**Exercise: Mini Project Using Unity Catalog and Data Governance**

**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.:**

**For the Marketing catalog, create schemas such as ads\_data and**

**customer\_data .**

USE Marketing;

CREATE SCHEMA ads\_data;

CREATE SCHEMA customer\_data;

**For the Engineering catalog, create schemas such as projects and**

**development\_data .**

USE Engineering;

CREATE SCHEMA projects;

CREATE SCHEMA development\_data;

**For the Operations catalog, create schemas such as logistics\_data and**

**supply\_chain** .

USE Operations;

CREATE SCHEMA logistics\_data;

CREATE SCHEMA 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 .**

**ads\_data.csv**

ad\_id,impressions,clicks,cost\_per\_click

101,5000,300,1.20

102,7000,450,1.10

103,10000,600,1.15

104,6000,320,1.30

105,8000,500,1.25

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

**project\_name , start\_date , end\_date**

**Projects.csv**

project\_id,project\_name,start\_date,end\_date

201,Website Redesign,2024-01-15,2024-04-20

202,Mobile App Development,2024-02-01,2024-07-30

203,Data Migration,2024-03-10,2024-06-15

204,AI Integration,2024-04-01,2024-09-01

205,Cloud Infrastructure Setup,2024-05-05,2024-08-25

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

**destination , status .**

**Logistics\_data.csv**

shipment\_id,origin,destination,status

301,New York,Los Angeles,In Transit

302,Chicago,Houston,Delivered

303,San Francisco,Seattle,Delayed

304,Miami,Dallas,In Transit

305,Atlanta,Boston,Delivered

**Task 5: Create Tables from the Datasets**

**Load the datasets into their respective schemas as tables.**

**Example: Create a table for ads\_data in the marketing catalog.**

USE Marketing;

CREATE TABLE ads\_data.ads (

ad\_id INT,

impressions INT,

clicks INT,

cost\_per\_click DECIMAL(5, 2) );

USING csv

OPTIONS (path 'dbfs:/mnt/path/marketing\_ads\_data.csv', header = true);

**Example: Create a table for projects in the engineering catalog.**

USE Engineering;

CREATE TABLE projects.projects (

project\_id INT,

project\_name VARCHAR(255),

start\_date DATE,

end\_date DATE );

USING csv

OPTIONS (path 'dbfs:/mnt/path/engineering\_projects\_data.csv', header = true);

**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**

**Marketing Role:**

CREATE ROLE marketing\_role;

**Engineering Role**

CREATE ROLE engineering\_role;

**Operations Role**

CREATE ROLE operations\_role;

**GRANT ACCESS:**

**Marketing Role:**

GRANT USAGE ON DATABASE Marketing TO marketing\_role;

GRANT USAGE ON SCHEMA ads\_data TO marketing\_role;

GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA ads\_data TO marketing\_role;

**Engineering Role:**

GRANT USAGE ON DATABASE Engineering TO engineering\_role;

GRANT USAGE ON SCHEMA projects TO engineering\_role;

GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA projects TO engineering\_role;

**Operations Role:**

GRANT USAGE ON DATABASE Operations TO operations\_role;

GRANT USAGE ON SCHEMA logistics\_data TO operations\_role;

GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA logistics\_data TO operations\_role;

**Assign Users to Roles:**

**Assign User to Marketing Role**

GRANT marketing\_role TO user1;

**Assign User to Engineering Role**

GRANT engineering\_role TO user2;

**Assign User to Operations Role**

GRANT operations\_role TO user3;

**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.**

**Marketing Department:**

USE CATALOG Marketing;

GRANT USAGE ON SCHEMA customer\_data TO ROLE marketing\_role;

GRANT SELECT ON TABLE Marketing.customer\_data TO ROLE marketing\_role;

REVOKE SELECT ON TABLE Marketing.ads\_data FROM ROLE marketing\_role;

**Engineering Department:**

USE CATALOG Engineering;

GRANT USAGE ON SCHEMA projects TO ROLE engineering\_role;

GRANT SELECT ON TABLE Engineering.projects TO ROLE engineering\_role;

REVOKE SELECT ON TABLE Engineering.development\_data FROM ROLE engineering\_role;

GRANT SELECT(ad\_id, impressions, clicks) ON TABLE Marketing.ads\_data TO ROLE marketing\_role;

REVOKE SELECT(cost\_per\_click) ON TABLE Marketing.ads\_data FROM ROLE marketing\_role;

**Assign Specific Users to Roles:**

GRANT ROLE marketing\_role TO `marketing\_user1;

GRANT ROLE engineering\_role TO `engineering;

**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.**

**Marketing:**

USE CATALOG Marketing;

USE SCHEMA ads\_data;

SELECT ad\_id, SUM(impressions) AS total\_impressions, SUM(clicks) AS total\_clicks

FROM ads\_data

GROUP BY ad\_id;

**Engineering:**

USE CATALOG Engineering;

USE SCHEMA projects;

SELECT p.project\_id, p.project\_name, d.development\_status

FROM projects p

JOIN development\_data d

ON p.project\_id = d.project\_id;

**Operations:**

USE CATALOG Operations;

USE SCHEMA logistics\_data;

SELECT status, COUNT(shipment\_id) AS total\_shipments

FROM logistics\_data

WHERE origin = 'New York'

GROUP BY status;

**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.**

CREATE TABLE audit\_log (

    log\_id INT IDENTITY(1,1),

    user\_name VARCHAR(100),

    action\_type VARCHAR(50),

    table\_name VARCHAR(100),

    query\_text VARCHAR(MAX),

    action\_time DATETIME DEFAULT CURRENT\_TIMESTAMP

);

CREATE TRIGGER trg\_ads\_data\_insert

ON ads\_data

AFTER INSERT

AS

BEGIN

    DECLARE @user\_name VARCHAR(100) = SYSTEM\_USER;  -- Gets the current user

    DECLARE @query\_text VARCHAR(MAX) = 'INSERT';    -- Logs the action type

    -- Insert action into audit log

    INSERT INTO audit\_log (user\_name, action\_type, table\_name, query\_text)

    VALUES (@user\_name, 'INSERT', 'ads\_data', @query\_text);

END;

CREATE TRIGGER trg\_ads\_data\_update

ON ads\_data

AFTER UPDATE

AS

BEGIN

    DECLARE @user\_name VARCHAR(100) = SYSTEM\_USER;

    DECLARE @query\_text VARCHAR(MAX) = 'UPDATE';

    INSERT INTO audit\_log (user\_name, action\_type, table\_name, query\_text)

    VALUES (@user\_name, 'UPDATE', 'ads\_data', @query\_text);

END;

CREATE TRIGGER trg\_ads\_data\_delete

ON ads\_data

AFTER DELETE

AS

BEGIN

    DECLARE @user\_name VARCHAR(100) = SYSTEM\_USER;

    DECLARE @query\_text VARCHAR(MAX) = 'DELETE';

    INSERT INTO audit\_log (user\_name, action\_type, table\_name, query\_text)

    VALUES (@user\_name, 'DELETE', 'ads\_data', @query\_text);

END;

View all actions logged in the audit log

SELECT \*

FROM audit\_log

ORDER BY action\_time DESC;

**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.**

SHOW CATALOGS;

SHOW SCHEMAS IN < Marketing>;

SHOW TABLES IN <catalog\_name>.<schema\_name>;

DESCRIBE <catalog\_name>.<schema\_name>.<table\_name>;

SELECT COUNT(\*) AS num\_rows

FROM <catalog\_name>.<schema\_name>.<table\_name>;

**Descriptions:**

ALTER CATALOG <catalog\_name>

SET COMMENT 'Description of the catalog';

ALTER SCHEMA <catalog\_name>.<schema\_name>

SET COMMENT 'Description of the schema';

ALTER TABLE <catalog\_name>.<schema\_name>.<table\_name>

SET COMMENT 'Description of the table';

ALTER TABLE <catalog\_name>.<schema\_name>.<table\_name>

SET TBLPROPERTIES ('property\_name' = 'property\_value');