Apache Hive

Apache Hive is a data warehouse system built on top of Hadoop.

It allows users to query large datasets stored in HDFS using a SQL like language called HiveQL (HQL)

Just with MR, working with hive is going to make our life a lot easy.

Why was Hive Created ?

Complex Java based MR program

SQL kind of interface to work on the data ⇒ HQL

developed at FB (2010)

SQL — no Java or MR code

Analogy

Librarian with a Mobile / IPad / smart device

PBs of data

How Hive Make Big Data Processing Easier

higher level abstraction layer
over Hadoop

SQL like
HQL → MR, Spark or Tez

```
public class CustomerCount extends Mapper<LongWritable,
    Text, Text, IntWritable> {
    public void map(LongWritable key, Text value,
        Context context) {
        String[] fields = value.toString().split(",");
        context.write(new Text(fields[2]), new
            IntWritable(1));  // City as key
    }
}

public class CustomerCountReducer extends Reducer<Text,
    IntWritable, Text, IntWritable> {
    public void reduce(Text key, Iterable<IntWritable>
        values, Context context) {
        int sum = 0;
        for (IntWritable val : values) {
            sum += val.get();
        }
        context.write(key, new IntWritable(sum));
    }
}
```

```
select city, count(*)
  from customers
  group by city
```

Architecture
in depth knowledge

Without Hive	With Hive
Writing Java-based MapReduce programs to process data	Writing SQL-like HiveQL queries
Complex and time-consuming development	Easier and faster query execution
Requires knowledge of Java and MapReduce	Anyone familiar with SQL can use Hive

Some Common Question / Misconception about Hive

1. Hive is a Database just like mySQL or PostgressSQL.

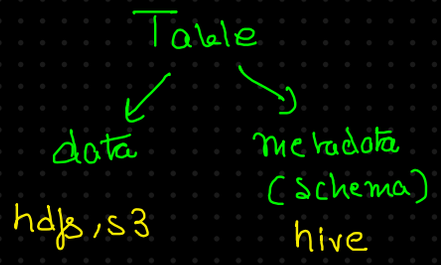
Hive is not a database but a data warehouse tool built on top of Hadoop.

not storing any data like MySQL
→ interface to query large datasets stored in HDFS

Storage
Query execution

hive is a metastore

Feature	Traditional Databases (MySQL, PostgreSQL)	Apache Hive
Storage	Stores data in structured tables on disks	Stores data in **HDFS** (distributed storage)
Query Processing	Executes queries in-memory	Uses **MapReduce, Tez, or Spark** for distributed processing
Schema	Schema-on-Write (Data must fit schema)	Schema-on-Read (Can query raw files)
Transaction Type	OLTP (Transactional systems)	OLAP (Batch processing, analytics)
Speed	Fast for small queries, real-time updates	Optimized for large-scale batch processing



* In big data world, any tech that supports Querying, might not be a DB

2. Hive is a replacement for Hadoop.

Hive is built on top of Hadoop & hadoop

Components
- HDFS
- YARN
- MR / Spark / Tez

like a translator
SQL like code
instead of MR

3. Hive Queries run like Normal SQL Queries in Databases

Hive does not process queries instantly
like mySQL ⟶ MR/Spark/Tez

~AI converts

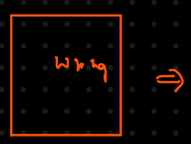
4. Hive can Perform Row-level Transaction like MySQL.

↓ dw olap

↓ dB oltp

PBs ⇓

Hive is not meant for row-level operations



5 Hive only works with HDFS

Wrong Hive can store & process data from

HDFS
S3
Azure datalake
GCS

6. Hive Tables Work Like Normal Database Tables.

Hive is a metastore and not storing tables.

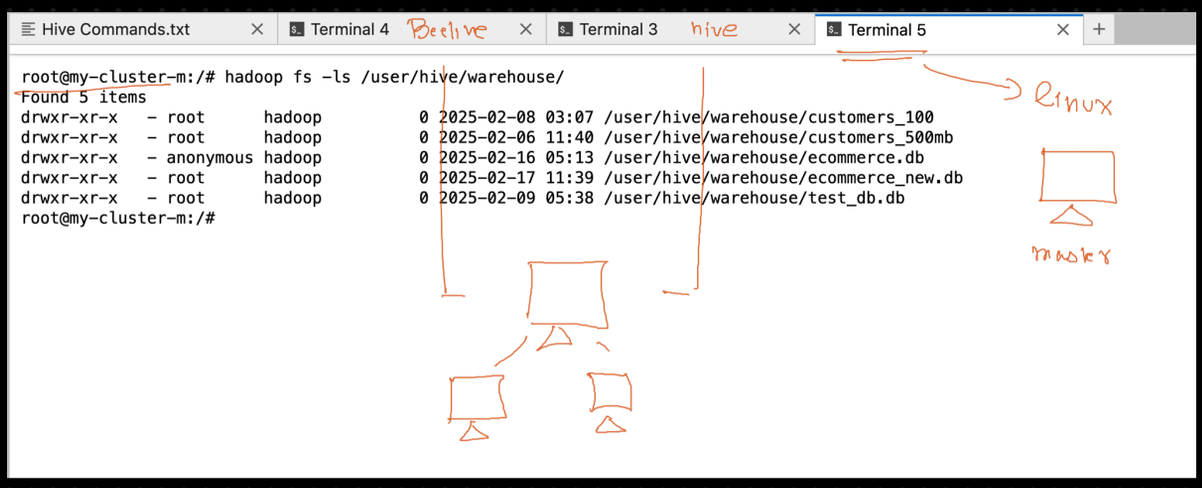
Feature	Hive Advantage
SQL-Like Interface	Allows non-programmers to work with big data.
Scalability	Handles **petabytes** of data.
Optimized Execution	Uses **MapReduce, Tez, or Spark** for distributed processing.
Storage Flexibility	Works with **HDFS, S3, Azure Blob, GCS**.
Schema Flexibility	Schema-on-read enables analyzing raw files.

Hive Practical on Google Dataproc

We can run hive commands via the hive terminal or the Beeline (recommended)

Feature	Hive CLI	Beeline (Recommended)
Connection Type	Directly connects to Hive	Uses JDBC for **remote** execution
Security	No authentication	Supports authentication (LDAP, Kerberos)
Multiple Connections	Single session only	Supports multiple concurrent sessions
Performance	More resource-heavy	Optimized for **query performance**
Recommended?	❌ Deprecated	✅ Yes, for production use

Hive Practical - 2



Accessing metadata in Hive