**NAME :- ANIKET SANJAYKUMAR BIYANI**

**DATA ENGINEERING BATCH -1**

**AZURE DATABRICKS CODING ASSESMENT**

**Q2.Explain Overview of 3 level namespace and creating Unity Catalog objects.**

* The term "3 level namespace" typically refers to a hierarchical structure for organising data within a distributed file system like UnityFS.
* UnityFS is a highly scalable and parallel file system designed for use in high-performance computing (HPC) and big data environments.
* Here's an overview of the 3 level namespace concept and creating Unity Catalog objects within this context: Hierarchy Levels:
* The 3 level namespace consists of three hierarchical levels: root, directory, and file.

1.Root Level:

* At the top level, there is a single root directory that serves as the starting point for the entire file system.
* At the top level of the hierarchy is the volume, which represents a logical storage unit or partition on a storage device.
* Volumes are typically associated with physical storage resources such as disks, partitions, or storage arrays.
* They provide a logical boundary for organizing and managing data within a storage system.
* Example: In a Unix-like file system, volumes are represented by mount points such as **/**, **/home**, **/mnt**, etc.

2.File System Level:

* Within each volume, one or more file systems can be created.
* A file system defines the rules and structures for organizing, storing, and accessing files and directories within the volume.
* It governs how data is stored, retrieved, and managed, including aspects such as file allocation, naming conventions, and security permissions.
* Example: Common file systems include NTFS (Windows), ext4 (Linux), HFS+ (macOS), and FAT32 (universal).

3. **Directory and File Level**:

* Within each file system, directories (folders) and files are organized in a hierarchical manner.
* Directories serve as containers for grouping related files and subdirectories, forming a tree-like structure.
* Files are individual data objects stored within directories, each identified by a unique name and location.
* Example: In a directory structure, directories and files are arranged hierarchically, with parent-child relationships defining their position within the hierarchy.

**Unity Catalog:**

* In the context of UnityFS, a catalog is a metadata repository that stores information about the file system's structure, attributes, and location of data.
* Unity Catalog maintains mappings between logical file paths (namespaces) and physical data locations within the storage system.
* Catalog objects represent various entities within the file system, such as directories, files, and attributes associated with them.

Define once, secure everywhere: Unity Catalog offers a single place to administer data access policies that apply across all workspaces.

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.

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.

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

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

**Creating Unity Catalog Objects:**

To create Unity Catalog Objects, developers or asset creators need to follow these steps:

* **Asset Creation**: Create the asset or content that you want to sell on the Unity Asset Store. This could be anything from 3D models, textures, scripts, shaders, etc., depending on your area of expertise and the needs of Unity developers.
* **Preparing for Publication**: Before publishing your asset, ensure that it meets the Asset Store's submission guidelines and quality standards. This may involve optimizing the asset, creating appropriate documentation, setting up demo scenes (if applicable), and packaging the asset for distribution.
* **Publishing on the Asset Store**: Once your asset is ready, you can submit it to the Unity Asset Store for review and publication. This involves creating a listing for your asset, providing descriptions, images, videos, and setting pricing and licensing terms.
* **Managing Catalog Objects**: After your asset is published, it becomes a Unity Catalog Object available for purchase and download by Unity developers through the Asset Store. You can manage your Catalog Objects, track sales, update content, and interact with customers through the Unity Developer Dashboard.
* Creating Unity Catalog Objects allows developers and content creators to showcase their work, reach a wider audience, and monetize their creations within the Unity ecosystem. It also provides Unity developers with access to a vast library of high-quality assets to enhance their projects.