



# APACHE HIVE BASICS

# Hive

- Large datasets [ Equi joins, more built in function, Row level updates, deletes as special case]
  - Parallel computations – Distributed systems with multiple machines
  - High Latency – Fetching a row will run a mapreduce that might take minutes
  - Read Operations – Data can be sourced into Hive tables for schema on read

- Hive is an Abstraction/Layer on top of Map Reduce
- Hive is an open source Data warehouse and part of Larger Hadoop ecosystem
- Hive runs on top of distributed computing framework
- Hive stores the data in HDFS
- Hive runs all processes in the form of MR jobs

Hive Metastore holds  
schema[metadata]

Warehouse directory :  
Data stored in Hdfs

# Hive Metastore

- Metastore is the bridge between data stored in HDFS as files and the tables exposed to users.
  - Stores metadata for all the tables in Hive.
  - Maps the files and directories in Hive to tables.
- Any database with JDBC driver can be used as Metastore
  - Derby database can be used for Dev environments

Default location :  
/user/hive/warehouse  
Check out : [hive-site.xml](#)



- Hive has 2 parts :
1. Data – It is stored in the form of Files
  2. Schema or metadata

Schema :  
Emp\_name  
Emp\_id

Data :  
Ram Kumar    25362  
Ashik         25566

Hive Metastore stores all Metadata in  
Derby database and MySQL is  
preferred choice

# Hive Complex datatypes

- Array[collection datatype with no fixed size and only arrays of primitive types were allowed]
- Map[ unordered collection of pairs & no fixed Size. Each entity is key & value pair]
  - Struct [ Logical grouping of data and can have different data types]

# Hive Built in functions

- UDF – User defined functions [ works on single row and output also single row] ex:  
trim(),concat(),length(),round(),floor()

- UDAF- User defined aggregate functions [ works on multiple rows and output is single row()]  
ex :  
count(\*),sum(),avg()

- UDTF- User defined table generating functions [ works on single and outputs multiple row] ex:  
explode(),posexplode()

# Hive explode function

- Flatten the data in arrays & maps

Manager_Name	Team members
suren	[Karthik,anish,suja]
Arun	[kamini,suresh,arjun]

Karthik
anish
suja
kamini
suresh
arjun



# Lateral View

1. Virtual table formed by exploded view which can be joined with the original tables to allow complex queries

Manager_Name	Team members	Team members
suren	[Karthik,anish,suja]	Karthik
		anish
Arun	[kamini,suresh,arjun]	suja
		kamini
		suresh
		arjun

# Types of Tables

1. Internal Tables or Managed Tables [ If you drop the table both data and metadata will be gone]
2. External Tables [ If we drop the table only metadata will be gone and data will still be available in HDFS location]

# Hive set operations

- Union & Union All
  - Minus
  - Intersect

- Subqueries [IN/NOT IN ]  
[ Exists/non Exists]
  - views

## How to create table and See the schema in Hive:

```
hive>  
>  
> create table customers  
> (  
> id bigint,  
> name string,  
> address string);  
OK  
Time taken: 1.269 seconds  
hive> describe customers;  
OK  
id                bigint  
name              string  
address           string  
Time taken: 0.791 seconds, Fetched: 3 row(s)  
hive> █
```

```
hive> describe formatted customers;  
OK  
# col_name          data_type          comment  
id                  bigint  
name                string  
address             string  
  
# Detailed Table Information  
Database:           testing  
Owner:              cloudera  
CreateTime:         Wed Jun 22 05:54:55 PDT 2022  
LastAccessTime:     UNKNOWN  
Protect Mode:       None  
Retention:          0  
Location:           hdfs://quickstart.cloudera:8020/user/hive/warehouse/testing.db/customers  
Table Type:         MANAGED_TABLE  
Table Parameters:  
    transient_lastDdlTime    1655902495  
  
# 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:  
    serialization.format    1  
Time taken: 0.247 seconds, Fetched: 28 row(s)  
hive> █
```

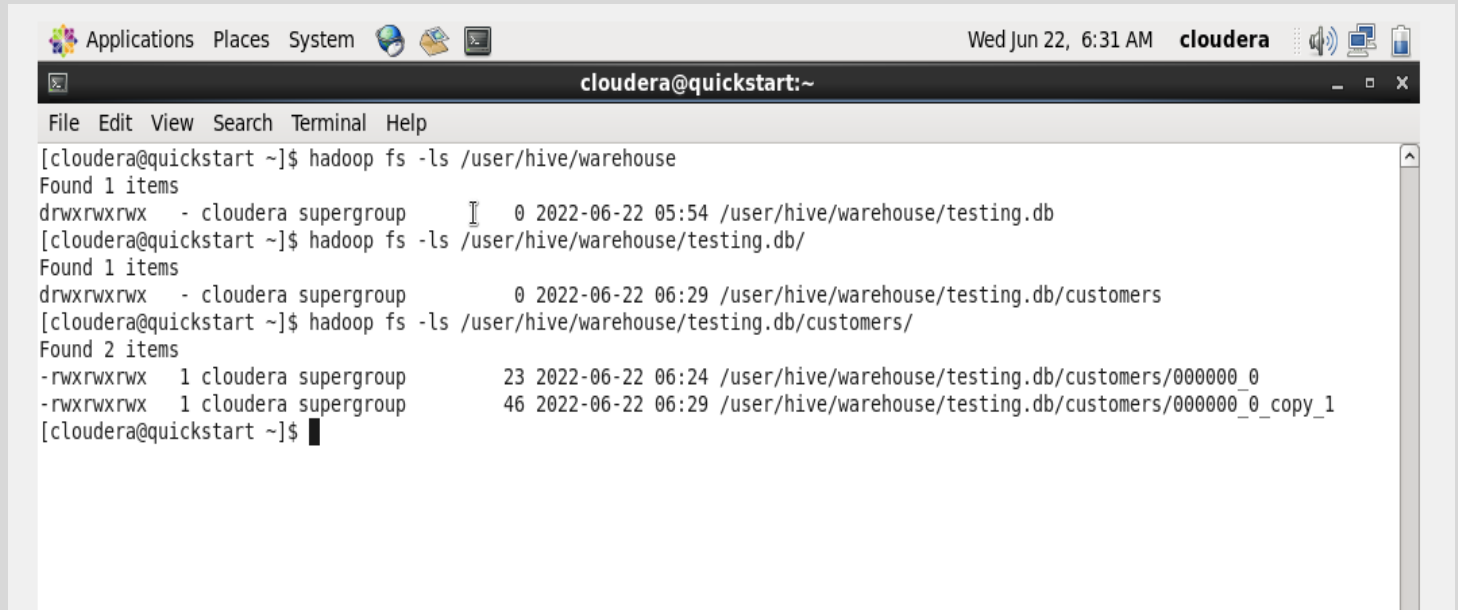
New Tab - Mozilla Firef...

cloudera@quickstart: ~

cloudera@quickstart: ~

Data stored in Warehouse directory & Look for the data inserted :

Database : Testing  
Tables: customers



A terminal window titled 'cloudera@quickstart:~' showing the execution of Hadoop file system commands. The window has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The terminal output shows the following commands and results:

```
[cloudera@quickstart ~]$ hadoop fs -ls /user/hive/warehouse
Found 1 items
drwxrwxrwx  - cloudera supergroup      0 2022-06-22 05:54 /user/hive/warehouse/testing.db
[cloudera@quickstart ~]$ hadoop fs -ls /user/hive/warehouse/testing.db/
Found 1 items
drwxrwxrwx  - cloudera supergroup      0 2022-06-22 06:29 /user/hive/warehouse/testing.db/customers
[cloudera@quickstart ~]$ hadoop fs -ls /user/hive/warehouse/testing.db/customers/
Found 2 items
-rwxrwxrwx  1 cloudera supergroup      23 2022-06-22 06:24 /user/hive/warehouse/testing.db/customers/000000_0
-rwxrwxrwx  1 cloudera supergroup      46 2022-06-22 06:29 /user/hive/warehouse/testing.db/customers/000000_0_copy_1
[cloudera@quickstart ~]$
```




Use Beeline to see data in structured Tabular format:

```
beeline -u  
jdbc:hive2://
```

```
[cloudera@quickstart ~]$ beeline -u jdbc:hive2://  
scan complete in 6ms  
Connecting to jdbc:hive2://  
Connected to: Apache Hive (version 1.1.0-cdh5.13.0)  
Driver: Hive JDBC (version 1.1.0-cdh5.13.0)  
Transaction isolation: TRANSACTION_REPEATABLE_READ  
Beeline version 1.1.0-cdh5.13.0 by Apache Hive  
0: jdbc:hive2://> select * from testing.customers limit 2;  
OK  
+-----+-----+-----+  
| customers.id | customers.name | customers.address |  
+-----+-----+-----+  
| 232252       | sunitha       | Newyork          |  
| 232952       | hasan        | London           |  
+-----+-----+-----+  
2 rows selected (1.9 seconds)  
0: jdbc:hive2://> █
```

## Running beeline query from Terminal:

```
beeline -u  
jdbc:hive2:// -e  
"select * from  
testing.customers  
limit 2"
```

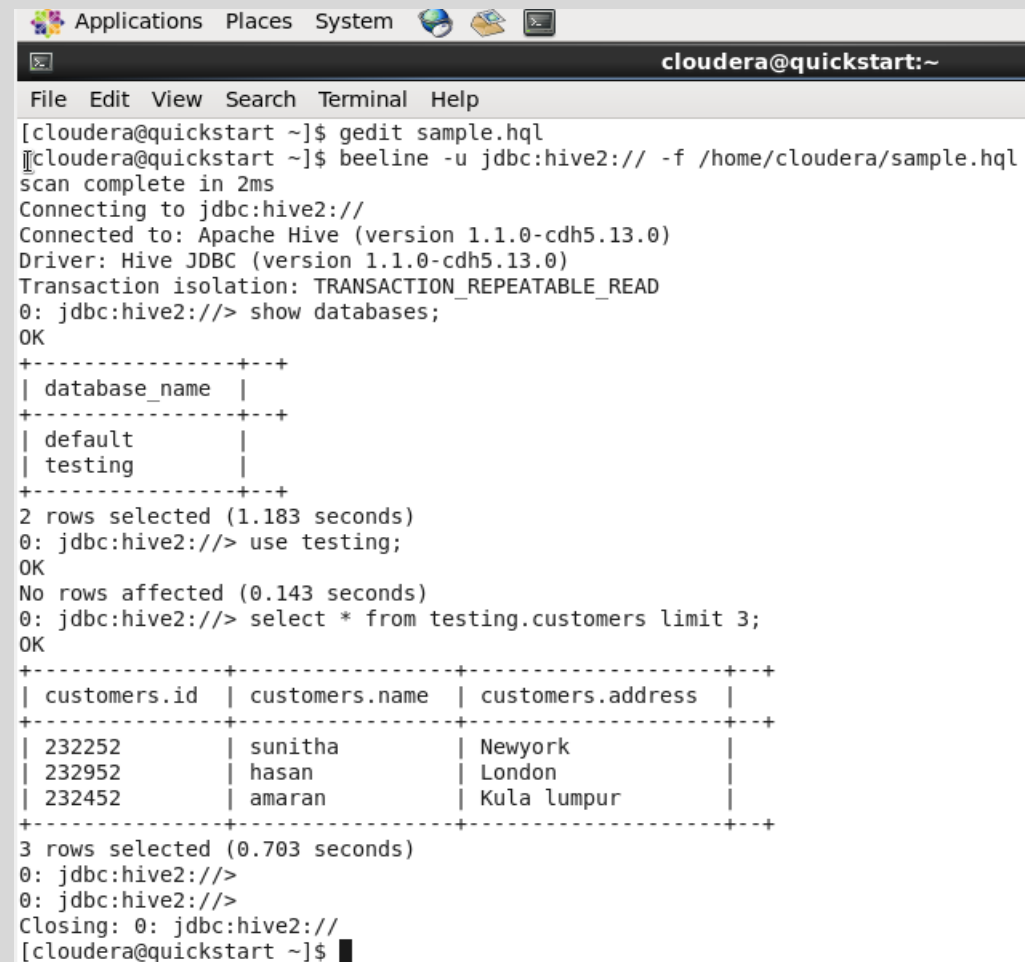


The screenshot shows a terminal window titled "cloudera@quickstart:~" with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal output shows the execution of a beeline query. The query is "select \* from testing.customers limit 2". The output shows the query was successful, returning 2 rows. The rows are displayed in a table format with columns: customers.id, customers.name, and customers.address. The first row has id 232252, name sunitha, and address Newyork. The second row has id 232952, name hasan, and address London. The terminal also shows the connection details for the Hive JDBC driver and the transaction isolation level.

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
[cloudera@quickstart ~]$ beeline -u jdbc:hive2:// -e "select * from testing.customers limit 2"  
scan complete in 1ms  
Connecting to jdbc:hive2://  
Connected to: Apache Hive (version 1.1.0-cdh5.13.0)  
Driver: Hive JDBC (version 1.1.0-cdh5.13.0)  
Transaction isolation: TRANSACTION_REPEATABLE_READ  
OK  
+-----+-----+-----+  
| customers.id | customers.name | customers.address |  
+-----+-----+-----+  
| 232252      | sunitha       | Newyork          |  
| 232952      | hasan         | London           |  
+-----+-----+-----+  
2 rows selected (2.288 seconds)  
Beeline version 1.1.0-cdh5.13.0 by Apache Hive  
Closing: 0: jdbc:hive2://  
[cloudera@quickstart ~]$
```

## Run a HQL script through Beeline:

```
beeline -u jdbc:hive2:// -f  
/home/cloudera/sample.hql
```



A terminal window titled "cloudera@quickstart:~" with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the execution of a Beeline command to run an HQL script. The script connects to a Hive database, shows databases, switches to the 'testing' database, and queries the 'customers' table, returning 3 rows.

```
[cloudera@quickstart ~]$ gedit sample.hql
[cloudera@quickstart ~]$ beeline -u jdbc:hive2:// -f /home/cloudera/sample.hql
scan complete in 2ms
Connecting to jdbc:hive2://
Connected to: Apache Hive (version 1.1.0-cdh5.13.0)
Driver: Hive JDBC (version 1.1.0-cdh5.13.0)
Transaction isolation: TRANSACTION_REPEATABLE_READ
0: jdbc:hive2://> show databases;
OK
+-----+
| database_name |
+-----+
| default      |
| testing      |
+-----+
2 rows selected (1.183 seconds)
0: jdbc:hive2://> use testing;
OK
No rows affected (0.143 seconds)
0: jdbc:hive2://> select * from testing.customers limit 3;
OK
+-----+-----+-----+
| customers.id | customers.name | customers.address |
+-----+-----+-----+
| 232252       | sunitha       | Newyork           |
| 232952       | hasan         | London            |
| 232452       | amaran        | Kula lumpur       |
+-----+-----+-----+
3 rows selected (0.703 seconds)
0: jdbc:hive2://>
0: jdbc:hive2://>
Closing: 0: jdbc:hive2://
[cloudera@quickstart ~]$
```

Thank You !!!!!!!!!!!!!