Part 1: Setting Up the Environment

**Task 1: Create a Metastore**

CREATE CATALOG IF NOT EXISTS my\_catalog;

USE CATALOG my\_catalog;

CREATE SCHEMA IF NOT EXISTS my\_schema;

Task 2: Create Department

CREATE CATALOG marketing\_data;

CREATE CATALOG engineering\_data;

CREATE CATALOG operations\_data;

Task 3: Create Schemas for Each Department

-- Marketing

CREATE SCHEMA marketing\_data.ads\_data;

CREATE SCHEMA marketing\_data.customer\_data;

-- Engineering

CREATE SCHEMA engineering\_data.projects;

CREATE SCHEMA engineering\_data.development\_data;

-- Operations

CREATE SCHEMA operations\_data.logistics;

CREATE SCHEMA operations\_data.supply\_chain;

Part 2: Loading Data and Creating Tables

Task 4: Prepare Datasets

Task 5: Create Tables from the Datasets

-- Marketing ads\_data

CREATE TABLE marketing\_data.ads\_data.ads (

  ad\_id INT,

  impressions INT,

  clicks INT,

  cost\_per\_click FLOAT

);

INSERT INTO marketing\_data.ads\_data.ads

VALUES

  (1, 1000, 50, 0.75),

  (2, 1500, 75, 0.60),

  (3, 2000, 90, 0.85);

-- Engineering projects\_data

CREATE TABLE engineering\_data.projects.projects\_data (

  project\_id INT,

  project\_name STRING,

  start\_date DATE,

  end\_date DATE

);

INSERT INTO engineering\_data.projects.projects\_data

VALUES

(301, 'Project Orion', '2024-07-01', '2024-12-01'),

(302, 'Project Phoenix', '2024-08-01', '2025-01-01'),

(303, 'Project Titan', '2024-09-01', '2025-02-01');

-- Operations logistics\_data

CREATE TABLE operations\_data.logistics.logistics\_data(

  shipment\_id INT,

  origin STRING,

  destination STRING,

  status STRING

);

INSERT INTO operations\_data.logistics.logistics\_data

VALUES

(1101, 'Miami', 'Dallas', 'In Transit'),

(1102, 'Boston', 'Philadelphia', 'Delivered'),

(1103, 'Atlanta', 'Denver', 'Pending');

Part 3: Data Governance Capabilities

Task 6: Create Roles and Grant Access

GRANT USAGE ON CATALOG marketing\_data TO `marketing\_group`;

GRANT SELECT ON SCHEMA marketing\_data.ads\_data TO `marketing\_group`;

GRANT SELECT ON TABLE marketing\_data.ads\_data.ads TO `marketing\_group`;

Task 7: Configure Fine-Grained Access Control

-- users in the marketing department can only access marketing data

GRANT SELECT ON TABLE marketing\_data.ads\_data.ads TO `user@example.com`;

-- engineers can only access project data.

REVOKE SELECT ON TABLE engineering\_data.projects.projects\_data FROM `user@example.com`;

-- operations\_role full access to logistics data

GRANT ALL PRIVILEGES ON TABLE operations\_data.logistics.logistics\_data TO `user@example.com`

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

Marketing Department – Ads Data Lineage:

USE CATALOG marketing\_data;

USE SCHEMA ads\_data;

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

FROM ads

GROUP BY ad\_id;

Engineering Department – Project Development Lineage:

USE CATALOG engineering\_data;

USE SCHEMA projects;

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

FROM projects\_data p

JOIN development\_data d

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

Operations Department – Logistics Data Lineage:

USE CATALOG operations\_data;

USE SCHEMA logistics;

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

FROM logistics\_data

WHERE origin = 'New York'

GROUP BY status;

Task 9: Monitor Data Access and Modifications

Create an Audit Log Table:

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 Triggers for Ads Data in Marketing Department:

CREATE TRIGGER trg\_ads\_data\_insert

ON marketing\_data.ads\_data.ads

AFTER INSERT

AS

BEGIN

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

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

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

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

END;

Update Trigger for ads\_data:

CREATE TRIGGER trg\_ads\_data\_update

ON marketing\_data.ads\_data.ads

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;

Delete Trigger for ads\_data:

CREATE TRIGGER trg\_ads\_data\_delete

ON marketing\_data.ads\_data.ads

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 Logged Actions

SELECT \*

FROM audit\_log

ORDER BY action\_time DESC;

Data Discovery

Task 10: Explore Metadata in Unity Catalog

-- View the schema of a table

DESCRIBE TABLE marketing\_data.ads\_data.ads;

-- Get extended metadata about the table

DESCRIBE EXTENDED engineering\_data.projects.projects\_data;

-- View the number of rows and other statistics

ANALYZE TABLE operations\_data.logistics.logistics\_data COMPUTE STATISTICS;

-- description to a catalog

COMMENT ON CATALOG marketing\_data IS 'Catalog for marketing-related data.';

-- Add a comment to the ads\_data schema

COMMENT ON SCHEMA marketing\_data.ads\_data IS 'This schema contains ads performance data for marketing.';

-- description to a table

COMMENT ON TABLE marketing\_data.ads\_data.ads IS 'This table stores ad performance data for marketing campaigns.';