access Control       ■  ■
■  ■ ■■ Lineage & Audit      ■  ■
■  ■ ■■ Metadata & Tags      ■  ■
■  ■ ■■ Compliance Policies  ■  ■
■  ■■■■■■■■■■■■■■■■■■■■■■■■■■■  ■
■       ↓          ↓              ■
■   Security   Trust & Sharing   ■
■   (ABAC, RLS, CLS)              ■
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
```

■■ Unity Catalog Architecture Overview

Unity Catalog serves as the centralized governance layer for all Databricks data and AI assets. It enforces uniform access control, lineage, and auditing across multiple workspaces and cloud environments.

```
                   ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
                   ■        Metastore (UC)       ■
                   ■  - Metadata, Policies       ■
                   ■  - Tags & ACLs              ■
                   ■  - Lineage & Audit Logs     ■
                   ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
                                ■
                  ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
                  ■     Databricks Workspaces    ■
                  ■ (Compute Clusters, Jobs)     ■
                  ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
                                ■
        ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
        ■              ■                  ■               ■
    Catalog A      Catalog B          Catalog C      Shared Catalog
(e.g. Finance)  (e.g. Marketing)  (e.g. ML Models)  (Delta Sharing)
```

■ Three■Level Hierarchy

• **Catalog** — top■level container representing a data domain (e.g., finance).

• **Schema (Database)** — logical grouping of related tables/views.

• **Table / View / Function** — actual data object with policies and lineage.

■ Privilege Model

```
GRANT SELECT ON TABLE finance.curated.transactions TO analyst_group;
REVOKE MODIFY ON TABLE finance.curated.transactions FROM readonly_role;
```

■ Fine■Grained Access Control (Row & Column Level)

Unity Catalog introduces native SQL syntax for row filters and column masks to enforce security at query time.

```
-- Row Filter Example
CREATE OR REPLACE ROW FILTER region_filter
AS (region = current_user_region())
ON TABLE finance.curated.sales;

-- Column Mask Example
CREATE OR REPLACE COLUMN MASK ssn_mask
AS (CASE WHEN is_role_in('hr_admin') THEN ssn ELSE 'XXX-XX-XXXX' END)
ON TABLE hr.employee_data (ssn);
```

■■ Attribute■Based Access Control (ABAC)

ABAC allows dynamic access rules using governed tags and user attributes such as department or sensitivity level.

```
-- Example Policy (conceptual)
IF user.department = 'Finance' AND column.tag = 'PII' THEN MASK
ELSE GRANT SELECT
```

■ Implementation Example — End■to■End Governance

```
-- Define Groups
CREATE GROUP finance_users;
CREATE GROUP hr_admins;

-- Create Table
CREATE TABLE finance.curated.sales (
    region STRING,
    amount DOUBLE,
    ssn STRING
);

-- Apply Column Mask & Row Filter
CREATE COLUMN MASK ssn_mask
AS (CASE WHEN is_member('hr_admins') THEN ssn ELSE 'XXX-XX-XXXX' END)
ON TABLE finance.curated.sales (ssn);

CREATE ROW FILTER region_filter
AS (region = current_user_region())
ON TABLE finance.curated.sales;

-- Assign Privileges
GRANT SELECT ON TABLE finance.curated.sales TO finance_users;
```

■ Data Governance & Lineage Features

- **Central Metastore** — One source of truth for metadata across all workspaces.

- **Audit Logs** — Record access, privilege changes, and data modifications.

- **Lineage Tracking** — Automatic end■to■end tracking (notebook → table → dashboard).

- **Tags & Classification** — Used for sensitivity labeling and ABAC enforcement.

- **Delta Sharing Integration** — Secure cross■account data sharing with full access control.

■ Unity Catalog vs Hive Metastore Comparison

Feature	Unity Catalog	Hive Metastore
Scope	Cross■workspace, multi■cloud	Per■workspace
Security	Row/Column level, ABAC	Table■level only
Lineage & Audit	Built■in tracking	Manual or external tools
Governed Tags	Supported (ABAC)	Not supported
Integration	Delta Sharing, external locations	Hive■only
Deployment	Managed service in Databricks	Cluster■local metadata DB

■ Interview Q&A; Highlights

Q: What is Unity Catalog?

A: A unified governance layer in Databricks for metadata, access control, lineage, and data sharing.

Q: How does Unity Catalog implement fine■grained security?

A: Through row filters, column masks, and ABAC policies using tags and user attributes.

Q: What are ABAC tags used for?

A: To classify data (e.g., PII, Department) and dynamically enforce policies based on user attributes.

Q: How is lineage captured?

A: Automatically via the Databricks platform—every job, SQL, or notebook execution updates lineage graphs.

Q: Can Unity Catalog control access to S3/ADLS data?

A: Yes, via external locations and storage credentials tied to Unity Catalog governance.

Q: Difference between Unity Catalog and Hive Metastore?

A: Unity Catalog is centralized, secure, and fine■grained; Hive is workspace■bound and limited.

Q: How does Delta Sharing integrate with Unity Catalog?

A: Unity Catalog governs what is shared externally and logs all share access.

Q: Why is ABAC preferred for large enterprises?

A: It scales governance by using dynamic attribute■based rules instead of per■object grants.

■ Best Practices

- Use separate catalogs for each business domain (Finance, HR, Marketing).
- Tag sensitive data columns for ABAC enforcement (PII, PCI, Confidential).
- Apply column masking and row filtering instead of duplicating views.
- Assign privileges to groups, not individuals.
- Monitor lineage and audit logs for compliance reports.
- Integrate Unity Catalog with enterprise IAM and SCIM provisioning.
- Use Delta Sharing for secure, external data collaboration.