

An SQL Query engine, which functions on top of Hadoop's MapReduce. Data stored on HDFS is used to perform queries provided to the engine.

Queries supported:

- PUT: Transfer data from local filesystem to HDFS
- SHOW TABLES: Show all database/tables loaded
- DESCRIBE: Show schema of database/table specified
- LOAD: Load the schema of respective data
 - Specify database/table.csv and schema
 - Optionally specify separator of csv file (default : ',')
- SELECT:
 - Select multiple columns
 - Aggregations supported: Max, Min, Count

ALGORITHM/DESIGN

Control flow:

- Get query; parse query
- Select type of query
- Check match between structure of data and submitted schema (LOAD)
- Store schema (LOAD)
- Change mapper and reducer files accordingly
- Perform Hadoop MapReduce job
- Get output of MapReduce job; Display the same

Structure of schema:

```
{column_name1 : (column_index1, datatype1),  
  column_name2 : (column_index2, datatype2),  
  "separator" : 'your_seperator'}
```

Schema storage: Using Python's **Pickle** module(object serialization to store as a dictionary) **on HDFS**

Order of operations: SELECT>>PROJECT >> AGGREGATE

EXPERIMENTAL RESULTS

```
Activities Terminal ▾ Fri 14:04 ● hduser@ubuntu: ~/myhive
File Edit View Search Terminal Help
hduser@ubuntu:~/myhive$ python3 myhive.py
Welcome to MyHive.
This is an SQL query engine using Hadoop MapReduce as backend.

Type "HELP" for Help

#>help
Please enter queries in one of these formats:

PUT tablename.csv
LOAD dbname/tablename.csv AS (colname1:dtype1, colname2:dtype2) SEP separator
SHOW TABLES
DESCRIBE tablename
SELECT * FROM dbname/tablename.csv
SELECT col1 col2 coln from dbname/tablename.csv WHERE cond
SELECT col from dbname/tablename.csv WHERE cond AGGREGATE_BY aggregation

#>show tables
d/fifa.csv

#>load d/height_weight.csv as (name:str,age:int,height:int,weight:int) sep ,
Found separator: ,

#>show tables
d/fifa.csv
d/height_weight.csv
```

```
Activities Terminal ▾ Fri 14:09 ● hduser@ubuntu: ~/myhive
File Edit View Search Terminal Help
Found separator: ,

#>show tables
d/fifa.csv
d/height_weight.csv

#>describe d/height_weight.csv
{'age': (1, 'int'),
 'height': (2, 'int'),
 'name': (0, 'str'),
 'separator': ',',
 'weight': (3, 'int')}

#>select * from d/height_weight.csv
Mac | 20 | 171 | 81
Zack | 21 | 155 | 78
Jack | 19 | 169 | 72

#>select name age from d/height_weight.csv where weight > 72
Mac | 20
Zack | 21

#>select * from d/height_weight.csv where weight > 72 aggregate_by count
count 2

#>select height from d/height_weight.csv aggregate_by max
max 171

#>select height from d/height_weight.csv aggregate_by min
min 155

#>
```