Introduction to Big Data

ECAP456

Dr. Rajni Bhalla

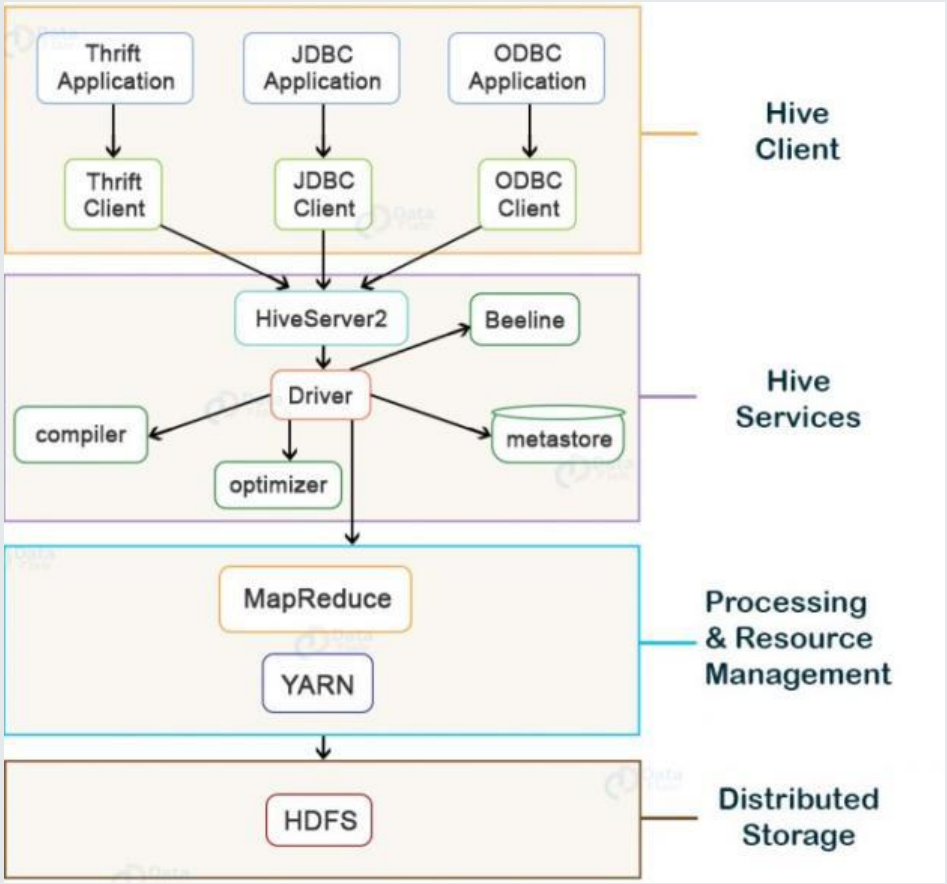
Associate Professor

Learning Outcomes

After this lecture, you will be able to

- learn services offered by HIVE
- explore concepts of HIVEQL

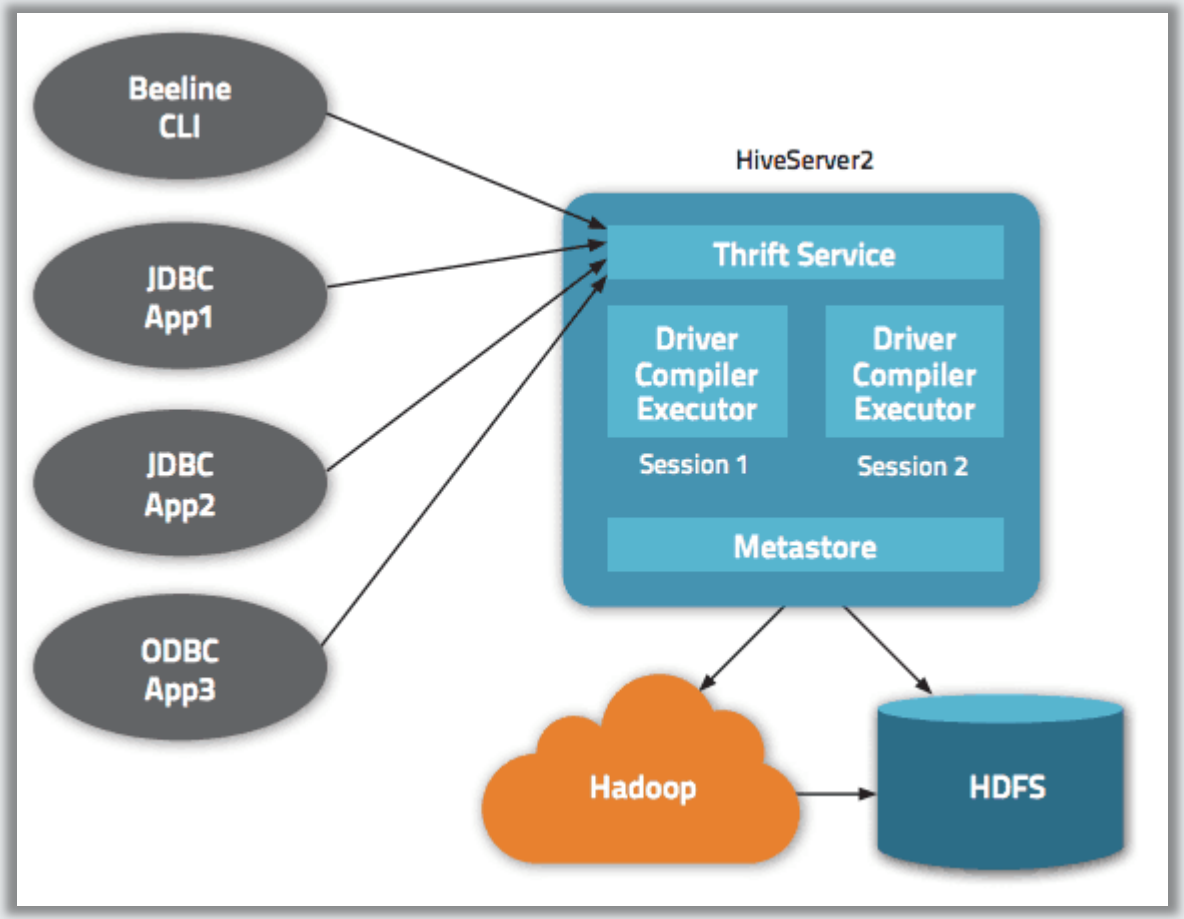
Introduction



HIVE Architecture

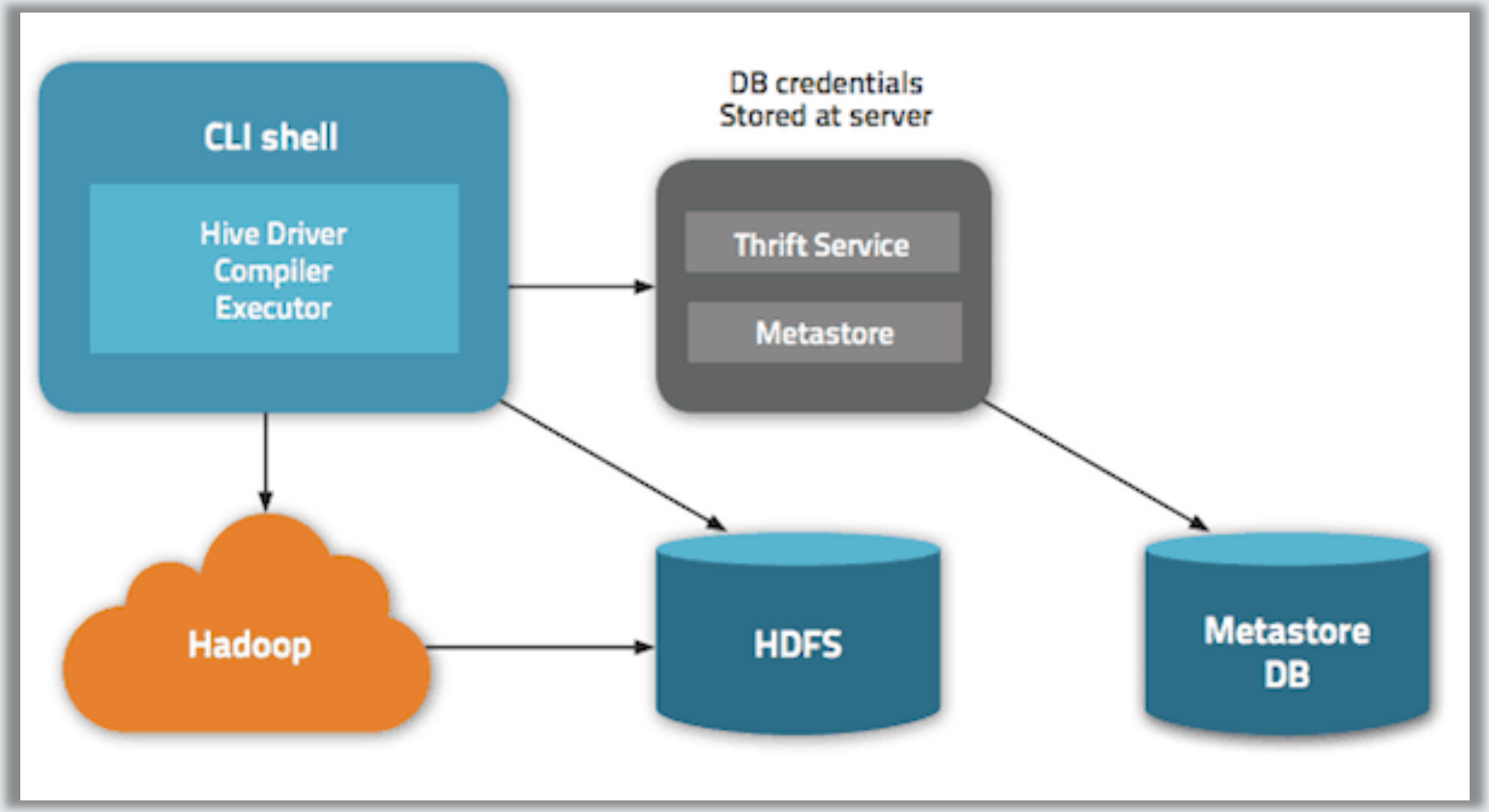
Services offered by HIVE

Beeline



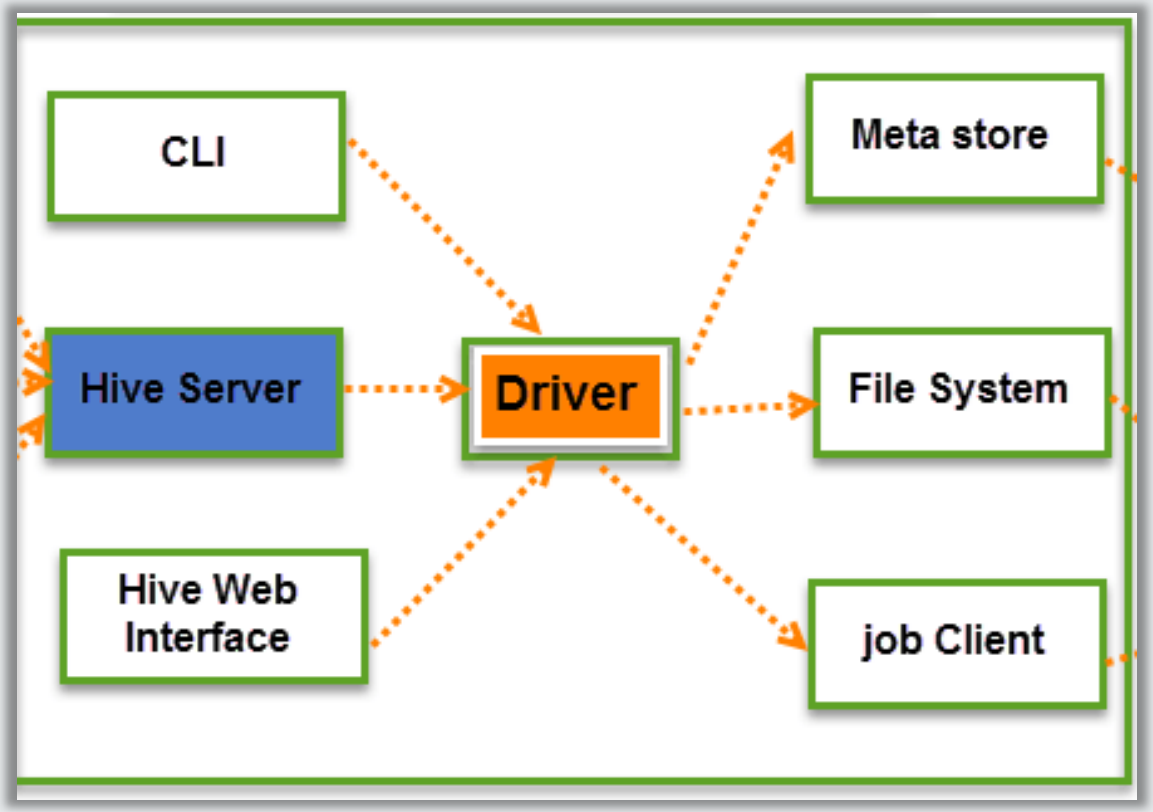
Services offered by HIVE

HIVE Server2



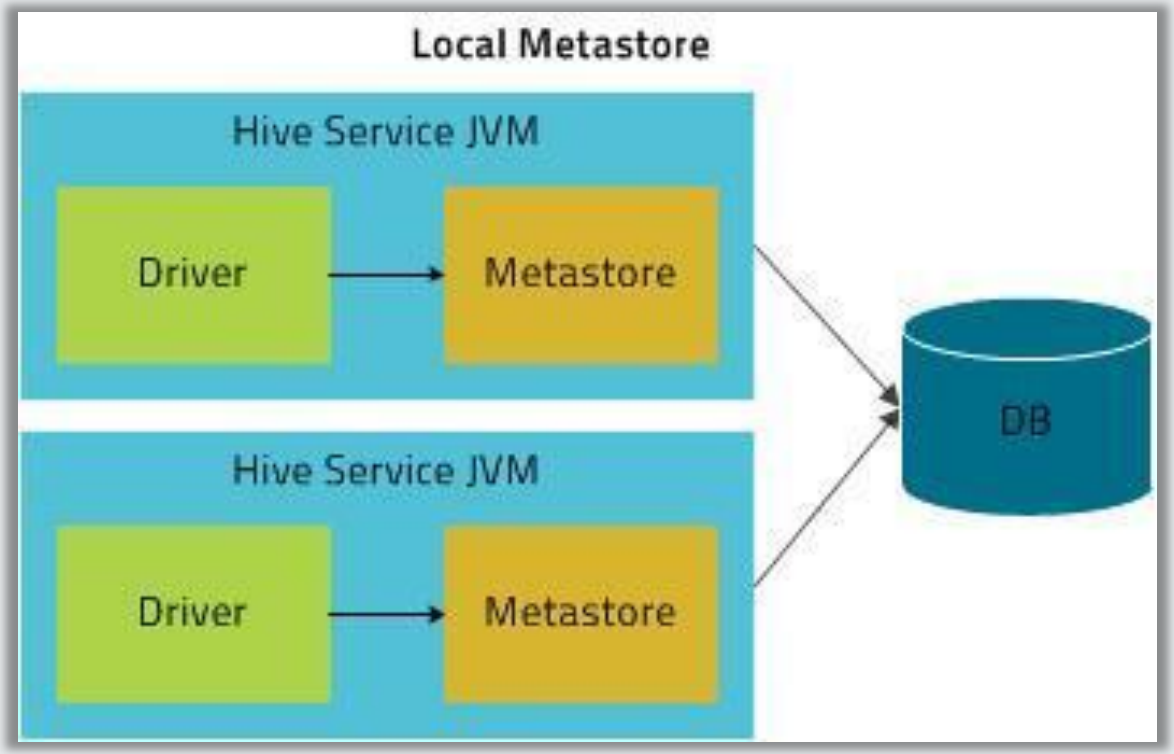
Services offered by HIVE

HIVE Driver



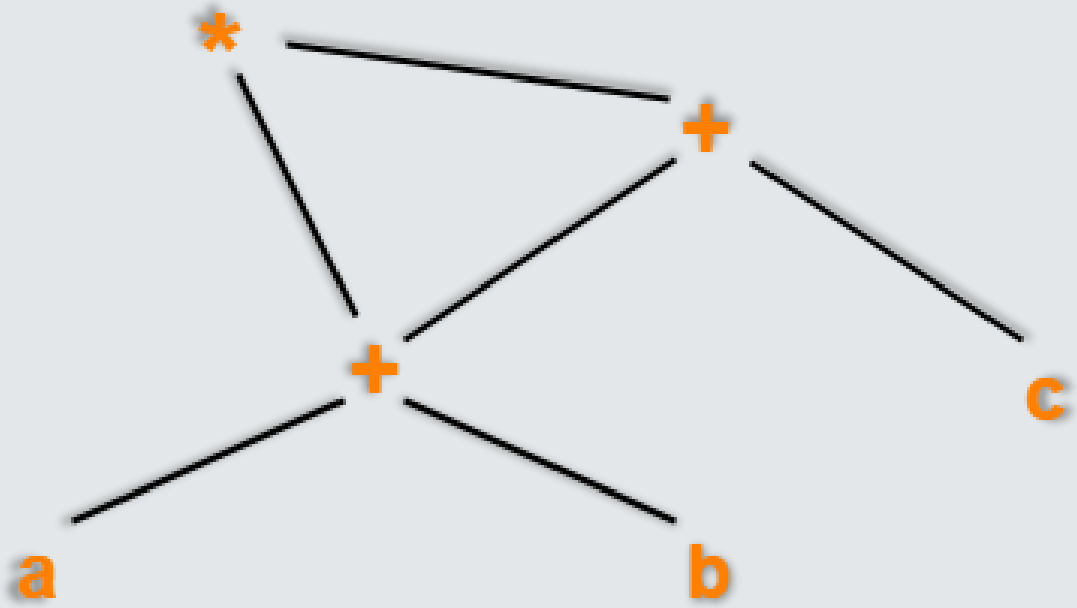
Services offered by HIVE

HIVE Complier



Services offered by HIVE

HIVE Complier



Directed Acyclic Graph

Services offered by HIVE

Optimizer

Services offered by HIVE

Optimization Techniques

- Tez-Execution Engine in Hive

Services offered by HIVE

Optimization Techniques

- Tez-Execution Engine in Hive
- Usage of Suitable File Format in Hive

Services offered by HIVE

Optimization Techniques

- Tez-Execution Engine in Hive
- Usage of Suitable File Format in Hive
- Hive Partitioning

Services offered by HIVE

Optimization Techniques

- Tez-Execution Engine in Hive
- Usage of Suitable File Format in Hive
- Hive Partitioning
- Bucketing in Hive

Services offered by HIVE

Optimization Techniques

- Tez-Execution Engine in Hive
- Usage of Suitable File Format in Hive
- Hive Partitioning
- Bucketing in Hive
- Vectorization In Hive

Services offered by HIVE

Optimization Techniques

- Tez-Execution Engine in Hive
- Usage of Suitable File Format in Hive
- Hive Partitioning
- Bucketing in Hive
- Vectorization In Hive
- Cost-Based Optimization in Hive (CBO)

Services offered by HIVE

Optimization Techniques

- Tez-Execution Engine in Hive
- Usage of Suitable File Format in Hive
- Hive Partitioning
- Bucketing in Hive
- Vectorization In Hive
- Cost-Based Optimization in Hive (CBO)
- Hive Indexing

Services offered by HIVE

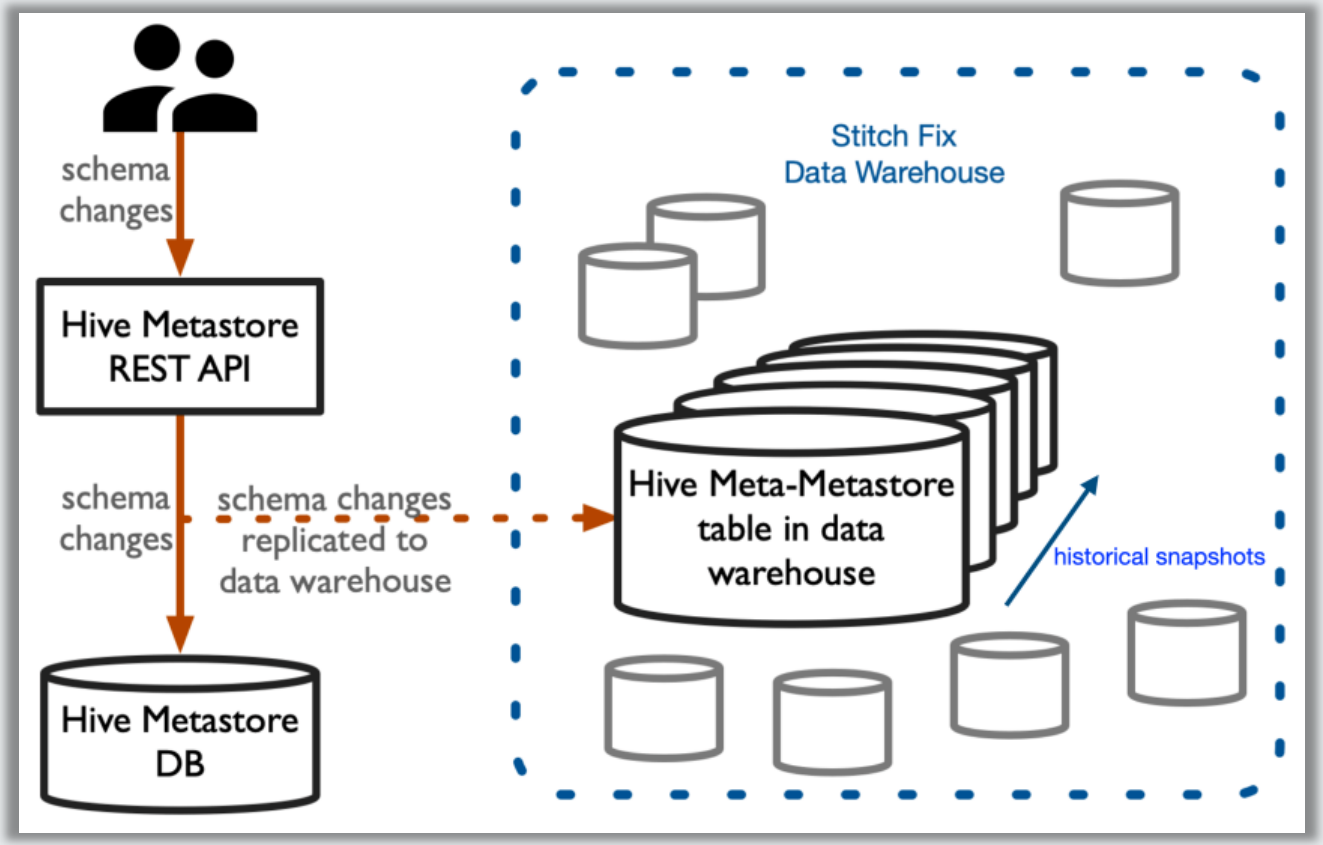
Services offered by HIVE

Execution Engine

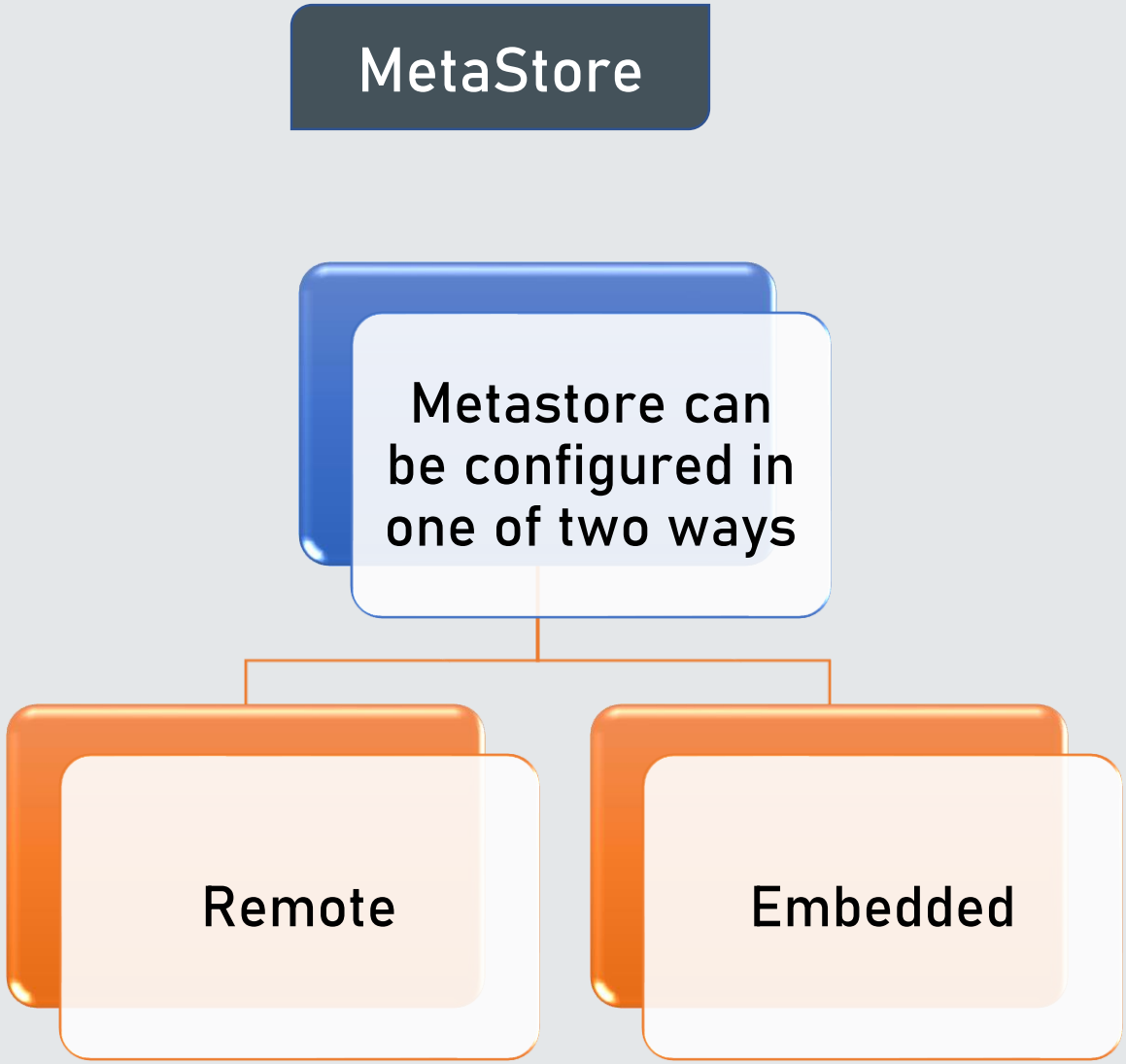
The execution engine uses Hadoop to execute the execution plan created by the compiler in order of their dependencies following the compilation and optimization processes.

Services offered by HIVE

MetaStore



Services offered by HIVE



Services offered by HIVE

MetaStore

It also holds serializer and deserializer metadata, which is essential for read/write operations, as well as HDFS files where data is kept. In most cases, this metastore is a relational database.For searching and altering Hive metadata, Metastore provides a Thrift interface.

Services offered by HIVE

MetaStore

It also holds serializer and deserializer metadata, which is essential for read/write operations, as well as HDFS files where data is kept. In most cases, this metastore is a relational database.For searching and altering Hive metadata, Metastore provides a Thrift interface.

Services offered by HIVE

MetaStore

Metastore can be configured in one of two ways:

- Remote: Metastore is a Thrift service in remote mode, which is suitable for non-Java applications.
- Embedded: In embedded mode, the client can use JDBC to interface directly with the metastore.

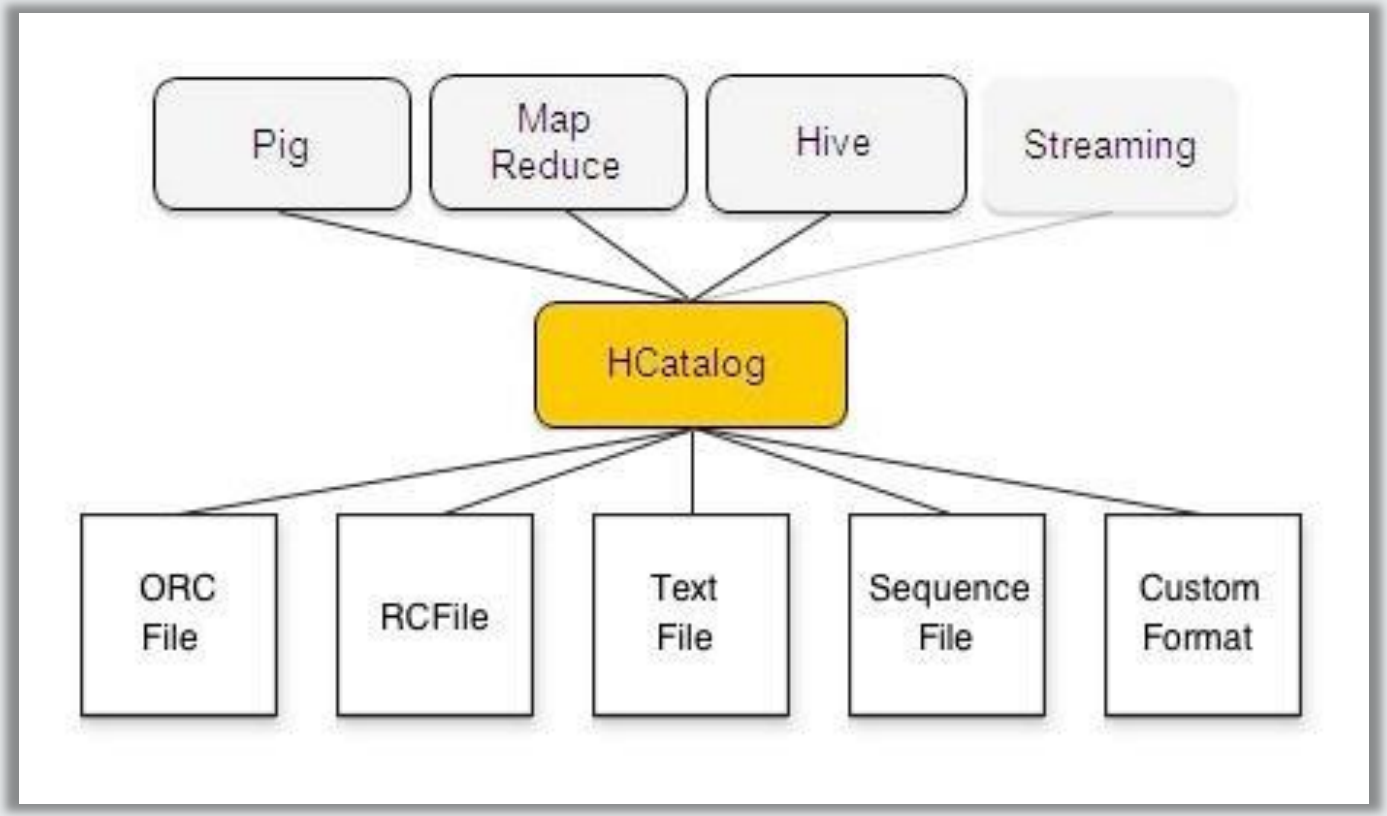
Services offered by HIVE

HCatalog

Hadoop's table and storage management layer is HCatalog. It allows users to read and write data on the grid using various data processing tools such as Pig, MapReduce, and others.It is based on Hive metastore and exposes Hive metastore's tabular data to other data processing tools.

Services offered by HIVE

HCatalog



Services offered by HIVE

HCatalog

HCatalog is a Hadoop table and storage management layer that allows users to read and write data on the grid more simply using various data processing tools such as Pig and MapReduce. Users can have a relational view of data in the Hadoop distributed file system (HDFS) thanks to HCatalog's table abstraction, which means they don't have to worry about where or how their data is stored — RCFile format, text files, SequenceFiles, or ORC files.

Services offered by HIVE

HCatalog

HCatalog supports reading and writing files in any format for which a SerDe (serializer-deserializer) can be written. By default, HCatalog supports RCFile, CSV, JSON, and SequenceFile, and ORC file formats. To use a custom format, you must provide the InputFormat, OutputFormat, and SerDe.

Services offered by HIVE

WebHCat

For HCatalog, WebHCat is the REST API. It's a Hive metadata operations HTTP interface. It lets users perform Hadoop MapReduce (or YARN), Pig, and Hive jobs.

Services offered by HIVE

WebHCat

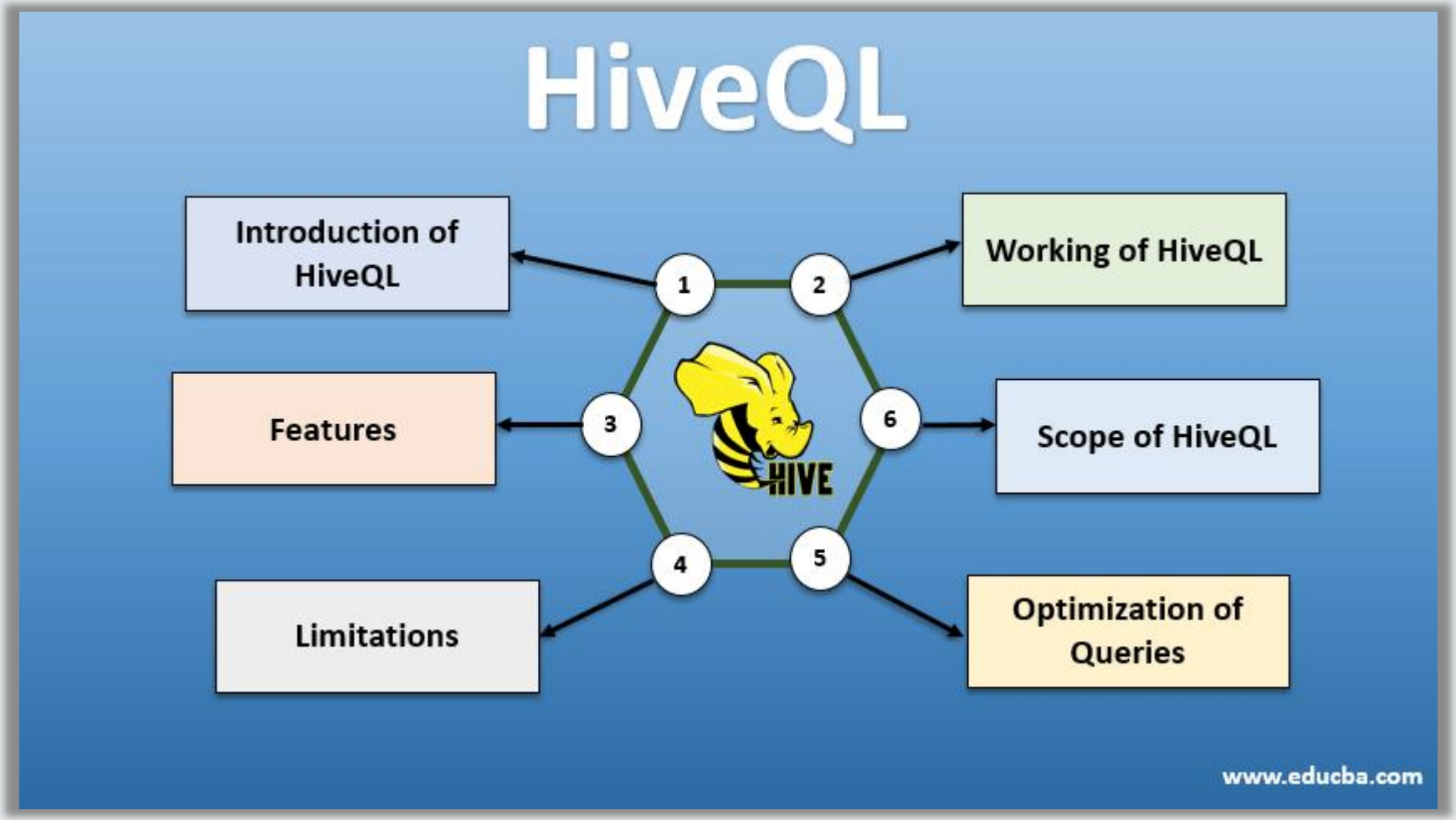
Developers use HTTP requests to access Hadoop MapReduce (or YARN), Pig, Hive, and HCatalog DDL from within applications, as demonstrated in the diagram below. HDFS stores the data and code utilised by this API. When HCatalog DDL commands are requested, they are immediately executed. WebHCat (Templeton) servers queue MapReduce, Pig, and Hive jobs, which may be monitored for progress or cancelled as needed. Pig, Hive, and MapReduce results are stored in HDFS, and developers select where they should be stored.

Services offered by HIVE

- URL Format

HIVE QL

HIVE QL


HiveQL
Introduction of HiveQL
1
Working of HiveQL
2
Features
3
HIVE
6
Scope of HiveQL
4
5
Limitations
Optimization of Queries
www.educba.com

HIVE QL

HIVE QL

- Syntax

HIVE QL

Example

Let us take an example for SELECT…WHERE clause. Assume we have the employee table as given below, with fields named Id, Name, Salary, Designation, and Dept. Generate a query to retrieve the employee details who earn a salary of more than Rs 30000.

ID	Name	Salary	Designation	Dept
1201	Gopal	45000	Technical manager	TP
1202	Manisha	45000	Proofreader	PR
1203	Masthanvali	40000	Technical writer	TP
1204	Krian	40000	Hr Admin	HR
1205	Kranthi	30000	Op Admin	Admin

HIVE QL

Query

ID	Name	Salary	Designation	Dept
1201	Gopal	45000	Technical manager	TP
1202	Manisha	45000	Proofreader	PR
1203	Masthanvali	40000	Technical writer	TP
1204	Krian	40000	Hr Admin	HR
1205	Kranthi	30000	Op Admin	Admin

HIVE QL

JDBC Program

```
import java.sql.SQLException;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.Statement;
import java.sql.DriverManager;

public class HiveQLWhere {
  private static String driverName = "org.apache.hadoop.hive.jdbc.HiveDriver";

  public static void main(String[] args) throws SQLException {

    // Register driver and create driver instance
    Class.forName(driverName);

    // get connection
    Connection con =
DriverManager.getConnection("jdbc:hive://localhost:10000/userdb", "", "");
```

HIVE QL

JDBC Program

```
 // create statement
      Statement stmt = con.createStatement();
// execute statement
      Resultset res = stmt.executeQuery("SELECT * FROM employee WHERE
salary>30000;");

      System.out.println("Result:");
System.out.println(" ID \t Name \t Salary \t Designation \t Dept ");

      while (res.next()) {
         System.out.println(res.getInt(1) + " " + res.getString(2) + " " + res.getDouble(3) +
" " + res.getString(4) + " " + res.getString(5));
      }
      con.close();
   }
}
```

HIVE QL

Save the program in a file named HiveQLWhere.java.

Commands to compile and execute this program.

HIVE QL

ORDER BY clause in a SELECT statement.

HIVE QL

ORDER BY

- Syntax

HIVE QL

Example

Let us take an example for SELECT...ORDER BY clause. Assume employee table as given below, with the fields named Id, Name, Salary, Designation, and Dept. Generate a query to retrieve the employee details in order by using Department name.

ID	Name	Salary	Designation	Dept
1201	Gopal	45000	Technical manager	TP
1202	Manisha	45000	Proofreader	PR
1203	Masthanvali	40000	Technical writer	TP
1204	Krian	40000	Hr Admin	HR
1205	Kranthi	30000	Op Admin	Admin

HIVE QL

Query

ID	Name	Salary	Designation	Dept
1201	Gopal	45000	Technical manager	TP
1202	Manisha	45000	Proofreader	PR
1203	Masthanvali	40000	Technical writer	TP
1204	Krian	40000	Hr Admin	HR
1205	Kranthi	30000	Op Admin	Admin

HIVE QL

- GROUP BY

HIVE QL

- Syntax

HIVE QL

Example

Let us take an example of SELECT…GROUP BY clause. Assume employee table as given below, with Id, Name, Salary, Designation, and Dept fields. Generate a query to retrieve the number of employees in each department.

ID	Name	Salary	Designation	Dept
1201	Gopal	45000	Technical manager	TP
1202	Manisha	45000	Proofreader	PR
1203	Masthanvali	40000	Technical writer	TP
1204	Krian	45000	Proofreader	PR
1205	Kranthi	30000	Op Admin	Admin

HIVE QL

Query

ID	Name	Salary	Designation	Dept
1201	Gopal	45000	Technical manager	TP
1202	Manisha	45000	Proofreader	PR
1203	Masthanvali	40000	Technical writer	TP
1204	Krian	45000	Proofreader	PR
1205	Kranthi	30000	Op Admin	Admin


That's all for now...