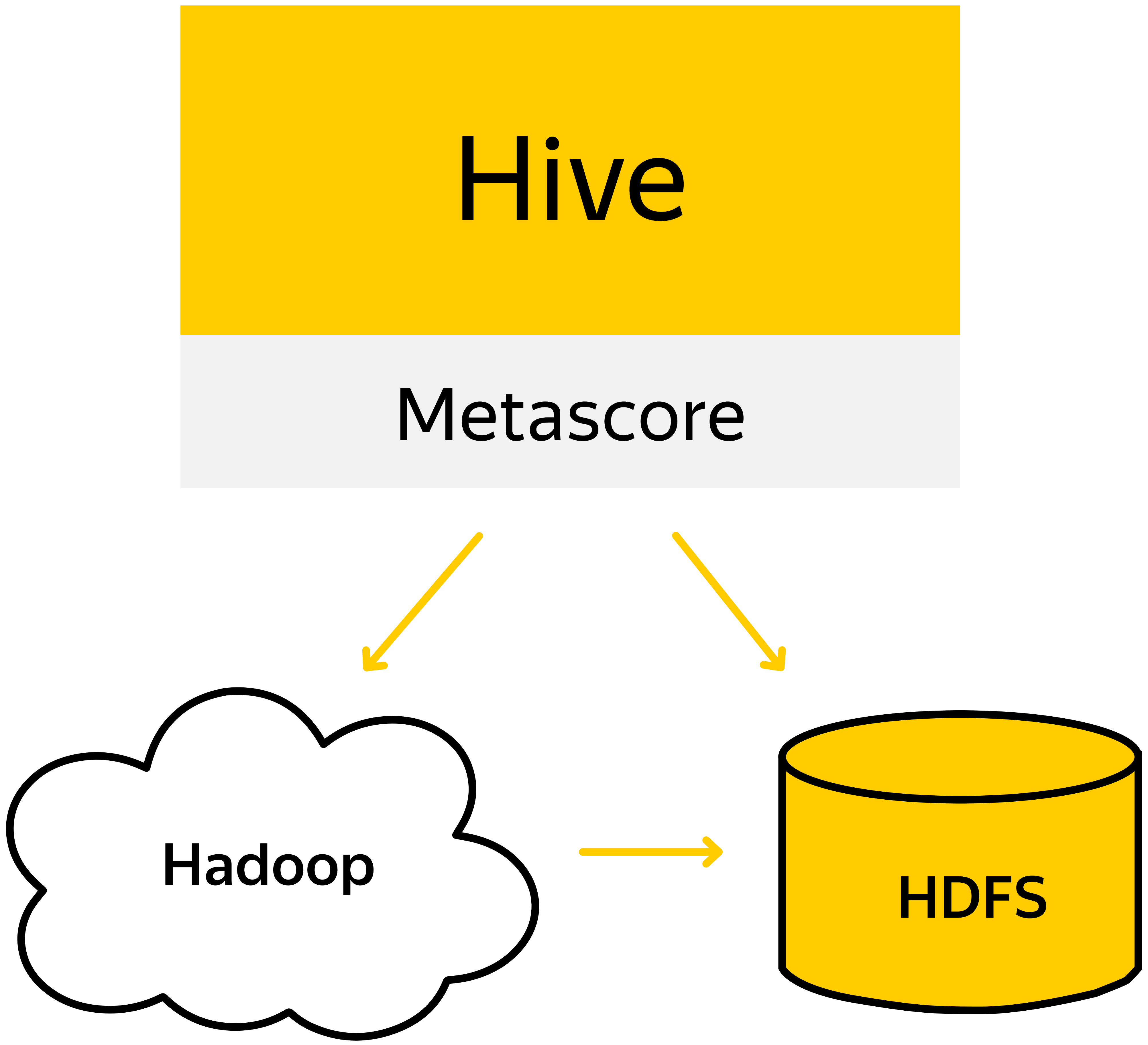
Yandex

SQL over BigData

Hive Data Manipulation Language (DML)



Hive

Metascore

Hadoop

HDFS

(1) DDL

(2) HiveQL

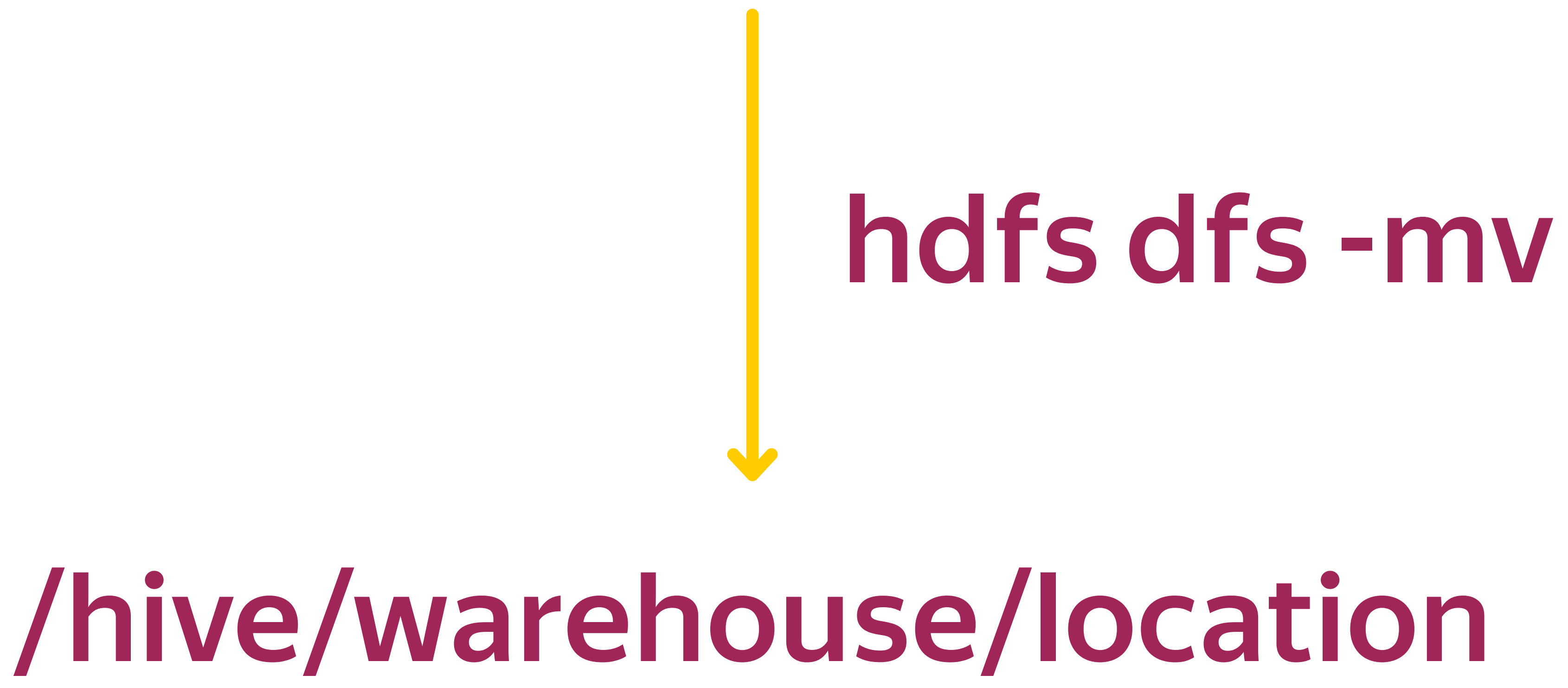
(3) DML

DML (import)

```
LOAD DATA INPATH '/local/path/employees-data'
INTO TABLE employees;
```

DML (import)

```
LOAD DATA INPATH '/local/path/employees-data'
INTO TABLE employees;
```

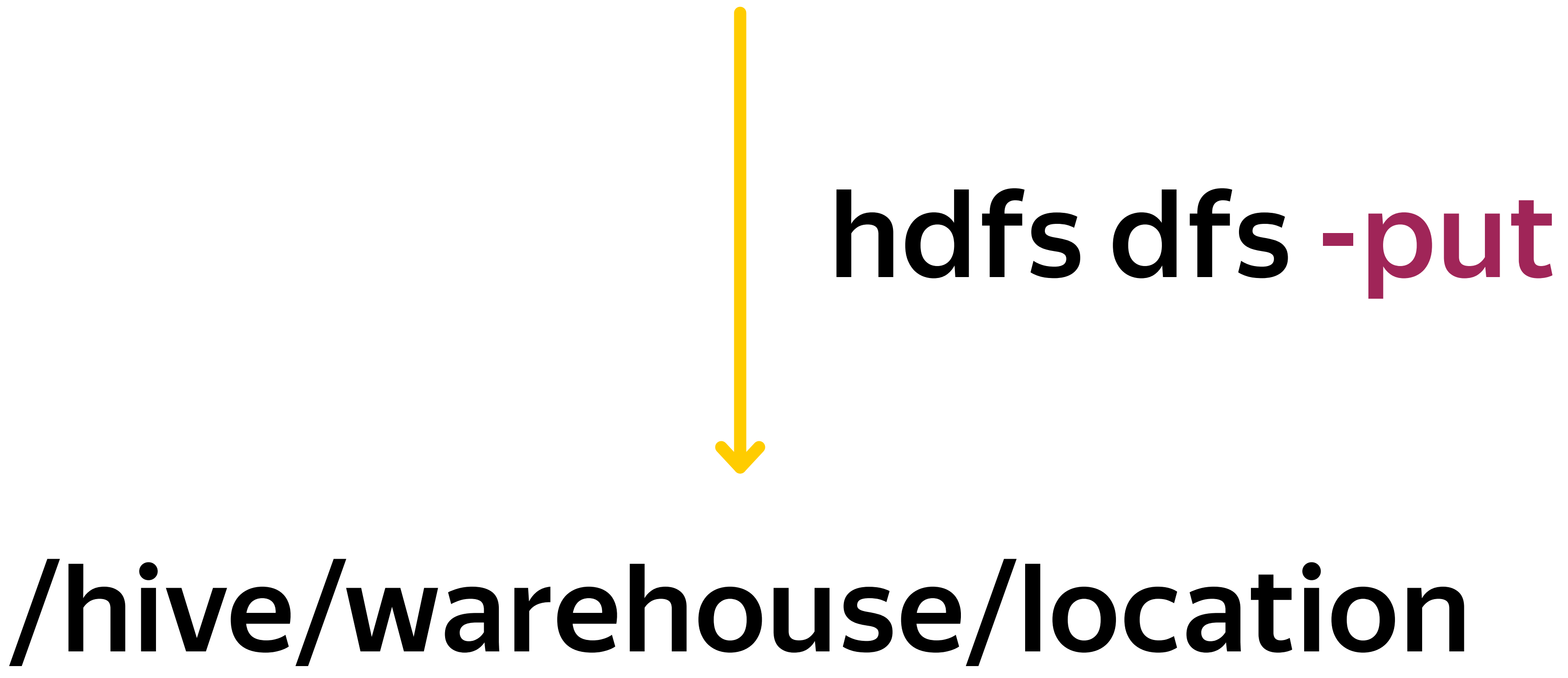


hdfs dfs -mv

/hive/warehouse/location

DML (import)

```
LOAD DATA LOCAL INPATH '/local/path/employees-data'
INTO TABLE employees;
```



hdfs dfs -put

/hive/warehouse/location

DML (import)

```
LOAD DATA [LOCAL] INPATH '/local/path/employees-data'
OWERWRITE INTO TABLE employees;
```

move

/hive/warehouse/location

erase HDFS folder
before “load”

DML (export)

```
INSERT OVERWRITE [LOCAL] DIRECTORY '/tmp/employees'
SELECT name, salary, address
FROM employees
WHERE …;
```

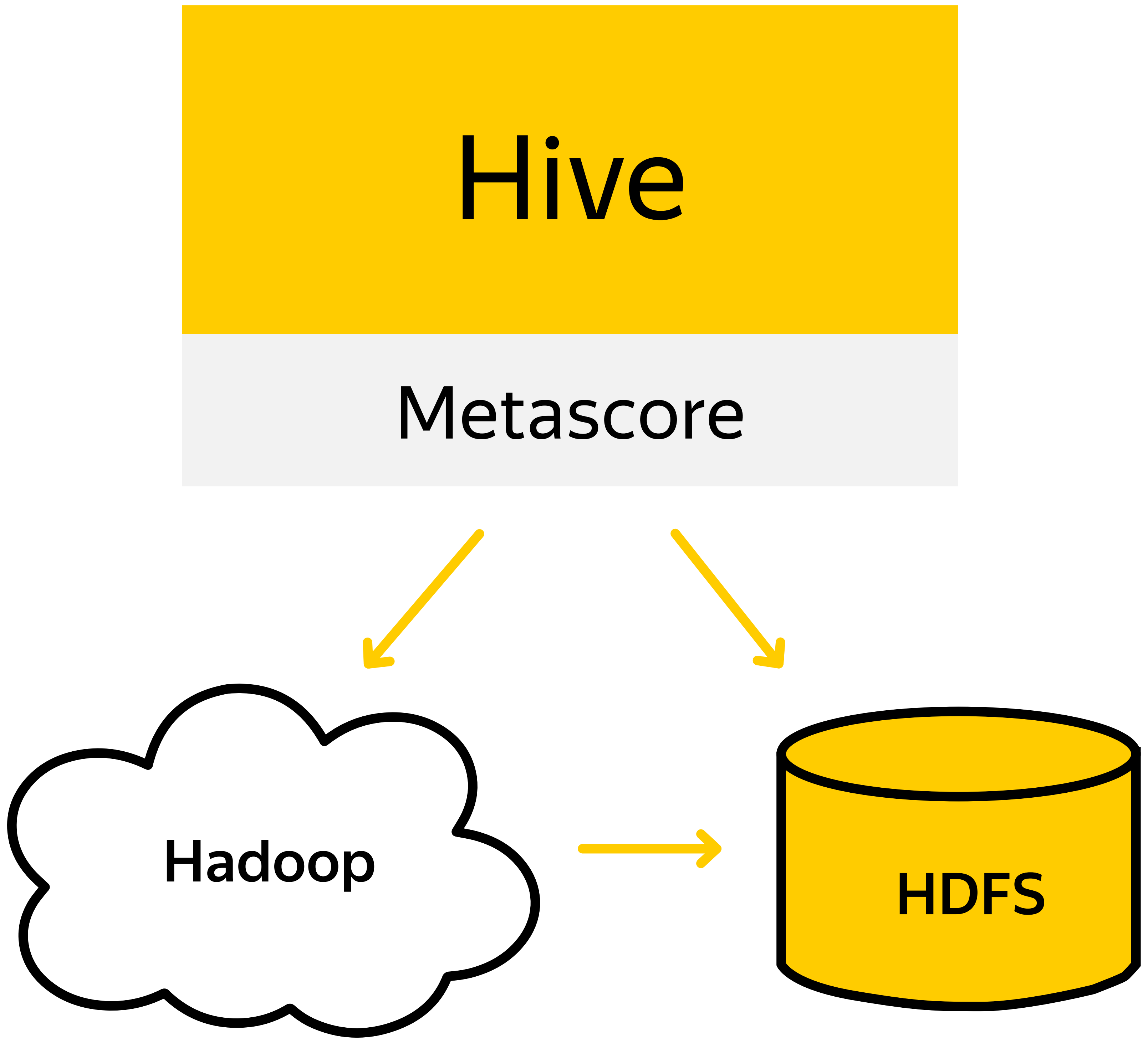
DML (export), multiple-insert statement

```
FROM employees
INSERT OVERWRITE [LOCAL] DIRECTORY '/tmp/ca_employees'
SELECT name, salary, address
WHERE state = 'CA'
INSERT OVERWRITE [LOCAL] DIRECTORY '/tmp/ny_employees'
SELECT name, salary, address
WHERE state = 'NY';
```

```
FROM raw_table
INSERT OVERWRITE TABLE us_employees
SELECT *
WHERE raw_table.country = 'US'
INSERT OVERWRITE TABLE uk_employees
SELECT *
WHERE raw_table.country = 'UK'
…;
```

DDL (CTAS)

```
CREATE TABLE ca_employees
AS SELECT name, salary, address
FROM employees
WHERE state = 'CA';
```



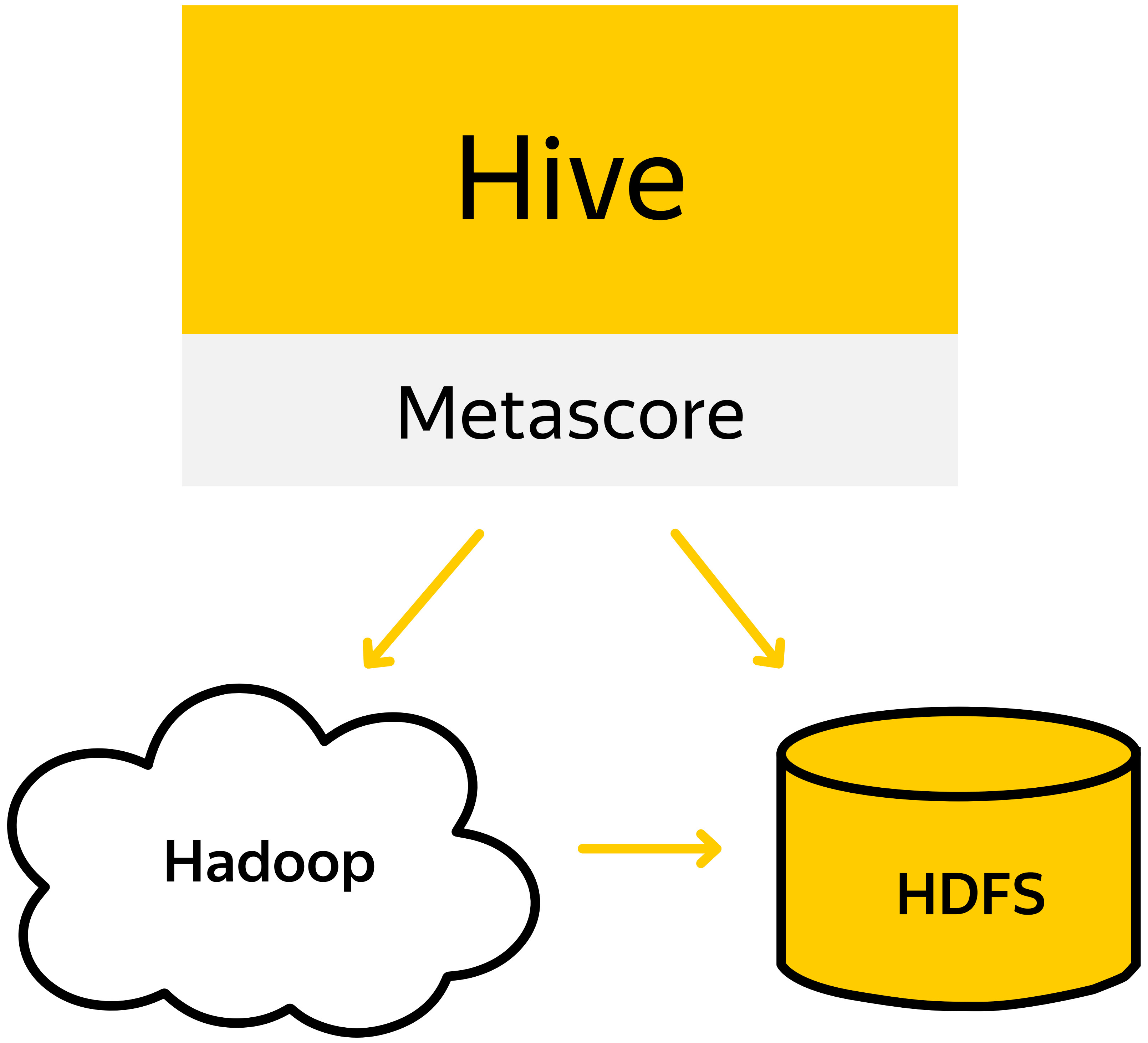
(1) DDL

(2) HiveQL (details)

(3) DML

MapReduce (?)

- SELECT .. FROM
- WHERE
- GROUP BY + HAVING
- JOIN
- ORDER BY / **SORT BY**



Hive

Metascore

Hadoop

HDFS

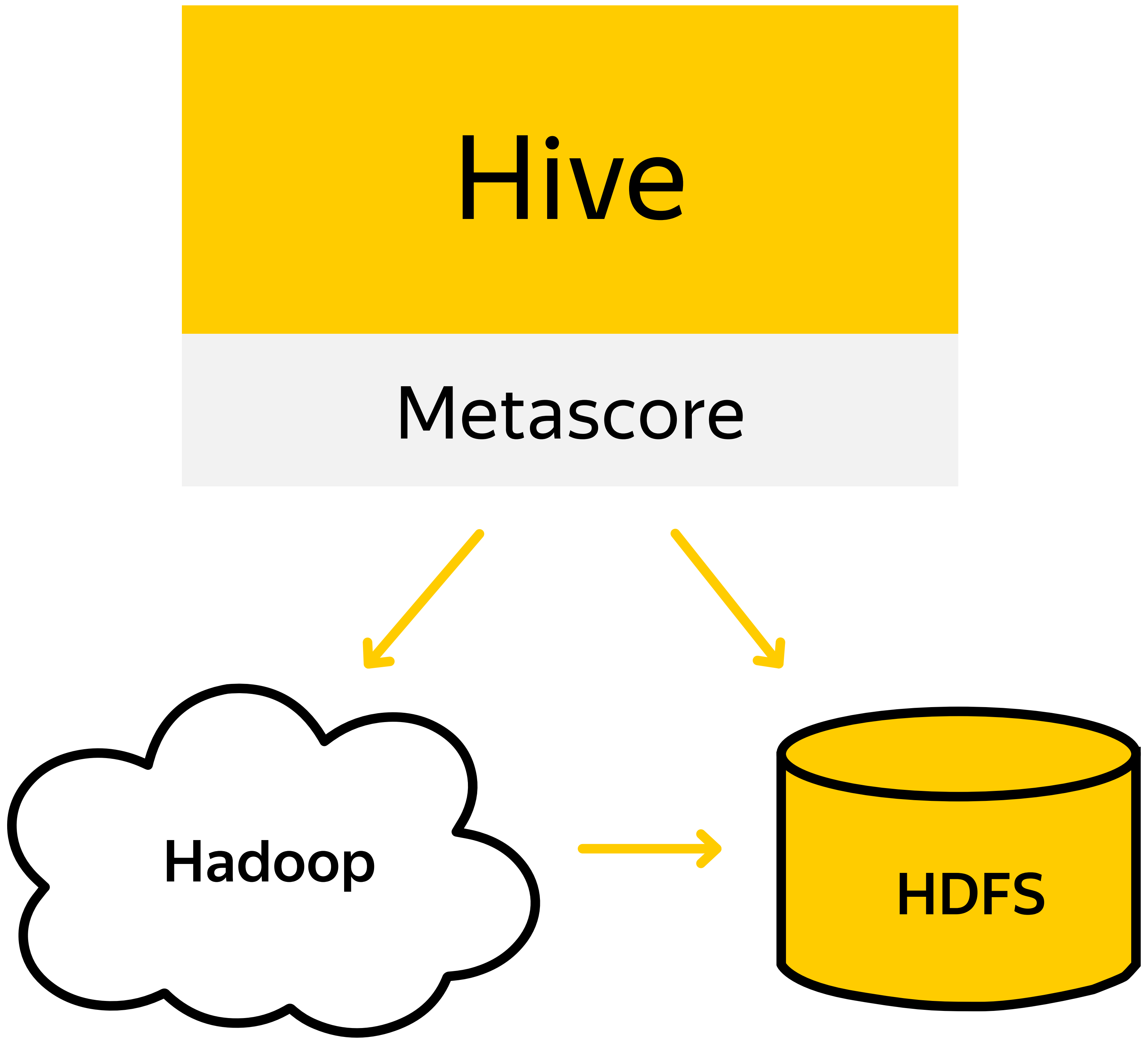
(1) DDL

(2) HiveQL (details)

(3) DML

MapReduce (?)

- SELECT .. FROM **[<-- Map]**
- WHERE
- GROUP BY + HAVING
- JOIN
- ORDER BY / **SORT BY**



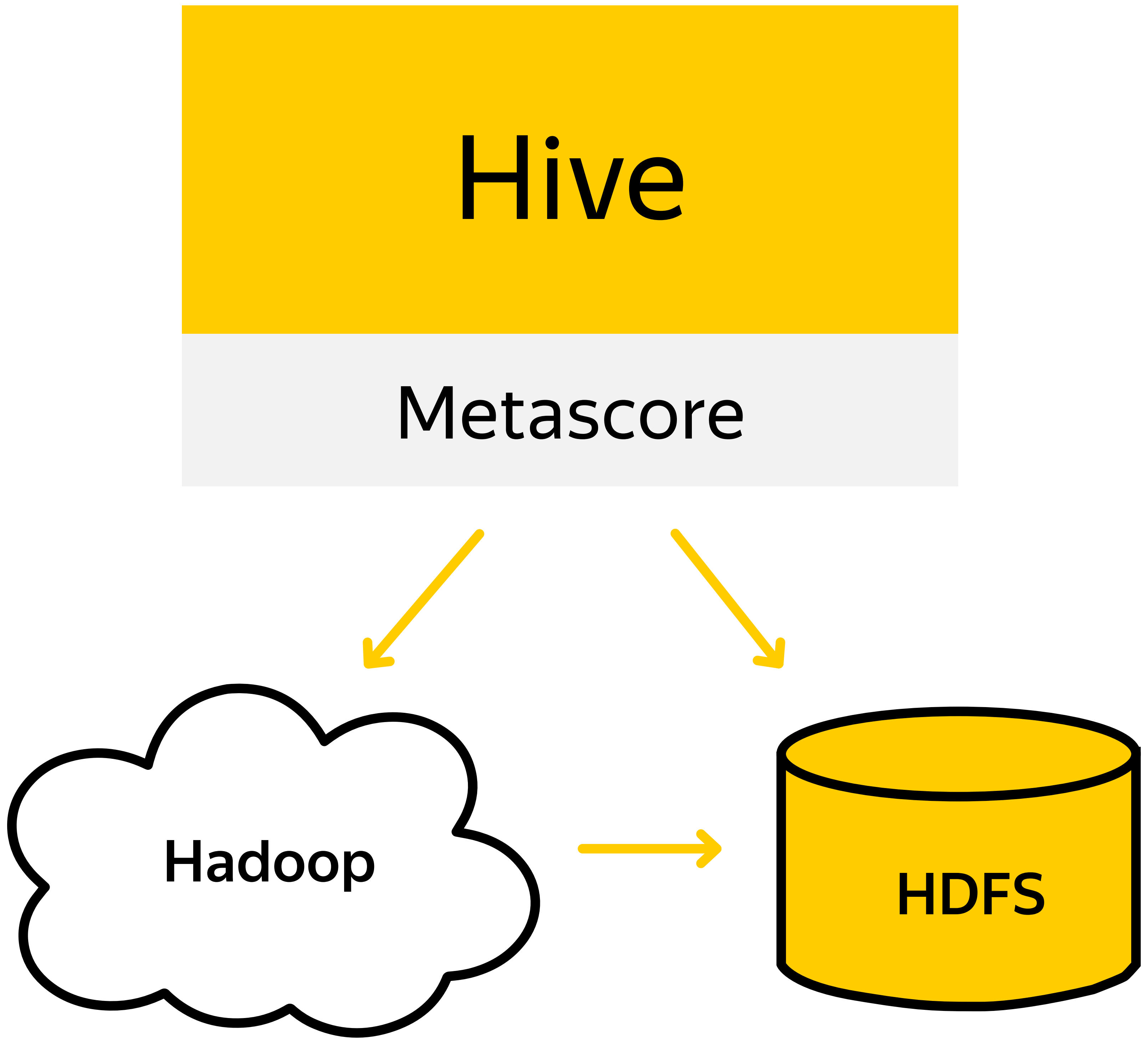
(1) DDL

(2) HiveQL (details)

(3) DML

MapReduce (?)

- SELECT .. FROM **[<-- Map]**
- WHERE **[<-- Map]**
- GROUP BY + HAVING
- JOIN
- ORDER BY / **SORT BY**



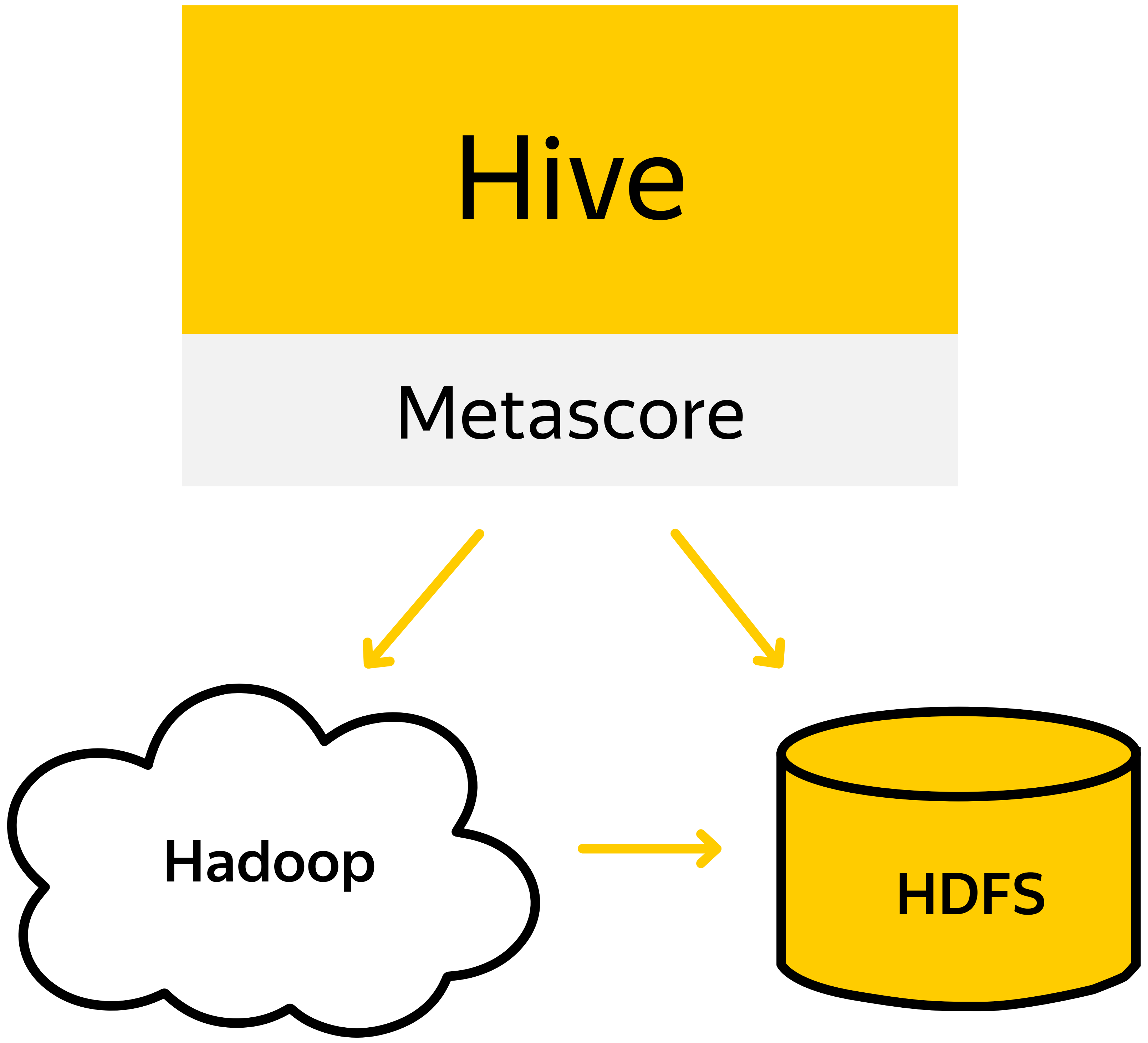
(1) DDL

(2) HiveQL (details)

(3) DML

MapReduce (?)

- SELECT .. FROM **[<-- Map]**
- WHERE **[<-- Map]**
- GROUP BY **[<-- Shuffle & Sort]** + HAVING
- JOIN
- ORDER BY / **SORT BY**



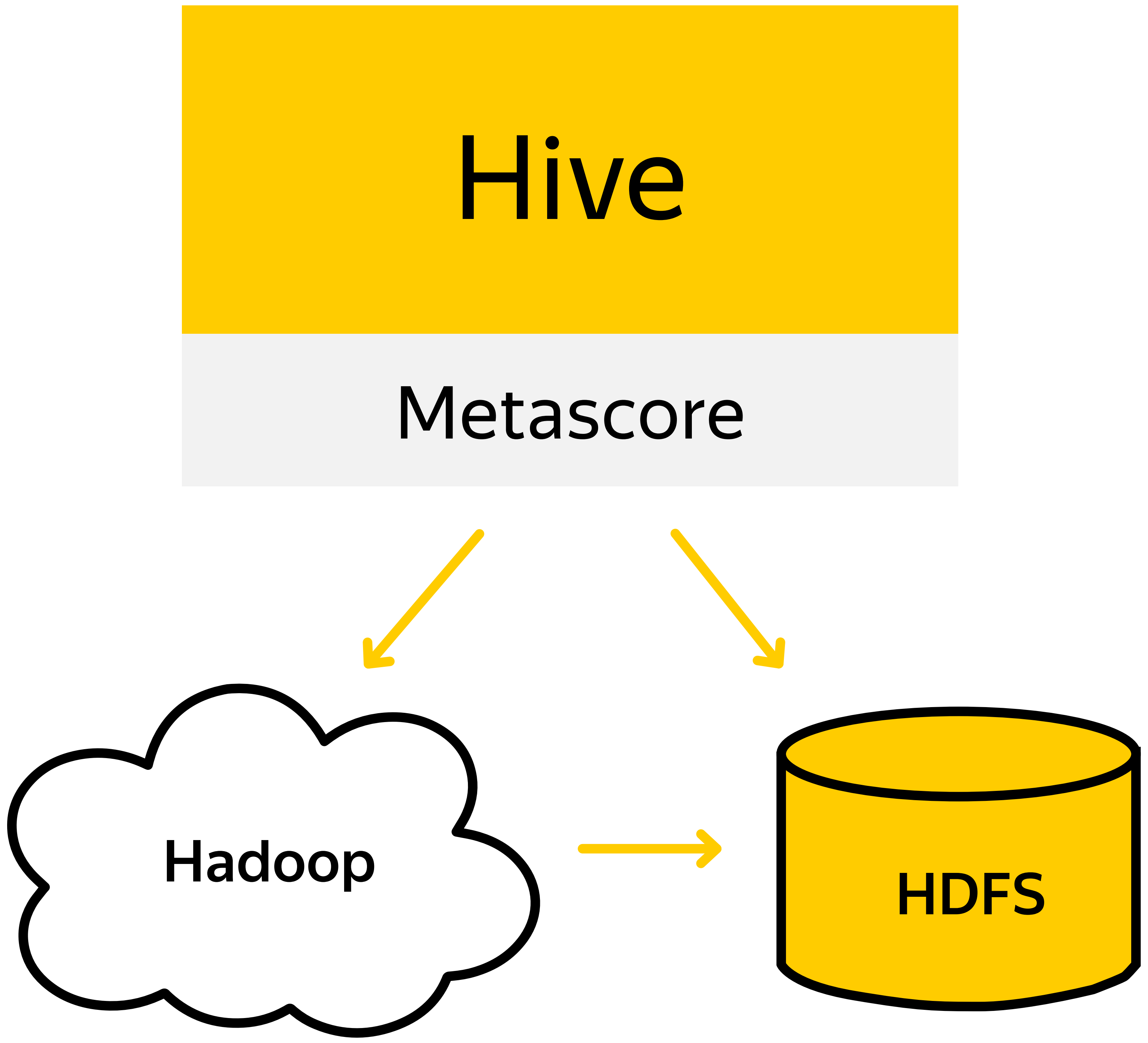
(1) DDL

(2) HiveQL (details)

(3) DML

MapReduce (?)

- SELECT .. FROM **[<-- Map]**
- WHERE **[<-- Map]**
- GROUP BY **[<-- Shuffle & Sort]** + HAVING **[<-- Reduce]**
- JOIN
- ORDER BY / **SORT BY**



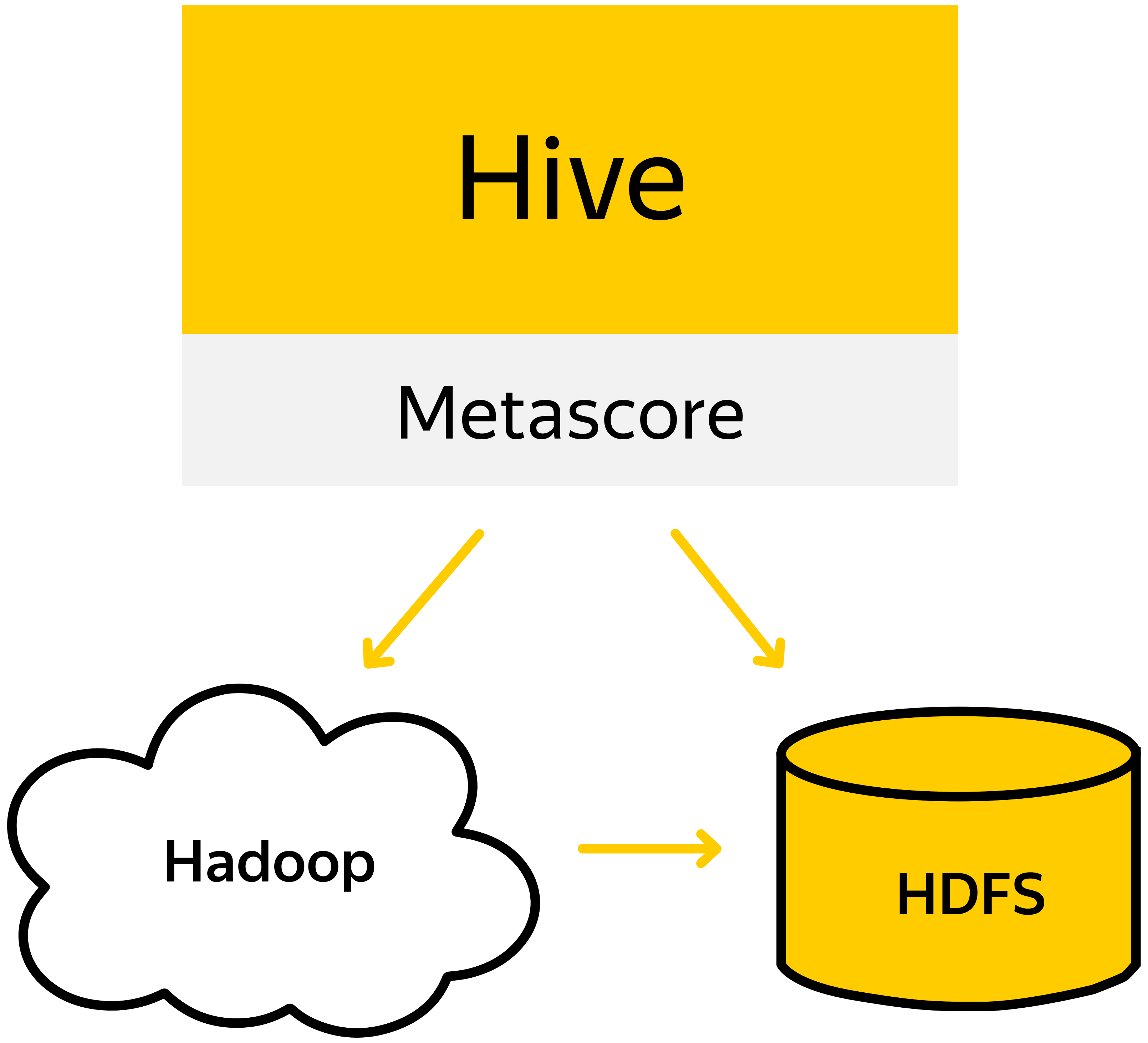
(1) DDL

(2) HiveQL (details)

(3) DML

MapReduce (?)

- SELECT .. FROM **[<-- Map]**
- WHERE **[<-- Map]**
- GROUP BY **[<-- Shuffle & Sort]** + HAVING **[<-- Reduce]**
- JOIN **[<-- Map / Reduce "-side"]**
- ORDER BY / **SORT BY**



Hive

Metascore

Hadoop

HDFS

(1) DDL

(2) HiveQL (details)

(3) DML

MapReduce (?)

- SELECT .. FROM **[<-- Map]**
- WHERE **[<-- Map]**
- GROUP BY **[<-- Shuffle & Sort]** + HAVING **[<-- Reduce]**
- JOIN **[<-- Map / Reduce "-side"]**
- ORDER BY / **SORT BY** **[<-- Reduce]**

```
EXPLAIN
FROM src
INSERT OVERWRITE TABLE dest_g1
SELECT src.key, sum(substr(src.value,4))
GROUP BY src.key;
```

```
EXPLAIN
FROM src
INSERT OVERWRITE TABLE dest_g1
SELECT src.key, sum(substr(src.value,4))
GROUP BY src.key;
```

(1) The Abstract Syntax Tree

```
ABSTRACT SYNTAX TREE:
  (TOK_QUERY (TOK_FROM (TOK_TABREF src))
 …
```

EXPLAIN

```
FROM src
INSERT OVERWRITE TABLE dest_g1
SELECT src.key, sum(substr(src.value,4))
GROUP BY src.key;
```

(1) The Abstract Syntax Tree

```
ABSTRACT SYNTAX TREE:
  (TOK_QUERY (TOK_FROM (TOK_TABREF src))
 …
```

(2) The Dependency Graph

```
  STAGE DEPENDENCIES:
  Stage-1 is a root stage
  Stage-2 depends on stages: Stage-1
  Stage-0 depends on stages: Stage-2
```

```
EXPLAIN
FROM src
INSERT OVERWRITE TABLE dest_g1
SELECT src.key, sum(substr(src.value,4))
GROUP BY src.key;
```

(1) The Abstract Syntax Tree

```
ABSTRACT SYNTAX TREE:
  (TOK_QUERY (TOK_FROM (TOK_TABREF src))
 …
```

(2) The Dependency Graph

```
 STAGE DEPENDENCIES:
  Stage-1 is a root stage
  Stage-2 depends on stages: Stage-1
  Stage-0 depends on stages: Stage-2
```

(3) The plans of each Stage

```
STAGE PLANS:
  Stage: Stage-1
    Map Reduce
      Alias -> Map Operator Tree:
        src
            Reduce Output Operator
              key expressions:
                    expr: key
                    type: string
              sort order: +
```

Summary

Summary

- You can **move** the data **in** and **out** of Hive warehouse

Summary

- You can **move** the data **in** and **out** of Hive warehouse
- You can **create** new Hive tables on-the-fly and **populate** existing ones

Summary

- You can **move** the data **in** and **out** of Hive warehouse
- You can **create** new Hive tables on-the-fly and **populate** existing ones
- You can **use** “explain” to get details of MapReduce Job(s) breakdown

Summary

- You can **move** the data **in** and **out** of Hive warehouse
- You can **create** new Hive tables on-the-fly and **populate** existing ones
- You can **use** “explain” to get details of MapReduce Job(s) breakdown

see: https://cwiki.apache.org/confluence/display/Hive/LanguageManual+DML#LanguageManualDML-Update
see: https://cwiki.apache.org/confluence/display/Hive/LanguageManual+Explain

BigDATAteam