Projet BDD + Big Data

Sujet :

La découverte de l’Apache Hive

Projet final du module.

Sommaire :

1. Introduction
2. What is
   * Big Data
   * Hadoop
   * Hive Architecture
3. Limits
4. Compare with technologies studied in class: MongoDB/Cassandra
5. Architecture
6. Different use cases: Netflix , Facebook
7. Conclusion

Introduction :

For the Big Data project we need to do a report. It is intended to discover a technology seen outside of the Big data and Database module. This report will contain explication and a part more "concrete" of a study of the utilization of Hive.

Apache Hive is a data warehouse software project built on top of Apache Hadoop for providing data query and analysis. Hive gives a SQL-like interface to query data stored in various databases and file systems that integrate with Hadoop. Traditional SQL queries must be implemented in the MapReduce Java API to execute SQL applications and queries over distributed data. Hive provides the basic SQL abstraction to integrate SQL-like queries (HiveQL) into the underlying Java without implementing queries in the low-level Java API. Since most data warehousing applications work with SQL-based querying languages, Hive aids portability of SQL-based applications to Hadoop. While initially developed by Facebook, Apache Hive is used and developed by other companies such as Netflix and the Financial Industry Regulatory Authority (FINRA). Amazon maintains a software fork of Apache Hive included in Amazon Elastic MapReduce on Amazon Web Services.

## What is Big Data?

Big data is more related to the three V: Variety., Velocity and Volume

Velocity is a reference to speed which new information (data) is created. On the other hand Variety, refers to the type of the data that been generated. However, we think about that the limit from which we "do BigData" is the one from which conventional approaches are no longer usable at reasonable price. In a practical way, it is when we exceed 10 TB of data, we are working in Big Data. This limit also move according to the complexity of the data.

The challenge of these day, has become to use more data, faster, whether already known, or from sources that are new to combine with the existing ones to give a richer context to them. It’s in this context that come Hadoop, showing an innovative solution to store and analyze big volumes of data in a scalable way, while keeping his budget controlled.

## Hadoop, what is it?

Hadoop is a free and open source framework for the development of distributed scalable applications. It allows the distributed processing of large volumes of data on a cluster of several hundred (or thousands) of standard machines called commodity hardware. It is based on:

HDFS (Hadoop Distributed File System) for the data storage part, MapReduce, which is a model of program (with an associated implementation) created by Google for parallel processing and distributed data on a cluster.

In general, the more complex the data model, the more harder it is to write a MapReduce job that handles it. The simple

example we can take of the count of Word we can find on the official documentation of Hadoop, the Java7 implementation does about a hundred lines with:

15 lines can map it,

12 lines can reducer it,

35 lines setup as well as utility methods for the parse of input data, 30 lines for the hand.

Yes, all for a Word Count with MapReduce!

To make the analysis easier of data stored in HDFS without going through MapReduce complexity,there is frameworks like Pig, Hive that appeared. They are offering high-level languages for launch ad-hoc queries on HDFS.

Hive advantage is to define a variety of data formats with a structure thus make more easy the possibility of querying them. It is therefore good adapted to a context of data analysis. Hive also offers distributed storage and access to files stored in HDFS (or other systems such as Apache HBase).

In terms of language, Hive offers access to HiveQL, a declarative language, like SQL, while Pig proposes Pig Latin, a data flow language oriented for "exploration" of big volumes of data.

## What is HIVE?

Apache Hive is a datawarehouse for Hadoop. It was created by Facebook to eventually become an open source Apache project. It is not a relational database or a classic data warehouse.

Hive not a database or a datawarehouse ?

This is a system maintaining metadata and describe the data stored in HDFS. It use for a relational database called metastore (Also Derby) to ensure the persistence of metadata. The composition of a table in Hive is essentially :

A schema stored in the metastore, Data stored in HDFS.

With the metastore data, Hive can let you manipulate the data as if it were persisted in tables (in the sense of a classic database management system) and to query them with its HiveQL language.

Hive change HiveQL queries to MapReduce or Tez jobs (from Hive version 0.13, a HiveQL query can be translated into an executable job on Apache Tez, that is a Hadoop framework that execute and

can replace MapReduce).

Thus, profiles that know SQL (analysts, data scientists, etc.) not know a developer background can develop their HiveQL queries to exploit the data save in HDFS without worrying about the programming jobs.

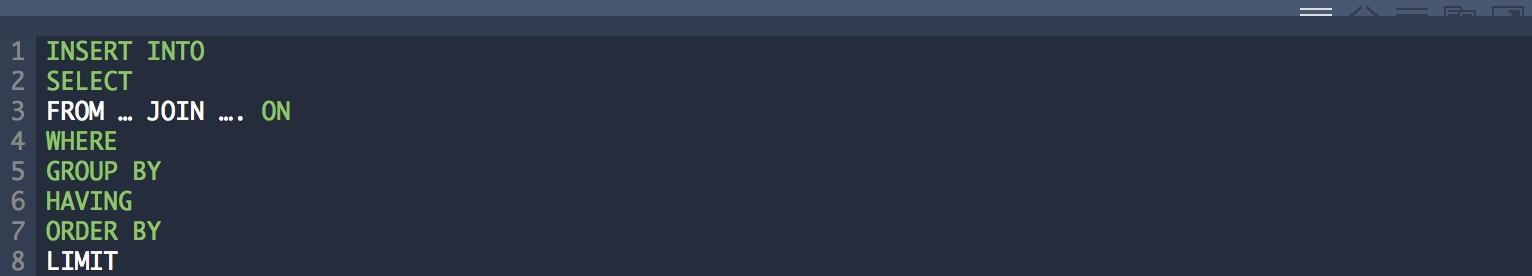
Hive supports standard SQL clauses from a syntax point of view (Fig 1) as well as structure definition commands : 

Fig 1 : Standard SQL clauses supported by HiveQL

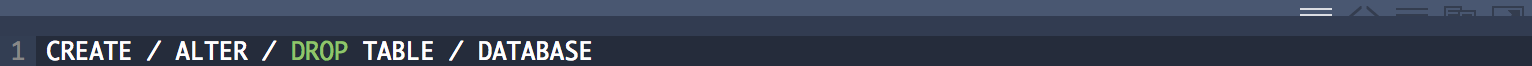


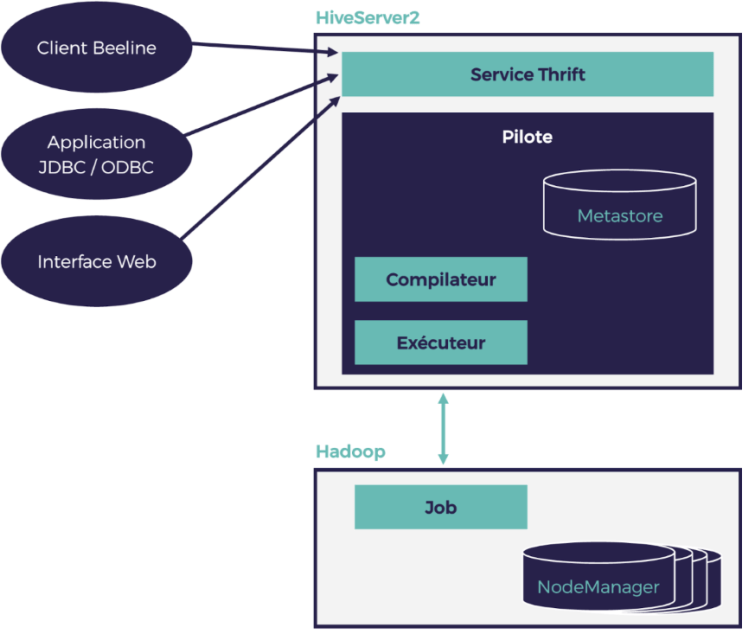
Figure 2: Example of structure definition commands with HiveQL

## HIVE Architecture :

The operation of Hive and its architecture to make it more easier to understand , we will analyse the execution of a Hive query. Indeed, the Hive / Hadoop communication is carried out according to the these three steps:

Sending the HiveQL request: using a Hive client (the shell client, a JDBC / ODBC client or a web interface), the request is sent to the Hive server,

The request is planned : the driver received the request. Then is it compiled, optimized and then planned as a job,

Job execution:the Hadoop cluster run the job 

Part (1): part of the client

We can submit requests to the server Hive in multiple ways. Using:

The Hive CLI (Hive Command Line Interface) client that let you enter commands directly from the shell of hive or a set of Hive commands written in a text file can be execute . This client is not compatible with the new version of Hive (HiveServer2) and has been replaced by Beeline the new command line client of Hive. It communicates with HiveServer2 via thrift,

The JDBC / ODBC client, The web client.

Part (2): part of the server

The successor of HiveServer is named HiveServer2 (now deprecated from version 1.0.0).The Hive runtime engine container and is commonly called a driver (or driver). It’s contain a executor, compiler and metastore.

HiveServer2 provides two new features: concurrent request management and client authentication management. For every client connection, HiveServer2 creates a new execution context (login + session). The server can associate the Hive execution context with the thread serving the client request allows by the new HiveServer2 RPC interface. A thrift service to communicate with clients and execute their queries is implements by this interface.

Party (3): Hadoop

It means the job execution on the Hadoop cluster.

* Partitions, queries in Hive and tables
* Data loading in Hive and tables
* In HDFS, a table in Hive can associate a structure with unstructured data.
* When we create a table in Hive, it’s exactly like creating a table in an RDBMS and we can do it with the command : CREATE TABLE.

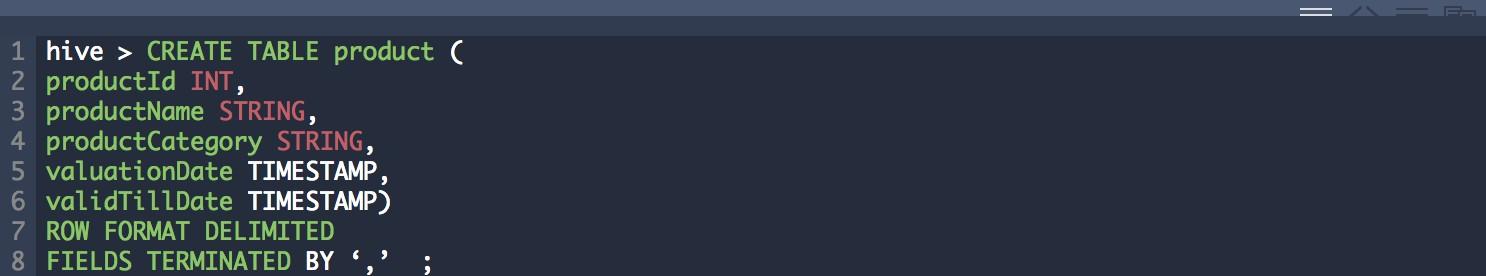
We can find two types of tables in Hive:

1. Managed table.
2. External table.

Hive Managed table and External table

In Hive, a Managed table resembles a table in the RDBMS sense. The difference between a managed table and an external table can be seen when the table is deleted : we notice a change in the management of the data.

The deletion of a Managed table results in the deletion of the metadata along with the data in HDFS (in the example that we chose underneath, the data is stored in HDFS by default under / apps / hive / warehouse / product). On the other hand, when we delete an external table we find as a result the deletion of the metadata. The good thing is that the External table is a way to protect data against accidental drop commands.

Figure 4: HiveQL command for creating a managed table product with 5 columns (productId, productName, productCategory, valuationDate, validTillDate)

Hive is also useful when it comes to specifying where to store data in HDFS. It can be achieved by adding the LOCATION clause during table creation ( Fig 5).

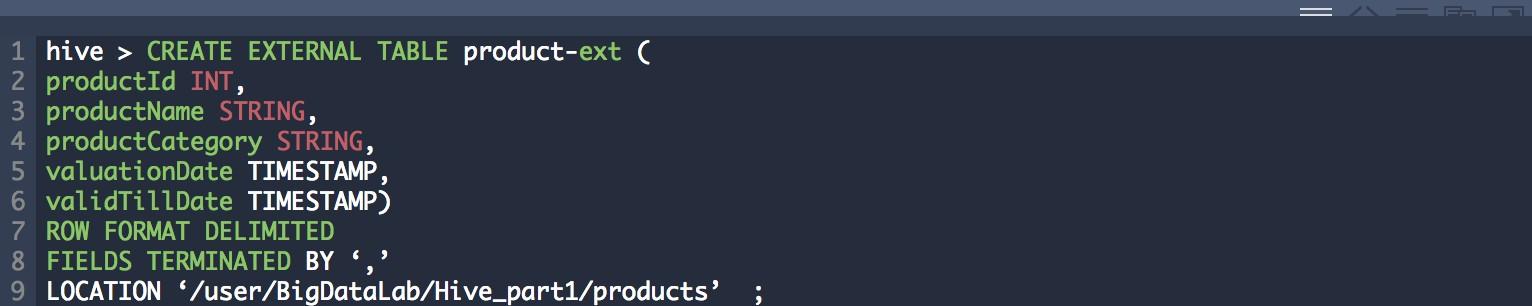


Figure 5: HiveQL command for the creation of a product-ext external table with specification of the data storage location in HDFS

## Loading data into a Hive table :

Hive is known for its ability to associate data with metadata, and the source of this data may be HDFS or the file system. To be able to do so, we use the LOAD DATA command. The question now is how is how can we do this loading ?

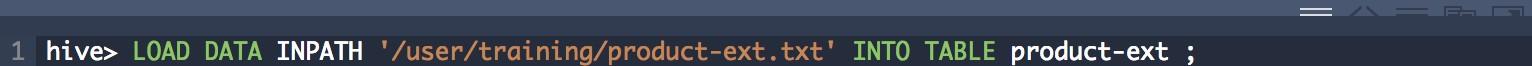
What is necessary to know, is that the data is not loaded but moved.

For a Managed table: the data is moved to the default root directory which is a apps /hive / warehouse (Fig 6). The meaning of the LOCAL keyword is that the input file is in the local file system. Once the LOCAL is omitted, it becomes an input file in HDFS.

The meaning of the OVERWRITE keyword is that the data in the product table will be removed. Once OVERWRITE is omitted, the data will be added to the data that already exists.

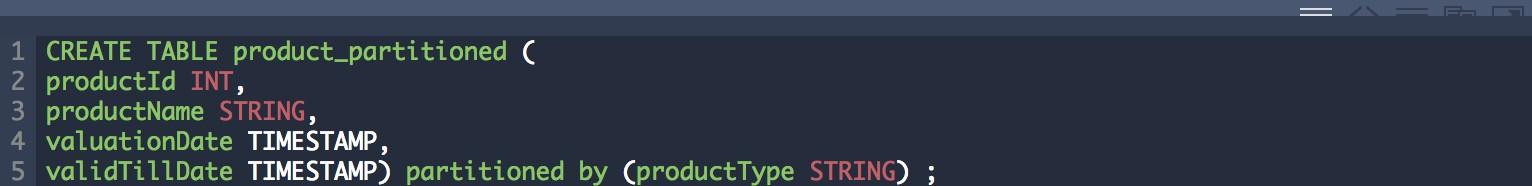
Figure 6: HiveQL command to load data from an input file into the file system in the product table with overwriting existing data

For an External table: the data is moved to the directory specified in the LOCATION clause (Fig 7). In our example, the product-ext.txt input file will be moved to / user /BigDataLab / Hive\_part1 / products since it is the location that we defined when we created the table (Fig 5).



## Partition Management :

Partitioning a table in Hive means separating files based on their columns that defines the partition key. One of the partitioning advantages is that it improves the performance of HiveQL queries since the files in HDFS have been separated based on the value of their columns. This separation is able to reduce the number of mappers and the number of shuffle / job operations that resulted from the job. The definition of a partition is almost like its definition in SQL:



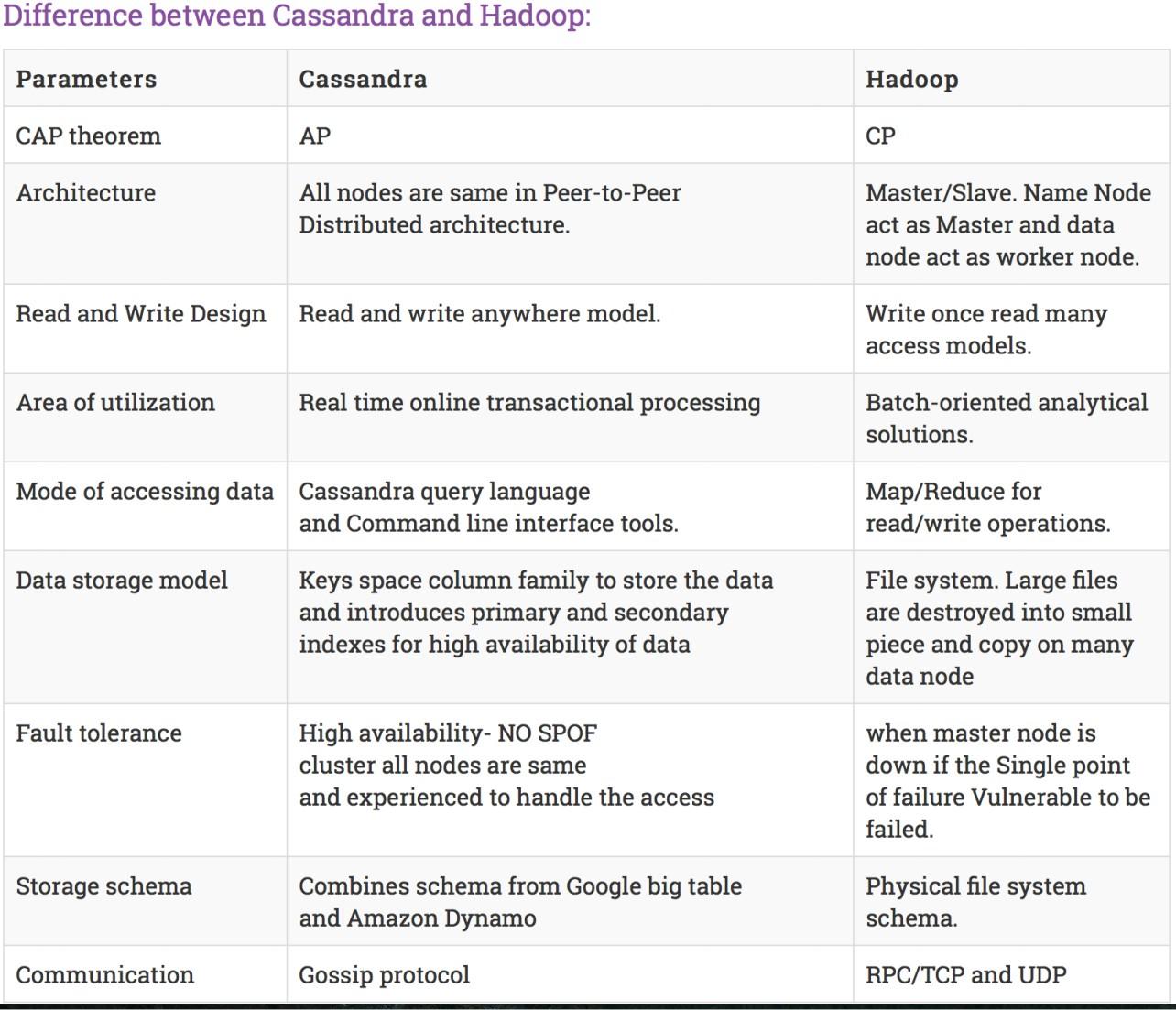
# Limits :

Hive can still not support "updates" of the statements. Also, when we use Hadoop, some queries can take a while before giving us back a result.

## Apache Hive vs Cassandra:

Comparing Hive and Cassandra won’t help us with anything in particular.

Hive is integrated on Hadoop and it is between Cassandra and Hadoop that comparisons can be really interesting.



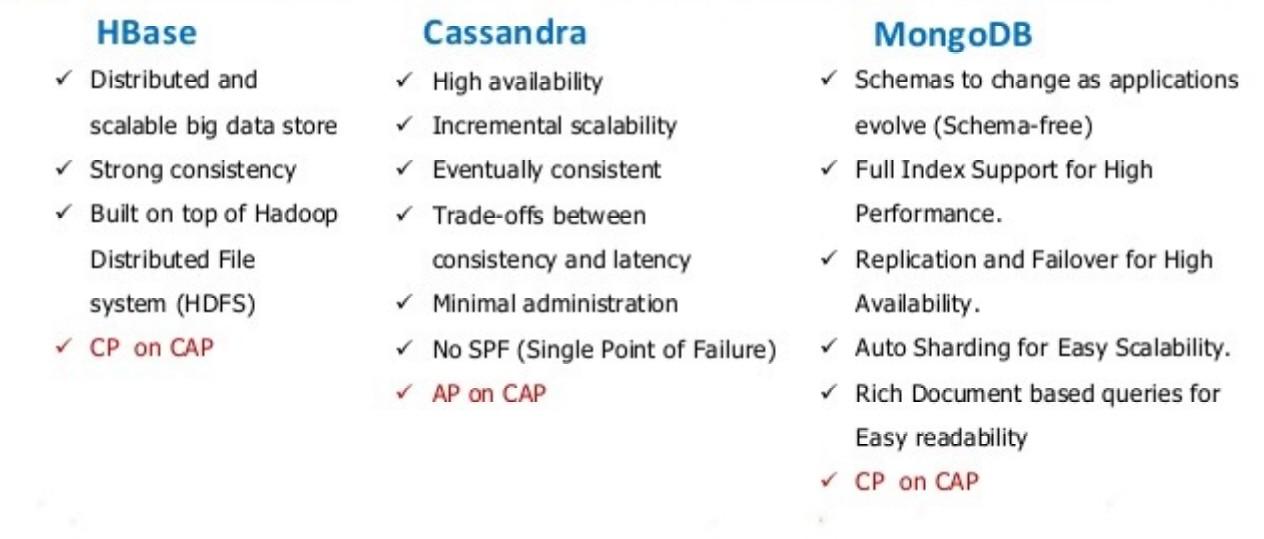
## Apache Hive vs MongoDB:

Hive and HBase are very similar, they’re both integrated on Hadoop, a

synthetic comparison between HBase, Cassandra and MongoDB allows us to understand better the nuances of the technologies seen.

Also, comparing Hbase and Hive is like comparing a network social and a search engine. They both work with metadata and are both useful for data unstructured, and sounds usually used together.

However, those are the only similarities and it becomes a question of being suspicious of the schema below.



## Different use cases:

As we have mentioned before, Hive is used in some of the most popular apps in the world : **Facebook and Netflix**.

## Facebook :

Facebook engineers used Hadoop in order to manage the multiplication of data. The problem is that the rigidity of Hadoop 1.0 only returned formats in map / reduce, so the latter decided to develop a simpler system that allowed to analyse the data without having to convert the code to map/reduce because it took a lot of time and they decided to run it on Hadoop.

Hive uses an interface that works like SQL. Its use is to organize data in partitions and tables then store them in directories of HDFS. Hive also combines the possibilities offered by tables and queries with the scalability capabilities of Hadoop.

Thanks to Hive, Facebook can maintain a simplicity of Hadoop despite the huge amount of data that keeps on expanding.

## Netflix:

Netflix uses approximately Hive for the same reasons as Facebook. For more than 10 years,while Netflix wasn’t famous yet, the company was already using Hive for its referral service. Netflix must study each user’s taste, based on complex and numerous algorithms. Everytime a user views a content it’s processed and then it posts recommendations for the user. It is therefore extremely important for the platform to have a big amount of data and to process them quickly and easily. Hive is very useful for its qualities of "query" processing "," BI "and" scalable log analysis ". Also it deals with hundreds of content for tens of millions of users constantly while updating regularly the catalog. At last, Netflix employees are also in the list of Apache Hive contributors.

# Conclusion

We have introduced through this article Apache Hive, its architecture and its query language HiveQL that we have noticed to be identical to SQL.

With HiveQL, analyzing big amounts of data becomes as easy as querying a relational database with SQL.

The strong point of Hive compared to other Big Data data analytics frameworks include : the active community that uses it, its maturity, and its compatibility with every new version of Hadoop.

A lot of advanced features such as joins and sorting is offers by Hive (Hive are defined in many join types such as Shuffle Join, Broadcast Join, and SMB Join). User Defined Functions (UDFs) is a part flexible by Hive, which are user-defined functions that extend the language and can be re-used as if it were an external library.