CSCI 620/Section 03/Mior, Introduction to Big Data, Spring 2215

Assignment 3 – Normalization

Your tasks

1. Provide a program to create a new relation which is the result of joining Movie, Movie_Genre, Genre, Member, Movie_Actor, and Actor_Movie_Character from assignment 2. Restrict yourself to the following attributes: movieId, type, startYear, runtime, avgRating, genreId, genre, memberId, birthYear, character. Only use movies whose runtimes are greater or equal than 90 minutes and those movies where all actors only play a single character. Explain your decisions.

   (Hint: When creating the new relation, you need to uniquely identify each tuple.) **(10 points)**

2. Provide a program implementing the naïve approach to discover functional dependencies on the relation from question 1. The naïve approach checks all possible dependencies against all pairs of rows. You can exclude trivial dependencies. Only include dependencies with one attribute on the right hand side. Run your program for a while and provide an estimate on the time it should take to complete. Explain your answer. **(20 points)**

3. Provide a program implementing the pruning approach to discover functional dependencies on the relation from question 1. (That is, compute the partitions for the input columns and use partition refinement and a lattice of possible dependencies.) Your program needs to discover functional dependencies with combinations of no more than two attributes on the left-hand side (and one on the right) in less than five hours. Report the functional dependencies your program finds and provide examples of pruning functional dependencies. Explain your answer. **(40 points)**

4. Assuming that there are no more minimal functional dependencies than the ones computed in Question 3 (combinations of no more than two attributes on the left-hand side), explain the outcome if we do not restrict that “the same actor only plays a single character in a given movie” as in question 1. **(10 points)**

5. Compute a 3NF decomposition of the relation from question 1 given the set of functional dependencies discovered in question 3. You may do this manually or write a program to do so. Provide the results (candidate keys, canonical cover, final decomposition).
   (20 points)