Databricks Coding Assessment

By Esaq A

1. Create the Unity Catalog Metastore (on Azure)

Prerequisites:

- A Databricks account with Admin privileges on a Premium plan or higher.
- An Azure subscription with permissions to create a storage account and an Access Connector for Azure Databricks.
- A unique name for your metastore.
- The Azure region where your workspaces are located.

Steps:

1. **Configure Storage:** In the Azure portal, create a new **Azure Data Lake Storage Gen2** account. Inside this storage account, create a container that will serve as the root storage for your metastore.
2. **Create an Access Connector for Azure Databricks:**
   - In the Azure portal, create an **Access Connector for Azure Databricks** resource. This connector will contain a system-assigned managed identity.
   - Navigate to your ADLS Gen2 storage account. Go to **Access control (IAM)** and grant the managed identity of the Access Connector the **Storage Blob Data Contributor** role.
3. **Create the Metastore and Attach a Workspace:**
   - Log in to your Databricks account console and go to the **Data** section.
   - Click **Create Metastore**.
   - Enter a **Name** for the metastore and select the **Region** where your workspaces are located.

    - For the storage path, enter the ADLS Gen2 container path in the format: `abfss://<container-name>@<storage-account-name>.dfs.core.windows.net/`.
    - Enter the Azure Resource ID of the Access Connector you created.
    - Assign the metastore to your workspace(s).

---

2. Enable Your Workspace for Unity Catalog

1. Log in to the Databricks account console as an account admin.
2. Click on the **Data** option in the navigation panel.
3. Access your metastore by clicking on its name.
4. Navigate to the **Workspaces** tab within the metastore.
5. Click the **Assign to workspaces** button.
6. In the dialog box, select the workspace(s) you want to enable.
7. Click **Assign**, then confirm by clicking **Enable**.
8. Verify the assignment by checking that your workspace is listed in the **Workspaces** tab.

---

3. Set Up and Manage Unity Catalog

Prerequisites:

- A Unity Catalog metastore associated with your workspace.
- Appropriate admin roles (account, metastore, or workspace).

Steps:

1. **Check Workspace Enablement:** Run the SQL query `SELECT CURRENT_METASTORE();` in a notebook. If it returns a metastore ID, your workspace is enabled.
2. **Add Users and Roles:** As a workspace or account admin, add users and groups to your workspace and assign any necessary admin roles.
3. **Set up Compute Resources:** Create SQL warehouses or clusters that comply with Unity Catalog's security requirements, using either **Shared Access Mode** or **Single User Access Mode**.

Grant User Privileges: Grant permissions to users or groups to create and access objects. For example, use the Catalog Explorer UI or run a SQL command:
SQL

```
GRANT CREATE SCHEMA ON CATALOG <my-catalog> TO `data-consumers`;
```

4.
5. **Create Catalogs and Schemas:** Create at least one catalog to begin organizing your data and AI assets.

---

4. Getting Started with Unity Catalog Objects

Prerequisites:

- A Unity Catalog-enabled workspace.
- Access to a compliant compute resource.
- Necessary privileges like `USE CATALOG`, `USE SCHEMA`, and `CREATE TABLE`.

Steps (SQL):

1 . Create a New Catalog: You must be a metastore admin or have the `CREATE CATALOG` privilege. Specify a managed location pointing to your ADLS Gen2 storage.

```
CREATE CATALOG IF NOT EXISTS <catalog_name>
MANAGED LOCATION
'abfss://<container-name>@<storage-account-name>.dfs.core.windows.net/catalogs/<catalog_name>';
```

2 . Select and Grant Permissions on the Catalog:

```
-- Set the current catalog
USE CATALOG <catalog_name>; [cite: 269]

-- Grant permissions to all account users
GRANT CREATE SCHEMA, CREATE TABLE, USE CATALOG
ON CATALOG <catalog_name>
TO `account users`; [cite: 271, 272, 273]
```

3 . Create and Manage Schemas: Create schemas to organize tables and views.

```
-- Create a new schema
CREATE SCHEMA IF NOT EXISTS <schema_name>
COMMENT "A new Unity Catalog schema"; [cite: 278, 279]
```

4 . Create a Managed Table:

```
-- Set the current schema
USE <schema_name>; [cite: 289]

-- Create a managed Delta table and insert records
CREATE TABLE IF NOT EXISTS <table_name> (columnA Int, columnB String); [cite: 291, 292]
INSERT INTO TABLE <table_name> VALUES (1, "one"), (2, "two"); [cite: 293, 294, 295, 296]
```

5 . Query the Table: Access tables using the three-level namespace.

```
-- Query using the fully qualified name
SELECT * FROM <catalog_name>.<schema_name>.<table_name>; [cite: 313]
```

6 . Manage Data Permissions (Optional): Use `GRANT` and `REVOKE` to manage access to data objects.

```
-- Grant SELECT privilege on a table
GRANT SELECT
ON TABLE <schema_name>.<table_name>
TO `account users`; [cite: 336, 337, 338]
```