Data set The data that you will use is the Last.fm (http://www.last.fm)data set released in HetRec2011 (http://ir.ii.uam.es/hetrec2011/). You can view the details of the data set and download it from http://grouplens.org/datasets/hetrec-2011/. The data set contains information about "social networking, tagging, and music artist listening information from a set of 2k users from Last.fm online music system". The data is organized as relational tables and is stored in several text files, each corresponding to a table. All files are of tab separated format. Unique IDs are assigned to artists, users and tags for easy referencing across files. Basic information about artists is stored in file artists.dat. Each line contains four columns: id, name of the artist, an url pointing to the description of the artist and another pictureURL pointing to a picture of the artist. The tag data is stored in another file tags.dat with only two columns: the tagID and the actual tagValue. The data set does not contain any personal information about user, hence there is no separate file for users. All other files in the data set contain information about some relationships. The user artists.dat stores information about listening count per user per artist. The user friends.dat stores the friend relations between users. The tag assignments of artists by users are stored in two files. The only difference between the files is the timestamp format. The file user taggedartists.dat stores the day, month and year in separate columns 1 while user taggedartists-timestamps.dat stores the unix timestamp in a single column. You only need to use one of the files for tag information.

Target Queries
Simple query

• given a user id, find all artists the user's friends listen.

• given an artist name, find the most recent 10 tags that have been assigned to it.

• given an artist name, find the top 10 users based on their respective listening counts of this artist. Display both the user id and the listening count

• given a user id, find the most recent 10 artists the user has assigned tag to.

Complex queries

• find the top 5 artists ranked by the number of users listening to it

• given an artist name, find the top 20 tags assigned to it.
The tags are ranked by the number of times it has been assigned to this artist

• given a user id, find the top 5 artists listened by his friends but not him. We rank artists by the sum of friends' listening counts of the artist.

• given an artist name, find the top 5 similar artists. Here similarity between a pair of artists is defined by the number of unique users that have listened both. The higher the number, the more similar the two artists are.

For MongoDB, load the complete data on MongoDB and set up proper indexes that will be used by the target queries. Design and implement all target queries. You may implement a query using shell command, a combination of JavaScript and shell command or as Python/Java program. For each query (or sub query), report execution statistics such as: which index is used, how many documents are examined to answer this query.