Table A and B both could be matched using an algorithm that compared the titles contained in each of the tables. If the title matched, then the other attributes could then be further compared to each other to see if an attribute like rating was similar. If the title from table A matched the title in table B, then the matching pair and their attributes would be saved into the new table C csv file. The table C csv file contains all the matching pairs between Table A and Table B based on the "title" column, formatted using the below header:

ID,atable_rank,btable_rank,title,atable_rating,atable_show_type,atable_votes,atable_episodes,btable_rating,btable_show_type,btable_votes,btable_episodes

A problem that could be present with this algorithm is that it is looking for an exact match in the title names between both tables. Therefore, if the titles were the same but had any differences in syntax or one contained a subtitle, it would not match and would not be saved to the new table of matching data.

Table A – 1000 tuples (1a-1000a)

Table B – 1000 tuples (1b-1000b)

Total number of tuple pairs in Cartesian product of A and B – 1,000,000 pairs

Total number of tuple pairs in table C – 113 tuples (ID 0-112)

Table A and B were both cleaned to make sure that they shared the same schema of attributes in the same order, this was done before the matching code was run. There was no further cleaning done other than making sure Table A and B shared the same attributes in the same column order.

Python Packages Used: panda import was used in the titleattributematching.py code so that DataFrames could be used to match the "title" attribute.