Asmita Porwal
Data Engineering
Batch-1
21/02/2024

Coding Challenge -4 Question-2

Explain Overview of 3 level namespace and creating Unity Catalog objects.

3 level namespace (catalog.schema.table)

In Unity Catalog, the hierarchy of primary data objects flows from metastore to table or volume:

- **Metastore**: The top-level container for metadata. Each metastore exposes a three-level namespace (`catalog.schema.table`) that organizes your data.
- **Catalog**: The first layer of the object hierarchy, used to organize your data assets.
- **Schema:** Also known as databases, schemas are the second layer of the object hierarchy and contain tables and views.
- **Tables, views, and volumes:** At the lowest level in the data object hierarchy are tables, views, and volumes. Volumes provide governance for non-tabular data.
- **Models:** Although they are not, strictly speaking, data assets, registered models can also be managed in Unity Catalog and reside at the lowest level in the object hierarchy.

Catalogs

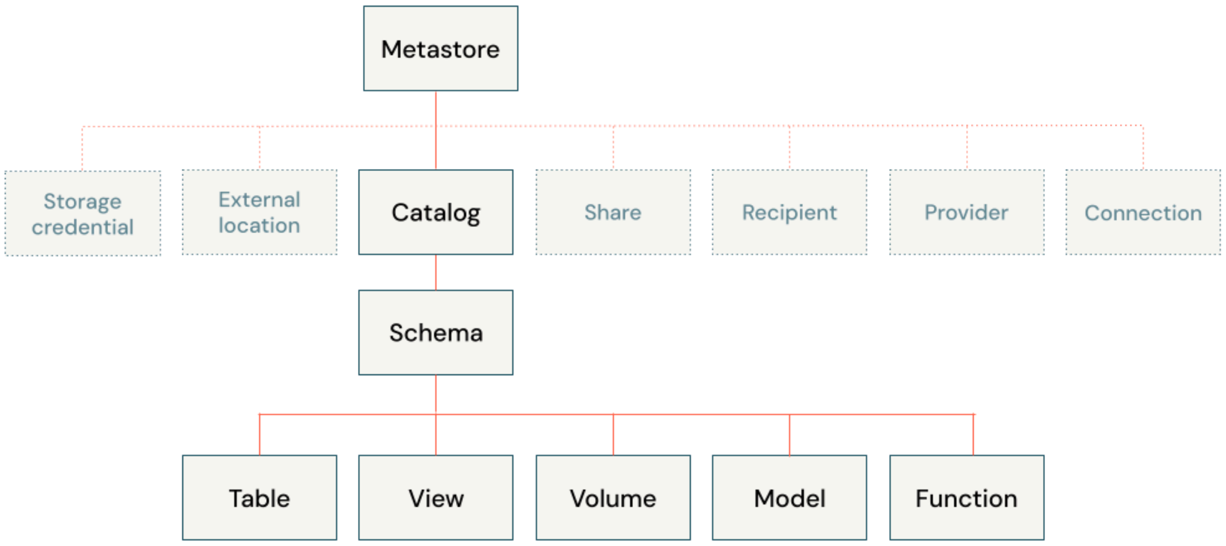
A catalog is the first layer of Unity Catalog's three-level namespace.
It's used to organize your data assets.
Users can see all catalogs on which they have been assigned the USE CATALOG.
Depending on how your workspace was created and enabled for Unity Catalog, your users may have default permissions on automatically provisioned catalogs, including either the main catalog or the workspace catalog (<workspace-name>)

Schemas

A schema (also called a database) is the second layer of Unity Catalog’s three-level namespace.
A schema organizes tables and views.
Users can see all schemas on which they have been assigned the USE SCHEMA permission, along with the USE CATALOG permission on the schema’s parent catalog.
To access or list a table or view in a schema, users must also have SELECT permission on the table or view.
If your workspace was enabled for Unity Catalog manually, it includes a default schema named default in the main catalog that is accessible to all users in your workspace.
If your workspace was enabled for Unity Catalog automatically and includes a <workspace-name> catalog, that catalog contains a schema named default that is accessible to all users in your workspace.

Tables

A table resides in the third layer of Unity Catalog’s three-level namespace.
It contains rows of data.
To create a table, users must have CREATE and USE SCHEMA permissions on the schema, and they must have the USE CATALOG permission on its parent catalog.
To query a table, users must have the SELECT permission on the table, the USE SCHEMA permission on its parent schema, and the USE CATALOG permission on its parent catalog.
A table can be managed or external.



creating Unity Catalog objects

Unity Catalog is a feature within Azure Databricks which is a cloud-based big data analytics platform built on Apache Spark that allows for efficient management and organization of data, providing capabilities such as creating catalogs, schemas, and managing privileges for users.
Unity Catalog provides a way to manage and organize data within Azure Databricks by offering a metadata management system.

Step 1: Confirm that your workspace is enabled for Unity Catalog:

> Checking if the workspace is already enabled for Unity Catalog through the account console or SQL query.

Step 2: Add users and assign the workspace admin role:

> Managing user roles and privileges, including syncing account-level identities from Microsoft Entra ID.

Step 3: Create clusters or SQL warehouses:

> Configuring compute resources that comply with Unity Catalog security requirements.

Step 4: Grant privileges to users:

> Defining default user and admin privileges and how to grant additional privileges.

Step 5: Create new catalogs and schemas:

> Creating catalogs, assigning managed storage, binding catalogs to workspaces, and granting privileges.

Unity Catalog is particularly useful for scenarios where you need to organize and govern your data assets within Azure Databricks. It provides a structured and efficient way to manage metadata, control access, and ensure data governance in big data analytics workflows.