**Coding Challenge-2**

**Question 2: Explain Overview of 3 level namespace and creating Unity Catalog objects.**

**SOLUTION:**

**Overview of 3 level namespaces:**

Using a three level namespace we can reference the data in the object model by using the syntax: Catalog.schema.assets , here assests cane be table ,view or volume.The three namespaces in Unity Catalog are :

* Catalog
* Schema
* Tables,views or volumes

**Catalog**: A catalog is the first layer of three-level namespace. It’s used to organize the data assets. Users can see all catalogs on which they have been assigned by using the syntax: USE CATALOG.

**Schema**: It 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 by using the syntax: 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,Views and Volumes:** They reside in the third layer of Unity Catalog’s three-level namespace. A table contains rows of data. A view is a read-only object created from one or more tables. Volumes contain directories and files for data stored in any format. Volumes provide non-tabular access to data, meaning that files in volumes cannot be registered as tables.

Tables can be of two types:

* **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:
* DELTA
* CSV
* JSON
* AVRO
* PARQUET
* ORC
* TEXT

**Unity Catalog**

Unity Catalog provides centralized access control, auditing, lineage, and data discovery capabilities across Azure Databricks workspaces. Key features of Unity Catalog include:

* Built-in auditing and lineage
* Data discovery
* System tables (Public Preview):
* Standards-compliant security model
* Define once, secure everywhere

**The Unity Catalog object model**

In Unity Catalog, the object model is as followed:

* **Metastore**: The top-level container for metadata. Each metastore exposes a three-level namespace that organizes the data.
* **Catalog**: It is the first layer in the object model which is used to organize data assets.
* **Schema**: They are also known as databases and are the second layer in the object model which contains tables and views.
* **Tables, views, and volumes**: These are the lowest level in the object model which store some tabular or non-tabular data
* **Models**: They are not a part of the object model but registered models can also be managed in Unity Catalog and they are present at the lowest level in the object model.