Part 1: Setting Up the Environment

Task 1 : Creating a meta Store from the admin console

Task 2: Create Department-Specific Catalogs

```
CREATE CATALOG marketing;

CREATE CATALOG engineering;

CREATE CATALOG operations;
```

Task 3: Create Schemas for Each Department

```
CREATE SCHEMA marketing.ads_data;

CREATE SCHEMA marketing.customer_data;


CREATE SCHEMA engineering.projects;

CREATE SCHEMA engineering. development_data;


CREATE SCHEMA operations.logistics_data;

CREATE SCHEMA operations. supply_chain;
```

Part 2: Loading Data and Creating Tables

Task 4: Prepare Datasets

Marketing's ads_data csv file created

Task 5: Create Tables from the Datasets

```
CREATE TABLE marketing.ads_data(

ad_id INT,

impressions INT,

clicks INT,

cost_per_click INT

);
```

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

CREATE TABLE operations.logistics_data (
shipment_id INT,
origin STRING,
destination STRING,
status STRING
);

INSERT INTO Marketing.ads_data.ad_details (ad_id, impressions, clicks,
cost_per_click)
VALUES
(1, 10000, 500, 0.25),
(2, 15000, 750, 0.30),
(3, 12000, 600, 0.20);

 INSERT INTO Marketing.customer_data.customer_detail (cust_id, ad_id)
VALUES
(101, 1),
(102, 2),
(103, 3);
```

```
 INSERT INTO Engineering.projects.project_data (project_id, project_name)
VALUES
(1, 'Website Redesign'),
(2, 'Mobile App Development'),
(3, 'Database Optimization');

 INSERT INTO Engineering.projects.development_data (dev_id, project_id,
start_data, end_date)
VALUES
(1, 1, '2024-01-01', '2024-06-30'),
(2, 2, '2024-03-15', '2024-12-31'),
(3, 3, '2024-02-01', '2024-04-30');

 INSERT INTO Operations.logistics_data.logistics (shipment_id, status)
VALUES
(1001, 'Delivered'),
(1002, 'In Transit'),
(1003, 'Processing');

 INSERT INTO Operations.supply_chain.supply_chain_data (Id_no, origin, destination,
shipment_id)
VALUES
(1, 'Chennai', 'Bangalore', 1001),
(2, 'Chennai', 'Hyderabad', 1002),
(3, 'Chennai', 'Mumbai', 1003);
```

Part 3: Data Governance Capabilities

Data Access Control

Task 6: Create Roles and Grant Access

```
CREATE ROLE marketing_role;

CREATE ROLE engineering_role;

CREATE ROLE operations_role;
```

Task 7: Configure Fine-Grained Access Control

```
GRANT SELECT ON TABLE marketing.ads_data TO ROLE marketing_role;

REVOKE SELECT ON TABLE marketing.customer_data FROM ROLE marketing_role;


GRANT SELECT ON TABLE Engineering.projects.project_data TO engineering_role;

REVOKE SELECT ON TABLE Engineering.projects.project_data FROM ROLE engineering_role;


GRANT SELECT ON TABLE operations.supply_chain.supply_chain_data TO operations_role;

REVOKE SELECT ON TABLE operations.supply_chain.supply_chain_data FROM ROLE operations_role;
```

Data Lineage

Task 8: Enable and Explore Data Lineage

Navigate to the databricks UI to Catalog Explorer to check the lineage of the tables we created

Task 9: Monitor Data Access and Modifications:

In the Admin Console, we can view the Audit logs for the operations performed.

Task 10: Explore Metadata in Unity Catalog

```
DESCRIBE TABLE marketing.ads_data. ad_details;;

DESCRIBE TABLE Engineering.projects.project_data;

DESCRIBE TABLE Operations.logistics_data.logistics;
```

```
SELECT COUNT(*) FROM marketing.ads_data.ad_details;

SELECT COUNT(*) FROM engineering.projects.project_data;

SELECT COUNT(*) FROM Operations.logistics_data.logistics;
```