HW #2 Grading Sheet

Student ID: ____123____________   Student Name: ______Nick Noteworthy______________

Problem 1a (20 pts) - Structure of the MovieLens data.

Questions	**Pts**	**Comments**
Is the schema homogeneous? - Movies - *yes* - Ratings - *yes* - Tags - *yes* - Links - *yes*	4	Most everyone got these.
How are fields accessed (by name or position)? - All files - *by position*	2	CSV's are kind of both - but most said name-based (due to header, which was a perfectly good answer.
How are records delimited? - All files - *newline*	2	Most everyone got this.
How are fields delimited? - All files - *comma*	2	Ditto.
How are field values encoded? - All files - *plain text*	2	Ditto.
How complex (primitive values vs. arrays)? What about relationships (e.g., are files 1NF or not)?		
- Movies - all *primitive* except genres (which is an *array*)	2	Most everyone got these; a few failed to mention genres being a pipe-delimited list-valued field.
- Ratings - all *primitive*	1	
- Tags - all *primitive*	1	
- Links - all *primitive*	1	
What are the semantics of the data, and are validity checks possible? - Could check that the *links* to other movie databases are *valid* (similar to FK checks) - Could check *movie ids* when used as *FKs* - Could check *user ids* when used as *FKs* - Could check *sensibility of timestamp values* (e.g., in a year range after movies first appeared in the world)	3	There were a number of thoughts possible and given here; in grading we were mostly just looking for some attention being paid to this question, including things that could be wrong and be checked (or not).
Total	20	

Problem 1b (10 pts) - Granularity of the MovieLens data.

Questions	Pts	Score	Comments
What kind of thing (person, object, etc.) do the records represent?			Most everyone got these.
- Movies - *movies*	2		
- Ratings - *user movie ratings*	2		
- Tags - *user annotations*	2		
- Links - *cross-references to two other movie databases*	2		
Are the records homogeneous?			
- All files - *yes*	1		Most everyone got this.
Any alternative interpretations of the records?			Here we just wanted something to be said about this question (showing that it was thought about).
- *No fixed tag interpretation(s)*	1		
Total	10		

Problem 1c (20 pts) - Scope of the MovieLens data.

Questions	Pts	Score	Comments
What characteristics of the things are captured by the record fields?			Most everyone got this, though a few forgot to mention the timestamps (which were the rating and tag timestamps, not movie timestamps).
- Movies - *title, genre(s)*	4		
- Ratings - *user, rating, tstamp*	6		
- Tags - *tag value, tstamp*	4		
- Links - *remote movie ids*	2		
Are related fields consistent?	(0)		Here we looked at comments but didn't grade the commentary.
Can additional characteristics be deduced from the characteristics?	(0)		Ditto.
Do records all have the same fields?			
- All files - *yes*	1		Most everyone got this.
Do the records represent the entire populations of their things?			Here the sought-after point was that not every movie ever made, or perhaps every user, was in the data set.
- Subset of *all movies*	1		
Are there multiple records for the same thing (requiring de-dup'ing)?			
- All files - *no*	1		Here we just looked for some thoughts.
Any heterogeneous sets of records?			
- All files - *no*	1		Most everyone got this.
Total	20		

Problem 2 (50 pts) - Data wrangling.

Questions	**Pts**	**Score**	**Comments**
a. Misplaced value to extract? - Movie year - Could use string regex	5 5		Here we wanted both identifying the value (year) and mentioning how it might be extracted.
b. Lifting of one->several fields? - IMDB: primary name - Could separate: first + last	5 5		Here we were thinking of IMDB actor names, but we also took other answers that were “lifting-like” as valid answers.
c. Aggregation - ratings example? - Could aggregate ratings: *movieId,avgRating* 31,2.75 66,4.23 ...	5 5		Most folks got the idea here. We wanted both a written example and a data example of what it would look like.
d. Movies file problems/issues? - Non-1NF field (genres) should be normalized - Misplaced value (year) should be extracted *movieId,title,year:* 1,Toy Story,1995 ... *movieId,genre:* 1,Adventure 1,Animation ...	5 5		Here we wanted all of the issues to be mentioned - both the misplaced year and the fact that genres being non-1NF would be a pain for queries/analysis.
e. How could reviews be combined across IMDB and MovieLens? - Need to ***union*** the ratings to get them into one dataset. - Must first ***standardize*** their values (e.g., to a 1-10 scale, multiplying MovieLens values by 2 after averaging them by movie): *movieId,avgRating* 31,3.50 66,6.37 ...	4 2 2 2		ATTENTION: Most folks did NOT get this, at least not in the way we wanted. To deal with scores across databases, scores on different scales, some work is needed. One DB had averages, so first the other DB had to be averaged. Also, the score scales were different (by 2x), so one would need to equalize the scales. Then - and only then - one could “union” the results, as in, combine the scores TOGETHER to have a score that reflects both. Aggregation and normalization were the big asks here.
Total	50		

Total Score: __________ (= ______ + ______ + ______ + ______)