Overview of Hive

Course Road Map

- Module 1: Big Data Fundamentals
- Module 2: Data Acquisition and Storage
- **Module 3: Data Access and Processing**
- Module 4: Data Unification
- Module 5: Data Analysis
- Module 6: Big Data Deployment Options

- Lesson 8: Introduction to MapReduce
- Lesson 9: Resource Management Using YARN
- Lesson 10: Apache Spark
- Lesson 11: Overview of Apache Hive
- Lesson 12: Overview of Cloudera Impala
- Lesson 13: Using Oracle XQuery for Hadoop
- Lesson 14: Overview of Solr

Objectives

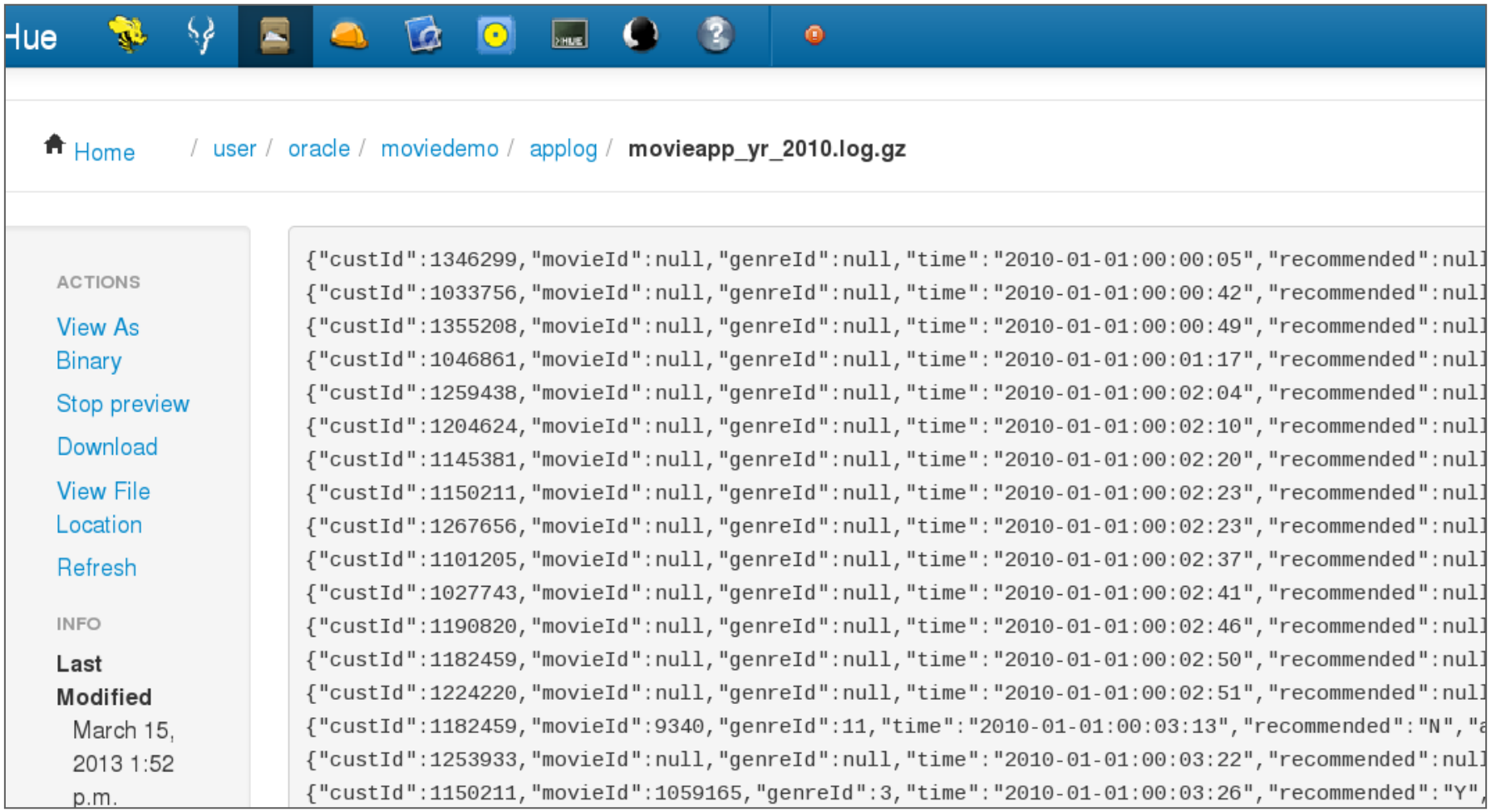
After completing this lesson, you should be able to:

- Define Hive
- Describe the Hive data flow
- Create a Hive database

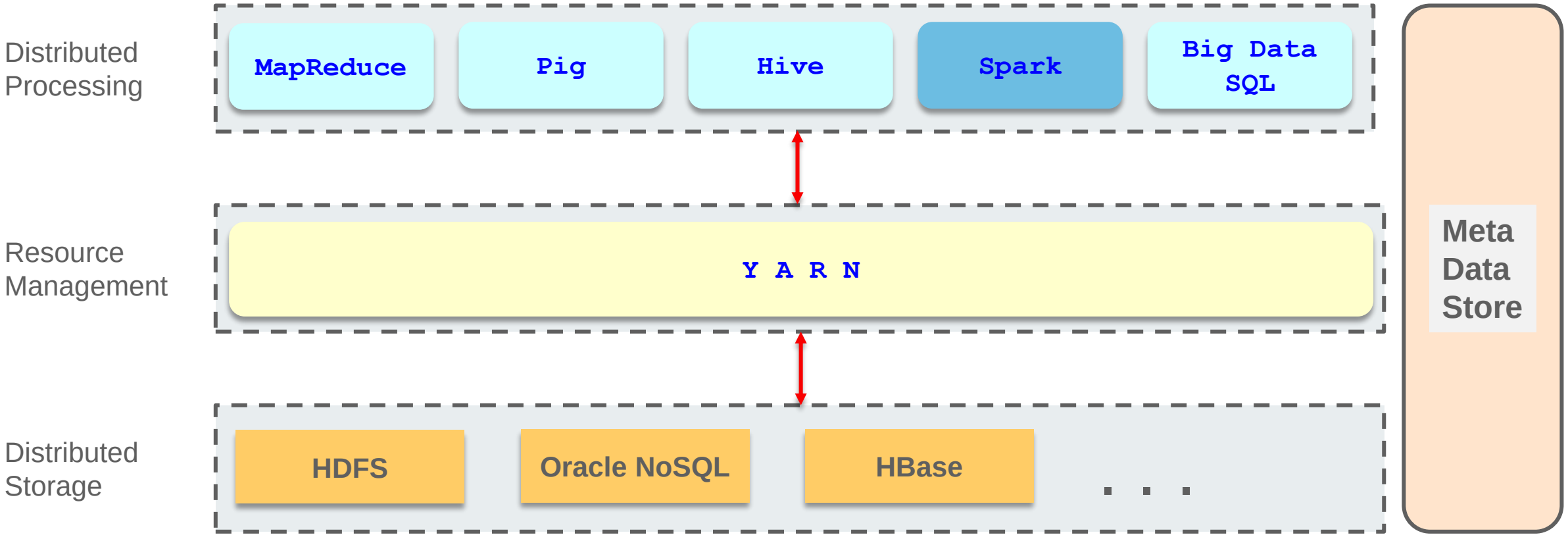
Hive

- Hive is an open source Apache project and was originally developed by Facebook.
- Hive enables analysts who are familiar with SQL to query data stored in HDFS by using HiveQL (a SQL-like language).
- It is an infrastructure built on top of Hadoop that supports the analysis of large data sets.
- Hive transforms HiveQL queries into one of the following:
  - MapReduce jobs (high-level abstraction on top of MapReduce)
  - Spark jobs
- This lesson covers Hive at a high level.

Use Case: Storing Clickstream Data

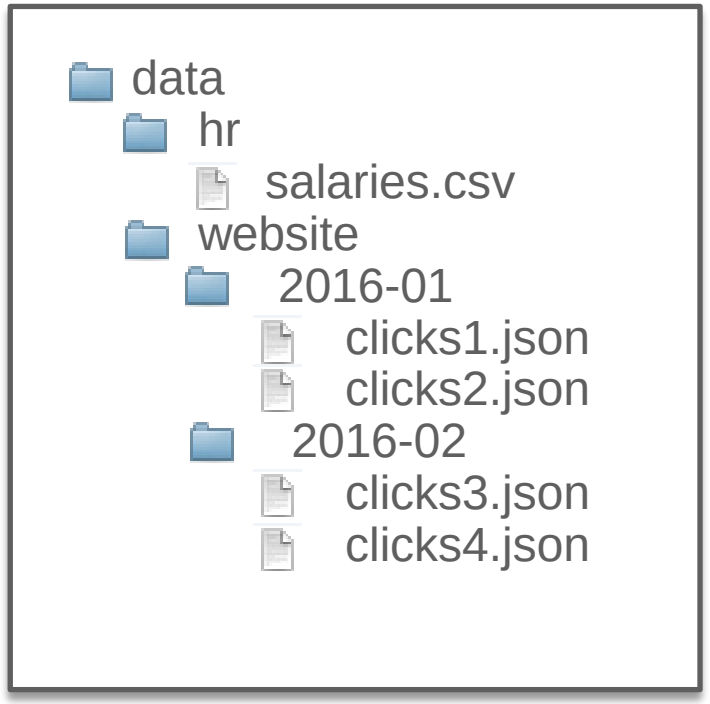


Hadoop Architecture

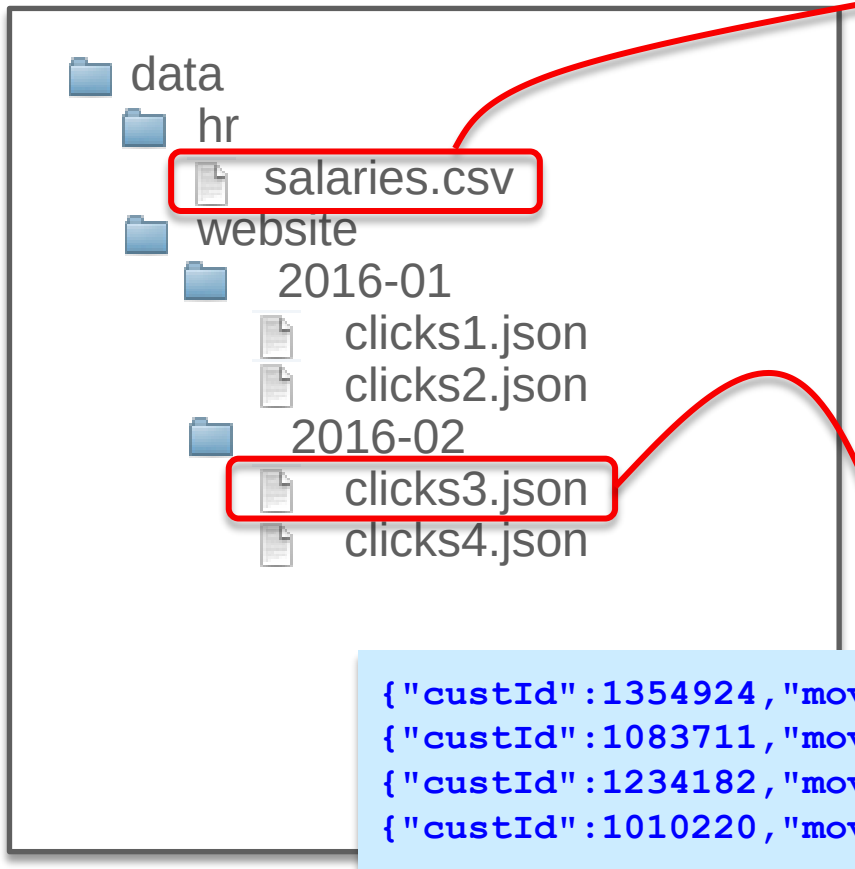


How Is Data Stored in HDFS?

- Data is stored in files and is organized into folders:
  - Can be of any file type
  - Replicated three times across the cluster
- Schema on Read
  - The tool reading the data interprets the data as it sees appropriate.



How Is Data Stored in HDFS

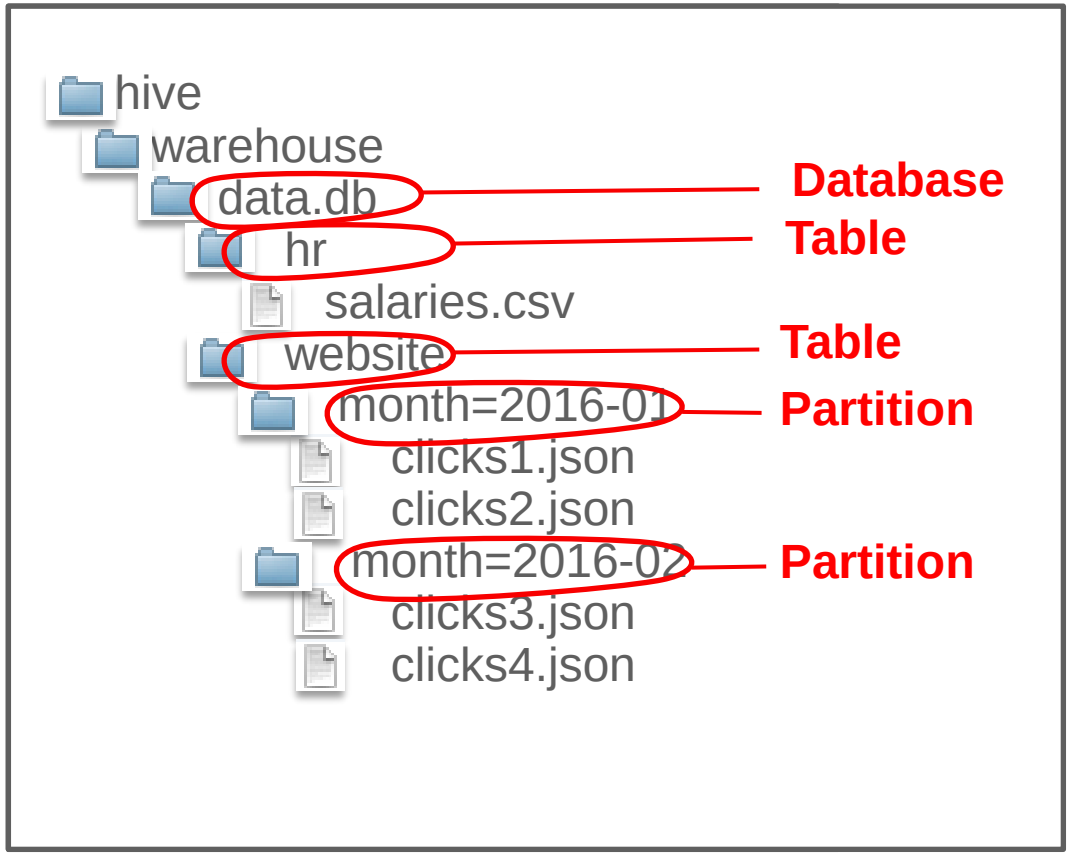



```
Hanks,Spielberg,1000000
Spielberg,Cameron,2500000
Cameron,Oprah,125000
Oprah,Boss,54000000
...
```

```
{"custId":1354924,"movieId":1948,"genreId":9,"time":"2012-07-01:00:00:22”}
{"custId":1083711,"movieId":null,"genreId":null,"time":"2012-07-01:00:00:26”}
{"custId":1234182,"movieId":11547,"genreId":44,"time":"2012-07-01:00:00:32”}
{"custId":1010220,"movieId":11547,"genreId":44,"time":"2012-07-01:00:00:42”}
...
```

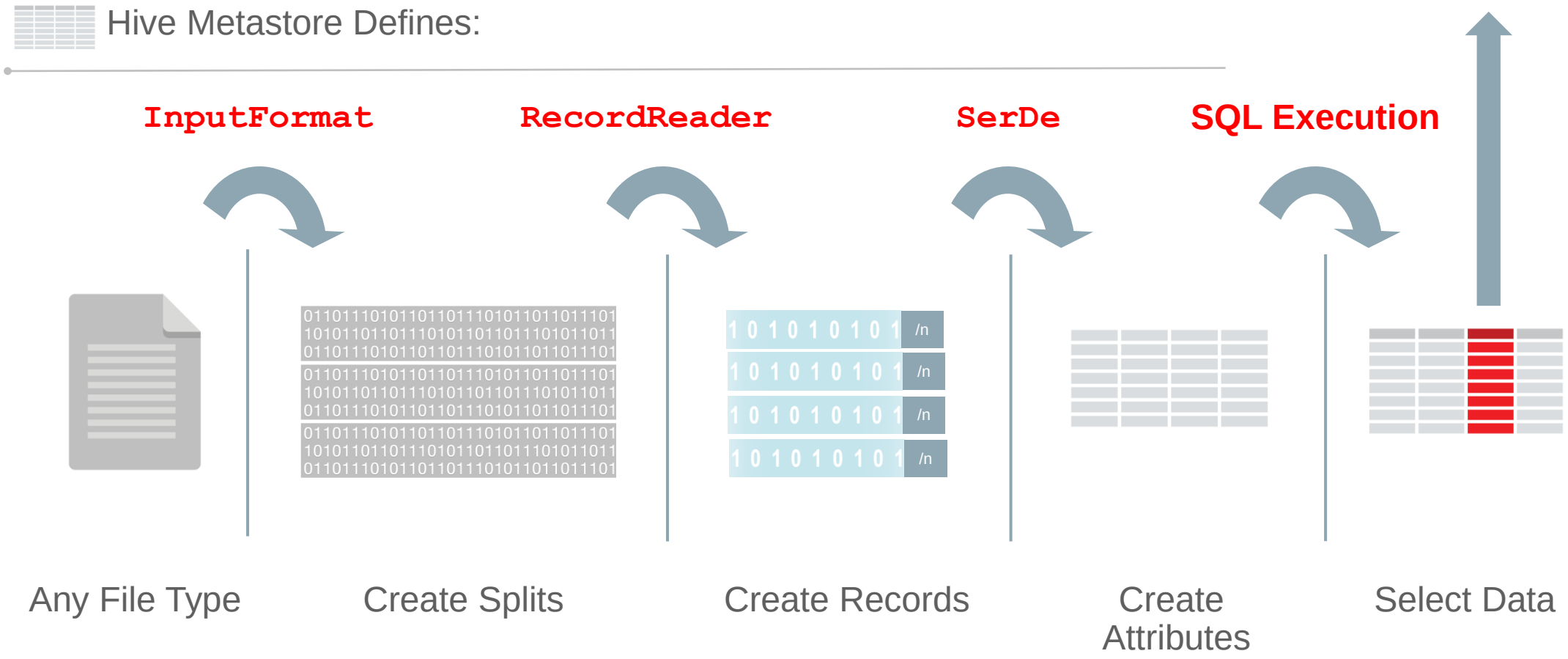
Organizing and Describing Data with Hive

- Information is captured in Hive Metastore.
- HDFS folders become:
  - *Databases*
  - *Tables*
  - *Partitions*
- *A **table*** includes metadata for parsing data files using Java classes:
  - `InputFormat` defines chunks called splits based on file type.
  - `RecordReader` creates rows out of splits.
  - `SerDe` creates columns.



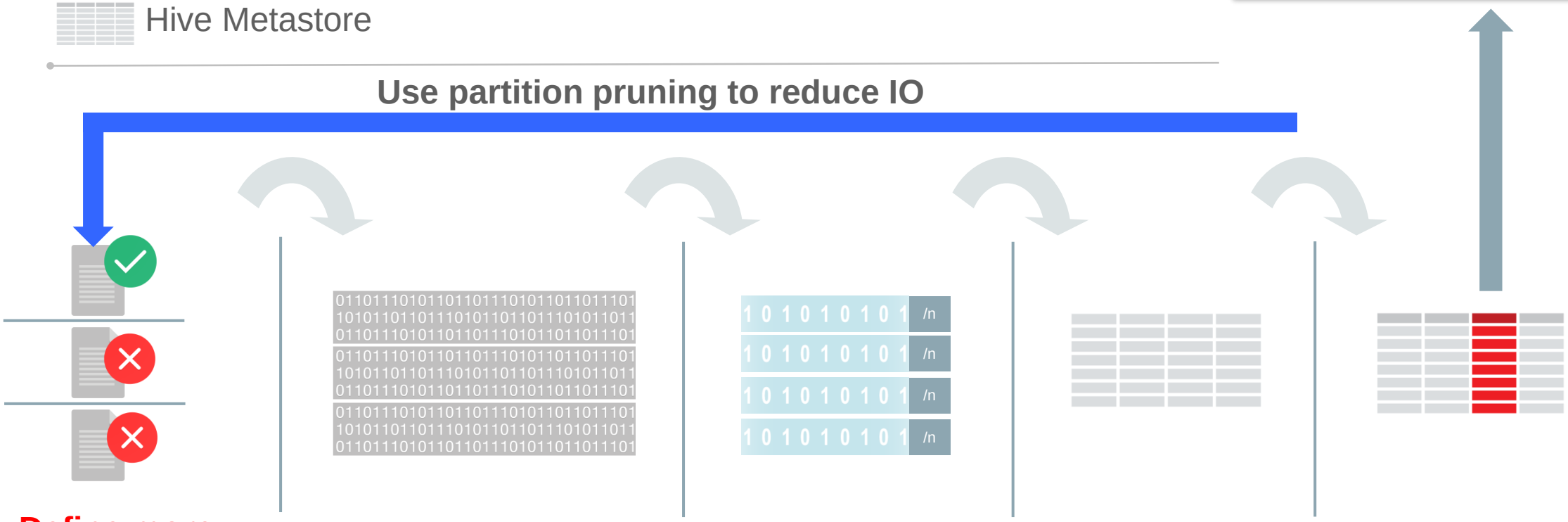
How Does Hive Read *ANY* Data?

```
SELECT name
FROM my_cust
WHERE month = '2016-01';
```



How Does Hive Read Data *FASTER*?

```
SELECT name
FROM my_cust
WHERE month = '2016-01';
```



Define more fine-grained schema on files: Partitioning

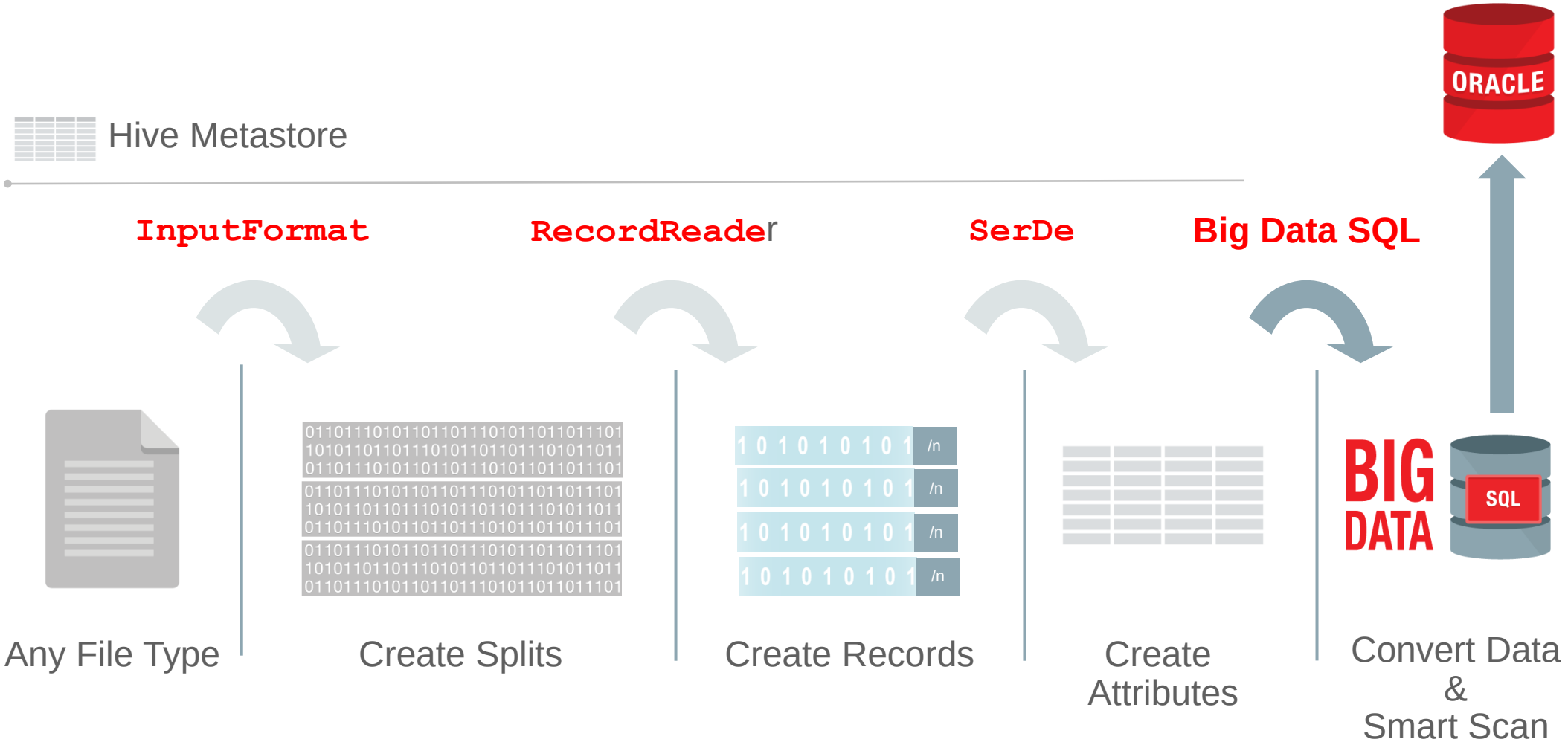
Create Splits

Create Records

Create Attributes

Select Data

Big Data SQL on Top of “Hive” Data



Defining Tables Over HDFS

```
22
23 -- Create table over source JSON
24 CREATE EXTERNAL TABLE IF NOT EXISTS movieapp_log_json (
25   custId INT,
26   movieId INT,
27   genreId INT,
28   time STRING,
29   recommended STRING,
30   activity INT,
31   rating INT,
32   price FLOAT
33 )
34 ROW FORMAT SERDE 'org.apache.hadoop.hive.contrib.serde2.JsonSerde'
35 LOCATION '/user/oracle/moviedemo/applog/';
```

Simple SQL syntax

SerDe option

A table in Hive is mapped to HDFS directories.

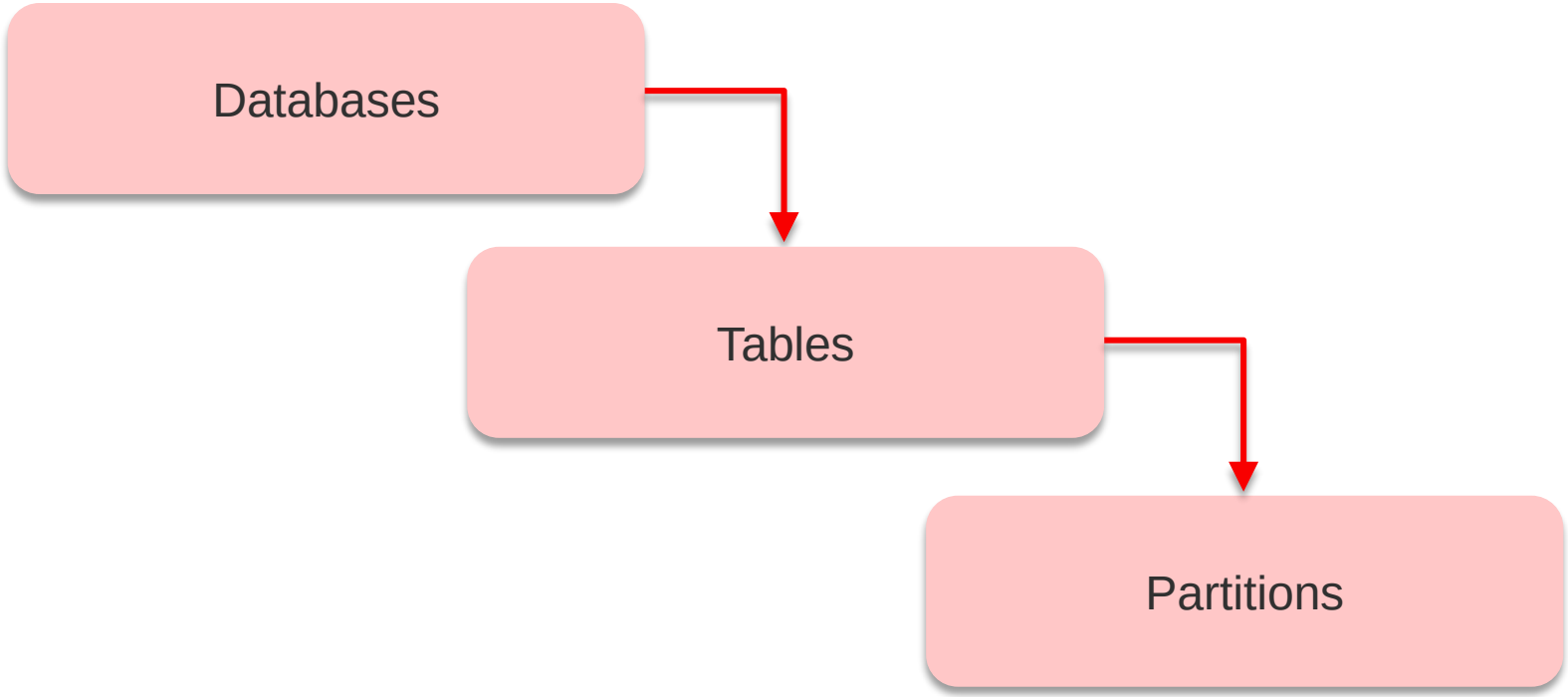
Defining Tables over HDFS

```
{"custid":1185972,"movieid":null,"genreid":null,"time":"2012-07-01:00:00:07","recommended":null,"activity":8}
{"custid":1354924,"movieid":1948,"genreid":9,"time":"2012-07-01:00:00:22","recommended":"N","activity":7}
...
```

HiveQL (simple SQL-Like SQL Syntax to query the click stream data)

```
CREATE EXTERNAL TABLE default.movieapp_log_json(
  custid int  ,
  movieid int  ,
  genreid int  ,
  time string  ,
  recommended string  ,
  activity int  ,
  rating int  ,
  price float  ,
  position int  )
ROW FORMAT SERDE
  'org.apache.hive.hcatalog.data.JsonSerDe'
STORED AS INPUTFORMAT
  'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION
  'hdfs://bigdatalite.localdomain:8020/user/oracle/moviework/applog_json'
```

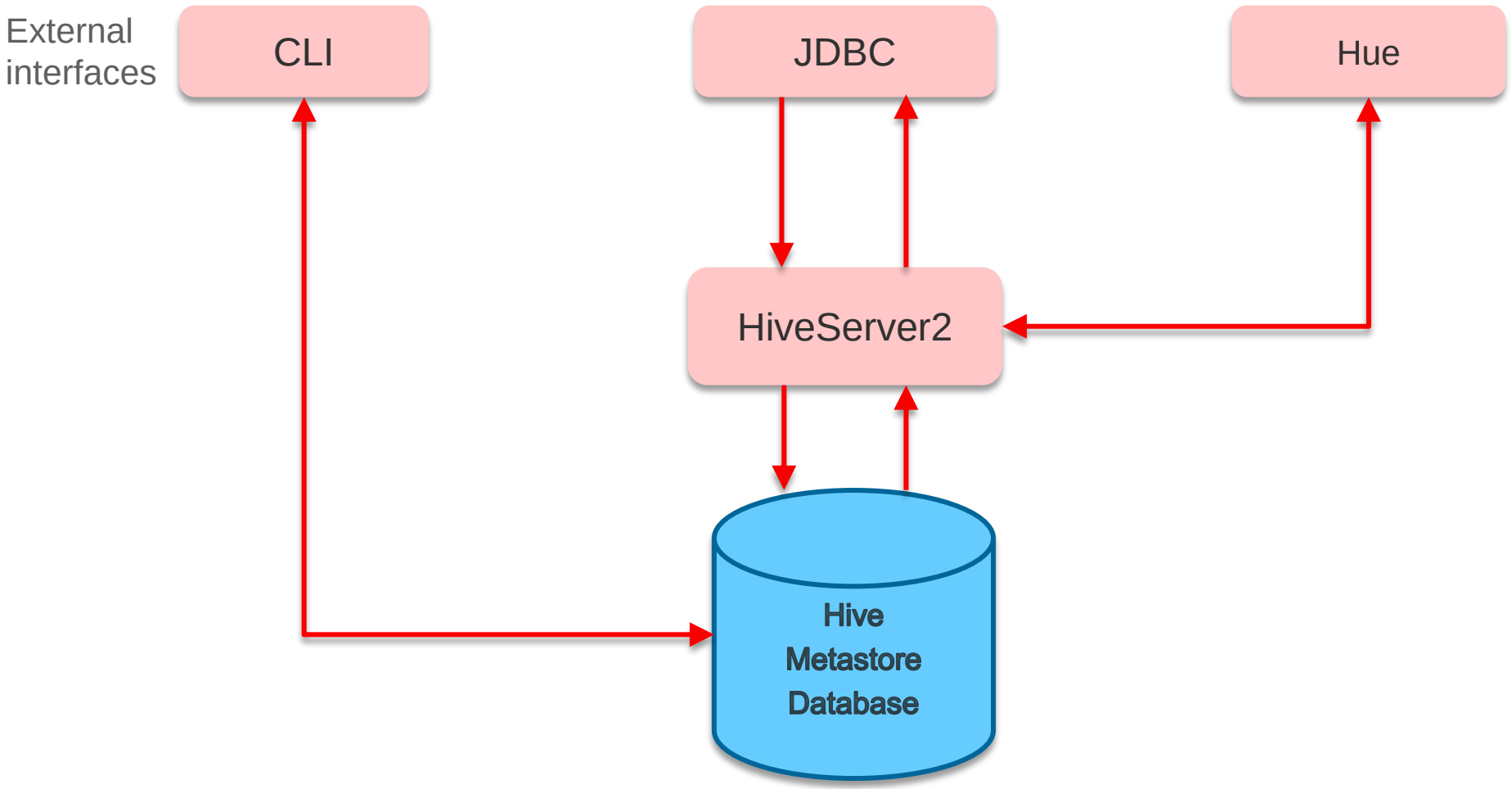
Hive: Data Units



Hive Metastore Database

- Contains metadata regarding databases, tables, and partitions
- Contains information about how the rows and columns are delimited in the HDFS files that are used in the queries
- Is an RDBMS database, such as MySQL, where Hive persists table schemas and other system metadata

Hive Framework



Creating a Hive Database

1. **Start hive.**

```
[oracle@localhost mapreduce]$ hive
Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.properties
Hive history file=/tmp/oracle/hive_job_log_oracle_201302071749_169058549.txt
hive>
```

2. **Create the database.**

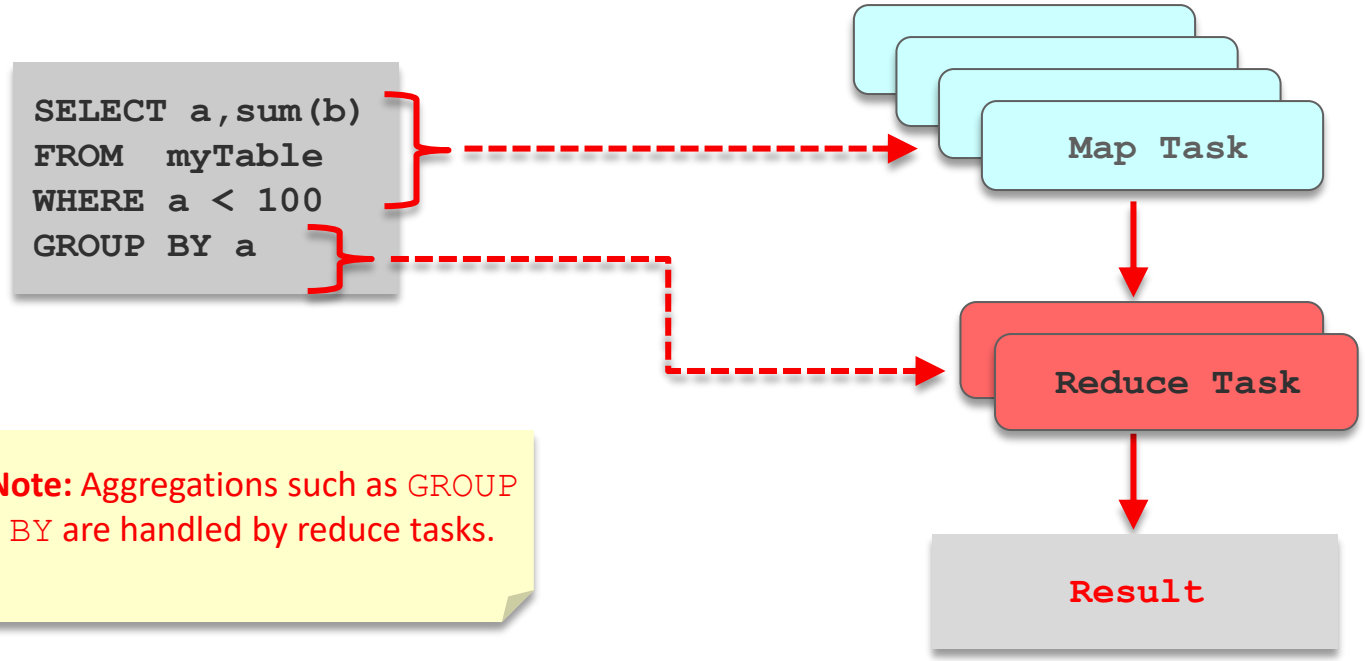
```
hive> create database moviework;
OK
Time taken: 4.288 seconds
hive>
```

3. **Verify the database creation.**

```
hive> show databases;
OK
default
moviedemo
moviework
Time taken: 1.281 seconds
hive>
```

Data Manipulation in Hive

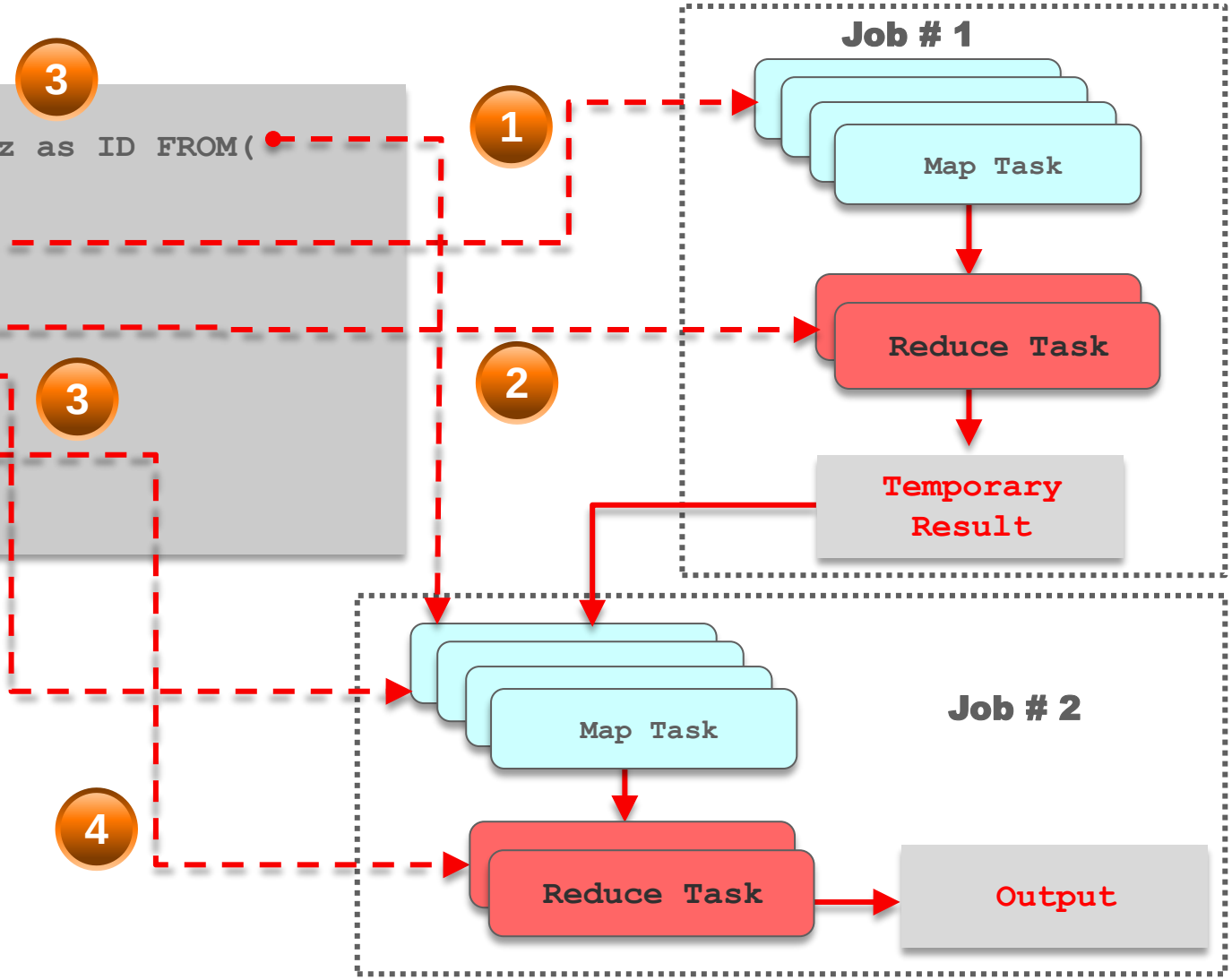
Hive `SELECT` with a `WHERE` clause:



Note: Aggregations such as `GROUP BY` are handled by reduce tasks.

Data Manipulation in Hive: Nested Queries

```
SELECT mt.a, mt.timesTwo,otherTable.z as ID FROM(
   SELECT a, sum(b*b) AS timesTwo
   FROM myTable
   GROUP BY a)mt
JOIN otherTable
ON otherTable.z = mt.a
GROUP BY mt.a,otherTable.z
```



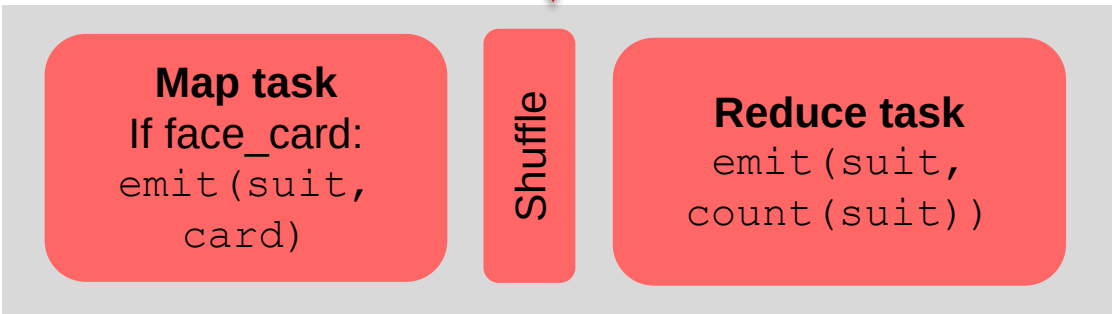
Notes:

- Subqueries are treated as sequential MapReduce jobs.
- Jobs execute from the innermost query outward.

Steps in a Hive Query

```
SELECT suit, COUNT(*)
FROM cards
WHERE face_value > 10
GROUP BY suit;
```

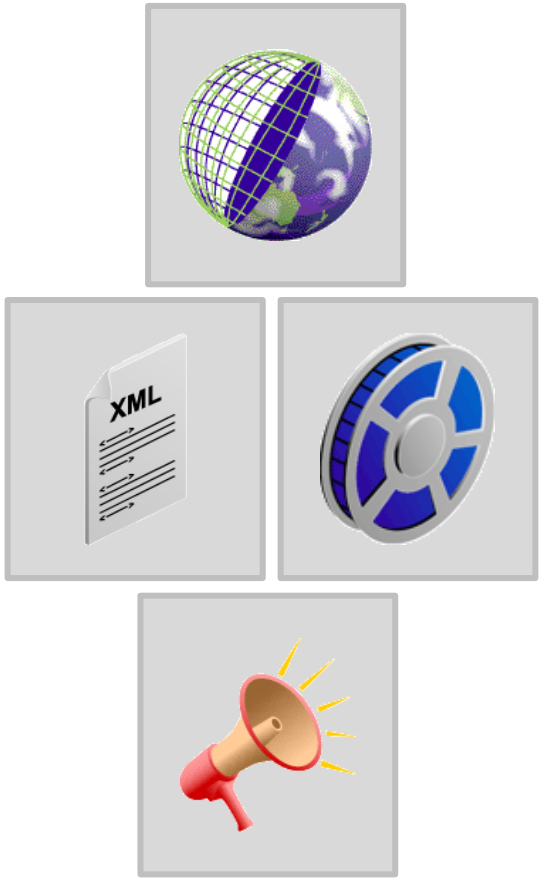
HiveQL



Hadoop Cluster (Job Tracker or Resource Manager)

Hive-Based Applications

- Log processing
- Text mining
- Document indexing
- Business analytics
- Predictive modeling



Hive: Limitations

- No support for materialized views
- No transaction-level support
- Not ideal for ad hoc work
- Limited subquery support
- Subset of SQL-92
- Immature optimizer

Summary

In this lesson, you should have learned how to:

- Define Hive
- Describe the Hive data flow
- Create a Hive database

Practice 11: Overview

This practice covers the following topics:

- Practice 11-1: Manipulating Data with Hive
- Practice 11-2: Extracting Facts by Using Hive