# Exercise: Mini Project Using Unity Catalog and Data Governance

## Objective:

Develop a mini project using Unity Catalog to demonstrate key data governance

capabilities such as **Data Discovery**, **Data Audit**, **Data Lineage**, and **Access Control**.

# Part 1: Setting Up the Environment

## Task 1: Create a Metastore

Set up a Unity Catalog metastore that will act as the central location to manage all catalogs and schemas.

## Task 2: Create Department-Specific Catalogs

Create separate catalogs for the following departments: Marketing

Engineering Operations

**CREATE CATALOG** Marketing;

**CREATE CATALOG** Engineering;

**CREATE CATALOG** Operations;

## Task 3: Create Schemas for Each Department

Inside each catalog, create specific schemas to store different types of data, e.g.:

ads\_data

For the Marketing catalog, create schemas such as

customer\_data .

For the Engineering catalog, create schemas such as

development\_data .

For the Operations catalog, create schemas such as

and

and

projects

and

logistics\_data

supply\_chain .

**CREATE SCHEMA** Marketing.ads\_data;

**CREATE SCHEMA** Marketing.customer\_data;

**CREATE SCHEMA** Engineering.projects;

**CREATE SCHEMA** Engineering.development\_data;

**CREATE SCHEMA** Operations.logistics\_data;

**CREATE SCHEMA** Operations.supply\_chain;

# Part 2: Loading Data and Creating Tables

## Task 4: Prepare Datasets

Use sample datasets for each schema (create CSV or JSON files if required): **Marketing - Ads Data**: Contains columns such as ad\_id , impressions , clicks , cost\_per\_click .

**Engineering - Projects**: Contains columns such as project\_id ,

project\_name , start\_date , end\_date .

**Operations - Logistics**: Contains columns such as shipment\_id , origin ,

destination , status .

## Task 5: Create Tables from the Datasets

Load the datasets into their respective schemas as tables.

ads\_data

projects

Example: Create a table for Example: Create a table for

**CREATE TABLE** Marketing.ads\_data.ads\_table (

ad\_id INT,

impressions INT,

clicks INT,

cost\_per\_click FLOAT

);

**CREATE TABLE** Engineering.projects.projects\_table ( project\_id INT,

project\_name STRING,

start\_date DATE,

end\_date DATE

);

**CREATE TABLE** Operations.logistics\_data.logistics\_table ( shipment\_id INT,

origin STRING,

destination STRING,

status STRING

);

in the marketing catalog. in the engineering catalog.

**INSERT INTO** Marketing.ads\_data.ads\_table (ad\_id, impressions, clicks, cost\_per\_click)

**VALUES**

(1, 1000, 100, 0.50),

(2, 1500, 150, 0.75),

(3, 2000, 200, 0.60),

(4, 2500, 250, 0.55);

**INSERT INTO** Engineering.projects.projects\_table (project\_id, project\_name, start\_date, end\_date)

**VALUES**

(1, 'Project Alpha', '2024-01-01', '2024-06-01'),

(2, 'Project Beta', '2024-02-01', '2024-08-01'),

(3, 'Project Gamma', '2024-03-01', '2024-09-01'),

(4, 'Project Delta', '2024-04-01', '2024-12-01');

**INSERT INTO** Operations.logistics\_data.logistics\_table (shipment\_id, origin, destination, status)

**VALUES**

(1, 'Warehouse A', 'Store X', 'Delivered'),

(2, 'Warehouse B', 'Store Y', 'In Transit'),

(3, 'Warehouse C', 'Store Z', 'Pending'),

(4, 'Warehouse D', 'Store W', 'Delivered');

# Part 3: Data Governance Capabilities

## Data Access Control

**Task 6**: **Create Roles and Grant Access**

Create specific roles for each department and grant access to the relevant catalogs and schemas.

For example: create roles such as marketing\_role , engineering\_role , and

operations\_role .

## Task 7: Configure Fine-Grained Access Control

Set up fine-grained access control, where users in the marketing department can only access customer-related data, while engineers can only access project

data. Define permissions accordingly.

**GRANT SELECT ON SCHEMA** projects.projects\_table TO `Project Alpha`;

**GRANT SELECT ON TABLE** projects.projects\_table TO ` Project Alpha`;

**REVOKE SELECT ON** **TABLE** projects.projects\_table FROM ` Project Alpha`;

## Data Lineage

**Task 8**: **Enable and Explore Data Lineage**

Enable data lineage for the tables created in Part 2.

Perform some queries (e.g., aggregate queries) on the datasets and examine how the data lineage feature traces the origin of data and tracks transformations.

## Data Audit

**Task 9**: **Monitor Data Access and Modifications**

Set up audit logging to track who is accessing or modifying the datasets.

Access the audit logs to view data access patterns and identify who performed which actions on the data.

## Data Discovery

**Task 10**: **Explore Metadata in Unity Catalog**

Explore the metadata of the tables you’ve created. Document information such as table schema, number of rows, and table properties for each department.

Make sure that the appropriate descriptions and properties are added to each catalog, schema, and table.

--ADVANCED CAPABILITIES

**VACUUM** Marketing.ads\_data.ads\_table RETAIN 168;

**DESCRIBE HISTORY** Marketing.ads\_data.ads\_table;

**SELECT \* FROM** Marketing.ads\_data.ads\_table sales **VERSION** AS OF 2;

# Deliverables:

Department catalogs, schemas, and tables created in Unity Catalog. Access roles and controls in place for each department.

Demonstrations of data governance capabilities such as **Data Lineage**, **Data Audit**, and **Data Discovery**.