[Hadoop]Hive01_Lab02_HiveWithCloudera

[Hadoop]Hive_Lab02_HiveWithCloudera

학습 목표

가. 실습을 통해 Hive의 기본동작을 이해할 수 있다.
나. Hive Query를 사용할 수 있다.
다. 테이블 생성 및 테이블로 데이터 가져오기를 이해할 수 있다.

목차

==========

1-1 데이터 다운로드 후, HDFS 데이터 올리기

(가) 웹에서 데이터 다운로드

- cloudera 계정으로 접속 후, wget명령을 이용한 데이터 다운로드

```
su - cloudera
wget http://www.grouplens.org/system/files/ml-100k.zip
```

위의 데이터 셋의 간단한 설명은 다음과 같다.
1998년 4월 1700편의 영화에 대한 1000명의 유저의 100,000 평이다.

https://grouplens.org/datasets/movielens/

older datasets

MovieLens 100K Dataset

Stable benchmark dataset. 100,000 ratings from 1000 users on 1700 movies. Released 4/1998.

- README.txt
- ml-100k.zip (size: 5 MB, checksum)
- Index of unzipped files

Permalink: http://grouplens.org/datasets/movielens/100k/

- 압축을 푼 후, 해당 디렉터리로 이동

```
unzip ml-100k.zip
cd ml-100k
```

(나) HDFS에 데이터 올리기

데이터 확인 명령
hdfs dfs -ls /
hdfs dfs -ls /user/cloudera # /user/cloudera의 디렉터리 리스트 확인

HDFS 시스템 디렉터리 만들기

hdfs dfs -mkdir movieinfo # movieInf(영화정보) 디렉터리 만들기
hdfs dfs -mkdir userinfo # userinfo (유저정보) 디렉터리 만들기

hdfs dfs -ls /user/cloudera # 디렉터리 확인

(만약 기존의 폴더를 지우고자 할때는 아래 명령어를 이용)
hdfs dfs -rmr movies
hdfs dfs -rmr movieinfo
hdfs dfs -rmr userinfo
hdfs dfs -ls /user/cloudera

(Ref) https://hadoop.apache.org/docs/r2.4.1/hadoop-project-dist/hadoop-common/FileSystemShell.html#rmr

디렉터리에 데이터 올리기

hdfs dfs -put u.item movieinfo # 영화정보 올리기
hdfs dfs -put u.user userinfo # 유저정보 올리기

hdfs dfs -ls -R /user/cloudera # 데이터 확인

======= 데이터 셋 확인

대상 데이터 셋은 u.item, u.data, u.user

u.item : (영화) 항목에 대한 정보

영화 ID | 영화 제목 | 출시일 | 비디오 출시일 |
IMDb URL | 알 수없는 | 액션 | 모험 | 애니메이션 |
어린이 | 코메디 | 범죄 | 다큐멘터리 | 드라마 | 판타지 |
필름 누아 | 공포 | 뮤지컬 | 미스테리 | 로맨스 | 공상 과학 |
스릴러 | 전쟁 | 서양 |
마지막 19 개의 필드는 장르

u.data : 전체 U데이터 세트, 1682 항목의 943명의 사용자가 100000개의 평가

사용자 ID | 상품 ID | 평가 | 타임 스탬프.

u.user : 사용자의 인구 통계 정보.

사용자 ID | 나이 | 성별 | 직업 | 우편 번호

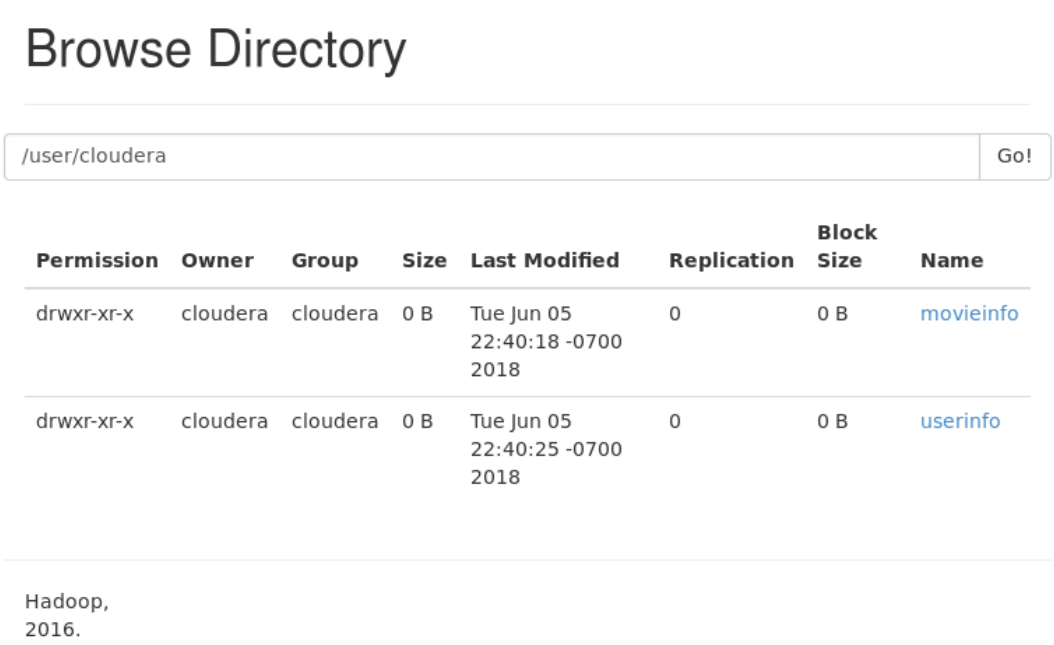
(다) 파일 확인(CLI)

hdfs dfs -ls -R /user/cloudera

(라) 파일 확인(Web)

웹 브라우저 실행 - Hadoop 선택 - Utilities - Browse Directory 선택 후,

 # Browse Directory
/user/cloudera



Browse Directory

/user/cloudera Go!

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	cloudera	cloudera	0 B	Tue Jun 05 22:40:18 -0700 2018	0	0 B	movieinfo
drwxr-xr-x	cloudera	cloudera	0 B	Tue Jun 05 22:40:25 -0700 2018	0	0 B	userinfo

Hadoop, 2016.

1-2 HDFS 데이터 Hive DB에 올리기

(가) hive 실행

[cloudera@quickstart ml-100k]$ hive
2018-06-05 23:08:44,617 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.

(나) 초기 설정

set hive.cli.print.current.db=true;

데이터 조회 시, DB 이름을 보이도록 한다.

hive> set hive.cli.print.current.db=true;
hive (default)>

(다) DB 생성 및 확인

DB 생성

create database hivedemo;
show databases;

 # 에러 메세지

FAILED: Execution Error, return code 1 from org.apache.hadoop.hive.ql.exec.DDLTask. Database hivedemo already exists
이미 DB가 존재.

*** 만약 DB가 있다면 삭제.**

drop database if EXISTS userdb;
drop database if EXISTS hivedemo;
drop database if EXISTS hivedemo CASCADE;

[실행결과]
hive (default)> create database hivedemo;
OK
Time taken: 0.199 seconds
hive (default)> show databases;
OK

```
default
hivedemo
Time taken: 0.378 seconds, Fetched: 2 row(s)
hive (default)>
```

(라) 그렇다면 DB는 HDFS에 어디에 생성되어 있을까?

(Linux 환경) hdfs dfs -ls /user/hive/warehouse;
(hive 환경) !hdfs dfs -ls /user/hive/warehouse;

[실행 결과]

```
hive (default)> !hdfs dfs -ls  /user/hive/warehouse;
Found 1 items
drwxr-xr-x   - cloudera hive          0 2018-06-05 23:20 /user/hive/warehouse/hivedemo.db
```

(마) hivedemo 에 movies 테이블 생성

```
CREATE TABLE movieinfo (
    movie_id INT,
    movie_title STRING,
    release_date STRING,
    video_release_date STRING,
    imdb_url STRING,
    unknown INT,
    action INT,
    adventure INT,
    animation INT,
    children INT,
    comedy INT,
    crime INT,
    documentary INT,
    drama INT,
    fantasy INT,
    film_noir INT,
    horror INT,
    musical INT,
    mystery INT,
    romance INT,
    sci_fi INT,
    thriller INT,
    war INT,
    Western INT
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '|'
STORED AS TEXTFILE;
```

hive (hivedemo)> show tables;

```
hive (default)> show tables;
OK
movieinfo
Time taken: 0.019 seconds, Fetched: 1 row(s)
```

- 테이블의 구조 확인. 현재 상태

```
hive (default)> describe  movieinfo;
OK
movie_id                int
movie_title             string
release_date            string
video_release_date      string
imdb_url                string
unknown                 int
```

```
action              int
adventure           int
animation           int
children            int
comedy              int
crime               int
documentary         int
drama               int
fantasy             int
film_noir           int
horror              int
musical             int
mystery             int
romance             int
sci_fi              int
thriller            int
war                 int
western             int
Time taken: 0.136 seconds, Fetched: 24 row(s)
```

HDFS 파일 시스템의 파일을 hive table로 옮긴다.

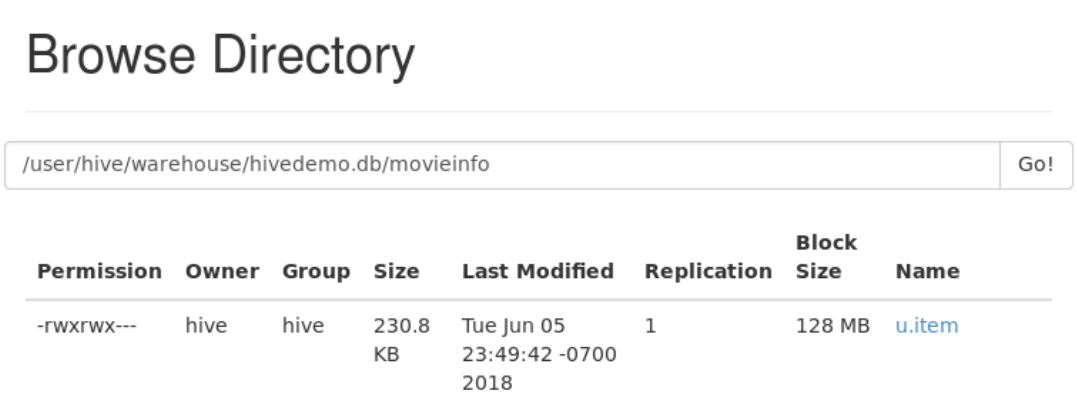
```
hive>!hdfs dfs -ls -R  /user/cloudera;
hive>LOAD DATA INPATH '/user/cloudera/movieinfo' INTO TABLE  movieinfo;
```

```
Loading data to table default.movieinfo
chgrp: changing ownership of 'hdfs://quickstart.cloudera:8020/user/hive/warehouse/movieinfo/u.item': Permission denied:
user=cloudera, access=EXECUTE, inode="/user/hive/warehouse":hive:hive:drwxrwx---
chmod: changing permissions of 'hdfs://quickstart.cloudera:8020/user/hive/warehouse/movieinfo/u.item': Permission denied:
user=cloudera, access=EXECUTE, inode="/user/hive/warehouse":hive:hive:drwxrwx---
Table default.movieinfo stats: [numFiles=1, totalSize=236344]
OK
```

========== 만약 데이터가 없고 문제가 있다면 권한 변경이 필요할 수 있음.

```
# 권한과 소유자를 변경해 준다.
# /user/hive/warehouse의 디렉터리 사용자와 그룹에게 모든 권한을 주고
# /user/hive/warehouse의 디렉터리 사용자와 그룹을 hive로 변경한다.
!sudo -u hdfs hdfs dfs -chmod -R 770 /user/hive/warehouse;
!sudo -u hdfs hdfs dfs -chown -R hive:hive /user/hive/warehouse;

# 확인
!hdfs dfs -ls -R   /user/hive/warehouse;    # hive 실행환경 아래.
# 정상적으로 hive로 데이터가 올라가게 되면 기존의 HDFS에 있는 데이터는 없어지고 옮겨지게 된다.
```




```
CREATE EXTERNAL TABLE users (
user_id INT,
age INT,
gender STRING,
```

```
occupation STRING,
zip_code STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '|'
STORED AS TEXTFILE;
```

두가지 방법, 둘 중에 하나의 방법 사용 가능.
데이터 올리기(LocalPC -> hiveDB)

```
LOAD DATA LOCAL INPATH '/home/cloudera/ml-100k/u.user' INTO TABLE users;
```

데이터 올리기(HDFS-> hiveDB)

```
LOAD DATA INPATH  '/home/cloudera/ml-100k/u.user' INTO TABLE users;
```

데이터 확인

select * from users limit 10;

```
hive (hivedemo)> select * from users limit 10;
OK
1    24    M    technician    85711
2    53    F    other    94043
3    23    M    writer    32067
4    24    M    technician    43537
5    33    F    other    15213
6    42    M    executive    98101
7    57    M    administrator    91344
8    36    M    administrator    05201
9    29    M    student    01002
10    53    M    lawyer    90703
```

select count(*) from users; # 결과 943명의 유저

정리된 결과

describe formatted users;

아티스티라는 직업을 가진 사람은 몇명인가?

select count(*) from users where occupation = 'artist';

```
Query ID = cloudera_20180606021515_3ad00e5f-6bb0-4e32-90b5-620362851d27
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1528261512214_0002, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1528261512214_0002/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1528261512214_0002
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-06-06 02:15:31,712 Stage-1 map = 0%,  reduce = 0%
2018-06-06 02:15:42,824 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.98 sec
2018-06-06 02:15:55,147 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 3.89 sec
MapReduce Total cumulative CPU time: 3 seconds 890 msec
Ended Job = job_1528261512214_0002
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 3.89 sec   HDFS Read: 52789 HDFS Write: 3 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 890 msec
OK
```

56

```
Time taken: 36.361 seconds, Fetched: 1 row(s)
```

1-4 HiveQL을 이용한 Query 사용

sub Query를 이용한 새로운 table 만들기
CREATE TABLE occupation_count
STORED AS RCFile AS
SELECT COUNT(*), occupation FROM users GROUP BY occupation;

확인 - 21개의 직업과 각각의 사람들의 수
hive (hivedemo)> **select * from occupation_count ;**

```
OK
158   administrator
56    artist
14    doctor
190   educator
134   engineer
36    entertainment
64    executive
32    healthcare
14    homemaker
24    lawyer
102   librarian
52    marketing
18    none
210   other
132   programmer
28    retired
24    salesman
62    scientist
392   student
54    technician
90    writer
Time taken: 0.066 seconds, Fetched: 21 row(s)
```

CREATE TABLE occupation_count2 LIKE occupation_count;
select * from occupation_count ;

```
hive (hivedemo)> select * from occupation_count ;
OK
158   administrator
56    artist
14    doctor
190   educator
134   engineer
36    entertainment
64    executive
32    healthcare
14    homemaker
24    lawyer
102   librarian
52    marketing
18    none
210   other
132   programmer
28    retired
24    salesman
62    scientist
392   student
54    technician
90    writer
Time taken: 0.064 seconds, Fetched: 21 row(s)
```

hive (hivedemo)> **!hdfs dfs -ls -R /user/hive/warehouse**
> ;

```
drwxrwx---   - hive hive          0 2018-06-06 02:19 /user/hive/warehouse/hivedemo.db
drwxr-xr-x   - cloudera hive          0 2018-06-06 01:54 /user/hive/warehouse/hivedemo.db/movieinfo
-rw-r--r--   1 cloudera cloudera     236344 2018-06-06 01:54 /user/hive/warehouse/hivedemo.db/movieinfo/u.item
drwxr-xr-x   - cloudera hive          0 2018-06-06 02:17 /user/hive/warehouse/hivedemo.db/occupation_count
-rw-r--r--   1 cloudera hive          336 2018-06-06 02:17
/user/hive/warehouse/hivedemo.db/occupation_count/000000_0
drwxr-xr-x   - cloudera hive          0 2018-06-06 02:19 /user/hive/warehouse/hivedemo.db/occupation_count2
```

다시 정리

```
hive> drop  database  hivedemo  cascade;
hive> show databases;
# hdfs dfs -rm -r  /user/cloudera/*
hdfs dfs -ls -R  /user/cloudera/*
# HDFS파일 지우기
```

처음상태로 돌아왔다.

다시 해 보자.

폴더만들기

movies, userinfo

```
[cloudera@quickstart ml-100k]$ hdfs dfs -mkdir movies
[cloudera@quickstart ml-100k]$ hdfs dfs -mkdir userinfo
```

데이터 올리기

```
 hdfs dfs -put u.item movies
 hdfs dfs -put u.user userinfo

[cloudera@quickstart ml-100k]$ hdfs dfs -put u.item movies
[cloudera@quickstart ml-100k]$ hdfs dfs -put u.user userinfo

# hive 에서 DB구축
[cloudera@quickstart ml-100k]$ hive
hive> use hivedemo ;

CREATE EXTERNAL TABLE users (
 user_id INT,
 age INT,
 gender STRING,
 occupation STRING,
 zip_code STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '|'
STORED AS TEXTFILE;

show tables;
```

데이터 넣기

```
LOAD DATA LOCAL INPATH '/home/cloudera/ml-100k/u.user' INTO TABLE users;
```

```
select * from users limit 10;
select count(*) from users where occupation = 'artist';
```

지워보기

```
hive> drop table users;
```

Web UI

스키마도 지우기

```
!hdfs dfs -rm -r /user/hive/warehouse/hivedemo.db/users;
```

[실습과제] 자신의 분석하고자 하는 데이터 셋을 Hive에 Table를 만들어 데이터를 올리고 Query를 이용하여 기본 데이터 조회를 수행해 보자.