Mini Project: Retail Sales Data Governance Platform

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.

-- SQL command to create a Metastore:
CREATE METASTORE retail_sales_metastore;

Task 2: Create Department-Specific Catalogs

Create separate catalogs for the following departments:
- Marketing
- Engineering
- Operations

SQL commands to create catalogs:
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.
- For the Marketing catalog, create schemas such as `ads_data` and `customer_data`.
- For the Engineering catalog, create schemas such as `projects` and `development_data`.
- For the Operations catalog, create schemas such as `logistics_data` and `supply_chain`.

-- SQL commands to create schemas:

For Marketing**:**
CREATE SCHEMA marketing.ads_data;
CREATE SCHEMA marketing.customer_data;

For Engineering**:**
CREATE SCHEMA engineering.projects;
CREATE SCHEMA engineering.development_data;

For Operations**:**
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: ad_id, impressions, clicks, cost_per_click.
- Engineering - Projects: project_id, project_name, start_date, end_date.
- Operations - Logistics: shipment_id, origin, destination, status.

Ads Data:

ad_id, impressions, clicks, cost_per_click

1, 1000, 50, 0.5

2, 2000, 80, 0.45

Projects:

project_id, project_name, start_date, end_date

P101, Alpha, 2023-01-01, 2023-06-30

P102, Beta, 2023-02-15, 2023-12-31

Logistics:

shipment_id, origin, destination, status

S101, NY, CA, Shipped

S102, TX, FL, Delivered

Task 5: Create Tables from the Datasets

Load the datasets into their respective schemas as tables. Example for Marketing:
-- SQL command to create a table for ads_data in Marketing catalog:

-- Ads data table in Marketing catalog

```
CREATE TABLE marketing.ads_data.ad_stats (
 ad_id INT,
 impressions INT,
 clicks INT,
 cost_per_click DECIMAL(5, 2)
)
USING DELTA;

-- Engineering projects table
CREATE TABLE engineering.projects.project_details (
 project_id STRING,
 project_name STRING,
 start_date DATE,
 end_date DATE
)
USING DELTA;

-- Operations logistics table
CREATE TABLE operations.logistics_data.shipments (
 shipment_id STRING,
 origin STRING,
 destination STRING,
 status STRING
)
USING DELTA;
```

Part 3: Data Governance Capabilities

Task 6: Create Roles and Grant Access

Create specific roles for each department and grant access to the relevant catalogs and schemas.
Example roles:
- `marketing_role`
- `engineering_role`
- `operations_role`

-- SQL commands to create roles and grant access:
CREATE ROLE marketing_role;
GRANT USAGE ON CATALOG marketing TO ROLE marketing_role;

CREATE ROLE engineering_role;
GRANT USAGE ON CATALOG engineering TO ROLE engineering_role;

CREATE ROLE operations_role;
GRANT USAGE ON CATALOG operations TO ROLE 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.

-- SQL command to grant fine-grained access:
GRANT SELECT ON SCHEMA marketing.customer_data TO ROLE marketing_role;
GRANT SELECT ON SCHEMA engineering.projects TO ROLE engineering_role;

Task 8: Enable and Explore Data Lineage

Enable data lineage for the tables created in Part 2.

-- Query to view data lineage:
SELECT * FROM system.data_lineage WHERE table_name = 'ad_stats';

Task 9: Monitor Data Access and Modifications

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

-- Command to enable audit logs:
ALTER SYSTEM SET 'spark.databricks.sql.audit.enabled' = true;

Task 10: Explore Metadata in Unity Catalog

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

-- SQL query to explore table metadata:
DESCRIBE TABLE marketing.ads_data.ad_stats;

-- Get table schema and properties:

DESCRIBE DETAIL marketing.ads_data.ad_stats;

DESCRIBE DETAIL engineering.projects.project_details;

-- Explore number of rows and basic statistics

ANALYZE TABLE marketing.ads_data.ad_stats COMPUTE STATISTICS;