

#### -----PERFORMANCE TUNING REPORT-----

My initial approach to parsing the three xml files was a naïve implementation that took up to 20 minutes to run. I made three connections for each of the xml files, and queried the database for every star and movie element in order to traverse the existing data and check for duplicates. This was very inefficient as these database calls were very expensive, which resulted in a very slow execution time since it would make these calls for over 12000 movie records.

My first method of optimizing this was to refactor my program to be contained in one class file, allowing me to retain a single database connection.

My second and most effective optimization technique was to utilize batch inserts and set the values of the insert statements within the same run as the xml tag iteration. As soon as data values were gathered for one movie or star object, I would pass them into the prepared statement and execute the batch insert after each iteration of a movie or star tag.

As a result, the xml parsing now takes 11.71 seconds.