Hive를 사용한 movielen 분석

(이삭엔지니어링 인턴 김형근, [hkkim@isaac-eng.com](mailto:hkkim@isaac-eng.com) )

(CDH 5.15.1, master1.isaac-eng.com에서 진행)

1. **MovieLens Dataset 가져오기**

$ wget http://files.grouplens.org/datasets/movielens/ml-1m.zip   
$ unzip ml-m1.zip  
$ cd ml-m1

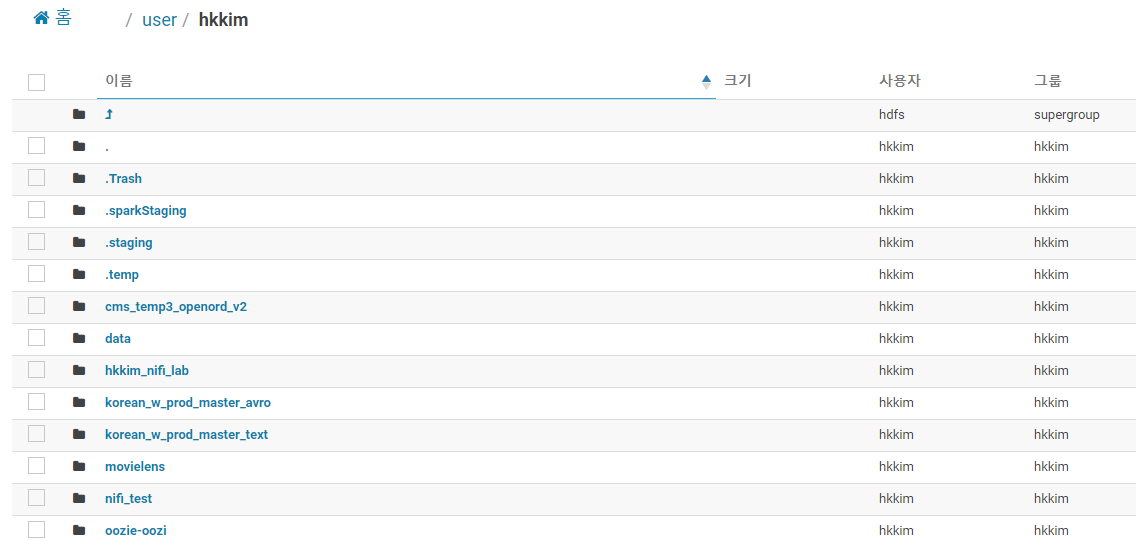
// ml-m1 디렉터리에는 movies.dat, users.dat, users.dat 라는 파일들이 존재할 것이다

$ sed -i 's/::/,/g' ml-1m/movies.dat  
$ sed -i 's/::/,/g' ml-1m/users.dat  
$ sed -i 's/::/,/g' ml-1m/ratings.dat  
// 파일 내부의 구분자를 “:”에서 “,”로 변경시킨다.

$ mv ml-1m/movies.dat /ml-1m/movies.csv  
$ mv ml-1m/ratings.dat /ml-1m/ratings.csv  
$ mv ml-1m/users.dat /ml-1m/users.csv  
// 각 파일들의 파일형식을 csv로 바꾼다.

1. **movielens 디렉터리 만들기**

$ hdfs dfs -mkdir /user/hkkim/movielens  
$ hdfs dfs -ls /user/hkkim



1. **SQL문 작성**  
   **(1) movies.sql**  
   DROP DATABASE IF EXISTS movielens CASCADE;

CREATE DATABASE movielens;

USE movielens;

CREATE EXTERNAL TABLE movies (MovieID INT,

Title varchar(60),

Genres varchar(60))

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

LINES TERMINATED BY "\n"

STORED AS TEXTFILE

LOCATION '/user/hkkim/movielens/ml-1m/mvs.txt';

LOAD DATA INPATH '/user/hkkim/movielens/ml-1m/movies.csv' INTO TABLE movies;

SELECT \* FROM movies LIMIT 10;  
  
**(2) ratings.sql**  
USE movielens;

CREATE EXTERNAL TABLE ratings (UserID INT,

MovieID INT,

Rating INT,

Timestamp STRING)

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

LINES TERMINATED BY "\n"

STORED AS TEXTFILE

LOCATION '/user/hkkim/movielens/ml-1m/rts.txt';

LOAD DATA INPATH '/user/hkkim/movielens/ml-1m/ratings.csv' INTO TABLE ratings;

SELECT \* FROM ratings LIMIT 10;

**(3) users.sql**USE movielens;

CREATE EXTERNAL TABLE users (UserID INT,

Gender STRING,

Age INT,

Occupation INT,

ZIP INT)

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

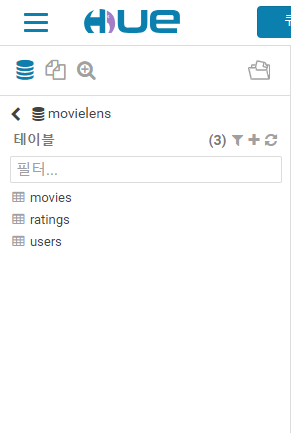
LINES TERMINATED BY "\n"

STORED AS TEXTFILE

LOCATION '/user/hkkim/movielens/ml-1m/usr.txt';

LOAD DATA INPATH '/user/hkkim/movielens/ml-1m/users.csv' INTO TABLE users;

SELECT \* FROM users LIMIT 10;

$ hive -f movies.sql  
$ hive -f ratings.sql  
$ hive -f users.sql  
// 위와 같이 작성 HiveQL문을 실행시킨다.  
  


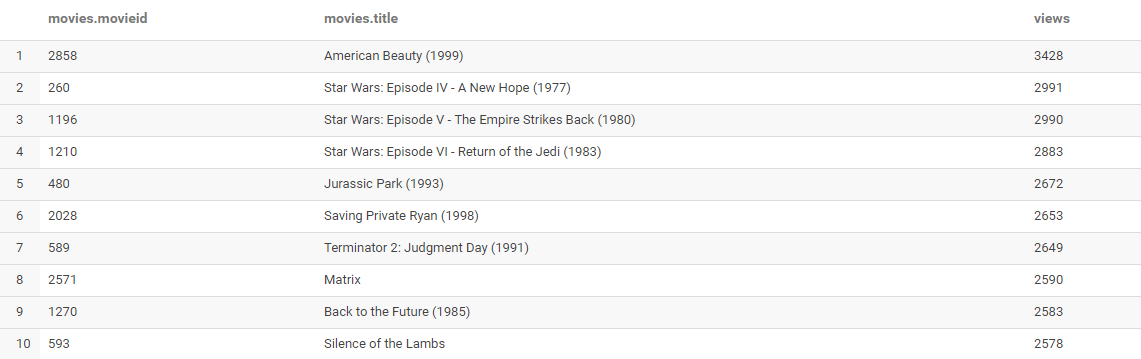
1. **영화 순위 분석  
   (1) Top 10 영화**

SELECT movies.MovieID,movies.Title,COUNT(DISTINCT ratings.UserID) as views

FROM movies JOIN ratings ON (movies.MovieID = ratings.MovieID)

GROUP BY movies.MovieID, movies.Title

ORDER BY views DESC

LIMIT 10;  
  
  
**(2) 관람횟수가 40번 이상인 Top 20 영화**  
SELECT movies.MovieID,movies.Title,AVG(ratings.Rating) as rtg,COUNT(DISTINCT ratings.UserID) as views

FROM movies JOIN ratings ON (movies.MovieID = ratings.MovieID)

GROUP BY movies.MovieID,movies.Title

HAVING views >= 40

ORDER BY rtg DESC

LIMIT 20;  


Reference : <https://towardsdatascience.com/getting-started-with-hive-ad8a93862f1a>