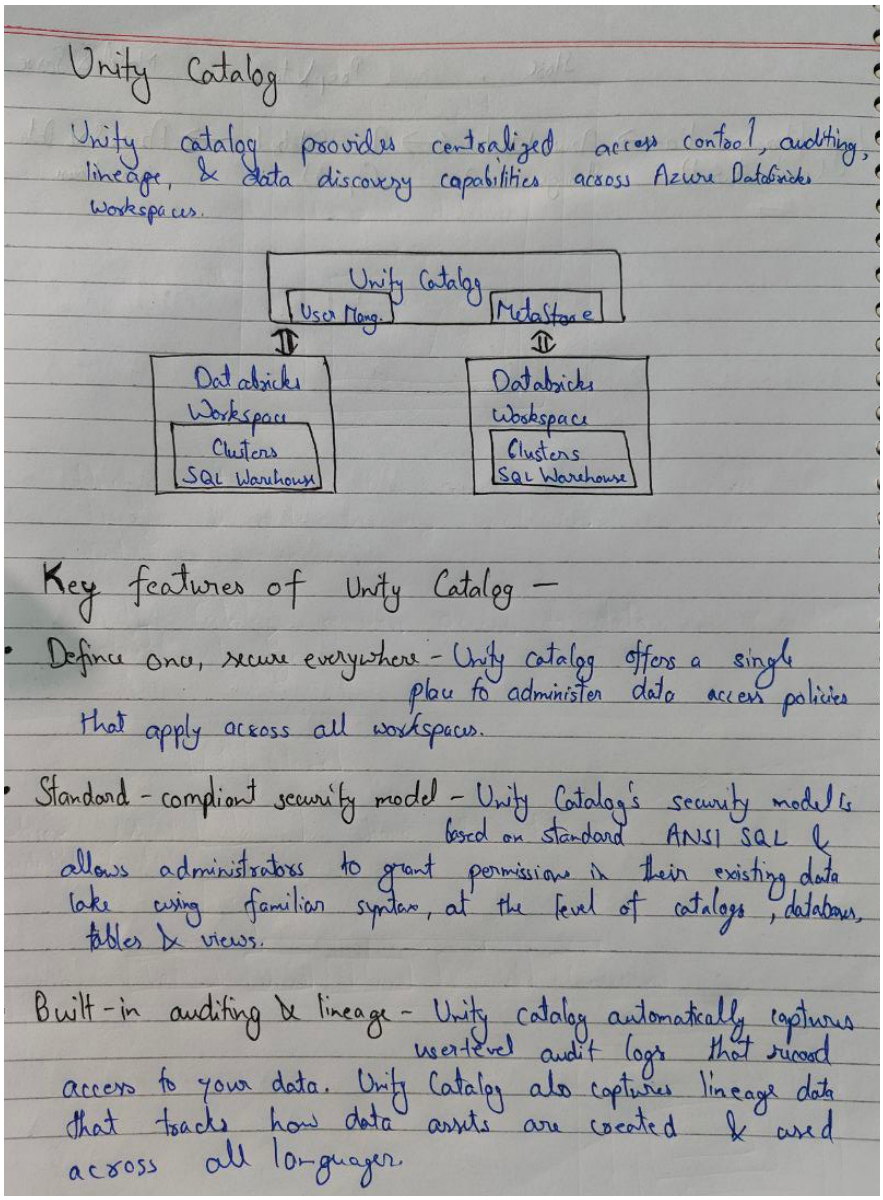
Assignment-21

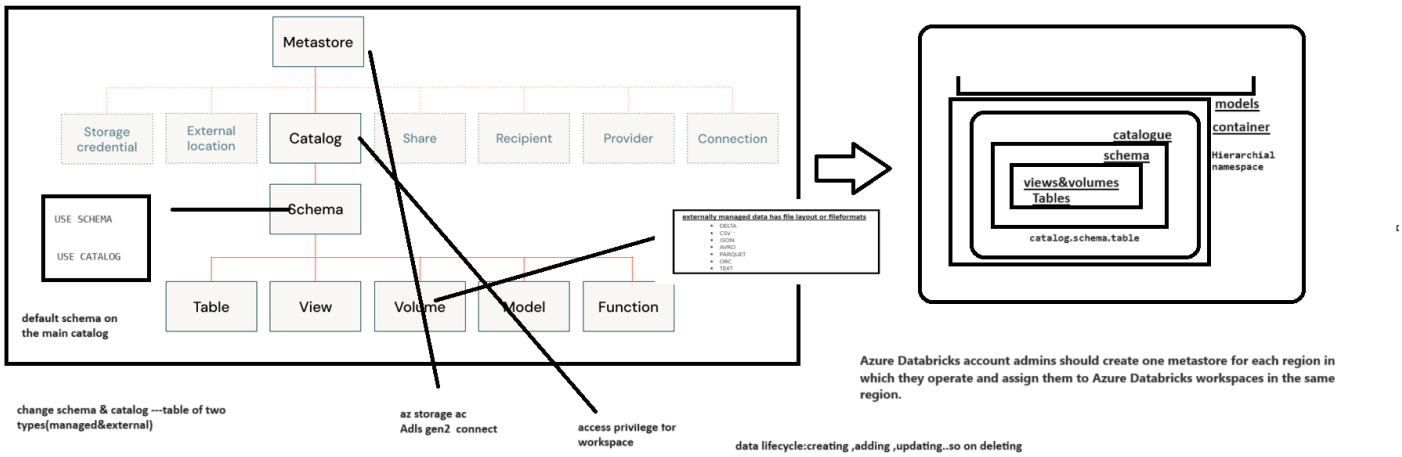
Name – Ajay Chaudhary
Batch-Data Engineering(Batch-1)

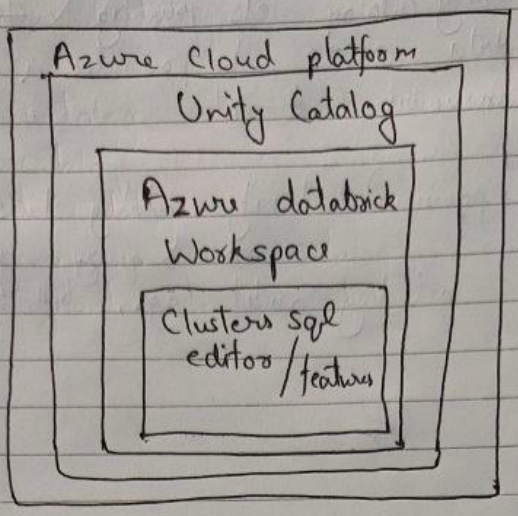
Handwritten notes-



Data discovery → Unity catalog lets you tag & ~~docum~~ document data assets, & provides a search interface to help data consumers find data.

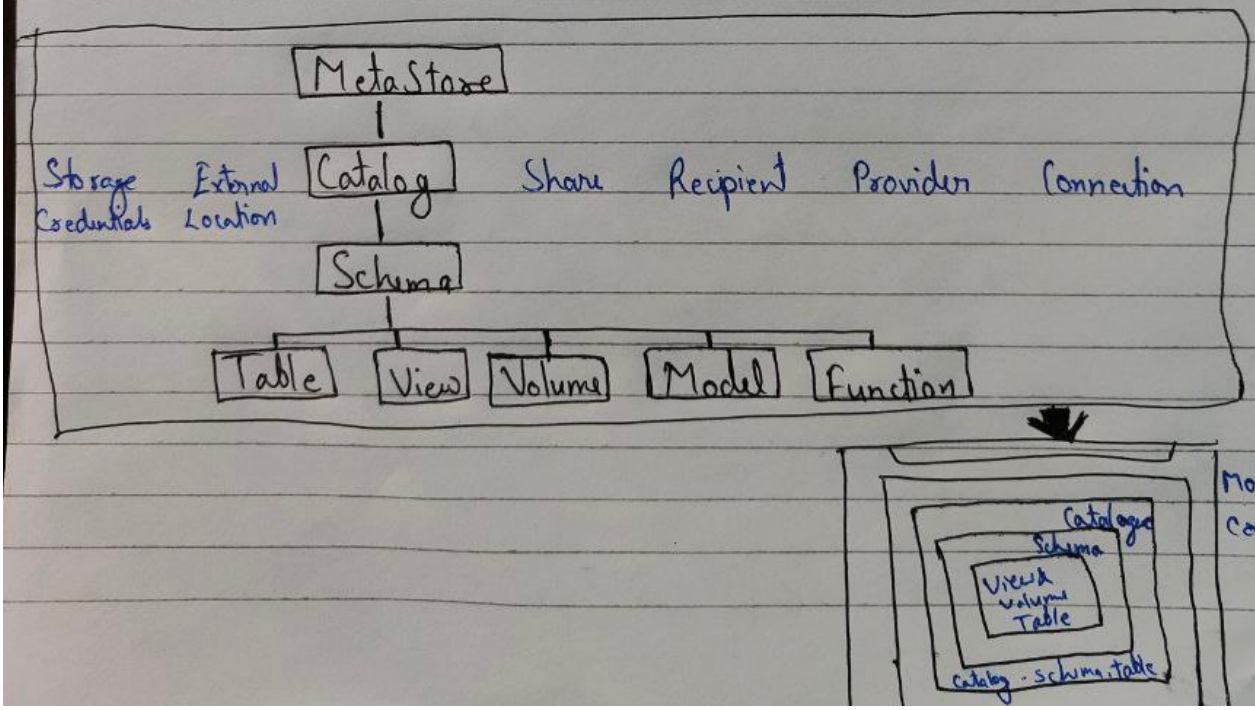
System tables (Public preview) → Unity catalog lets you easily access & query your account's operational data, including audit logs, billable usage & lineage.





Unity catalog provides centralised access control, auditing, lineage & data discovery capabilities across Azure Databricks workspaces.

Unity Catalog govern access to data in cloud object storage.



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 & USE Schema permission on the schema, & they must have 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 & file layout for these tables. You should not use tools outside of Databricks

By default stored in root storage location.

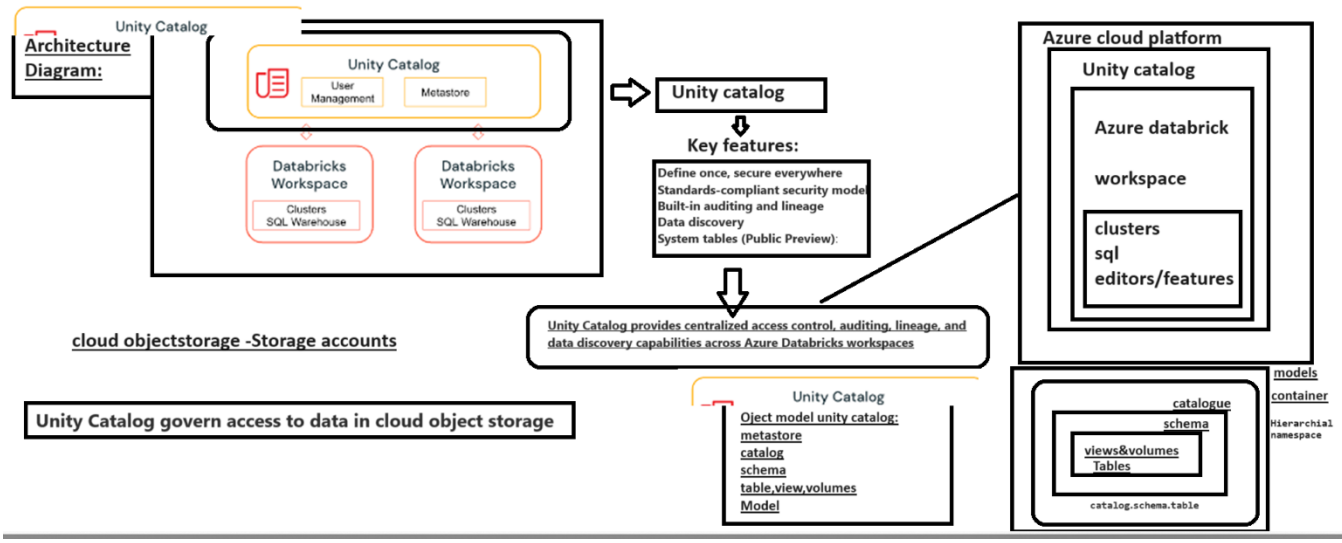
External tables

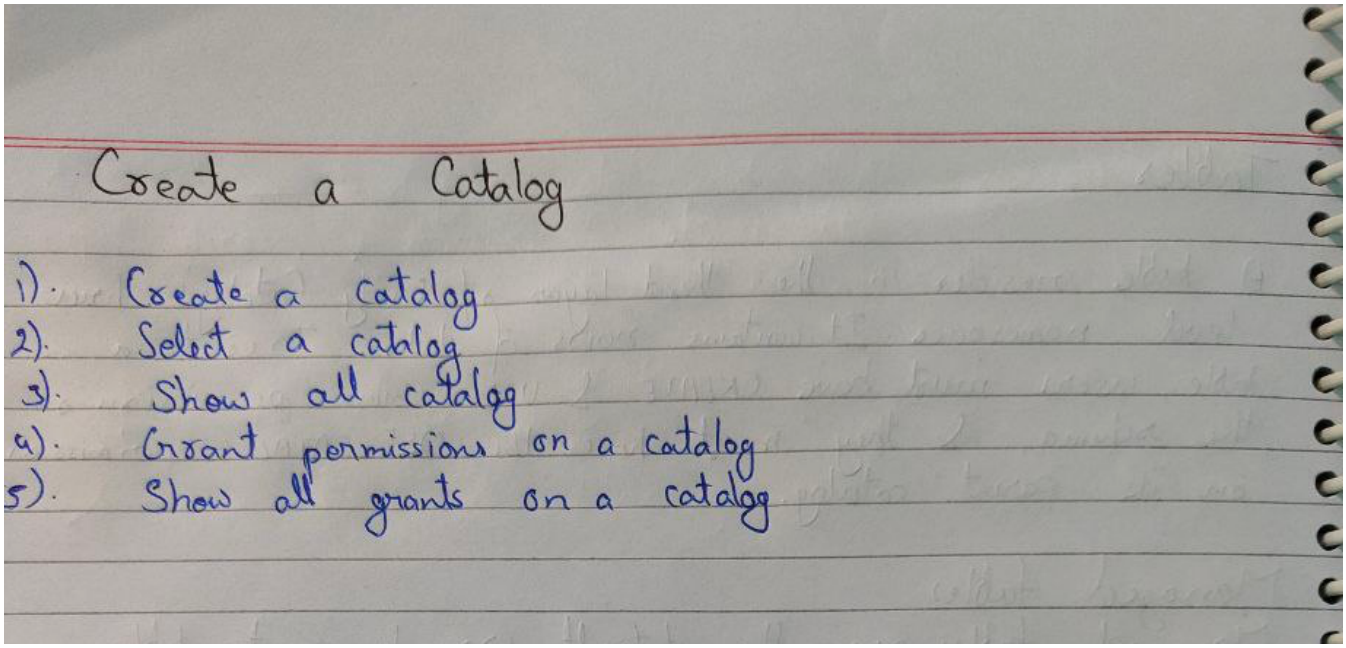
Ext. Tables are those tables whose data lifecycle & file layout are not managed by Unity Catalog.

Use external tables to register large amount of existing data in Unity Catalog or if you require direct access to the data using tools outside of Azure Databricks clusters or ~~Databricks~~ Databricks SQL warehouse.

External managed data has file layout ~~for~~ or fileformats

- Delta
- CSV
- JSON
- AVRO
- PARQUET
- ORC
- TEXT





databricks

unity-catalog-quickstart-python (Python)

Import Notebook

```
# Create a catalog.
spark.sql("CREATE CATALOG IF NOT EXISTS quickstart_catalog")

# Create a catalog and specify the managed location
# spark.sql("CREATE CATALOG IF NOT EXISTS quickstart_catalog MANAGED LOCATION '<location-path>'")
```

```
# Set the current catalog.
spark.sql("USE CATALOG quickstart_catalog")
```

```
# Show all catalogs in the metastore.
display(spark.sql("SHOW CATALOGS"))
```

```
# Grant create and use catalog permissions for the catalog to all users on the account.
# This also works for other account-level groups and individual users.
spark.sql("""
  GRANT CREATE, USE CATALOG
  ON CATALOG quickstart_catalog
  TO `account users`""")
```

```
# Show grants on the quickstart catalog.
display(spark.sql("SHOW GRANT ON CATALOG quickstart_catalog"))
```