

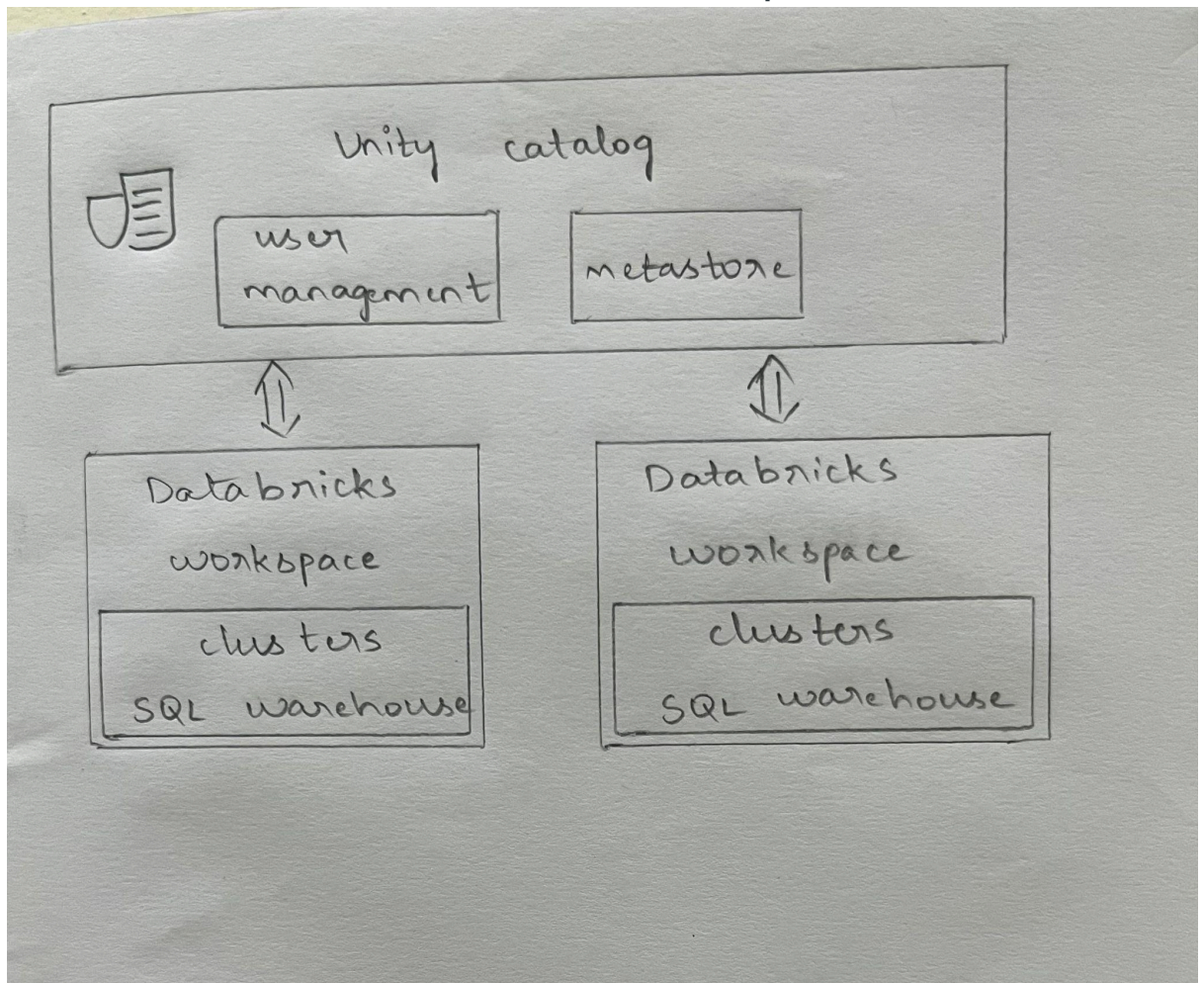
NAME : AKULA SHARATH CHANDRA

BATCH: DATA ENGINEERING

DATE :16-02-2024

TOPIC :UNITY CATALOG

1)UNITY CATALOG :Unity Catalog provides centralized access control, auditing, lineage, and data discovery capabilities across Azure Databricks workspaces.



—>KEY FEATURES:

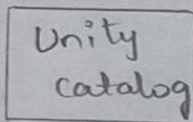
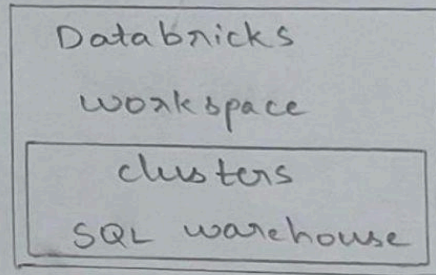
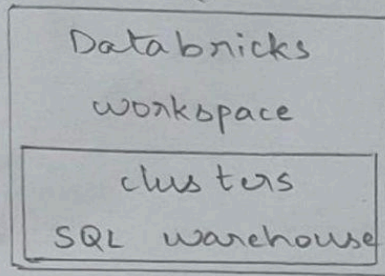
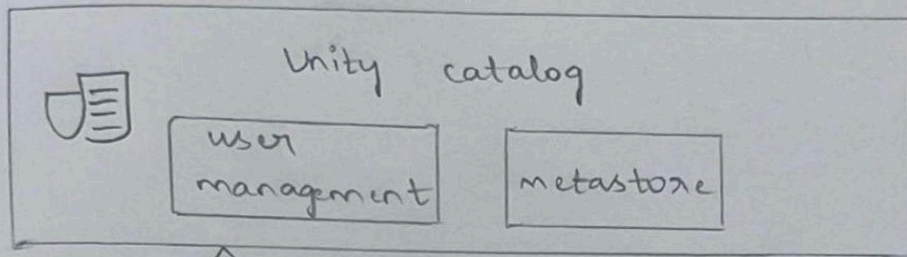
- i)**Define once, secure everywhere:** Unity Catalog offers a single place to administer data access policies that apply across all workspaces.
- ii)**Standards-compliant security model:** Unity Catalog's security model is based on standard ANSI SQL and allows administrators to grant permissions in their existing data lake using familiar syntax, at the level of catalogs, databases (also called schemas), tables, and views.

iii) **Built-in auditing and lineage:** Unity Catalog automatically captures user-level audit logs that record access to your data. Unity Catalog also captures lineage data that tracks how data assets are created and used across all languages.

iv) **Data discovery:** Unity Catalog lets you tag and document data assets, and provides a search interface to help data consumers find data.

v) **System tables (Public Preview):** Unity Catalog lets you easily access and query your account's operational data, including audit logs, billable usage, and lineage.

Architecture diagram:

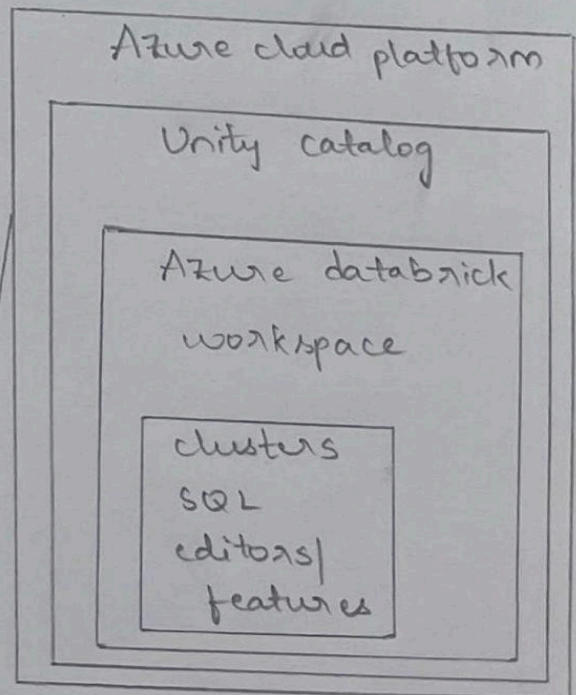


Key features:

- Define once, secure everywhere
- Standards-compliant security model
- Data discovery
- System tables (Public Preview)



Unity catalog provides centralised access control, auditing, lineage and data discovery capabilities across Azure Databricks workspace



2)The Unity Catalog object model:

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

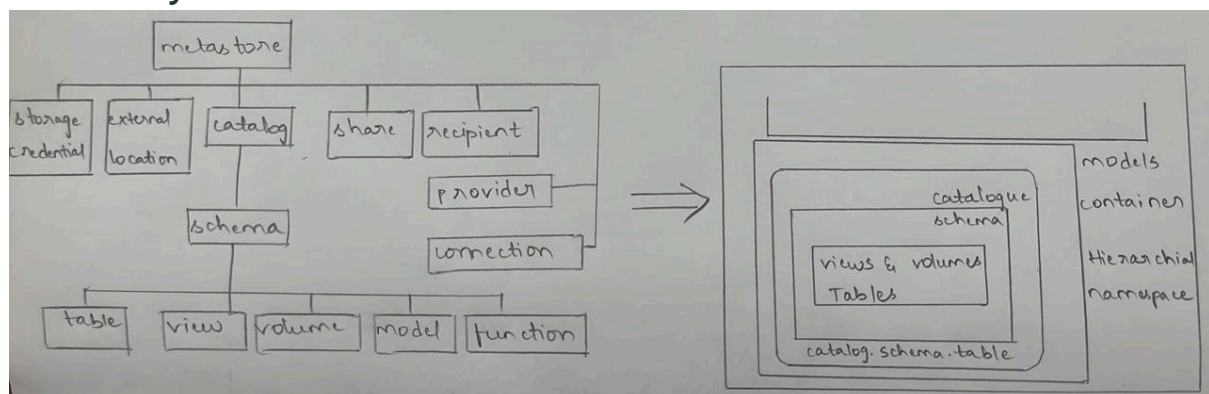
→**Metastore:** The top-level container for metadata. Each metastore exposes a three-level namespace (catalog.schema.table) that organizes your data.

→**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.



Metastores:

A metastore is the top-level container of objects in Unity Catalog. It registers metadata about data and AI assets and the permissions that govern access to them. Azure Databricks account admins should create one metastore for each region in which they operate and assign them to Azure Databricks workspaces in the same region. For a

workspace to use Unity Catalog, it must have a Unity Catalog metastore attached.

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 data permission.

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.

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.

Managed tables:

Managed tables are the default way to create tables in Unity Catalog. Unity Catalog manages the lifecycle and file layout for these tables. You should not use tools outside of Azure Databricks to manipulate files in these tables directly.

External tables:

External tables are tables whose data lifecycle and file layout are not managed by Unity Catalog. Use external tables to register large amounts of existing data in Unity

Catalog, or if you require direct access to the data using tools outside of Azure Databricks clusters or Databricks SQL warehouses.

External tables can use the following file formats:

- 1) CSV
- 2) JSON
- 3) AVRO
- 4) PARQUET
- 5) ORC
- 6) TEXT
- 7) DELTA

Views:

A view is a read-only object created from one or more tables and views in a metastore. It resides in the third layer of Unity Catalog's three-level namespace. A view can be created from tables and other views in multiple schemas and catalogs.

Volumes:

A volume resides in the third layer of Unity Catalog's three-level namespace. Volumes are siblings to tables, views, and other objects organized under a schema in Unity Catalog.