Data objects in Lakehouse :-

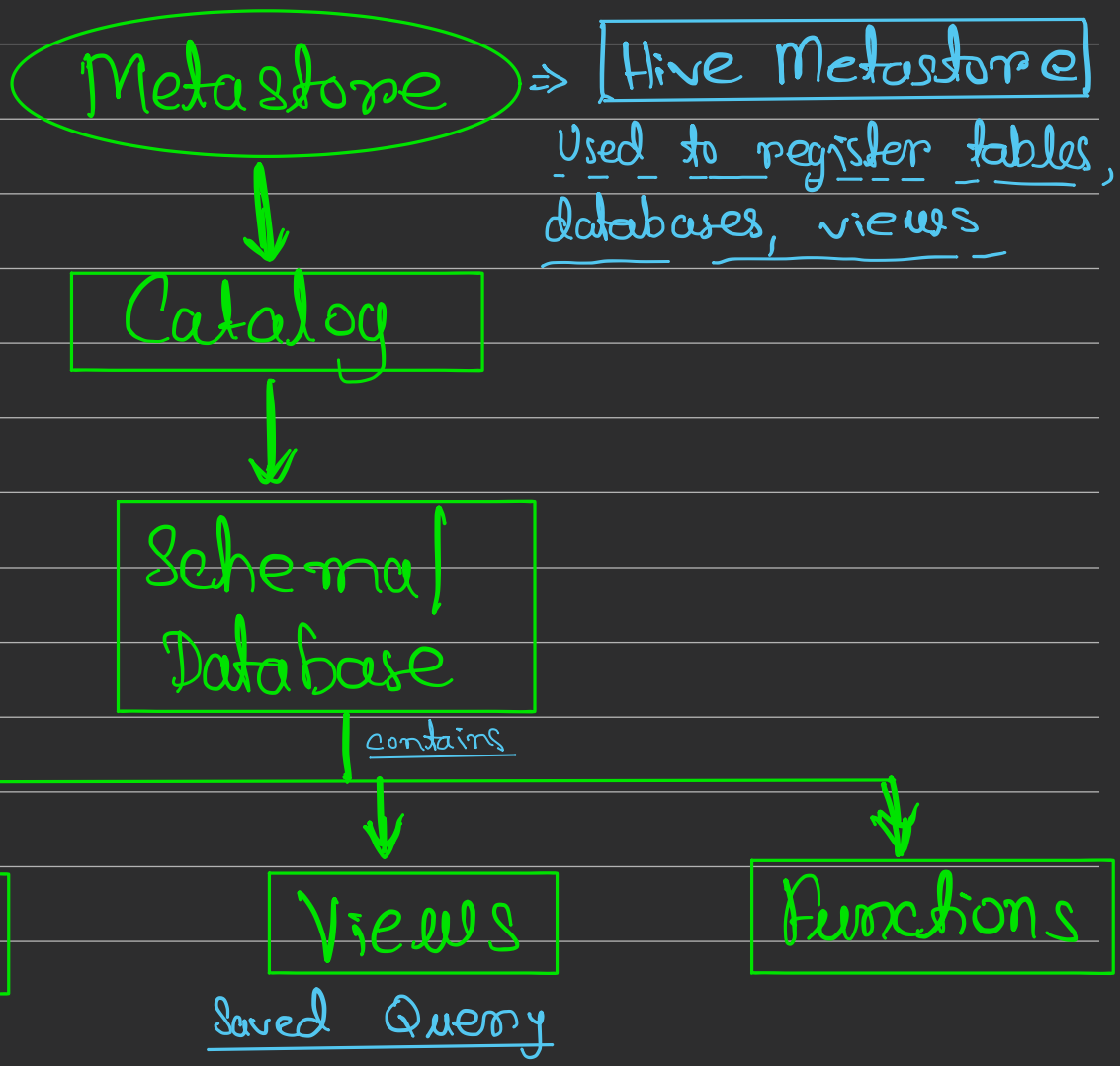
Databricks Lakehouse Architecture combines data stored with Delta Lake protocol in cloud object storage with metadata registered to a metastore.

5 primary objects in Databricks Lakehouse :-

a) Catalog : Group of databases.

b) Database / Schema : Group of objects. Resides in a Catalog.

c) Tables, d) Views, e) Functions.



Tables :- → Managed Table
→ External Table

stored as data files

Managed Table :

- Managed Tables are based on files stored in the managed storage location that is configured into the metastore.
Both data and metadata of manged table is managed.
- When a managed table is dropped, both the metadata & underlying data is deleted.

External Table :

- External Tables are tables whose data files are stored in a cloud storage location outside of manged storage locations.
Only the metadata is managed (not the data).
- When external table is dropped, the underlying data files remains.

Extracting Data Directly from Files using SparkSQL :-

Here we will directly apply SQL commands to fetch and display data directly from files.

To specify the files, we can use **absolute** or **relative** path.

The relative path will contain a path of the part contained in a variable, say, test, and in the path, it will be placed as :- ${test}

will be replaced by the string value (of path) it contains.

Can be json, parquet, delta

using back-ticks is must

Syntax :-

```
SELECT * FROM <file_format>.<filepath.extension>
```

Ex :-

1)
```
SELECT * FROM json.`/FileStore/flight-data.json`
```

backticks

we are querying from JSON file, thus json

absolute file path

② `SELECT * FROM json.` `${DA.paths}/001.json`

relative path

⊛ Querying All Files in a Directory :-

⇒ All the files in the directory must have same format and schema.

Provide the directory path instead of the filepath

```
SELECT * FROM json.`/filestore/folder/`
```

. all json files in this folder will be queried

⊛ Saving above Result as VIEW :-

```
CREATE OR REPLACE TEMP VIEW test_view
AS SELECT * FROM json.`/filestore/folder`
```

We can use this view in later point, to get this entire data.

```
SELECT * FROM test_view;
```

⇒ Views can also be temporary → CREATE OR REPLACE TEMP VIEW

≥ Temporary views only last till the current SparkSession thus isolated to current notebook, job.

Self-describing Formats	Not Self-describing formats
json, parquet, delta, orc, avro	csv, text ↓ not useful if we query these files directly

Ex:-

```
SELECT * FROM text.`${DA.path}/file.json`
```

will give each line as raw string with a column named value

≥ This is useful when we want to use custom text-parsing functions, to parse the data in files.

⊛ Extract Raw Bytes & Metadata of Files:-

```
SELECT * FROM binaryFile.`<path>`
```

↓

provides metadata & binary representation of files. Useful for images, unstructured data.