Apache flume: use to stream data can be used to load real time data into hdfs. In example it streaming geolocation data of trucks

It is used to stream live data which consist of time, trucked, driver id, latitudes, longitudes, dangerous events and velocity.

Apache sqoop: Used to load data into hdfs from a structured database. In example it is data related to vehicles we own. Deliveries done, and other data.

Apache Hive: used to query the HDFS. Queries are somewhat similar to sql.

Apache oozie: used to create jobs in hdfs.

Apache pig: used to process data using pig scripts.

Questions

Which trucks are idling and wasting fuel?

And which drivers are driving unsafely?

Configuring odbc driver .

[https://2xbbhjxc6wk3v21p62t8n4d4-wpengine.netdna-ssl.com/wp-*content/uploads/2016/08/Hortonworks-Hive-ODBC-Driver-User-Guide.pdf*](https://2xbbhjxc6wk3v21p62t8n4d4-wpengine.netdna-ssl.com/wp-content/uploads/2016/08/Hortonworks-Hive-ODBC-Driver-User-Guide.pdf)

## HDFS BACKDROP

A single physical machine gets saturated with its storage capacity as the data grows. This growth drives the need to partition your data across separate machines. This type of File system that manages storage of data across a network of machines is called Distributed File Systems. [HDFS](https://hortonworks.com/blog/thinking-about-the-hdfs-vs-other-storage-technologies/) is a core component of Apache Hadoop and is designed to store large files with streaming data access patterns, running on clusters of commodity hardware. With Hortonworks Data Platform HDP 2.2, HDFS is now expanded to support [heterogeneous storage](https://hortonworks.com/blog/heterogeneous-storage-policies-hdp-2-2/)  media within the HDFS cluster.

## APACHE HIVE BASICS

Apache Hive provides SQL interface to query data stored in various databases and files systems that integrate with Hadoop. Hive enables analysts familiar with SQL to run queries on large volumes of data. Hive has three main functions: data summarization, query and analysis. Hive provides tools that enable easy data extraction, transformation and loading (ETL)

Apache Hive presents a relational view of data in HDFS. Hive can represent data in a tabular format managed by Hive or just stored in HDFS irrespective in the file format their data is stored in. Hive can query data from RCFile format, text files, ORC, JSON, parquet, sequence files and many of other formats in a tabular view. Through the use of SQL you can view your data as a table and create queries like you would in an RDBMS.

There are six tabs to interact with SQL:

* **QUERY**: This is the interface shown above and the primary interface to write, edit and execute new SQL statements
* **JOBS**: This allows you to look at past queries or currently running queries.  It also allows you to see all SQL queries you have authority to view.  For example, if you are an operator and an analyst needs help with a query, then the Hadoop operator can use the History feature to see the query that was sent from the reporting tool.
* **TABLES**: Provides one central place to view, create, delete, and manage tables of whichever databases that you select.
* **SAVED QUERIES**: shows all the queries that have been saved by the current user. Click the gear icon to the right of the query list to view the history of a query or to delete it.
* **UDFs**: User-defined functions (UDFs) can be added to queries by pointing to a JAR file on HDFS and indicating the Java classpath, which contains the UDF definition. After the UDF is added here, an Insert UDF button appears in the Query Editor that enables you to add the UDF to your query.
* **SETTINGS**: Allows you to append settings to queries that you execute in Hive View.

Queries

select \* from trucks limit 100;

* show tables; – List the tables created in the database by looking up the list of tables from the metadata stored in HCatalogdescribe
* describe {table\_name}; – Provides a list of columns for a particular table

describe geolocation;

* show create table {table\_name}; – Provides the DDL to recreate a table

show create table geolocation;

* describe formatted {table\_name}; – Explore additional metadata about the table. For example you can verify geolocation is an ORC Table, execute the following query:

describe formatted geolocation;

CREATE TABLE truck\_mileage STORED AS ORC AS SELECT truckid, driverid, rdate, miles, gas, miles / gas mpg FROM trucks LATERAL VIEW

stack(54,'jun13',jun13\_miles,jun13\_gas,'may13',may13\_miles,may13\_gas,'apr13',apr13\_miles,apr13\_gas,'mar13',mar13\_miles,mar13\_gas,'feb13',feb13\_miles,feb13\_gas,

'jan13',jan13\_miles,jan13\_gas,'dec12',dec12\_miles,dec12\_gas,'nov12',nov12\_miles,nov12\_gas,'oct12',oct12\_miles,oct12\_gas,'sep12',sep12\_miles,sep12\_gas,

'aug12',aug12\_miles,aug12\_gas,'jul12',jul12\_miles,jul12\_gas,'jun12',jun12\_miles,jun12\_gas,'may12',may12\_miles,may12\_gas,'apr12',apr12\_miles,apr12\_gas,

'mar12',mar12\_miles,mar12\_gas,'feb12',feb12\_miles,feb12\_gas,'jan12',jan12\_miles,jan12\_gas,'dec11',dec11\_miles,dec11\_gas,'nov11',nov11\_miles,nov11\_gas,

'oct11',oct11\_miles,oct11\_gas,'sep11',sep11\_miles,sep11\_gas,'aug11',aug11\_miles,aug11\_gas,'jul11',jul11\_miles,jul11\_gas,'jun11',jun11\_miles,jun11\_gas,

'may11',may11\_miles,may11\_gas,'apr11',apr11\_miles,apr11\_gas,'mar11',mar11\_miles,mar11\_gas,'feb11',feb11\_miles,feb11\_gas,'jan11',jan11\_miles,jan11\_gas,

'dec10',dec10\_miles,dec10\_gas,'nov10',nov10\_miles,nov10\_gas,'oct10',oct10\_miles,oct10\_gas,'sep10',sep10\_miles,sep10\_gas,'aug10',aug10\_miles,aug10\_gas,

'jul10',jul10\_miles,jul10\_gas,'jun10',jun10\_miles,jun10\_gas,'may10',may10\_miles,may10\_gas,'apr10',apr10\_miles,apr10\_gas,'mar10',mar10\_miles,mar10\_gas,

'feb10',feb10\_miles,feb10\_gas,'jan10',jan10\_miles,jan10\_gas,'dec09',dec09\_miles,dec09\_gas,'nov09',nov09\_miles,nov09\_gas,'oct09',oct09\_miles,oct09\_gas,

'sep09',sep09\_miles,sep09\_gas,'aug09',aug09\_miles,aug09\_gas,'jul09',jul09\_miles,jul09\_gas,'jun09',jun09\_miles,jun09\_gas,'may09',may09\_miles,may09\_gas,

'apr09',apr09\_miles,apr09\_gas,'mar09',mar09\_miles,mar09\_gas,'feb09',feb09\_miles,feb09\_gas,'jan09',jan09\_miles,jan09\_gas ) dummyalias AS rdate, miles, gas;