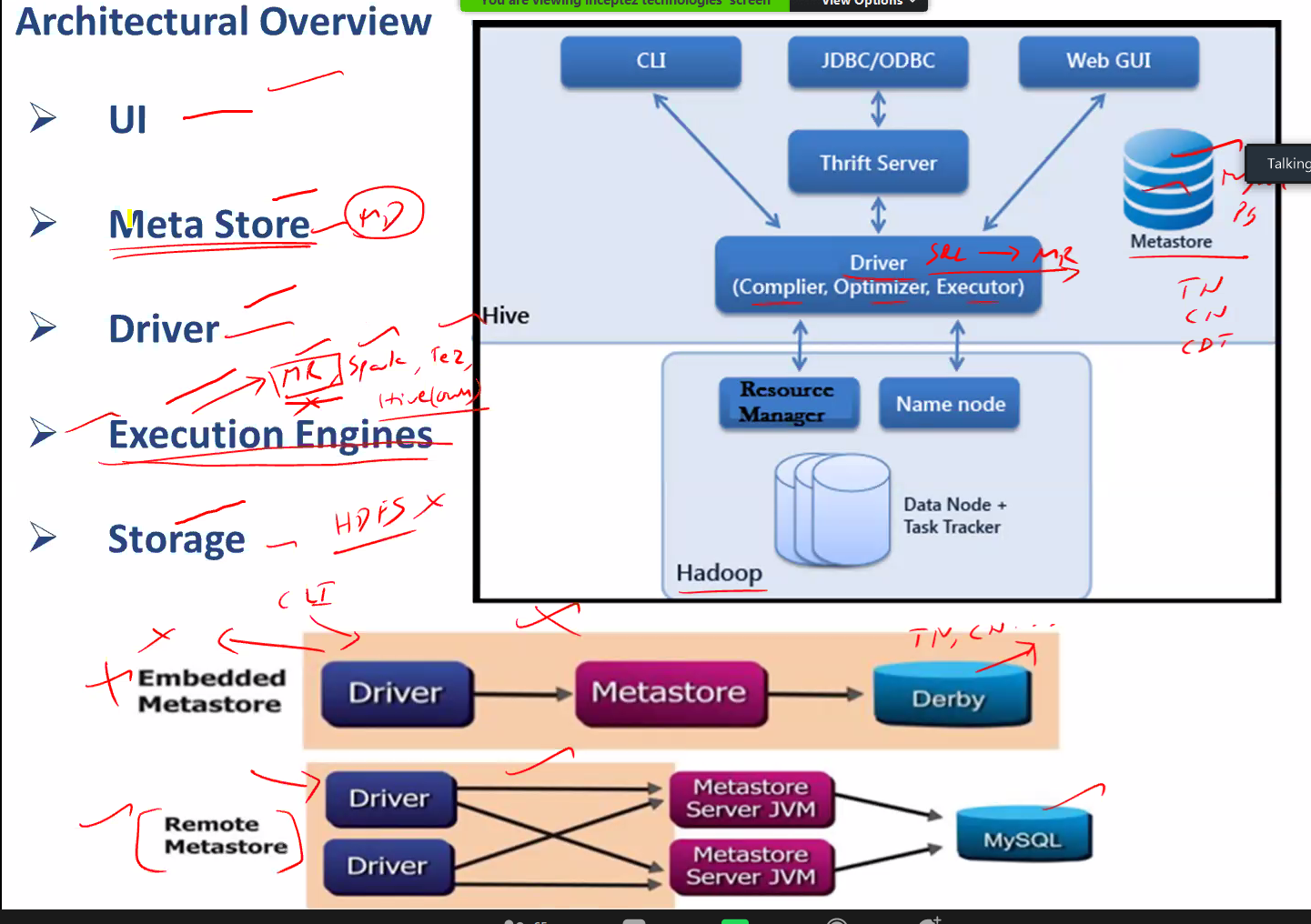
# HIVE

Hive is SQL layer on top of Hadoop. Its called HQL, Hive Query Language.

Hive Architecture:



### Metastore:

Metastore is required to store the Hadoop table structure information. It cannot be saved in Hadoop itself.

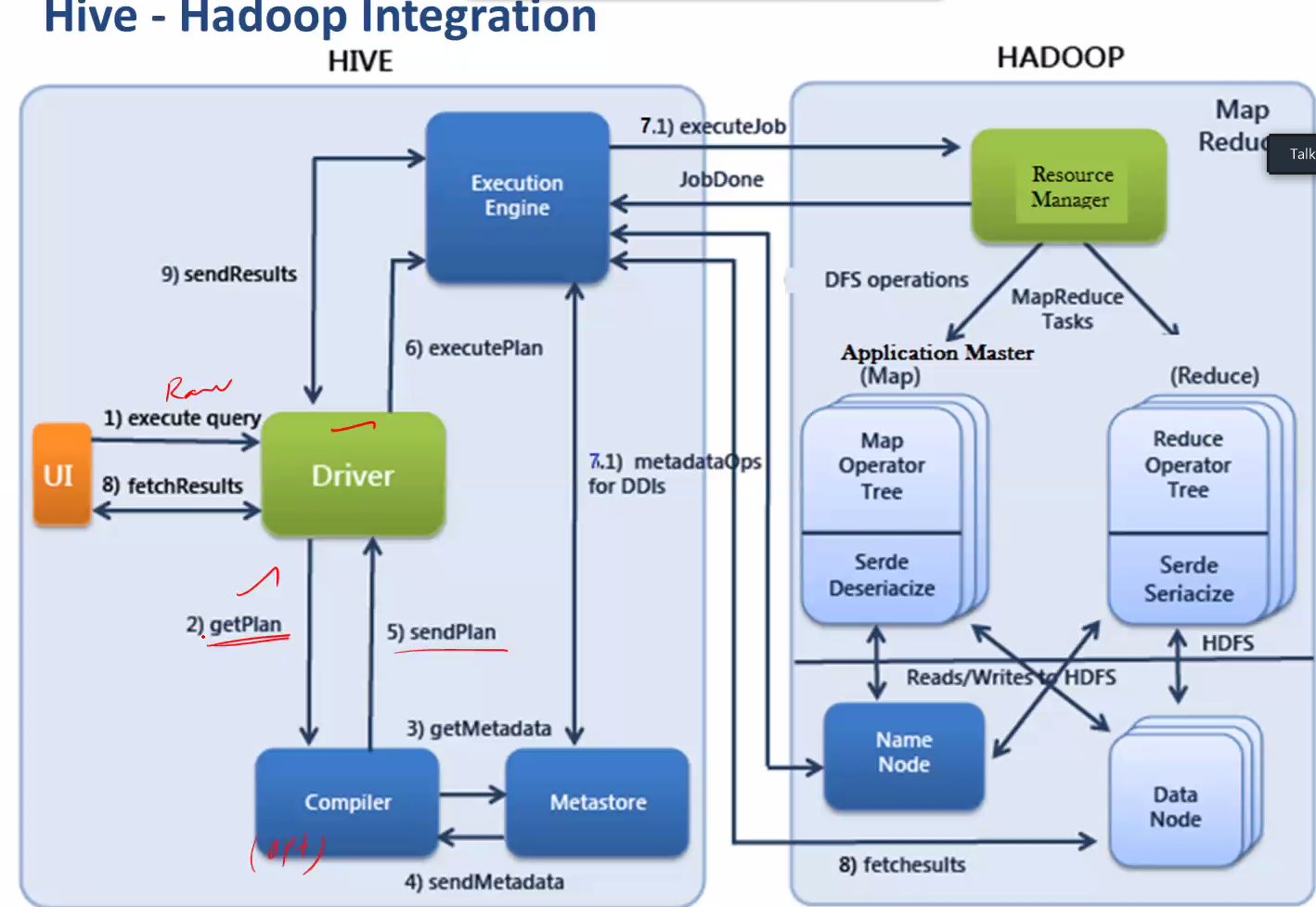
#### Embedded metastore:

Comes default with Hadoop, which contains Derby. Derby has limitation which can be used for Development and testing alone. No two sessions can be started at same time.

#### Metastore:

Hive supports many databases to store the metadata of table information. The metastore server jvm allows to start session with Hive

## Hive – Hadoop integration



5. Execution plan is generated, the best way to reach the database

6. Submit the plan to execution engire

7. Execute the job converting the job in to RM or Spark or Tez depending on the execution engine

NOTE: Some of the Queries can be directly fetched from Hadoop without involving any Hadoop execution engine. Hive to Hadoop directly. No use of RM, Spark or Tez

# SerDes - Serialization and Deserialization

Serialization: Converting the original data to byte code

Deserialization: Converting the byte code to original data

What is a SerDe?

* SerDe is a short name for "Serializer and Deserializer."
* Hive uses SerDe (and FileFormat) to read and write table rows.
* HDFS files --> InputFileFormat --> <key, value> --> Deserializer --> Row object
* Row object --> Serializer --> <key, value> --> OutputFileFormat --> HDFS files

# Hive Workout:

Create database:

CREATE (DATABASE|SCHEMA) [IF NOT EXISTS] database\_name

  [COMMENT database\_comment]

  [LOCATION hdfs\_path]

  [MANAGEDLOCATION hdfs\_path]

  [WITH DBPROPERTIES (property\_name=property\_value, ...)];

From above ( ) – is a mandatory parameter

From above [] – is a optional parameter

DATABASE|SCHEMA: These are the same thing. These words can be used interchangeably.

[IF NOT EXISTS]: This is an optional clause. If not used, an error is thrown when there is an attempt to create a database that already exists.

[COMMENT]: This is an optional clause. This is used to place a comment for the database. This comment clause can be used to add a description about the database. The comment must be in single quotes.

[LOCATION]: This is an optional clause. This is used to override the default location with the preferred one.

[WITH DBPROPERTIES]: This is an optional clause. This clause is used to set properties for the database. These properties are key-value pairs that can be associated with the database to attach additional information with the database.

To get the database details:

DESCRIBE DATABASE [EXTENDED] db\_name;

### Hive Data Hierarchy

Complex Data Type:

* ARRAY - This is a list of items of the same type - ["apple","orange","mango"]
* MAP - This is a set of key-value pairs - {1: "apple",2: "orange"}
* STRUCT - This is a user-defined structure of any type of field, such as {val1, val2, val3, and so on} - {1, "apple"}

We have employee database:

Michael|Montreal,Toronto|Male,30|DB:80|Product:Developer:Lead

Will|Montreal|Male,35|Perl:85|Product:Lead,Test:Lead

Shelley|New York|Female,27|Python:80|Test:Lead,COE:Architect

Lucy|Vancouver|Female,57|Sales:89,HR:94|Sales:Lead

Create table Employee:

CREATE TABLE employee (

name STRING,

work\_place ARRAY<STRING>,

gender\_age STRUCT<gender:STRING,age:INT>,

skills\_score MAP<STRING,INT>,

depart\_title MAP<STRING,ARRAY<STRING>>

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '|'

COLLECTION ITEMS TERMINATED BY ','

MAP KEYS TERMINATED BY ':'

STORED AS TEXTFILE;

Select \* from table looks like

employee.name employee.work\_place employee.gender\_age employee.skills\_score employee.depart\_title

Michael ["Montreal","Toronto"] {"gender":"Male","age":30} {"DB":80} {"Product":["Developer:Lead"]}

Will ["Montreal"] {"gender":"Male","age":35} {"Perl":85} {"Product":["Lead"],"Test":["Lead"]}

Shelley ["New York"] {"gender":"Female","age":27} {"Python":80} {"Test":["Lead"],"COE":["Architect"]}

Lucy ["Vancouver"] {"gender":"Female","age":57} {"Sales":89,"HR":94} {"Sales":["Lead"]}

#### How to access Array:

work\_place

["Montreal","Toronto"]

["Montreal"]

["New York"]

["Vancouver"]

hive (retail)> SELECT

> work\_place[0] as col\_1, work\_place[1] as col\_2,

> work\_place[2] as col\_3

> FROM employee;

col\_1 col\_2 col\_3

Montreal Toronto NULL

Montreal NULL NULL

New York NULL NULL

Vancouver NULL NULL

#### How to access Struct:

hive (retail)> SELECT gender\_age FROM employee;

gender\_age

{"gender":"Male","age":30}

{"gender":"Male","age":35}

{"gender":"Female","age":27}

{"gender":"Female","age":57}

Access individual items:

hive (retail)> SELECT gender\_age.gender, gender\_age.age FROM employee;

gender age

Male 30

Male 35

Female 27

Female 57

How to access Map items:

hive> SELECT skills\_score FROM employee;

{"DB":80}

{"Perl":85}

{"Python":80}

{"Sales":89,"HR":94}

How to access individual items:

hive> SELECT

> name, skills\_score['DB'] as DB, skills\_score['Perl'] as Perl,

> skills\_score['Python'] as Python,

> skills\_score['Sales'] as Sales,

> skills\_score['HR'] as HR

> FROM employee;

Michael 80 NULL NULL NULL NULL

Will NULL 85 NULL NULL NULL

Shelley NULL NULL 80 NULL NULL

Lucy NULL NULL NULL 89 94

### Database:

The database in Hive describes a collection of tables. If the database is not specified, the **default** database is used and **uses /user/hive/warehouse** in HDFS as its root directory. This path is configurable by the hive.metastore.warehouse.dir property in hive-site.xml. Whenever a new database is created, Hive creates a new directory for each database under /user/hive/warehouse. For example, the myhivebook database is located at /user/hive/datawarehouse/myhivebook.db. In addition, DATABASE has a name alias, SCHEMA, meaning they are the same thing in HQL

#### Create Database:

CREATE (DATABASE|SCHEMA) [IF NOT EXISTS] database\_name

  [COMMENT database\_comment]

  [LOCATION hdfs\_path]

  [MANAGEDLOCATION hdfs\_path]

  [WITH DBPROPERTIES (property\_name=property\_value, ...)];

#### Drop database:

DROP (DATABASE|SCHEMA) [IF EXISTS] database\_name [RESTRICT|CASCADE];

* A database cannot be dropped until all the tables inside it are empty, unless CASCADE is used

#### Alter Database:

ALTER (DATABASE|SCHEMA) database\_name SET LOCATION hdfs\_path;

### Tables:

#### Managed Table:

1. All the data in the table is stored in this hive user-manageable directory (full permission)
2. When keeping data in the internal tables, the table fully manages the data in it. When an internal table is dropped, its data is deleted together
3. the internal table is often used as an intermediate table during data processing, since it is quite powerful and flexible when supported by HQL.

#### External Table:

1. When data is already stored in HDFS, an external table can be created to describe the data. It is called external because the data in the external table is specified in the LOCATION property
2. when an external table is dropped, the data is not deleted
3. use external tables for source read-only data or sharing the processed data to data consumers giving customized HDFS locations

DDL Statement:

CREATE TABLE IF NOT EXISTS employee\_internal (

name STRING COMMENT 'this is optinal column comments',

work\_place ARRAY<STRING>, -- table column names are NOT casesensitive

gender\_age STRUCT<gender:STRING,age:INT>,

skills\_score MAP<STRING,INT>, -- columns names are lower case

depart\_title MAP<STRING,ARRAY<STRING>> -- No "," for the last column

)

COMMENT 'This is an internal table' -- This is optional table comments

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '|' -- Symbol to seperate columns

COLLECTION ITEMS TERMINATED BY ',' -- Seperate collection elements

MAP KEYS TERMINATED BY ':' -- Symbol to seperate keys and values

STORED as TEXTFILE; -- Table file format

To view details description of table – command “Describe formatted <<table name>>”

hive> describe formatted employee\_internal;

OK

# col\_name data\_type comment

name string this is optinal column comments

work\_place array<string>

gender\_age struct<gender:string,age:int>

skills\_score map<string,int>

depart\_title map<string,array<string>>

# Detailed Table Information

Database: retail

Owner: hduser

CreateTime: Mon Sep 21 00:19:16 IST 2020

LastAccessTime: UNKNOWN

Protect Mode: None

Retention: 0

Location: hdfs://localhost:54310/user/hive/warehouse/retail.db/employee\_internal

Table Type: MANAGED\_TABLE

Table Parameters:

comment This is an internal table

transient\_lastDdlTime 1600627756

# Storage Information

SerDe Library: org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe

InputFormat: org.apache.hadoop.mapred.TextInputFormat

OutputFormat: org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat

Compressed: No

Num Buckets: -1

Bucket Columns: []

Sort Columns: []

Storage Desc Params:

colelction.delim ,

field.delim |

mapkey.delim :

serialization.format |

### Load Data:

Load from local

hive> LOAD DATA LOCAL INPATH '/home/hduser/employee.txt' overwrite INTO TABLE employee\_internal;

Loading data to table retail.employee\_internal

Load from hdfs

hive> LOAD DATA inpath '/user/hduser/employee\_internal.txt' overwrite INTO TABLE employee\_internal;

Loading data to table retail.employee\_internal

Table retail.employee\_internal stats: [numFiles=1, numRows=0, totalSize=227, rawDataSize=0]