Fig: HBa

Unit -3

=> Introduction to Hive:-

-> What is HIVE?

- * Data warehousing tool that sits on top of Hadoop.
- * Used to process structured data in Hadoop.
- * 3 main tasks performed by Apache Hive are:-
  1. Summarization
  2. Querying
  3. Analysis
- * Hive makes use of:-
  1. HDFS for storage
  2. MapReduce for execution.
  3. Store metadata/ schemas is RDBMS.

* Hive provides HQL (Hive Query Language) or HiveQL which is similar do SQL.

* Hive compiles SQL queries into MapReduce jobs & then runs the job in the Hadoop Cluster.

* designed to support OLAP (Online Analytical Processin

* provides extensive data type functions & formats for data summarization & analysis.

- Is not RDBMS
- not designed to support OLTP
- not designed for real time queries
- Not designed to support row-level updates.

-> History & Recent Release of Hive:-

2007 — Hive was born at Facebook to analyze their incoming log data.

2008 — Hive became Apache Hadoop subproject

Fig:- History of Hive

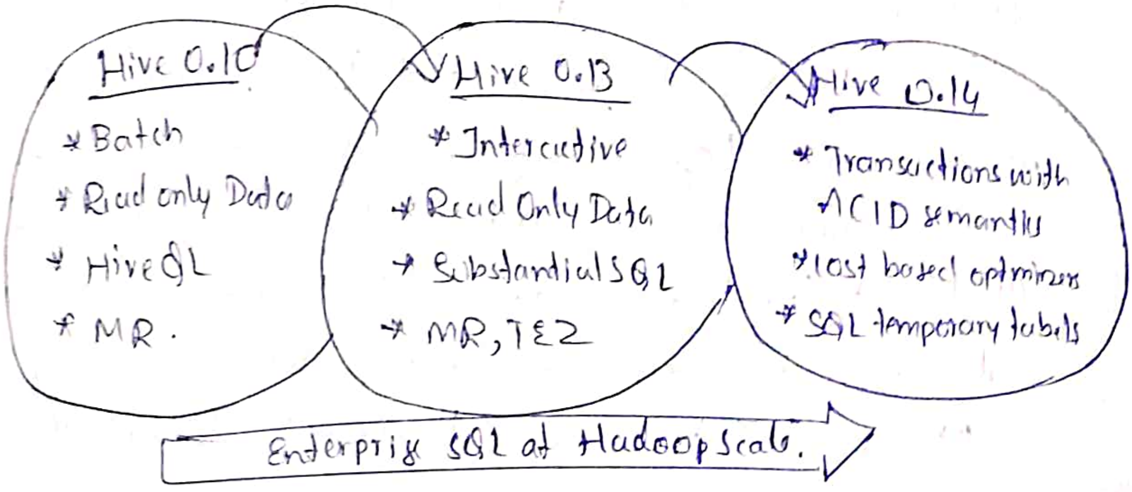

Fig:- Recent Release of Hive

-> Features:-

* similar to SQL
* is easy to code.
* supports rich data types such as structs, lists & maps.
* supports SQL filters, groupby & order by clauses
* custom Types, custom functions can be defined.

-> Hive Integration & work flow:-

Explanation of workflow:- Hourly log Data can be stored directly into HDFS and then data cleansing is performed on the log file. Finally, Hive table(s) can be created to query the log file.

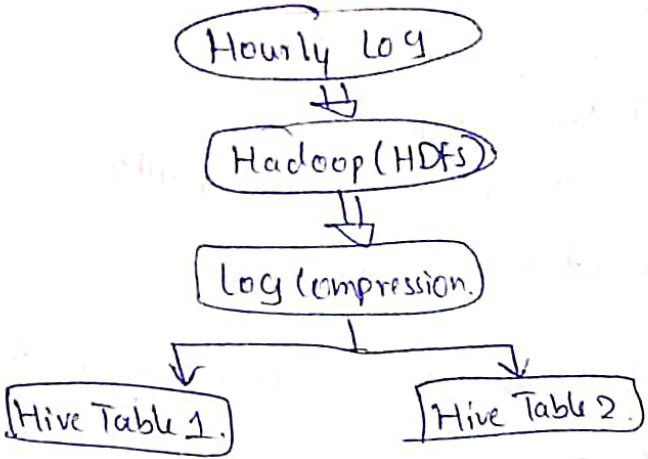

Fig:. Flow of log Analysis file

→ **Hive Data Units:-**

1. Databases: The namespace for tables.
2. Tables: set of records that have similar schema
3. Partitions:- Logical separation of data based on classification of given information as per specific attributes
4. Buckets (or clusters):- Similar to partitions but uses hash function to segregate data & determines the cluster or bucket into which the record should be placed.

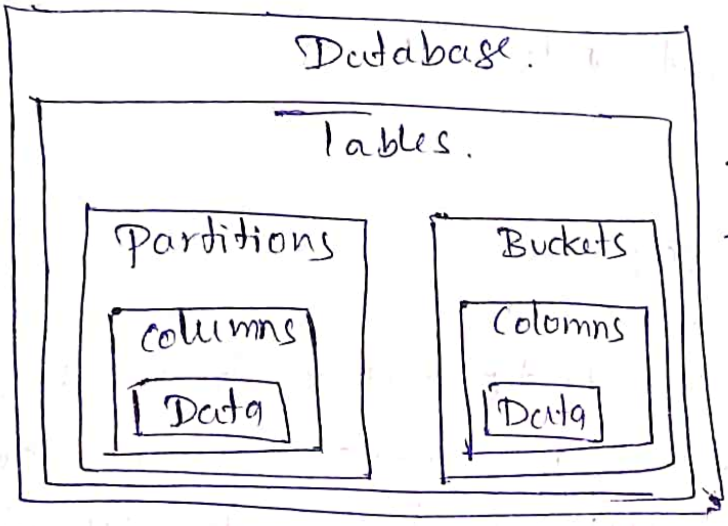

Fig: Data units as arranged in a Hive.

→ **Hive Architecture:-**

1. Hive Command-Line Interface (Hive CLI): most commonly used interface to interact with Hive.
2. Hive Web Interface:- It's a simple Grahic User Interface to interact with Hive & to execute query.
3. Hive Server:- an optional server, can be used to submit

Hive Jobs from a remote client.

4. JDBC/ODBC: Jobs can be submitted from a JDBC client. One can write a Java code to connect to Hive and submit jobs on it.

5. Driver:- Hive queries are sent to the driver for compilat-ion, optimization & execution.

6. Metastore:- table definitions & mapping to the data are Stored in a Metastore. A metastore consists of:-

* Metastore service:- Offers interface to the Hive
* Database: Store data definition, mappings to the data & others.

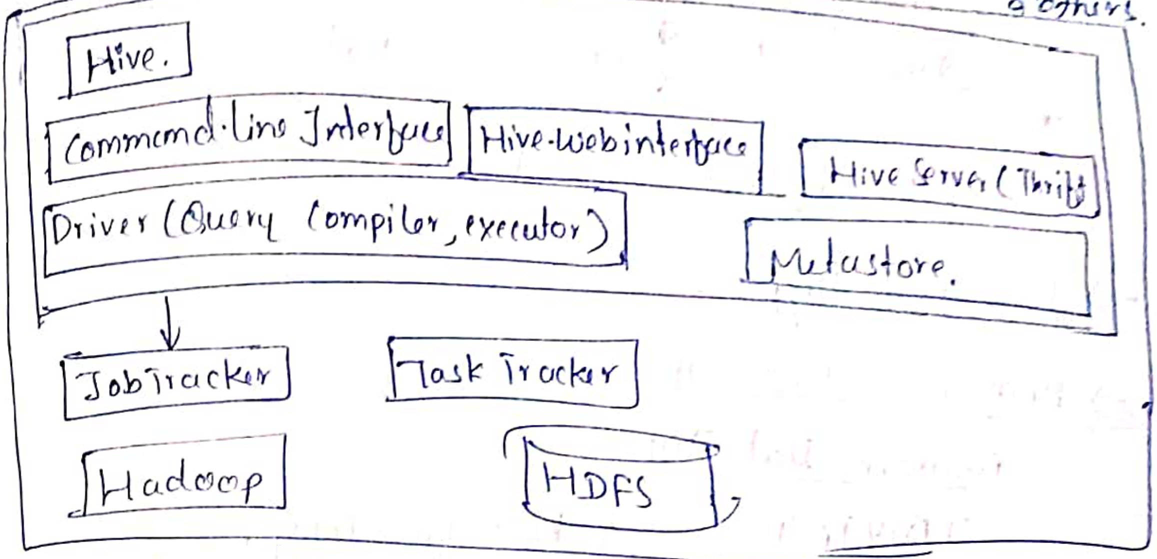

Fig:- Hive Architecture

-> The metastore is updated whenever a table is created or deleted from Hive. 3 kinds of metastore:-

① Embedded metastore :- used bor unit tests

② Local Metastore : Metadata stored in RDBMS (MySQL)

③ Remote MetaStore:- driver & metastore interface run on dibberent JVMs.

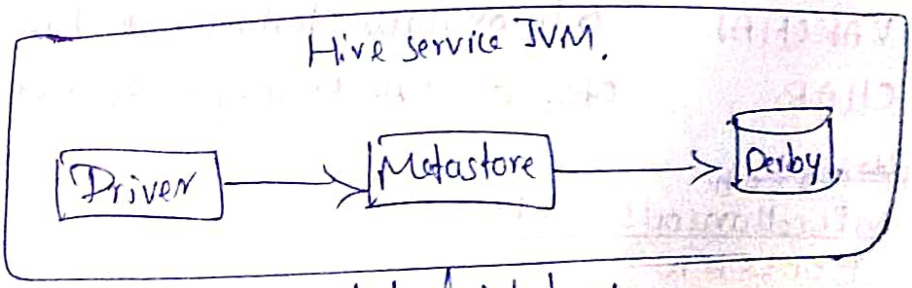

Fig:- Embedded Metastore.

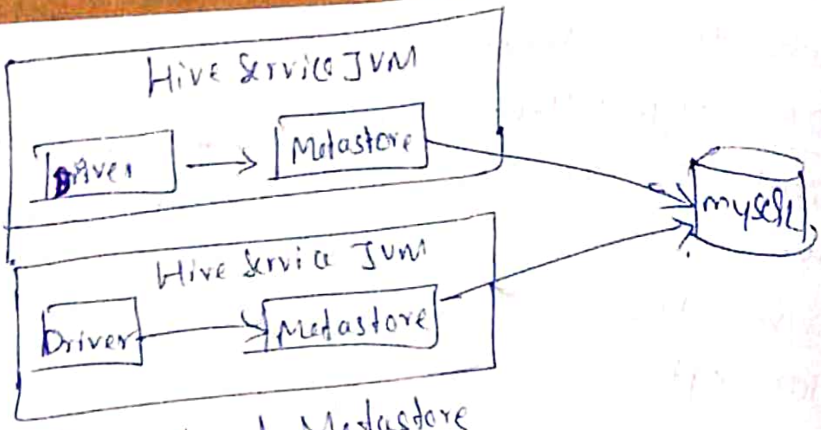

fig:- Local Metastore

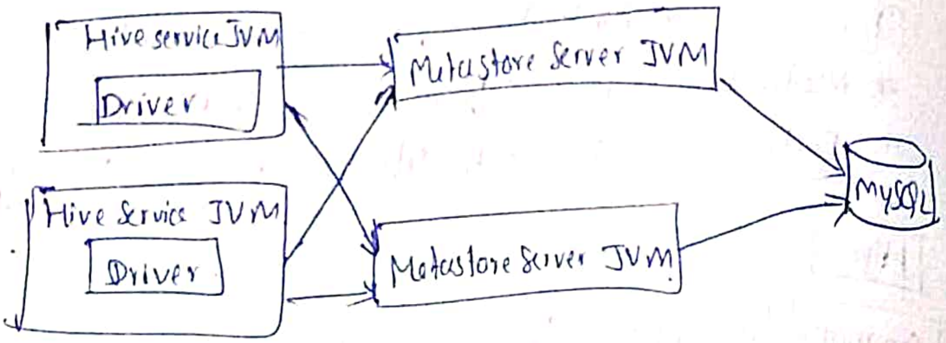

fig:- Remote Metastore

→ Hive Data Types:-

→ Primitive DataTypes:-

Numeric Data Type

TINYINT	1-byte signed integer
SMALLINT	2-byte signed integer
INT	4-byte signed integer
BIGINT	8-byte signed integer
FLOAT	4-byte single-precision floating-point.
DOUBLE	8-byte double-precision floatingpoint number.

String Types ('') or ("")

STRING	
VARCHAR	only available starting with Hive 0.12.0
CHAR	only available starting with Hive 0.13.0

~~String cab~~

Miscellaneous Types

BOOLEAN	
BINARY	only available starting with Hive

→ collection Data Types:-

struct :- Similar to 'C' struct, Fields are accessed using dot notation.

MAP :- A collection of key-value pairs, Fields are accessed using [] notation.

ARRAY :- ordered sequence of same types. Fields are accessed using array index

→ Hive File Format:-

* specify how records are encoded in a file.

→ Text file:-

* each record is a line in the file.
* different control characters are used as delimiters.
* The delimeters are ^A (octal 001, separates all fields), ^B (octal 002, separates the elements in the array or struct), ^C (octal 003, separates key-value pair), and \n.
* "field" is used when overriding the default ^A delimiter.
* Supported text files are CSV & TSV, also JSON or XML

→ Sequential file:-

* flat files that stores binary key-value pairs, It includes compression supports which reduces the CPU, I/O requirement.

→ RCFile:- (Record columnar File):-

* stores data in column oriented manner which ensures that Aggregation operation is not an expensive operation.