# Concepts

## Data storage

Databricks doesn’t store data for us. We need to use an external data storage and Databricks only helps us to manage this data and make calculations on it.

The most common is object storage (S3, ADLS, HDFS) but it can also work with data saved in SQL and other databases.

## Unity catalog

Unity catalog has a few main purposes:

* Storing tables metadata – enabling using SQL queries on them, similarly to Hive
* Providing access control – Regulates access to data
* Providing data lineage
* Providing three level namespace: catalog.schema.table – Catalogs consinst of schemas and schemas consist of tables.
* Providing metadata management through API – Using Databricks Rest API we can manage catalogs, schemas, permissions.

## Volumes

Volumes help with managing data (saved in for example ADLS).

We create volumes in Unity Catalog’s catalog / schema namespace: catalog.schema.volume\_name.

Each volume refers to a folder in an external data storage. For example if we have a volume catalog.schema.volume1 which refers to folder1/folder2 in an external storage, then in order to access the folder1/folder2/folder3/file.csv file in databricks by using this path:

* /Volumes/catalog/schema/volume1/folder3/file.csv

Instead of a full path:

* folder1/folder2/folder3/file.csv

Volumes are available only when using Unity Catalog. We can also manage volumes like tables in Unity Catalog (permissions etc).

## Managed tables

When we create a managed table then we don’t specify where in an external storage that table’s data will be stored. Databricks will handle where to save this data for us.

If we drop a managed table then Databricks deletes the data saved in external location.

## Unmanaged tables (external tables)

When we create unmanaged table then we need to specify where in an external storage data of that table will be stored.

If we drop an unmanaged table then Databricks doesn’t delete the data saved in external location. Only the metadata is removed.

## Delta lake

Databricks support Delta lake. We can use it natively in Databricks.