

# 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

### Task 2: Create Department-Specific Catalogs

- Create separate catalogs for the following departments:
  - Marketing

```
CREATE CATALOG marketing_data;
```

Engineering

```
CREATE CATALOG engineering_data;
```

Operations

- ```
CREATE CATALOG operations_data;
```

### Task 3: Create Schemas for Each Department

For Marketing:

```
CREATE SCHEMA marketing_data.ads_data;
```

```
CREATE SCHEMA marketing_data.customer_data;
```

For Engineering:

```
CREATE SCHEMA engineering_data.projects;
```

```
CREATE SCHEMA engineering_data.development_data;
```

For operations:

```
CREATE SCHEMA operations_data.logistics_data;
```

```
CREATE SCHEMA operations_data.supply_chain;
```

## Part 2: Loading Data and Creating Tables

### Task 4: Prepare Dataset

## **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 DECIMAL(5, 2)  
);
```

```
INSERT INTO marketing_data.ads_data.ads (ad_id, impressions, clicks, cost_per_click)  
VALUES  
(1, 1000, 50, 0.25),  
(2, 1500, 75, 0.30),  
(3, 2000, 100, 0.20),  
(4, 2500, 150, 0.35),  
(5, 3000, 180, 0.40);
```

### **Engineering - Projects:**

```
CREATE TABLE engineering_data.projects.project (  
    project_id INT,  
    project_name STRING,  
    start_date DATE,  
    end_date DATE  
);
```

```
INSERT INTO engineering_data.projects.project (project_id, project_name, start_date, end_date)  
VALUES  
(1, 'AI Development', '2023-01-01', '2023-12-31'),  
(2, 'Mobile App Redesign', '2023-02-01', '2023-08-31'),  
(3, 'Cloud Migration', '2023-03-01', '2023-10-30'),  
(4, 'Data Warehouse Setup', '2023-04-01', '2023-11-30'),  
(5, 'Automation Initiative', '2023-05-15', '2023-09-30');
```

### **Operations - Logistics:**

```
CREATE TABLE operations_data.logistics_data.shipment (  
    shipment_id INT,
```

```
origin STRING,  
destination STRING,  
status STRING  
);
```

```
INSERT INTO operations_data.logistics_data.shipment (shipment_id, origin, destination, status)  
VALUES  
(1, 'New York', 'Los Angeles', 'In Transit'),  
(2, 'San Francisco', 'Chicago', 'Delivered'),  
(3, 'Houston', 'Miami', 'In Transit'),  
(4, 'Seattle', 'Denver', 'Delivered'),  
(5, 'Boston', 'Dallas', 'Pending');
```

### **Part 3: Data Governance Capabilities**

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

```
CREATE ROLE marketing_role;  
  
GRANT USAGE ON CATALOG marketing_data TO marketing_role;  
  
GRANT SELECT ON SCHEMA marketing_data.ads_data TO marketing_role;
```

```
CREATE ROLE engineering_role;  
  
GRANT USAGE ON CATALOG engineering_data TO engineering_role;  
  
GRANT SELECT ON SCHEMA engineering_data.projects TO engineering_role;
```

```
CREATE ROLE operations_role;  
  
GRANT USAGE ON CATALOG operations_data TO operations_role;  
  
GRANT SELECT ON SCHEMA operations_data.logistics_data TO operations_role;
```

#### **Task 7: Configure Fine-Grained Access Control**

```
GRANT SELECT ON TABLE marketing_data.ads_data.ads TO marketting@gmail.com;
```

```
GRANT SELECT ON TABLE engineering_data.projects.project TO engineering_role;
```

## **Data Lineage**

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

```
SELECT project_name, COUNT(*)  
FROM engineering_data.projects.project  
GROUP BY project_name;
```

## **Data Audit**

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

- In the Databricks Admin Console, enable audit logs to monitor data access and changes

## **Data Discovery**

### **Task 10**

```
DESCRIBE TABLE marketing_data.ads_data.ads;
```

#### **Add descriptions to your tables:**

```
COMMENT ON TABLE marketing_data.ads_data.ads IS 'Stores ad data for marketing department';
```