

Modeling and Managing Data with Impala and Hive

Hive, Impala operate query on tables, just like in an RDBMS

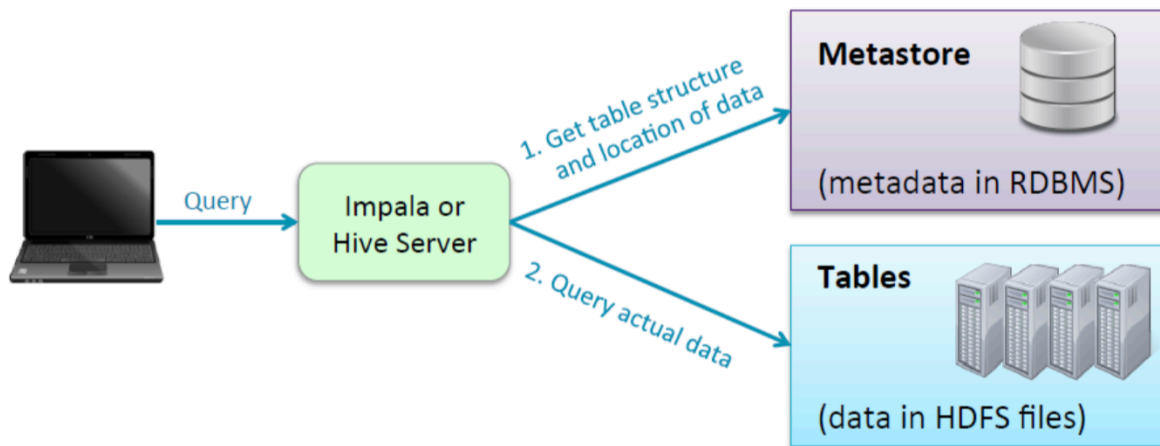
- Default path : /user/hive/warehouse/<table_name>
 - Table name 이 없으면 없이 저장됨

structure and location of tables

- these are specified when tables are created
- the metadata is stored in the Metastore
 - contained in an RDBMS such as Mysql

Tables in HDFS, metadata in the Metastore

- Hive, Impala work with same data



Metadata vs Data

Metadata		cust_id	name	country
		001	Alice	us
		002	Bob	ca
		003	Carlos	mx
	
Data		392	Maria	it
		393	Nigel	uk
		394	Ophelia	dk
	

- Data refers to the information you store and process
- Metadata describes important aspect of that data

DDL

- very similar to standard SQL DDL

creating a database

```
create database loudacre;
create database if not exists loudacre;
```

- adds the database definition to the metastore
- creating a storage directory in HDFS
 - user/hive/warehouse/loudacre.db

removing a database

```
drop database loudacre;
drop database if exists loudacre;

//
drop database loudacre cascade;
```

데이터 타입

데이터 타입	설명
TINYINT	1바이트 크기 정수
SMALLINT	2바이트 크기 정수
INT	4바이트 크기 정수
BIGINT	8바이트 크기 정수
BOOLEAN	TRUE 또는 FALSE
FLOAT	4바이트 부동소수점
DOUBLE	8바이트 부동소수점
STRING	문자열
TIMESTAMP	타임스탬프 (1970년 1월 1일 0시 이후 흘러간 시간)

기타: BINARY, STRUCT, MAP, ARRAY

```
CREATE TABLE tablename (colname DATATYPE, ...)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY char
STORED AS {TEXTFILE|SEQUENCEFILE|...}
```

- 첫번째 줄 : specify a name for the table, and list the column names and datatypes
- 해당 테이블 내의 데이터가 어떠한 형식으로 저장되는지, 그 밑줄에 field는 char로 구분되어진다.
- stored as 절은 데이터 저장 파일 포맷을 의미한다. 하이버는 textfile과 sequencfile을 지원한다.

```
create table jobs_archived like jobs
-- 같은 테이블의 구조를 가지나 데이터를 다르게 보관하기 위해서 쓰임

create table ny_customers as select cust_id, fname, lname from customers where
state = 'ny';
-- select table 의 결과를 가지고 만들어짐, 데이터도 포함함

location '/loudacre/jobs' --옵션을 넣으면 어디에 저장될지 설정가능
```

External table

- 외부테이블은 hive.metastore.warehouse.dir속성이 가리키는 디렉터리에 데이터를 보존하지 않고 테이블

설정시 설정한 경로로 데이터를 저장한다. 사용자가 실수로 테이블을 drop 했더라도 데이터가 보존된다 (메타 데이터만 사라짐)

- location 옵션을 포함해야함

```
CREATE EXTERNAL TABLE adclicks
( campaign_id STRING,
  click_time TIMESTAMP,
  keyword STRING,
  site STRING,
  placement STRING,
  was_clicked BOOLEAN,
  cost SMALLINT)
LOCATION '/loudacre/ad_data';
```

1. Make a table in Impala-Shell

- Check the result for SHOW TABLES
- Check the result for DESCRIBE and DESCRIBE FORMATTED
- Check the result for SHOW CREATE TABLE
- Check the result in Beeline

2. Check the difference between Impala-Shell and Beeline

- Make another table in Beeline
- Check the result using the commands above
- Check the result in Impala-Shell using the commands above
- Execute “invalidate metadata”, then check the result in Impala-Shell

- invalidate metadata를 안쓰면 caching 으로 인해 결과값이 현재 결과와 다를 수 있음.

Data validation

- unlike an rdbms, they do not validate data on insert
 - files are simply moved into place
 - loading data into tables is therefore very fast
 - missing or invalid data는 null로 표시

1. Make a simple python script to generate a given numbers of strings (one per line)
 - python random_strings.py 100
2. Make three files using the python script to generate random strings with 10, 100, and 1000 numbers, respectively (called, ten_strings.txt, hundred_strings.txt, and thousand_strings.txt)
3. Upload the generated files into HDFS
4. Create a table in Impala to store the generated data
5. Load a file, called thousand_strings.txt, into the table and check the total inserted number of data
6. Append a file, called hundred_strings.txt, into the table and check the total inserted number of data
7. Overwrite a file, called ten_strings.txt into the table and check the total inserted number of data

Appending selected records to a table

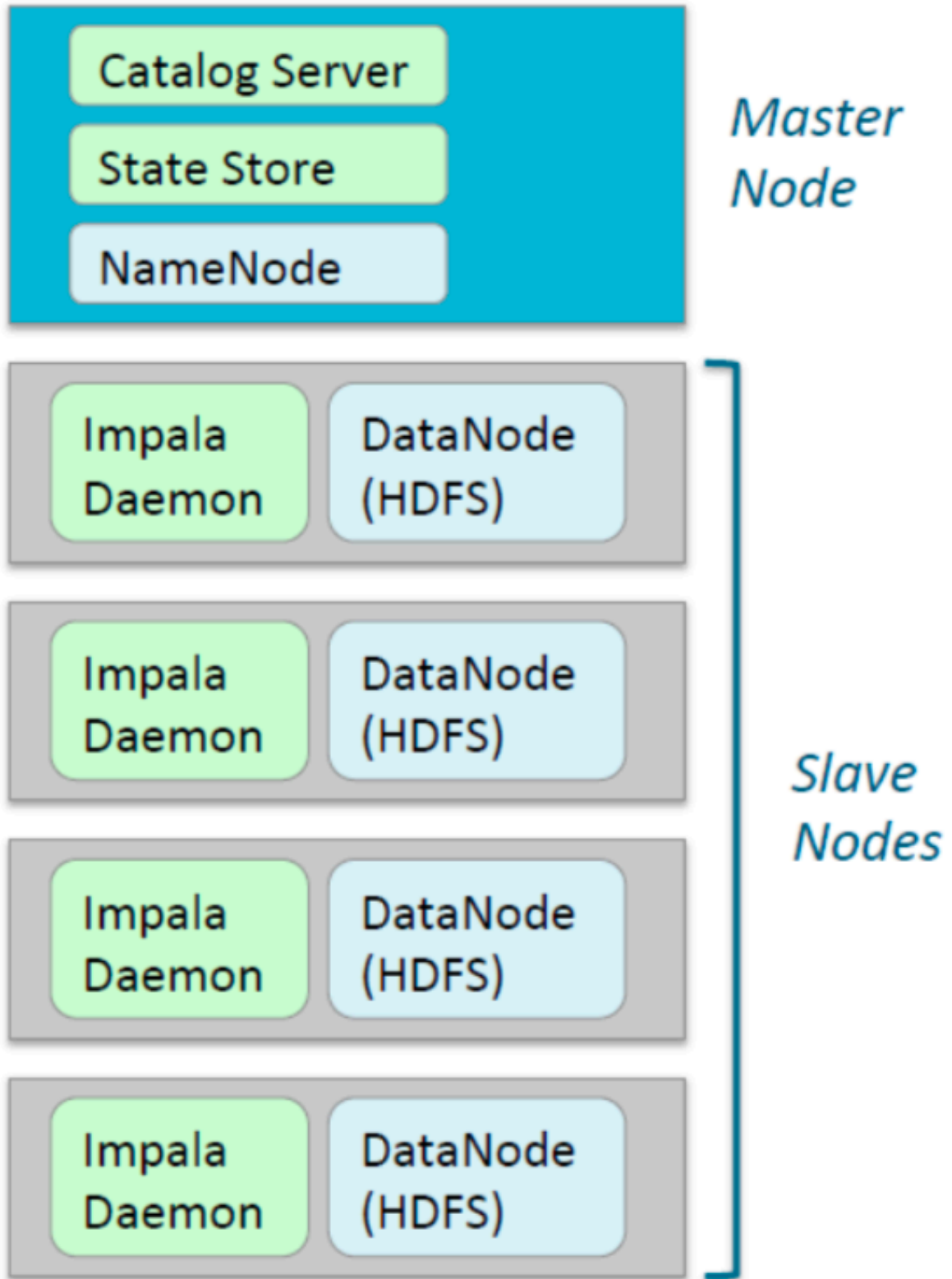
```
-- use insert into to add results to an existing hive table
Insert into table accounts_copy select * from accouts;

-- specify a where clause
Insert into table accounts_copy select * from accouts where
year(acct_create_dt) = 2008 and acct_close_dot is null;
```

Hcatalog

- hive에서 meta data를 관리할 수 있음
- command line 과 rest api로 접근
- allows you to define tables using (hiveQL DDL) syntax
- drop table 할 경우 external table도 다 지워짐 (메타데이터만 지우는 것이 아니라)

Impala in the Cluster



node에 impala daemon을 따로 두는 이유는 한 곳에 impala demon를 두면 네트워크 병목 현상이 일어남

- 한개만 있는 경우 필터링이 어려움, 데이터 처리를 할 수 있는 로직이 한 곳에만 있다 보니깐 미리 필터링을 해놓을 수가 없다. 내부적으로 데이터 처리 로직이 있어서 SQL을 쓸 수 있는 것이다. 노드 내에서 최대한 필터링을 하고 결과를 리턴시키는 것이 이상적임.

데이터 관리 기술의 핵심 : 단순히 키워드가 있는 데이터를 찾아 주는 것이 아니라 내부에 있는 데이터가 굉장히 많다고 가정할때 찾으려는 데이터를 최대한 필터링을 해서 연관된 데이터를 뽑아 주는 것

State store - 여러개의 impala daemon을 loop up, check status (임팔라 데몬이 잘 동작해야하만 데이터 손실없이 처리할 수 있음)

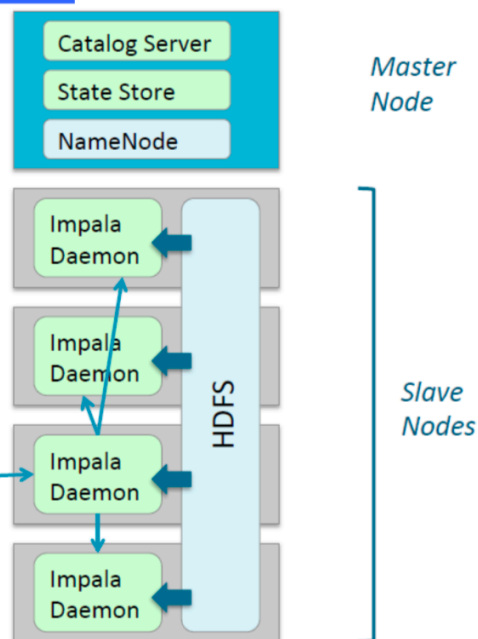
Catalog - relays metadata changes to all the impala daemons in a cluster

- 데이터를 파티션 해도 처리하는 경우 아무문제가 없어야 되는데 메타데이터는 테이블에 대한 정보가 한개여서 나눠서 처리할 수 없다.
- horizontal partitioning, vertical partitioning
 - 일반적으로 horizontal 을 많이 사용했으나 column store라는 서비스가 생성되었을때 많이 접근하는 column의 그룹이 있다는 가정하에 vertical partitionaing을 사용하게 되었고 하둡에서도 HBase가 있다.
 - 근본적으로는 horizontal partitioning을 따르고 있기 때문에 metadata를 나눌 수 없음

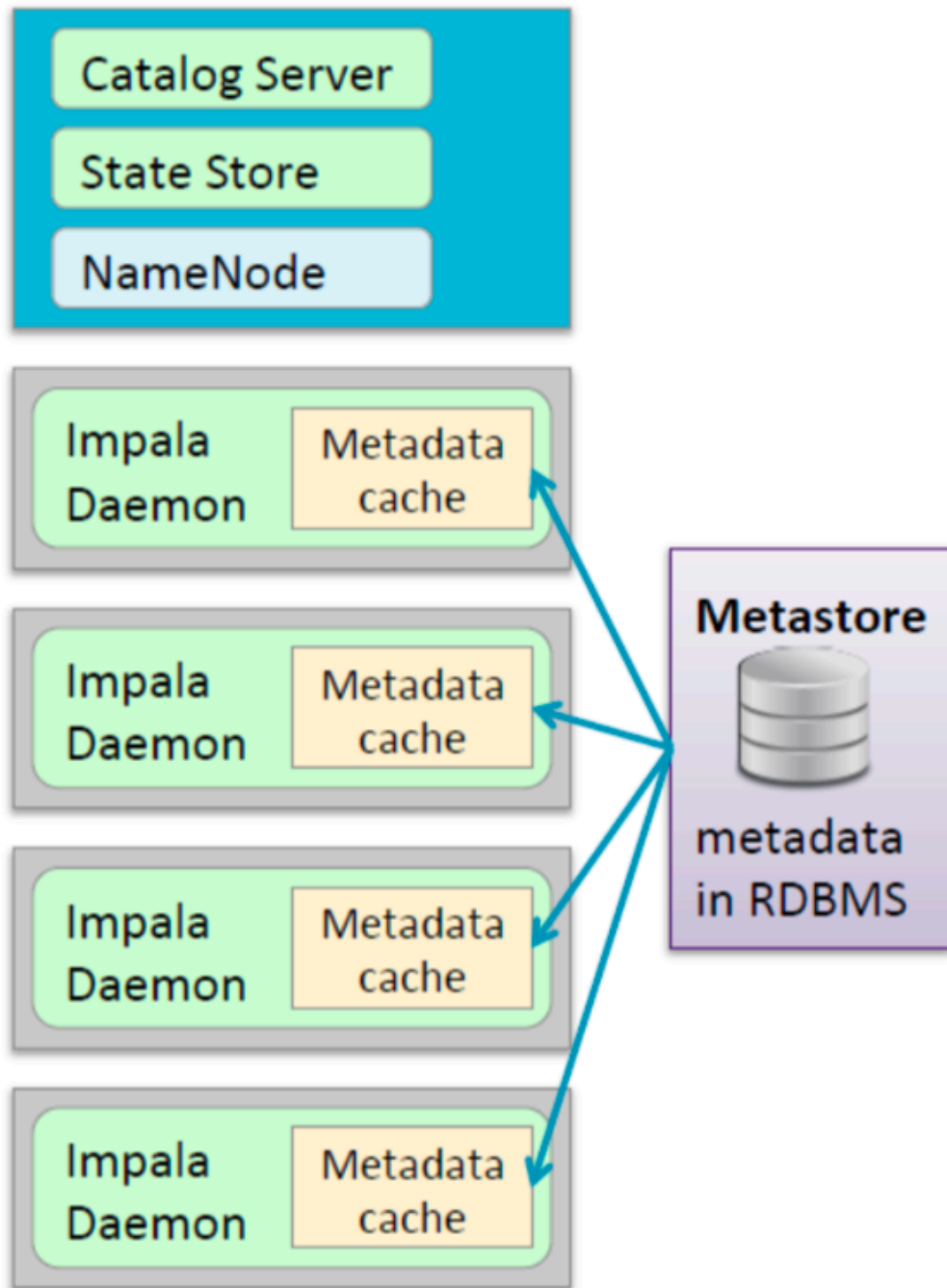
How Impala Executes a Query

■ Impala daemon plans the query

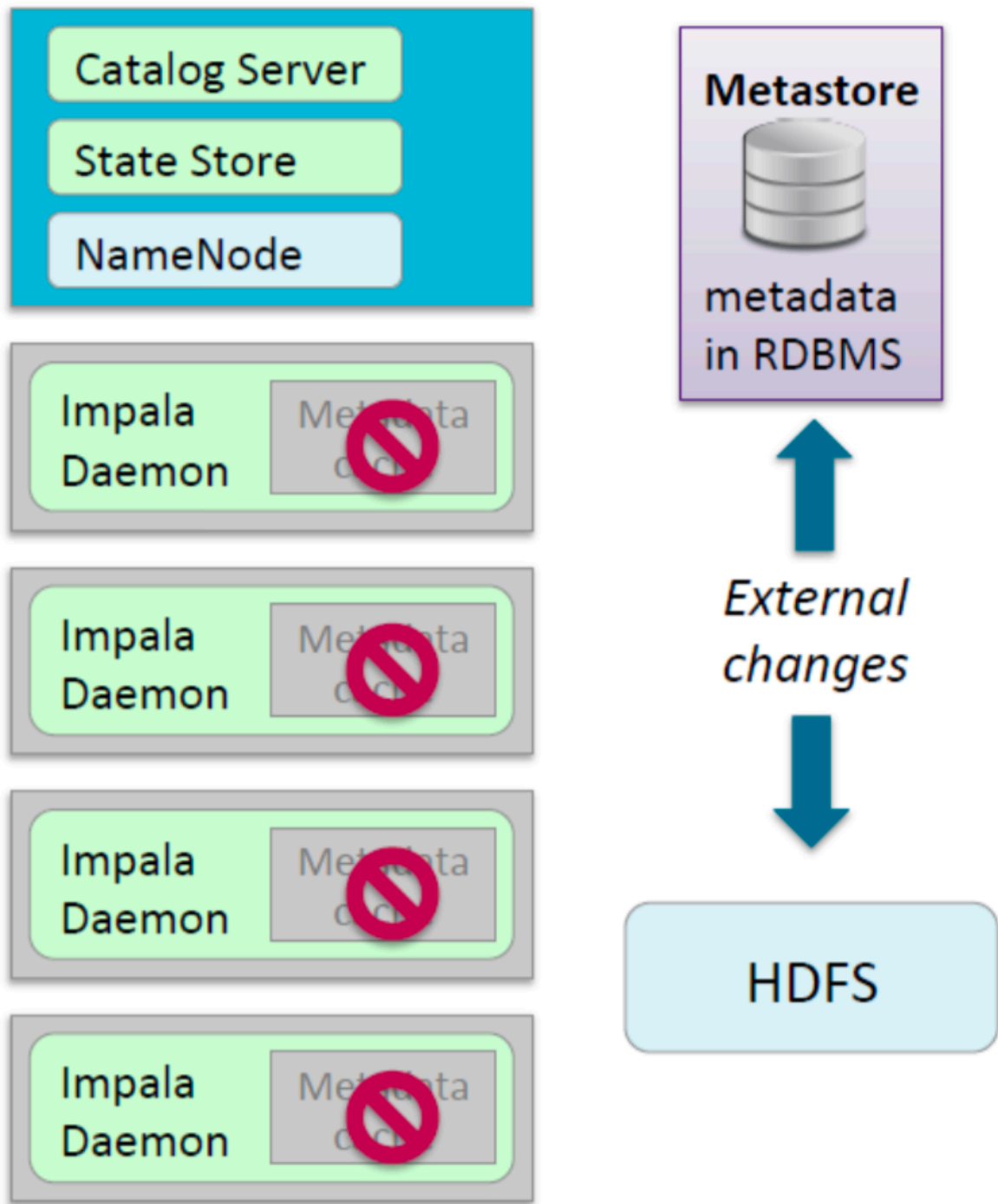
- Client (impala-shell or Hue) connects to a local impala daemon
 - This is the *coordinator*
- Coordinator requests a list of other Impala daemons in the cluster from the State Store
- Coordinator distributes the query across other Impala daemons
- Streams results to client



- Available한 노드 아무노드나 질의를 받아서 coordinator역할을 하게됨



- 메타데이터를 접근해야되는데 그때마다 catalog server와 통신하는 것을 막기위해서 metastore를 캐싱함.
 - 이런 구조로 성능이 빨라짐



- invalidate metadata를 사용해야됨
 - metadata는 outside of impala에서 수정되어짐
 - 그래서 meta data 캐시가 invalid가 됨

External Metadata Change	Required Action	Effect on Local Caches
New table added	INVALIDATE METADATA (with no table name)	Marks the entire cache as stale; metadata cache is reloaded as needed.
Table schema modified <i>or</i> New data added to a table	REFRESH <table>	Reloads the metadata for one table <i>immediately</i> . Reloads HDFS block locations for new data files only.
Data in a table extensively altered, such as by HDFS balancing	INVALIDATE METADATA <table>	Marks the metadata for a single table as stale. When the metadata is needed, all HDFS block locations are retrieved.