

# [Hadoop]Hive\_기본다지기1

## [Hadoop]Hive 기본 다지기

### 학습 목표

가. Hive 기본적인 명령어를 이해하고 기본 DB와 테이블을 만들어 볼 수 있다.

### 목차

#### [Hadoop]Hive 기본 다지기

- 1-1 시작하기
- 1-2 database 확인
- 1-3 table 생성
- 1-4 여러 줄 쿼리 입력
- 1-5 나가기
- 1-5 hive 실행 위치
- 1-6 hive 설정 파일
- 1-7 hive 명령 옵션
- 1-8 hive의 변수
- 1-9 hive의 동작과 관련된 속성 설정
- 1-10 하이버 한번에 select 문 실행시키기
- 1-11 쿼리 결과를 파일로 바로 담기
- 1-12 하이버 쿼리 실행하기
- 1-13 hive의 자동 완성
- 1-14 셸 실행
- 1-15 hive에서 하둡 dfs 명령 수행하기
- 1-16 hive 스크립트에서 주석 달기
- 1-17 table 만들기
- 1-18 데이터 베이스 생성
- 1-19 LIKE연산자를 이용한 일부 db만 보기
- 1-20 IF EXISTS 이용하기

#### 1-1 시작하기

\$ hive

하이버는 대소문자를 구분하지 않는다.

```
[cloudera@quickstart ~]$ hive
```

```
2018-06-10 15:16:53,928 WARN [main] mapreduce.TableMapReduceUtil: The hbase-prefix-tree module jar containing PrefixTreeCodec is not present. Continuing without it.
```

```
Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.properties
```

```
WARNING: Hive CLI is deprecated and migration to Beeline is recommended.
```

```
hive>
```

#### 1-2 database 확인

```
hive> show databases;
```

```
hive> show databases;
```

```
OK
```

```
default
hivedemo
Time taken: 2.85 seconds, Fetched: 2 row(s)
```

첫번째 줄은 성공적 수행 OK 출력  
그 다음은 수행 결과,  
맨 마지막은 수행 시간이 나타난다.

### 1-3 table 생성

**hive> CREATE TABLE x (a INT);**

```
hive> CREATE TABLE x (a INT);
OK
Time taken: 0.649 seconds
```

이름이 a이고 INT(4 바이트 정수)형 컬럼을 갖는 테이블 x를 생성한다.

### 1-4 여러 줄 쿼리 입력

**hive> select \* from x; <- 한줄 쿼리**

**hive> SELECT \***  
**> FROM x; <- 여러줄 쿼리**

```
hive> select * from x;
OK
Time taken: 1.531 seconds
hive> SELECT *
> FROM x;
OK
Time taken: 0.204 seconds
```

여러줄 쿼리 입력이 가능하다.

### 1-5 나가기

**\$ exit;**

hive 입력 환경을 벗어난다.

### 1-5 hive 실행 위치

**\$ which hive**

```
[cloudera@quickstart ~]$ which hive
/usr/bin/hive
```

```
[cloudera@quickstart bin]$ cd /usr/lib/hive
[cloudera@quickstart hive]$ cd conf
```

```
[cloudera@quickstart conf]$ ls -ltr
total 20
-rw-r--r-- 1 root root 3505 Jun 16 2016 hive-log4j.properties
-rw-r--r-- 1 root root 2662 Jun 16 2016 hive-exec-log4j.properties
-rw-r--r-- 1 root root 2378 Jun 16 2016 hive-env.sh.template
-rw-r--r-- 1 root root 1196 Jun 16 2016 beeline-log4j.properties.template
-rw-rw-r-- 1 root root 1937 Aug 10 2016 hive-site.xml
```

## 1-6 hive 설정 파일

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<configuration>

  <!-- Hive Configuration can either be stored in this file or in the hadoop configuration files -->
  <!-- that are implied by Hadoop setup variables. -->
  <!-- Aside from Hadoop setup variables - this file is provided as a convenience so that Hive -->
  <!-- users do not have to edit hadoop configuration files (that may be managed as a centralized -->
  <!-- resource). -->

  <!-- Hive Execution Parameters -->

  <property>
    <name>javax.jdo.option.ConnectionURL</name>
    <value>jdbc:mysql://127.0.0.1/metastore?createDatabaseIfNotExist=true</value>
    <description>JDBC connect string for a JDBC metastore</description>
  </property>

  <property>
    <name>javax.jdo.option.ConnectionDriverName</name>
    <value>com.mysql.jdbc.Driver</value>
    <description>Driver class name for a JDBC metastore</description>
  </property>

  <property>
    <name>javax.jdo.option.ConnectionUserName</name>
    <value>hive</value>
  </property>

  <property>
    <name>javax.jdo.option.ConnectionPassword</name>
    <value>cloudera</value>
  </property>

  <property>
    <name>hive.hwi.war.file</name>
    <value>/usr/lib/hive/lib/hive-hwi-0.8.1-cdh4.0.0.jar</value>
    <description>This is the WAR file with the jsp content for Hive Web Interface</description>
  </property>

  <property>
    <name>datanucleus.fixedDatastore</name>
    <value>true</value>
  </property>

  <property>
    <name>datanucleus.autoCreateSchema</name>
    <value>false</value>
  </property>

  <property>
    <name>hive.metastore.uris</name>
    <value>thrift://127.0.0.1:9083</value>
    <description>IP address (or fully-qualified domain name) and port of the metastore host</description>
  </property>
</configuration>
```

## 1-7 hive 명령 옵션

## \$ hive --help

hive 명령 도움말 확인

```
[cloudera@quickstart conf]$ hive --help
2018-06-10 15:36:49,174 WARN [main] mapreduce.TableMapReduceUtil: The hbase-prefix-tree module jar containing
PrefixTreeCodec is not present. Continuing without it.
Usage ./hive <parameters> --service serviceName <service parameters>
Service List: beeline cli help hiveburninclient hiveserver2 hiveserver hwi jar lineage metastore metatool orcfiledump rcfilecat
schemaTool version
Parameters parsed:
  --auxpath : Auxillary jars
  --config : Hive configuration directory
  --service : Starts specific service/component. cli is default
Parameters used:
  HADOOP_HOME or HADOOP_PREFIX : Hadoop install directory
  HIVE_OPT : Hive options
For help on a particular service:
  ./hive --service serviceName --help
Debug help: ./hive --debug --help
```

### (ex) CLI 명령 옵션 보기

\$ hive --service cli --help

\$ hive -h

```
[cloudera@quickstart conf]$ hive --service cli --help
2018-06-10 15:38:41,912 WARN [main] mapreduce.TableMapReduceUtil: The hbase-prefix-tree module jar containing
PrefixTreeCodec is not present. Continuing without it.
usage: hive
-d,--define <key=value>      Variable substitution to apply to hive
                             commands. e.g. -d A=B or --define A=B
--database <databasename>    Specify the database to use
-e <quoted-query-string>     SQL from command line
-f <filename>                SQL from files
-H,--help                    Print help information
--hiveconf <property=value> Use value for given property
--hivevar <key=value>        Variable substitution to apply to hive
                             commands. e.g. --hivevar A=B
-i <filename>                Initialization SQL file
-S,--silent                  Silent mode in interactive shell
-v,--verbose                  Verbose mode (echo executed SQL to the
                             console)
```

## 1-8 hive의 변수

hive> set env:HOME;

```
hive> set env:HOME;
env:HOME=/home/cloudera
```

SET명령을 이용하여 변수를 나타내고 변경한다.

hive> set;

전체 변수 목록 보기

## 1-8 hive의 변수 설정 후, 사용

\$ hive set --define foo=bar

hive> set foo;

```
hive> set foo;
foo=bar
```

```
hive> create table vartable (i int, ${foo} string);
hive> describe vartable;
```

```
hive> describe vartable;
OK
i                int
bar              string
Time taken: 0.647 seconds, Fetched: 2 row(s)
```

```
hive> drop table vartable;
```

```
hive> drop table vartable;
OK
Time taken: 1.914 seconds
```

테이블 삭제하기

## 1-9 hive의 동작과 관련된 속성 설정

```
$ hive --hiveconf hive.cli.print.current.db=true
```

CLI 프롬프트에 현재 작업 중인 데이터 베이스 이름을 출력한다.

```
[cloudera@quickstart conf]$ hive --hiveconf hive.cli.print.current.db=true
2018-06-10 15:48:32,867 WARN [main] mapreduce.TableMapReduceUtil: The hbase-prefix-tree module jar containing PrefixTreeCodec is not present. Continuing without it.
```

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.properties  
WARNING: Hive CLI is deprecated and migration to Beeline is recommended.

```
hive (default)>
```

```
$ hive --hiveconf y=5
hive> set y;
hive> CREATE TABLE whatit(i int);
```

```
[cloudera@quickstart conf]$ hive --hiveconf y=5
2018-06-10 15:52:01,230 WARN [main] mapreduce.TableMapReduceUtil: The hbase-prefix-tree module jar containing PrefixTreeCodec is not present. Continuing without it.
```

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.properties  
WARNING: Hive CLI is deprecated and migration to Beeline is recommended.

```
hive> set y
> ;
y=5
hive> set y=7;
hive> set y
> ;
y=7
```

## 1-10 하이브 한번에 select 문 실행시키기

```
$ hive -e "SELECT * FROM sample_07 LIMIT 3";
```

sample\_07 테이블의 3줄을 조회한다.

```
[cloudera@quickstart conf]$ hive -e "SELECT * FROM sample_07 LIMIT 3";
2018-06-10 15:55:38,758 WARN [main] mapreduce.TableMapReduceUtil: The hbase-prefix-tree module jar containing
PrefixTreeCodec is not present. Continuing without it.
```

```
Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.properties
OK
```

```
00-0000 All Occupations 134354250 40690
11-0000 Management occupations 6003930 96150
11-1011 Chief executives 299160 151370
```

```
Time taken: 3.967 seconds, Fetched: 3 row(s)
```

```
WARN: The method class org.apache.commons.logging.impl.SLF4JLogFactory#release() was invoked.
```

```
WARN: Please see http://www.slf4j.org/codes.html#release for an explanation.
```

## 1-11 쿼리 결과를 파일로 바로 담기

```
$ hive -S -e "SELECT * FROM sample_07 LIMIT 3" > /tmp/myquery
$ cat /tmp/myquery
```

hive 의 수행 결과를 표준 출력으로 출력한다.  
단, HDFS가 아닌 로컬 파일 시스템으로 결과를 출력한다.

```
[cloudera@quickstart conf]$ hive -S -e "SELECT * FROM sample_07 LIMIT 3" > /tmp/myquery
2018-06-10 15:57:51,993 WARN [main] mapreduce.TableMapReduceUtil: The hbase-prefix-tree module jar containing
PrefixTreeCodec is not present. Continuing without it.
```

```
[cloudera@quickstart conf]$ cat /tmp/myquery
00-0000 All Occupations 134354250 40690
11-0000 Management occupations 6003930 96150
11-1011 Chief executives 299160 151370
```

```
WARN: The method class org.apache.commons.logging.impl.SLF4JLogFactory#release() was invoked.
```

```
WARN: Please see http://www.slf4j.org/codes.html#release for an explanation.
```

## 1-12 하이브 쿼리 실행하기

```
$ hive -f /path/to/file/query01.hql
hive> /path/to/file/query01.hql
```

query01.hql의 query를 실행한다.

## 1-12 하이브 쿼리 작성하기

```
$ cd ~
$ nano query02.hql
SELECT * FROM sample_07 LIMIT 3;
$ hive -f query02.hql
```

```
[cloudera@quickstart ~]$ hive -f query02.hql
2018-06-10 16:04:54,832 WARN [main] mapreduce.TableMapReduceUtil: The hbase-prefix-tree module jar containing
PrefixTreeCodec is not present. Continuing without it.
```

```
Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.properties
OK
```

```
00-0000 All Occupations 134354250 40690
11-0000 Management occupations 6003930 96150
11-1011 Chief executives 299160 151370
```

```
Time taken: 4.229 seconds, Fetched: 3 row(s)
```

```
WARN: The method class org.apache.commons.logging.impl.SLF4JLogFactory#release() was invoked.
```

```
WARN: Please see http://www.slf4j.org/codes.html#release for an explanation.
```

hive 입력 환경을 벗어난다.

### 1-13 hive의 자동 완성

\$ hive

hive> SELE[tab키]

SELE를 누르고 탭 키를 누르면 SELECT 단어가 자동 완성된다.

### 1-14 셸 실행

hive > !pwd;

!을 이용하여 기본 셸 명령이 이용가능하다.

```
hive> !pwd
> ;
/home/cloudera
```

### 1-15 hive에서 하둡 dfs 명령 수행하기

hive > dfs -ls /;

hadoop 또는 hdfs의 명령어를 빼고 마지막에 세미콜론(;)을 넣기만 하면 된다.

```
hive> dfs -ls /;
Found 6 items
drwxrwxrwx - hdfs supergroup      0 2016-08-10 13:07 /benchmarks
drwxr-xr-x - hbase supergroup      0 2018-06-09 21:32 /hbase
drwxr-xr-x - solr solr             0 2016-08-10 13:09 /solr
drwxrwxrwt - hdfs supergroup      0 2018-06-09 22:55 /tmp
drwxr-xr-x - hdfs supergroup      0 2018-06-06 07:35 /user
drwxr-xr-x - hdfs supergroup      0 2016-08-10 13:09 /var
```

### 1-16 hive 스크립트에서 주석 달기

문자열 '--'로 시작하는 주석을 하이브 스크립트에 넣을 수 있다.

```
-- Copyright (c) 2018 KK
-- This is a hive script
```

```
SELECT * FROM sample_07 limit 3
```

### 1-17 table 만들기

```
create table some_data (
  name STRING,
  salary FLOAT
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ',';
```

some\_data 테이블 생성, ','을 이용하여 필드를 분리하겠다는 의미

```
hive> create table some_data (  
  > name STRING,  
  > salary FLOAT  
  > )  
  > ROW FORMAT DELIMITED  
  > FIELDS TERMINATED BY ',';  
OK  
Time taken: 1.175 seconds
```

## ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';

','을 이용하여 필드를 분리하겠다는 의미

### 1-18 데이터 베이스 생성

```
hive> create database hivedb;
```

**hivedb** 를 생성한다.

```
hive> show databases;  
OK  
default  
hivedb  
hivedemo  
Time taken: 0.317 seconds, Fetched: 3 row(s)
```

### 1-19 LIKE연산자를 이용한 일부 db만 보기

```
hive> SHOW DATABASES LIKE 'h.*';
```

h로 시작하고 어떠한 문자로 끝날 수 있는 데이터 베이스의 이름만 나열한다.  
default db가 안보임.

```
hive> SHOW DATABASES LIKE 'h.*';  
OK  
hivedb  
hivedemo  
Time taken: 0.075 seconds, Fetched: 2 row(s)
```

### 1-20 IF EXISTS 이용하기

```
hive> DROP DATABASE IF EXISTS hivedb;
```

```
hive> DROP DATABASE IF EXISTS hivedb;  
OK  
Time taken: 0.324 seconds
```