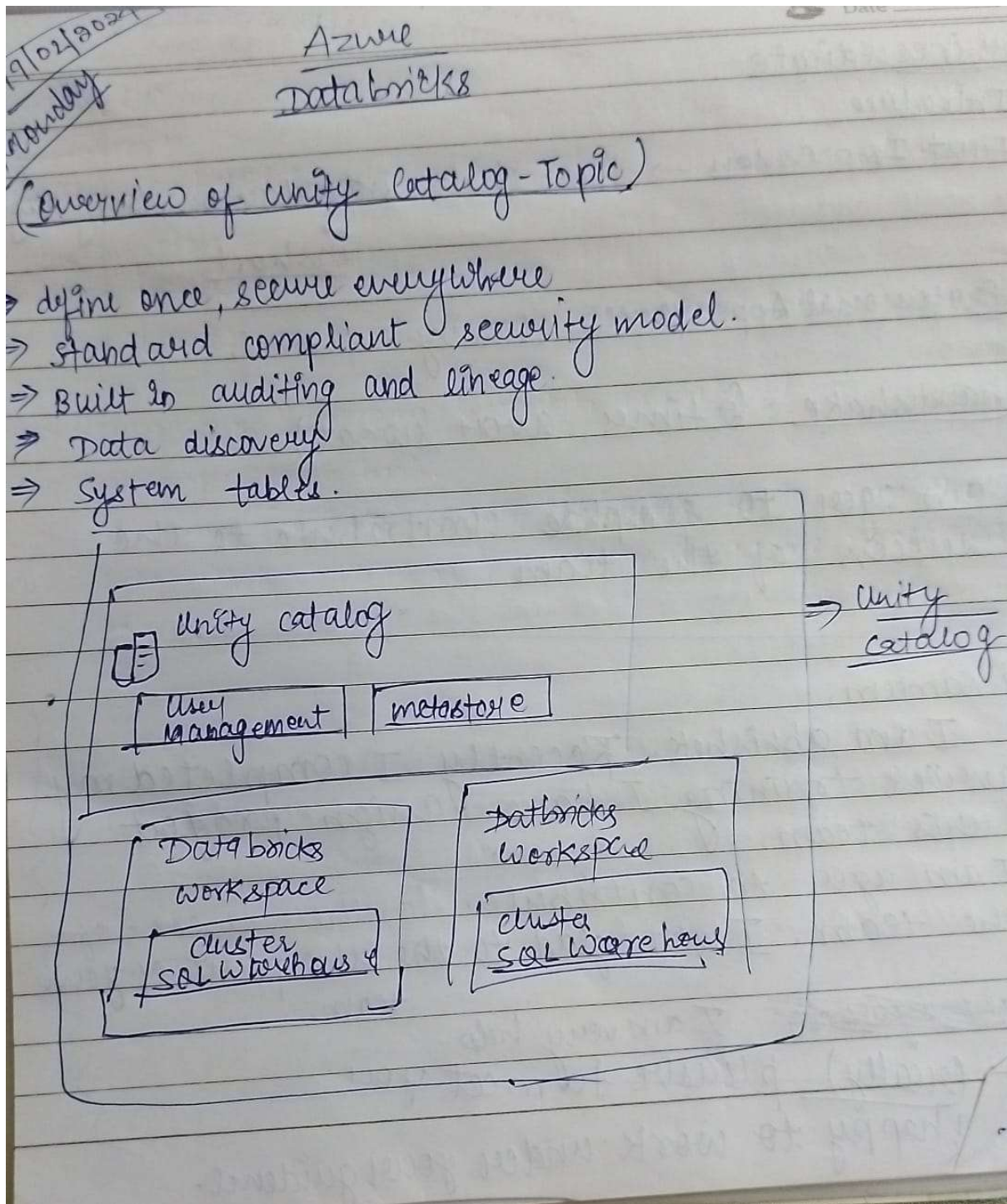


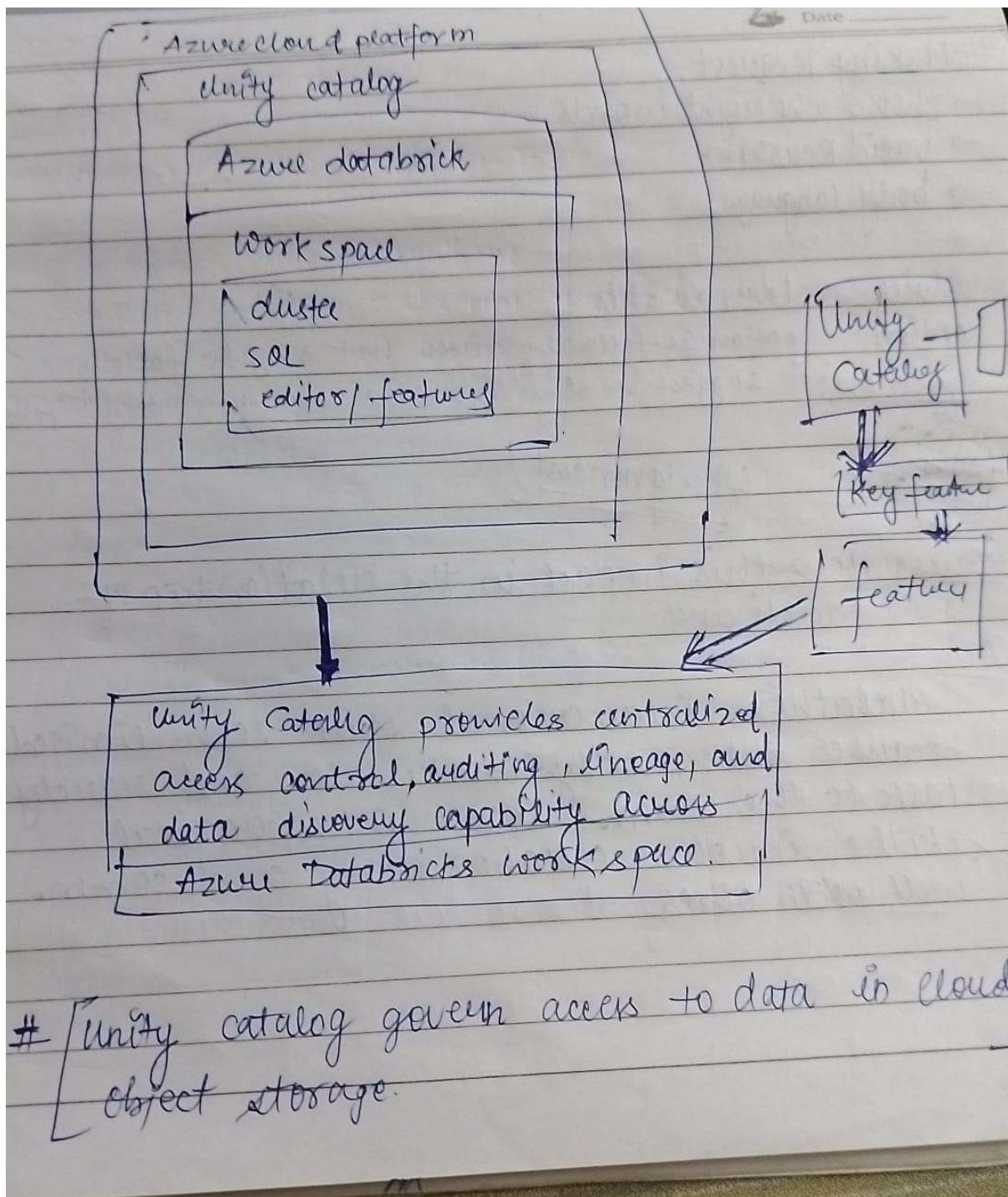
Name: Abhishek Kanoujia

DATA ENGINEERING BATCH 1

DAY 22 ASSIGNMENT

Class hand written notes:-







## object model unity catalog:

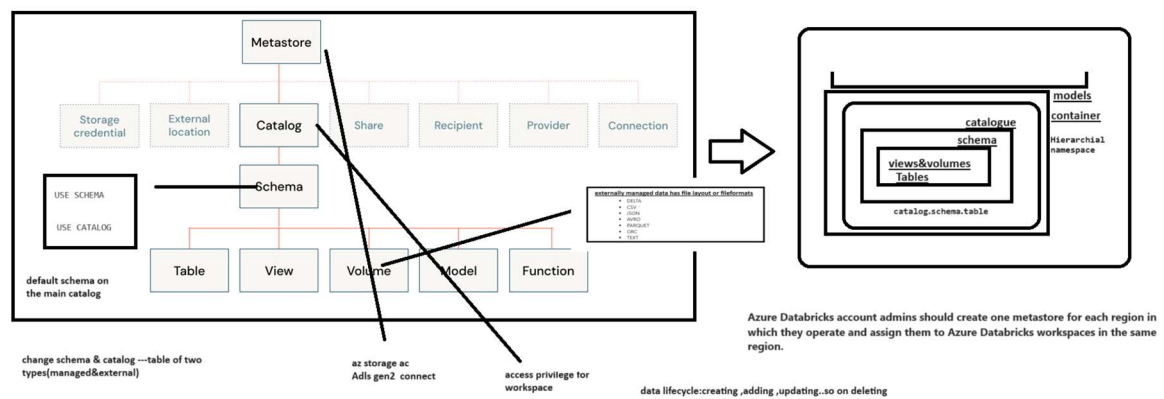
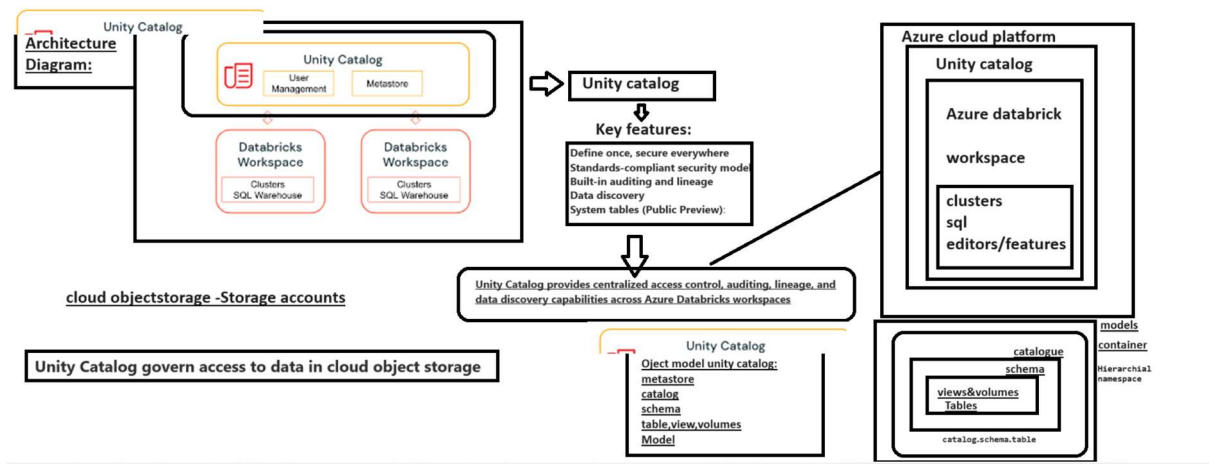
- Metastore
- catalog
- Schema
- table, view, volumes.
- Model.

## Extental table:

[data lifecycle: creating, adding, updating, deleting]

Externally managed data has file layout or file formats:

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



# Create a catalog

**# 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"))
```

# Create and manage schemas (databases)

**Schemas, also called databases, are the second level of the Unity Catalog three-level namespace. Schemas logically organize tables and views.**

**# Create a schema in the catalog that was set earlier.**

```
spark.sql("""  
CREATE SCHEMA IF NOT EXISTS quickstart_schema  
COMMENT 'A new Unity Catalog schema called quickstart_schema'""")
```

**# Show schemas in the catalog that was set earlier.**

```
display(spark.sql("SHOW SCHEMAS"))
```

**# Describe the schema.**

```
display(spark.sql("DESCRIBE SCHEMA EXTENDED quickstart_schema"))
```

**# Grant create table, and use schema permissions for the schema to all users on the account.**

**# This also works for other account-level groups and individual users.**

```
spark.sql("""
GRANT CREATE TABLE, USE SCHEMA
ON SCHEMA quickstart_schema
TO `account users`""")
```

## Create a managed table

Managed tables are the default way to create table with Unity Catalog. The table is created in the managed storage location configured for the metastore, catalog, or schema.

The following commands show how to:

1. Select a schema.
2. Create a managed table and insert records into it.
3. Show all tables in a schema.
4. Describe a table.

**# Set the current schema.**

```
spark.sql("USE quickstart_schema")
```

**# Show the current database (also called a schema).**

```
spark.catalog.currentDatabase()
```

**# Create a managed Delta table in the catalog that was set earlier.**

```
spark.sql("CREATE OR REPLACE TABLE quickstart_table (id STRING)")
```

**# Grant select and modify permissions for the table to all users on the account.**

**# This also works for other account-level groups and individual users.**

```
spark.sql("""
  GRANT SELECT, MODIFY
  ON TABLE quickstart_table
  TO `account users`""")
```

**# List the available tables in the catalog that was set earlier.**

```
display(spark.sql("SHOW TABLES"))
```

**# Insert 10 rows into the table.**

```
spark.range(10).selectExpr("id").write.insertInto("quickstart_table")
```

**# Show the table.**

```
display(spark.table("quickstart_table"))
```

**# Show all of the available tables in the schema.**

```
display(spark.sql("SHOW TABLES in quickstart_schema"))
```