**Data Set for Hive Practice**

1. Take sample data source for use case from below link:  
     
   [**http://www.grouplens.org/system/files/ml-1m.zip**](http://www.grouplens.org/system/files/ml-1m.zip)
2. It contains data around *movies, users, ratings*.  unzip it.
3. Below are the 3 files in archive:  
     
     movies.dat, ratings.dat, users.dat
4. Files in above are delimited by '::' just to have better readability (and one example to handle delimiter) change the delimiter to something other, you can keep the same, I am changing it to '#'  
     
   sed 's/::/#/g' movies.dat  
   sed 's/::/#/g' users.dat  
   sed 's/::/#/g' ratings.dat  
     
   Contents of the file would be:  
     
   **movies:**  
    **structure:**  
   id#name#genre  
     
   **sample data :**  
   1#Toy Story (1995)#Animation|Children's|Comedy  
   2#Jumanji (1995)#Adventure|Children's|Fantasy  
   3#Grumpier Old Men (1995)#Comedy|Romance  
   4#Waiting to Exhale (1995)#Comedy|Drama  
     
   **users:**  
     
   **structure:**  
   id#gender#age#occupationid#zipcode  
     
   **sample data:**  
   1#F#1#10#48067  
   2#M#56#16#70072  
   3#M#25#15#55117  
   4#M#45#7#02460  
   5#M#25#20#55455  
     
    **ratings:**  
     
   **structure:**  
   userid#movieid#rating#tmstmp  
     
   **Sample Data:**  
   1#1193#5#978300760  
   1#661#3#978302109  
   1#914#3#978301968  
   1#3408#4#978300275  
   1#2355#5#978824291

**just to have meaningful data, create an occupation data set**  
 **create a file named occupation.dat with below data:**  
 *vim occupation.dat*  
  
copy paste below and save the file.  
  
0#other/not specified  
1#academic/educator  
2#artist  
3#clerical/admin  
4#college/grad student  
5#customer service  
6#doctor/health care  
7#executive/managerial  
8#farmer  
9#homemaker  
10#K-12 student  
11#lawyer  
12#programmer  
13#retired  
14#sales/marketing  
15#scientist  
16#self-employed  
17#technician/engineer  
18#tradesman/craftsman  
19#unemployed  
20#writer  
  
  
Move the above files into the HDFS:  
  
I have created 4 directories in /hive/data named user, movie, rating, occupation  
  
*hadoop fs -put occupations.dat /hive/data/occupation*  
*hadoop fs -put users.dat /hive/data/user*  
*hadoop fs -put movies.dat /hive/data/movie*  
*hadoop fs -put ratngs.dat /hive/data/rating*

1. if the data set up is done now let's do the hive stuff:  
    **1. create a separate database named movielens**          create database movielens;  
             use movielens;  
            
   **2. create tables to hold data**       
             CREATE EXTERNAL TABLE ratings (  
                      userid INT,  
                      movieid INT,  
                      rating INT,  
                      tstamp STRING  
                       ) ROW FORMAT DELIMITED  
                       FIELDS TERMINATED BY '#'  
                       STORED AS TEXTFILE  
                       LOCATION '/hive/data/rating';  
                      
            
             CREATE EXTERNAL TABLE movies (  
                       movieid INT,  
                       title STRING,  
                       genres ARRAY<STRING>  
                     ) ROW FORMAT DELIMITED  
                     FIELDS TERMINATED BY '#'  
                     COLLECTION ITEMS TERMINATED BY "|"  
                     STORED AS TEXTFILE  
                     LOCATION '/hive/data/movie';  
                    
             CREATE EXTERNAL TABLE users (  
                       userid INT,  
                       gender STRING,  
                       age INT,  
                       occupation\_id INT,  
                       zipcode STRING  
                     ) ROW FORMAT DELIMITED  
                     FIELDS TERMINATED BY '#'  
                     STORED AS TEXTFILE  
                     LOCATION '/hive/data/user';  
              
              CREATE EXTERNAL TABLE occupations (  
                       id INT,  
                       occupation STRING  
                     ) ROW FORMAT DELIMITED  
                     FIELDS TERMINATED BY '#'  
                     STORED AS TEXTFILE  
                     LOCATION '/hive/data/occupation';  
                    
   **3. see if data is loaded**  
     
   hive> select \* from users limit 2;  
   OK  
   1 F 1 10 48067  
   2 M 56 16 70072  
   Time taken: 0.278 seconds, Fetched: 2 row(s)  
     
   hive> select \* from movies limit 2;  
   OK  
   1 Toy Story (1995) ["Animation","Children's","Comedy"]  
   2 Jumanji (1995) ["Adventure","Children's","Fantasy"]  
   Time taken: 0.352 seconds, Fetched: 2 row(s)  
     
   hive> select \* from ratings limit 2;  
   OK  
   1 1193 5 978300760  
   1 661 3 978302109  
   Time taken: 0.28 seconds, Fetched: 2 row(s)  
     
   hive> select \* from occupations limit 2;  
   OK  
   0 other/not specified  
   1 academic/educator  
   Time taken: 0.245 seconds, Fetched: 2 row(s)  
     
   if you are all good till here than lets practice hiveQL stuffs.  
     
   NOTE: in each case to maintain readabilty I will limit the output to 10 only.  
     
   Use Case 1:  
   Find out Occupation of all the users:  
     
   Solution:  
   select u.\*, o.occupation from users u, occupations o where u.occupation\_id= o.id limit 10;

OUTPUT:  
  
1 F 1 10 48067 K-12 student  
2 M 56 16 70072 self-employed  
3 M 25 15 55117 scientist  
4 M 45 7 02460 executive/managerial  
5 M 25 20 55455 writer  
6 F 50 9 55117 homemaker  
7 M 35 1 06810 academic/educator  
8 M 25 12 11413 programmer  
9 M 25 17 61614 technician/engineer  
10 F 35 1 95370 academic/educator  
  
  
Use Case 2:  
Find out numbers of non-adults as per Indian standard, who has rated movies:  
  
Solution: select count(\*) from users where age < 18;  
222  
  
Use case 3:  
Find out the no of users with same occupation and having age more than 25 along with occupation details:  
  
Solution:  
select o.occupation, count(1)  from users u join occupations o where u.occupation\_id= o.id AND u.age > 24 group by o.occupation;  
  
K-12 student 3  
academic/educator 479  
artist 220  
clerical/admin 155  
college/grad student 222  
customer service 94  
doctor/health care 227  
executive/managerial 660  
farmer 15  
homemaker 86  
lawyer 121  
other/not specified 578  
programmer 328  
retired 141  
sales/marketing 263  
scientist 130  
self-employed 223  
technician/engineer 448  
tradesman/craftsman 60  
unemployed 30  
writer 232  
  
  
Use Case 4: Find the age of the most rated user with counts of rating;  
  
Solution:  
  
select u.userid, u.age, x.count from users u join ( select r.userid, count(rating) count from ratings r group by (r.userid) order by count DESC limit 1) x where u.userid = x.userid;  
  
4169 50 2314