Introduction to Database Design, F2016
IT University of Copenhagen
Carsten Schürmann

Project 4

This project assignment is a group assignment.

Wednesday, 14 December 2016, 23:55

Rules:

- Submit your groups work as a single PDF file with a file name that includes the groups name.
- Upload your work on `learnit.itu.dk`.
- Late hand-ins will not be considered.
- The grade of all project assignments will also be averaged and determine the grade of assignment 4, which also counts with 15% towards your grade.

In this hand-in you will be doing analytics on a data set of movie ratings from MovieLens (movielens.umn.edu). It contains about 1 million user ratings of movies (on the scale 1-5), each associated with a unique user ID and the date of the rating. For each user, gender, approximate age, occupation, and zip code is recorded. Finally, information about the genre(s) of each movie is available. The main data is available as three relations:

```
user(id,gender,age,occupation,zip)
rating(userId,movieId,rating,time)
movieGenre(genreId,movieId)
```

These relations use numerical codes for occupation, movie ID, and genres. Textual descriptions of these codes are available as:

```
occupation(id,description)
movie(id,title)
genre(id,name)
```

Finally, a separate relation with information about zip codes is available:

```
zipcode(zip,city,state,lattitude,longitude,timezone,dst)
```

From the course homepage you can download a (ziped) file movielens.sql containing this data, prepared for importing into MySQL. (There is also a bigger file, movielensXL.sql, if you feel like trying some of the tasks with a bigger, but not as detailed, data set.)

Tasks

Your first task is to perform *data cleaning*. In particular, we wish to remove all data that lacks proper foreign key references. For example, there are tuples in user that do not refer to a tuple in zipcode. Another issue is that age approximations can be made more accurate. In the data set, age 1 means Under 18, age 18 means 18- 24, age 25 means 25-34, etc., up to 56 which means 56+. Modify the ages so that they are a best guess for the range; clearly, this is not an exact science.

Second, you should choose a way of *enriching* the data set. For example, you could make a table with the median or average household income for each zip code. This data can be downloaded in XLS format from `http://www.psc.isr.umich.edu/dis/census/Features/tract2zip/index.html` To get the data into MySQL, export as a comma-separated (CSV) file, and use MySQL's

```
LOAD DATA LOCAL INFILE
```

syntax to load it into a table. Alternatively, integrate with information in the IMDB database you worked with previously. (NB! Movie IDs and titles are not identical.)

The third task is to create a relational OLAP model for the data, according to the principles discussed in the class. Since MySQL does not support materialized views, you will need to construct separate tables with pre-aggregated data, and put indexes on these. B-tree indexes will suffice, even though they may be slower than bitmap indexes in this context. Motivate your choice of pre-aggregation with a few usage scenarios, and give corresponding example SQL queries (probably working on the pre-aggregated tables rather than the whole data set).