Unity Catalog Exercise:

**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 Catalog**

For Marketing Catalog:

* CREATE SCHEMA Marketing.ads\_data;
* CREATE SCHEMA Marketing.customer\_data;

For Engineering Catalog:

* CREATE SCHEMA Engineering.projects;
* CREATE SCHEMA Engineering.development\_data;

For Operations Catalog:

* CREATE SCHEMA Operations.logistics\_data;
* CREATE SCHEMA Operations.supply\_chain;

**Task 4 & 5: Creating Tables and loading dataset:**

For Marketing Catalog:

* CREATE TABLE Marketing.ads\_data.ad\_details (

ad\_id int,

impressions int,

clicks int,

cost\_per\_click double);

* CREATE TABLE Marketing.customer\_data.customer\_detail(

cust\_id int,

ad\_id int);

For Engineering Catalog:

* CREATE TABLE Engineering.projects.project\_data(

project\_id int,

project\_name string);

* CREATE TABLE Engineering.projects.development\_data(

dev\_id int,

project\_id int,

start\_data date,

end\_date date);

For Operations Catalog:

* CREATE TABLE Operations.logistics\_data.logistics (

shipment\_id int,

status string);

* CREATE TABLE Operations.supply\_chain.supply\_chain\_data(

Id\_no int,

origin string,

destination string,

shipment\_id int);

**Inserting Data:**

* 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);

**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:**

**For Marketing role:**

GRANT SELECT ON TABLE Marketing.customer\_data.customer\_detail TO marketing\_role;

GRANT SELECT ON TABLE Marketing.ads\_data.ad\_details TO marketing\_role;

**For Engineering role:**

GRANT SELECT ON TABLE Engineering.projects.project\_data TO engineering\_role;

GRANT SELECT ON TABLE Engineering.projects.development\_data TO engineering\_role;

**For Operations role:**

GRANT SELECT ON TABLE operations.logistics\_data.logistics TO operations\_role;

GRANT SELECT ON TABLE operations.supply\_chain.supply\_chain\_data TO operations\_role;

**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:**

For Marketing Tables:

DESCRIBE TABLE Marketing.ads\_data.ad\_details;

DESCRIBE TABLE Marketing.customer\_data.customer\_detail;

SELECT COUNT(\*) FROM marketing.ads\_data.ad\_details;

SELECT COUNT(\*) FROM marketing.customer\_data.customer\_detail;

For Engineering Tables:

DESCRIBE TABLE Engineering.projects.project\_data;

DESCRIBE TABLE Engineering.projects.development\_data;

SELECT COUNT(\*) FROM engineering.projects.project\_data;

SELECT COUNT(\*) FROM engineering.projects.development\_data;

For Operations Tables:

DESCRIBE TABLE Operations.logistics\_data.logistics;

DESCRIBE TABLE Operations.supply\_chain.supply\_chain\_data;

SELECT COUNT(\*) FROM Operations.logistics\_data.logistics;

SELECT COUNT(\*) FROM Operations.supply\_chain.supply\_chain\_data;