Apache Hive

Opracowanie programu

Streszczenie

Opracowanie zawiera przedstawienie programu, jego instalacje oraz opis podstawowych komend

Adam Janowski

Spis treści

Wstęp do Hive**:**

Apache Hive to rozproszony system hurtowni danych, umożliwiający przetwarzanie ustrukturyzowanych i częściowo ustrukturyzowanych danych w Hadoop na masową skalę. Hurtownia danych stanowi centralny magazyn informacji, które można łatwo analizować w celu podejmowania decyzji opartych o dane. Hive umożliwia użytkownikom odczytywanie, zapisywanie i zarządzanie petabajtami danych przy użyciu języka HQL.

Hive jest oparty na Apache Hadoop, który jest platformą open source używaną do wydajnego przechowywania i przetwarzania dużych zasobów danych. To, co czyni Hive wyjątkowym, to możliwość wykonywania zapytań z wykorzystaniem Apache Tez lub MapReduce z interfejsem podobnym do SQL.

Hive został stworzony, aby umożliwić osobom nie będącym programistami znającymi język SQL pracę z petabajtami danych przy użyciu podobnego do SQL interfejsu o nazwie HiveQL. Hive wykorzystuje przetwarzanie wsadowe, dzięki czemu działa szybko w bardzo dużej rozproszonej bazie danych. Hive przekształca zapytania HiveQL w zadania MapReduce lub Tez, które działają w ramach rozproszonego planowania zadań Apache Hadoop. Hive wysyła zapytania do danych przechowywanych w rozproszonym takim jak rozproszony system plików Hadoop (HDFS). Dzięki temu można użyć HiveQL do wykonywania zapytań dotyczących danych bez znajomości języka Java lub MapReduce.

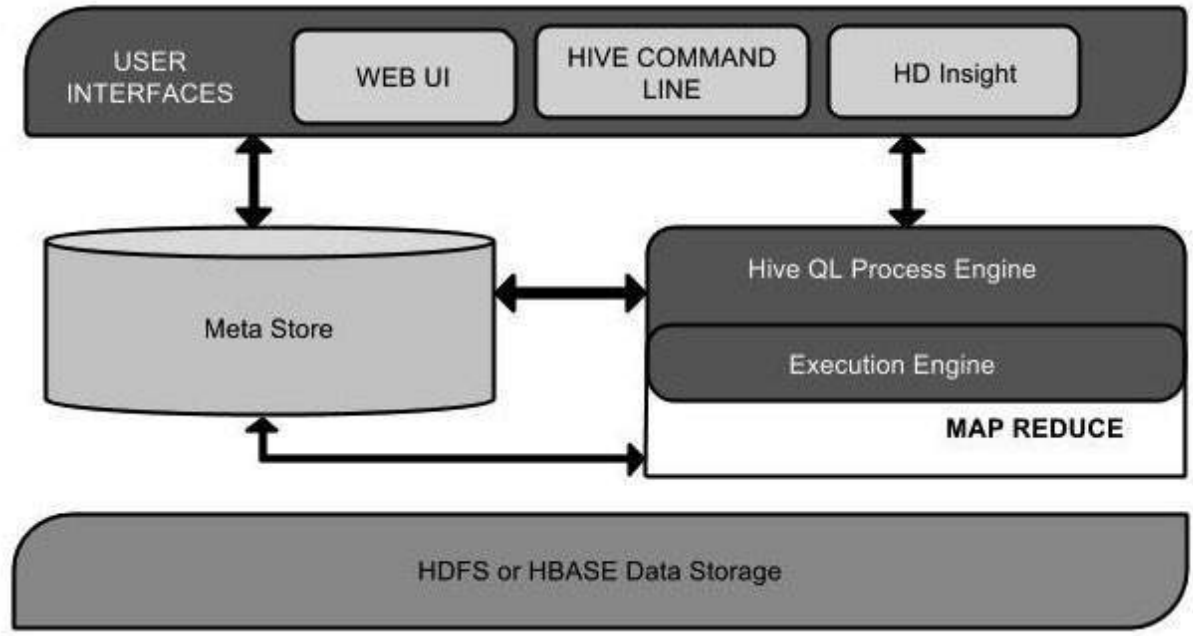
Korzyści z używania Hive:

Szybkość:
Hive jest przeznaczony do szybkiego przetwarzania petabajtów danych przy użyciu przetwarzania wsadowego.

Znajomość:
Hive zapewnia znajomy interfejs podobny do języka SQL, dostępny dla osób niebędącymi programistami.

Skalowanie:
Hive jest łatwy do dystrybucji i skalowania w zależności od potrzeb.

Struktura Hive:



User Interface (Interfejs użytkownika) - Hive daje możliwość interakcji między użytkownikiem, a systemem plików HDFS. Interfejsy użytkownika obsługiwane przez Hive to Hive Web UI, Hive Command Line i Hive HD Insight (na serwerze Windows).

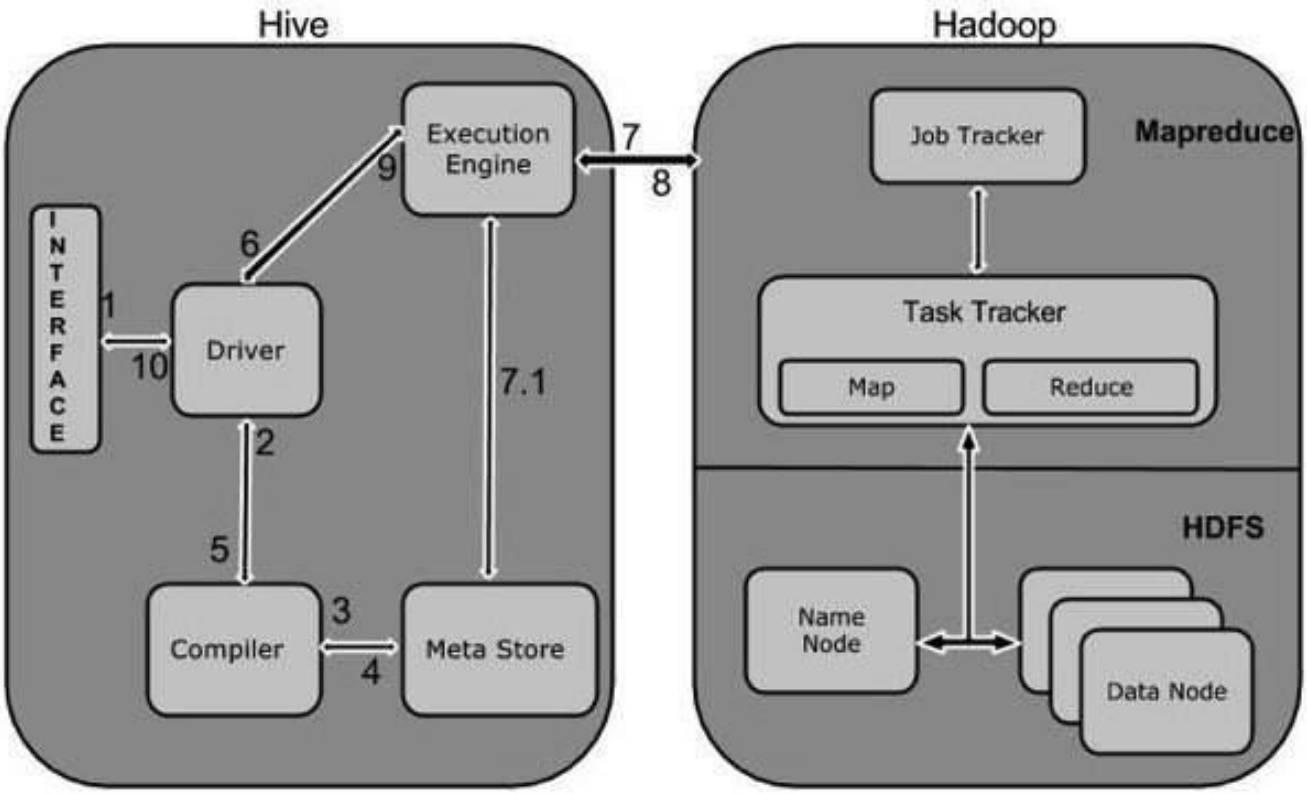
Meta Store - przechowuje metadane dla każdej tabeli, takie jak ich schemat i lokalizacja. Zawiera również metadane partycji, które pomagają sterownikowi śledzić postęp różnych zestawów danych rozproszonych w klastrze.

HiveQL Process Engine (Silnik procesów HiveQL) - HiveQL Process Engine to składnik, który używa HiveQL do wysyłania zapytań do Metastore. Daje użytkownikowi możliwość przeszukiwania dużych zbiorów danych bez konieczności pisania wielu kodów MapReduce. Zapytania są faktycznie konwertowane na zadania MapReduce w celu obliczenia wyniku końcowego.

Execution Engine (Silnik wykonawczy) - Execution Engine jest pomostem między HiveQL Process Engine i MapReduce. Execution Engine przetwarza zapytanie napisane w HiveQL i używa struktury MapReduce do wygenerowania wyniku.

HDFS or HBASE Data Storage (Przechowywanie danych HDFS lub HBASE) - Systemy plików, które umożliwiają przechowywanie dużych danych w wielu węzłach w klastrze Hadoop.

Działanie Hive:



1. **Wykonanie zapytania:**
Interfejs Hive (wiersz poleceń lub Web UI) wysyła zapytanie do sterownika (dowolnego sterownika bazy danych np. JDBC, ODBC) w celu jego wykonania.

2. **Uzyskanie planu:**
Sterownik korzysta z pomocy kompilatora zapytań, który analizuje zapytanie w celu sprawdzenia składni i planu zapytania lub wymagań zapytania.

3. **Pobieranie metadanych:**
Kompilator wysyła żądanie metadanych do Metastore.

4. **Wysyłanie metadanych:**
Metastore wysyła metadane jako odpowiedź do kompilatora.

5. **Wysyłanie planu:**
Kompilator sprawdza wymagania i wysyła plan do sterownika. Następuje zakończenie analizowania i kompilowania zapytania.

6. **Wykonanie planu:**
Sterownik przesyła plan wykonania do silnika wykonawczego.

7. **Wykonanie zadania:**
Wewnętrznie proces wykonania zadania jest zadaniem MapReduce. Silnik wykonawczy wysyła zadanie do JobTracker'a, który znajduje się w Name Node i przypisuje to zadanie do TaskTracker'a, który znajduje się w Data Node. Tutaj zapytanie wykonuje MapReduce.

7.1 Operacje na metadanych:
W międzyczasie silnik wykonawczy może wykonywać operacje na metadanych za pomocą Metastore.

8. **Pobieranie wyniku:**
Silnik wykonawczy otrzymuje wyniki z Data Node.

9. **Wysyłanie wyniku:**
Silnik wykonawczy wysyła wartości wynikowe do sterownika.

10. **Wysyłanie wyniku:**
Sterownik wysyła wyniki do interfejsu Hive.

Instalacja Hive:

Jeżeli mamy już poprawnie zainstalowane i skonfigurowane programy Java oraz Hadoop. Możemy przystąpić do instalacji Hive.

Należy zacząć od utworzenia odpowiedniego katalogu, gdzie plik powinien zostać pobrany.

```
$ sudo mkdir usr/local/hive
$ cd /usr/local/hive
```

Następnie możemy przejść do pobrania pliku. Można tego dokonać poprzez poniższą komendę:

```
$ sudo wget
https://apache.mirrors.tworzy.net/hive/stable-2/ apache-
hive-2.3.8-bin.tar.gz
```

Po ukończeniu pobierania następnym krokiem jest rozpakowanie:

```
$ sudo tar -xvf apache-hive-2.3.8-bin.tar.gz
```

Aby uzyskać dostęp do Hive z dowolnej ścieżki, jej zmienna środowiskowa musi być ustawiona w pliku .bashrc. Dostać się do tego pliku można za pomocą poniższej komendy:

```
$ sudo nano ~/.bashrc
```

Po otworzeniu pliku .bashrc należy dodać poniższy fragment:

```
# Set HIVE_HOME
export HIVE_HOME=/usr/local/hive/apache-hive-2.3.8-bin
PATH=$PATH:$HIVE_HOME/bin
```

Kolejno trzeba odświeżyć plik .bashrc podaną komendą:

```
$ source ~/.bashrc
```

Konfiguracja Hive.

Aby skonfigurować Hive z Hadoop, należy edytować plik hive-env.sh, który jest umieszczony w katalogu $HIVE_HOME/conf. Należy kolejno przejść do folderu konfiguracyjnego oraz skopiować szablonowy plik. Można to zrobić następującymi komendami:

```
$ cd $HIVE_HOME/conf
$ sudo cp hive-env.sh.template hive-env.sh
```

Następnie należy wejść w plik hive-env.sh używając komendy:

```
$ sudo nano hive-env.sh
```

Oraz dodać poniższy fragment:

```
export HADOOP_HOME=/usr/local/hadoop
export HADOOP_HEAPSIZE=512
export HADOOP_CONF_DIR=/usr/local/hadoop/conf
```

Kolejno należy skonfigurować plik hive-site.xml używając komend:

```
$ cd $HIVE_HOME/conf
$ sudo cp hive-default.xml.template hive-site.xml
$ sudo nano hive-site.xml
```

Po otworzeniu pliku należy upewnić się, że plik hive-site.xml zawiera poniższy fragment:

```
<property>
     <name>hive.metastore.warehouse.dir</name>
     <value>/user/hive/warehouse</value>
     <description>location of default database for the
warehouse</description>
</property>
```

Następnym plikiem do skonfigurowania jest plik hive-conf.sh znajdujący się w folderze bin. Można przejść do niego poprzez komendę:

```
$ sudo nano $HIVE_HOME/bin/hive-config.sh
```

Będąc w pliku należy dodać do niego następującą ścieżkę:

```
export HADOOP_HOME=/usr/local/hadoop
```

Gdy pliki konfiguracyjne są już poprawnie dostosowane, kolejnym krokiem jest utworzenie katalogu Hive w HDFS. Katalog „warehouse” to lokalizacja do przechowywania tabel lub danych związanych z Hive. Najpierw, jednakże trzeba zainicjować Hadoop. Aby to zrobić należy przejść do użytkownika, gdzie znajduje się hadoop, odpalić go oraz wyłączyć save mode.

```
$su hadoop
password:

$ start-all.sh
$ hadoop dfsadmin -safemode leave

$ hdfs dfs -mkdir -p /user/hive/warehouse
```

Następnie należy dodać uprawnienia do zapisu i wykonywania członkom grupy.

```
$ hdfs dfs -chmod g+w /user/hive/warehouse
```

Dodatkowo powinno się utworzyć katalog tmp. W tym katalogu będą przechowywane pośredniczące dane, które Hive wysyła do HDFS:

```
$ hdfs dfs -mkdir /tmp
```

Tak samo należy jeszcze dodać uprawnienia do zapisu i wykonywania członkom grupy.

```
$ hdfs dfs -chmod g+w /tmp
```

Na końcu należy poinformować Hive której bazy danych ma używać do definiowania schematu. Poniższe polecenie informuje Hive, aby używał bazy danych Derby jako bazy danych metastore.

```
$ $HIVE_HOME/bin/schematool –initschema –dbtype derby
```

Jeśli planuje się użyć innego rozwiązania bazodanowego, takiego jak MySQL lub PostgreSQL, możesz określić typ bazy danych w pliku hive-site.xml.

Jeśli baza danych Derby nie zostanie pomyślnie zainicjowana, może pojawić się błąd o następującej treści:

```
“Exception in thread “main” java.lang.NoSuchMethodError:
com.google.common.base.Preconditions.checkArgument(ZLjava
/lang/String;Ljava/lang/Object;)V”
```

Ten błąd wskazuje, że najprawdopodobniej występuje problem niezgodności między wersjami Hadoop i Hive guava. Aby rozwiązać ten problem należy na wstępie zlokalizować plik guava jar w katalogu biblioteki Hive:

```
$ ls $HIVE_HOME/lib
```

Następnie należy zlokalizować również plik guava jar w katalogu biblioteki Hadoop:

```
$ ls $HADOOP_HOME/share/hadoop/hdfs/lib
```

Dwie wersje nie są ze sobą zgodne i powodują błąd. Należy usunąć istniejący plik guava z katalogu biblioteki Hive:

```
$ sudo rm $HIVE_HOME/lib/guava-14.0.1.jar
```

Kolejno trzeba skopiować plik guava z katalogu lib Hadoop do katalogu lib Hive:

```
$ sudo cp $HADOOP_HOME/share/hadoop/hdfs/lib/guava-27.0-
jre.jar $HIVE_HOME/lib/
```

Tak skonfigurowany Hive jest gotowy do pracy. Aby uruchomić Hive Shell do wykonywania zadań wystarczy wpisać polecenie:

```
$ hive
```

Typy danych w Hive

W Hive znajduje się 5 typów danych:

1. **Typy liczbowe**

TINYINT
INT
BIGINT
FLOAT
DOUBLE
DECIMAL

2. **Typy daty / godziny**

TIMESTAMP
DATE

3. **Typy danych tekstowych**

STRING
VARCHAR
CHAR

4. **Różne typy**

BOOLEAN
BINARY

5. **Typy złożone**

tablice: ARRAY <typ_danych>
mapy: MAP <typ_prymitywny, typ_danych>
union: UNIONTYPE <typ_danych, typ_danych, ...>

Komendy w Hive

Polecenia DDL(Data Definition Language)

Polecenia DDL to instrukcje odpowiedzialne za definiowanie i zmianę struktury bazy danych lub tabeli w programie Hive używając HQL. Gdy Hive Shell został już pomyślnie uruchomiony, zapytania Hive mogą być wykonywane.

Polecenia z bazą danych

-Create

Pierwszym podstawowym zapytaniem jakie należy zrobić to stworzyć nową bazę danych. Bazę danych robi się w następujący sposób:

```
hive> create database baza;
OK
Time taken: 0.26 seconds
```

Istnieje również drugi sposób tworzenia bazy danych, w którym najpierw sprawdza się, czy baza danych już istnieje. Jeśli nie, Hive tworzy taką bazę.

```
hive> create database if not exists nowabaza;
OK
Time taken: 0.041 seconds
```

Jeśli istnieje już baza danych o nazwie baza, pierwsze polecenie zgłosi wyjątek, podczas gdy drugie polecenie utworzenia bazy danych nie zgłosi żadnego wyjątku.

Możliwa jest dodatkowa opcja dodania komentarza do bazy danych, aby uzyskać szybki przegląd tego co tam się znajduje.

```
hive> create database test comment "Miejsce na komentarz";
OK
Time taken: 0.064 seconds
```

-Show

Kolejnym podstawowym poleceniem w Hive jest pokazanie wszystkich istniejących baz używając komendy show databases.

```
hive> show databases;
OK
baza
nowabaza
test
Time taken: 0.053 seconds, Fetched: 3 row(s)
```

Polecenia show databases można używać wraz z wyrażeniami w celu wyszukania baz danych o określonej nazwie. Jest to pomocne, gdy istnieje wiele baz danych i trzeba wyświetlić listę baz danych o określonym wzorcu w nazwie bazy danych.

```
hive> show databases like 'nowa*';
OK
nowabaza
Time taken: 0.022 seconds, Fetched: 1 row(s)
```

-Drop

Do usunięcia już stworzonej bazy danych w programie Hive służy polecenie drop.

```
hive> drop database if exists nowabaza CASCADE;
OK
Time taken: 0.082 seconds
```

Używając polecenia drop mamy możliwość ustawienia trybu restrict lub cascade. Domyślnym trybem w Hive jest ustawienie restrict, oznacza to że baza danych może być usunięta tylko gdy jest pusta. Natomiast tryb cascade oznacza usunięcie bazy danych wraz ze wszystkimi tabelami znajdującymi się wewnątrz.

-Describe

To polecenie służy do sprawdzania wszelkich powiązanych metadanych dla konkretnej bazy danych.

```
hive> describe database test;
OK
test            Miejsce na komentarz
hdfs://localhost:9000/user/hive/warehouse/test.db     hadoop
USER
Time taken: 0.043 seconds, Fetched: 1 row(s)
```

Powyższe polecenie describe pokazuje komentarz, który podaliśmy podczas tworzenia bazy danych, który mówi - „Miejsce na komentarz”. Polecenie describe podaje również ścieżkę do bazy danych, w której jest przechowywana na HDFS, w tym przypadku ścieżka to „hdfs://localhost:9000/user/hive/warehouse/test.db”

-Alter

Chcąc zmienić metadane którejkolwiek z baz danych, można użyć polecenia alter.

```
hive> Alter database baza SET DBPROPERTIES ( ‘ owner ’ = ‘
hadoop ’);
OK
Time taken: 0.021 seconds
hive> describe database baza;
OK
baza
hdfs://localhost:9000/user/hive/warehouse/baza.db    hadoop
USER        { owner = hadoop }
Time taken: 0.014 seconds, Fetched: 1 row(s)
```

-Use

Następnym podstawowym poleceniem w Hive jest komenda use służąca do ustawienia bazy danych do dalszego wykonywania zapytań.

```
hive> use baza;
OK
Time taken: 0.046 seconds
```

Teraz Hive ustawi bieżącą bazę danych jako baza w celu wykonania wszystkich kolejnych instrukcji HiveQL. Jeśli nie określono nazwy bazy danych, Hive ustawi domyślną bazę danych do wykonywania zapytań.

Polecenia z tabelami

-Create

W celu stworzenia tabeli używa się polecenia create table, po czym podaje się nazwę tabeli oraz w nawiasie podaje się nazwy kolumn wraz z typem danych w nich występujących.

```
hive> create table kursy(id_kursu int,nazwa_kursu
string,zapisani_studenci int);
OK
Time taken: 0.285 seconds
```

Można również stworzyć tabelę poprzez replikację już istniejącej, należy jednak pamiętać, że replikowana jest sama tabela baz danych znajdujących się w niej.

```
hive> create table kursy2 like kursy;
OK
Time taken: 0.292 seconds
```

-Drop

Poleceniem drop można usunąć tabelę wraz z danymi znajdującymi się wewnątrz.

```
hive> drop table if exists kursy2 PURGE;
OK
Time taken: 0.26 seconds
```

Polecenie drop usuwa określoną tabelę. Dane są zwykle przenoszone do katalogu .Trash/Current, jeśli Trash jest skonfigurowany. Jednakże jeśli podano opcję PURGE, dana tabela nie trafi do katalogu Trash i nie będzie możliwości ich odzyskania w przypadku błędnego wykonania polecenia drop.

-Truncate

To polecenie hive służy do obcinania wszystkich wierszy znajdujących się w tabeli.

```
hive> truncate table kursy;
OK
Time taken: 0.155 seconds
```

-Alter

Za pomocą polecenia alter można zmodyfikować strukturę oraz metadane tabeli po jej utworzeniu.

```
hive> alter table kursy rename to kursy1;
OK
Time taken: 0.266 seconds
```

-Describe

Po utworzeniu tabeli można sprawdzić jej opis w następujący sposób:

```
hive> describe kursy1;
OK
id_kursu                int
nazwa_kursu             string
zapisani_studenci       int
Time taken: 0.047 seconds, Fetched: 3 row(s)
```

-Show

Polecenie show podaje listę istniejących tabel w bieżącej bazie danych.

```
hive> show tables;
OK
kursy1
Time taken: 0.072 seconds, Fetched: 1 row(s)
```

-Insert

Następnie wstawimy rekord do tabeli kursy1 z id_kursu=1, nazwa_kursu=NoSQL oraz zapisani_studenci=30

```
hive> INSERT INTO TABLE kursy1 VALUES (1,'NoSQL',30);
.
.
OK

Time taken: 22.521 seconds
```

Wstawianie kolejnego rekordu wygląda identycznie:

```
hive> INSERT INTO TABLE kursy VALUES (2,'WstepNoSQL',100);
.
.
OK

Time taken: 16.214 seconds
```

-Select

Aby wyświetlić wszystkie wstawione rekordy należy skorzystać z zapytania select.

```
hive> select * from kursy1;
OK
1    NoSQL          30
2    WstepNoSQL     100
Time taken: 0.152 seconds, Fetched: 2 row(s)
```

Polecenia DML (Data Manipulation Language)

Polecenia DML służą do wstawiania, pobierania, modyfikowania, usuwania i aktualizowania danych w tabeli programu Hive po zdefiniowaniu tabeli i bazy danych za pomocą poleceń Hive DDL.

-Load

Polecenie load w Hive służy do przenoszenia plików danych do lokalizacji odpowiadających tabelom programu Hive. Jeśli podano słowo kluczowe local, to komenda load będzie szukać ścieżki do pliku w lokalnym systemie plików. Natomiast jeśli nie określono słowa kluczowego local, Hive będzie potrzebować bezwzględnego identyfikatora URL pliku. W przypadku określenia słowa kluczowego overwrite, zawartość docelowej tabeli zostanie usunięta i zastąpiona plikami, do których odwołuje się ścieżka pliku. Jednakże, jeśli nie określono słowa kluczowego overwrite, pliki do których odnosi się ścieżka pliku, zostaną dołączone do tabeli.

Syntax:

```
> LOAD DATA [LOCAL] INPATH 'filepath' [OVERWRITE] INTO TABLE
tablename [PARTITION (partcol1=val1, partcol2=val2 ...)];
```

-Select

Polecenie select w programie Hive podobnie jak w języku SQL służy do pobierania danych z bazy danych.

Syntax:

```
> SELECT col1,col2 FROM tablename;
```

-Insert

Polecenie insert w programie Hive ładuje dane do tabeli. Istnieją trzy możliwości użycia tego polecenia.

Polecenie **insert into** dołącza dane do istniejących danych w tabeli.

Syntex:

```
INSERT INTO TABLE tablename1 [PARTITION (partcol1=val1,
partcol2=val2 ...)] select_statement1 FROM from_statement;
```

Polecenie **insert overwrite** zastępuje istniejące dane w tabeli.

Syntex:

```
INSERT OVERWRITE TABLE tablename1 [PARTITION (partcol1=val1,
..) [IF NOT EXISTS]] select_statement FROM from_statement;
```

Polecenie **insert ..values** wstawia dane do tabeli bezpośrednio z SQL.

Syntex:

```
INSERT INTO TABLE tablename [PARTITION (partcol1[=val1],
partcol2[=val2] ...)] VALUES values_row [, values_row ...];
```

-Delete

Polecenie delete w Hive usuwa dane tabeli. Jeśli podano klauzulę where, usuwa wiersze spełniające podany warunek. To polecenie może być używana tylko w tabelach, które obsługują ACID.

Syntax:

```
DELETE FROM tablename [WHERE expression];
```

-Update

Polecenie update modyfikuje dane już zapisane w Hive. W zależności od warunku określonego w opcjonalnej klauzurze where, polecenie update może wpływać na każdy wiersz tabeli.

Syntax:

```
UPDATE tablename SET column = value [, column = value ...]
[WHERE expression];
```

-Export

Polecenie export eksportuje dane tabeli wraz z metadanymi do określonej lokalizacji wyjściowej w systemie plików HDFS.

Syntax:

```
EXPORT TABLE tablename [PARTITION (part_column="value"[,
...])]
TO 'export_target_path' [ FOR replication('eventid') ];
```

-Import

Polecenie import importuje dane z określonej lokalizacji do nowej tabeli lub już istniejącej tabeli.

Syntax:

```
IMPORT [[EXTERNAL] TABLE new_or_original_tablename [PARTITION
(part_column="value"[, ...])]]
FROM 'source_path' [LOCATION 'import_target_path'];
```

Bibliografia

https://hive.apache.org/index.html

https://cwiki.apache.org/confluence/display/Hive

https://docs.cloudera.com/documentation/enterprise/5-8-x/PDF/cloudera-hive.pdf

https://www.javatpoint.com/hive

https://data-flair.training/blogs/apache-hive-tutorial/

https://www.guru99.com/hive-query-language-built-operators-functions.html

https://www.geeksforgeeks.org/apache-hive/

https://www.dezyre.com/hadoop-tutorial/install-hive

https://www.tutorialspoint.com/hive/index.htm

https://www.edureka.co/blog/hive-tutorial/

https://medium.com/plumbersofdatascience/hive-architecture-in-depth-ba44e8946cbc

https://www.journaldev.com/20606/hive-commands