Mini Project: Building a Secure Data Platform with Unity Catalog

Task 1: Set Up Unity Catalog for Multi-Domain Data Management

1. Create a new catalog

```
>>  CREATE CATALOG enterprise_data_catalog;
```

2. Create Schemas for Each Department

```
>>  CREATE SCHEMA enterprise_data_catalog.marketing_data;
>>  CREATE SCHEMA enterprise_data_catalog.operations_data;
>>  CREATE SCHEMA enterprise_data_catalog.it_data;
```

3. Create tables in each schema

```
- - For Marketing data
>>  CREATE TABLE enterprise_data_catalog.marketing_data.marketing_table(
    CampaignID INT,
    CampaignName STRING,
    Budget DECIMAL(10,2),
    StartDate DATE
    );

- - For Operations Data
>>  CREATE TABLE enterprise_data_catalog.operations_data.operations_table(
    OrderID INT,
    ProductID INT,
    Quantity INT,
    ShippingStatus STRING
    );
- - For IT Data
>>  CREATE TABLE enterprise_data_catalog.it_data.it_table(
    IncidentID STRING,
    ReportedBy STRING,
    IssueType STRING,
    ResolutionTime INT
    );
```

Task 2: Data Discovery and Classification

1. **Search for Data Across Schemas:**

```
>>  SHOW TABLES IN enterprise_data_catalog;
```

2. **Tag Sensitive Information**

```
>>  ALTER TABLE enterprise_data_catalog.marketing_data.marketing_table
    SET TAG 'sensitive' ON COLUMN Budget;
>>  ALTER TABLE enterprise_data_catalog.it_data.it_table
    SET TAG 'sensitive' ON COLUMN ResolutionTime;
```

3. **Data Profiling**

```
>>  SELECT AVG(Budget), MIN(Budget), MAX(Budget) FROM
    enterprise_data_catalog.marketing_data.marketing_table;
>>  SELECT COUNT(ShippingStatus), ShippingStatus FROM
    enterprise_data_catalog.operations_data.operations_table GROUP BY ShippingStatus;
```

Task 3: Data Lineage and Data Auditing

1. **Track Data Lineage Across Schemas:**
   - - Link the marketing_data with the operations_data by joining campaign performance with product orders.

```
>>  CREATE TABLE enterprise_data_catalog.reporting.campaign_orders_report AS
    SELECT m.CampaignID, m.CampaignName, m.Budget, o.OrderID, o.ProductID, o.Quantity
    FROM enterprise_data_catalog.marketing_data.campaigns m
    JOIN enterprise_data_catalog.operations_data.orders o
    ON m.CampaignID = o.ProductID;
```

2. **Enable and Analyze Audit Logs:**
   - - Enabling audit logs for operations performed on the tables
   - Navigate to admin console in databricks
   - Go to audit logs tab and enable audit logs

Task 4: Implement Fine-Grained Access Control

1. **Create User Roles and Groups**

```
>>  CREATE GROUP MarketingTeam;

>>  GRANT USAGE ON SCHEMA enterprise_data_catalog.marketing_data TO
    MarketingTeam;

>>  GRANT USAGE ON SCHEMA enterprise_data_catalog.marketing_data TO
    OperationsTeam;


>>  CREATE GROUP OperationsTeam;

>>  GRANT USAGE ON SCHEMA enterprise_data_catalog.operations_data TO
    OperationsTeam;


>>  CREATE GROUP ITSupportTeam;

>>  GRANT USAGE ON SCHEMA enterprise_data_catalog.it_data TO ITSupportTeam;

>>  GRANT UPDATE ON TABLE enterprise_data_catalog.it_data.it_data TO ITSupportTeam;
```

2. **Implement Column-Level Security**

```
>>  GRANT SELECT ON COLUMN Budget TO MarketingTeam;
```

3. **Row-Level Security**

```
>>  CREATE ROW ACCESS POLICY operations_team_policy ON
    enterprise_data_catalog.operations_data.orders
    FOR EACH ROW
    WHEN current_user = operations_rep;
```

Task 5: Data Governance and Quality Enforcement

1. **Set Data Quality Rules**:

```
    - - Campaign budget greater than Zero(0).
>>  SELECT * FROM enterprise_data_catalog.marketing_data.marketing_table
     WHERE Budget <= 0;


    - - Shipping status is valid (e.g.,'Pending', 'Shipped', 'Delivered').
>>  SELECT * FROM enterprise_data_catalog.operations_data.operations_table
    WHERE ShippingStatus NOT IN ('Pending', 'Shipped', 'Delivered');
```

- - Issue resolution times are recorded correctly and not negative..

```
>>   SELECT * FROM enterprise_data_catalog.it_data.it_table WHERE ResolutionTime < 0;
```

2. **Apply Delta Lake Time Travel**

```
>>   RESTORE TABLE enterprise_data_catalog.operations_data.operations_table

      TO VERSION AS OF 1;
```

Task 6: Performance Optimization and Data Cleanup

1. **Optimize Delta Tables**

```
>>    OPTIMIZE enterprise_data_catalog.operations_data.operations_table;
>>    OPTIMIZE enterprise_data_catalog.it_data.it_table;
```

2. **Vacuum Delta Tables**

```
>>    VACUUM enterprise_data_catalog.operations_data.operations_table

      RETAIN 168 HOURS;

>>     VACUUM enterprise_data_catalog.it_data.it_table RETAIN 168 HOURS;
```