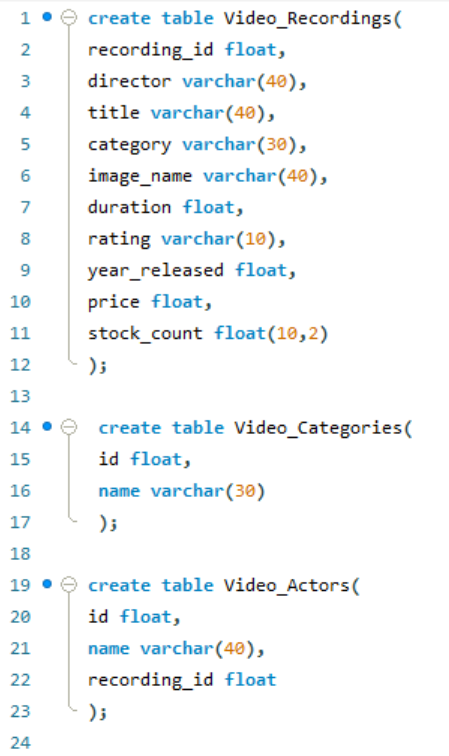
* Part I – Creating a Video Database

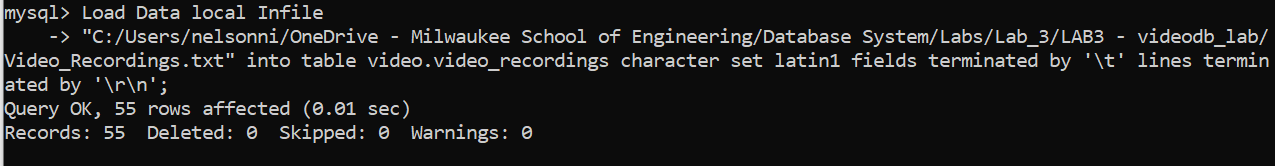
3.

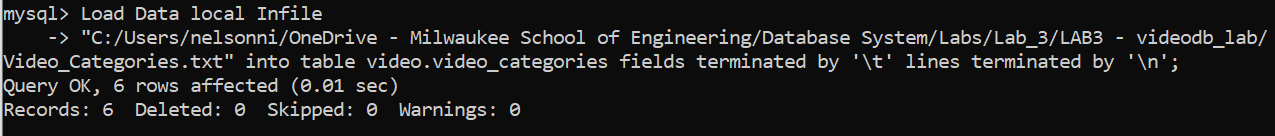


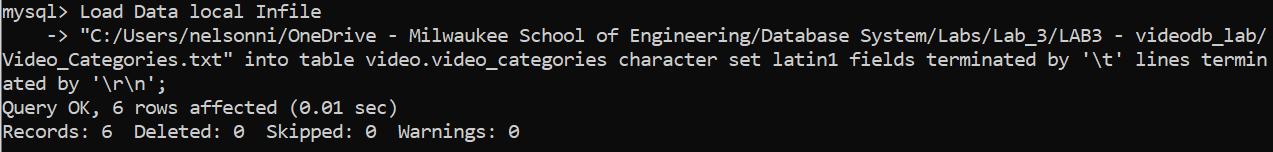
Update Video\_categories

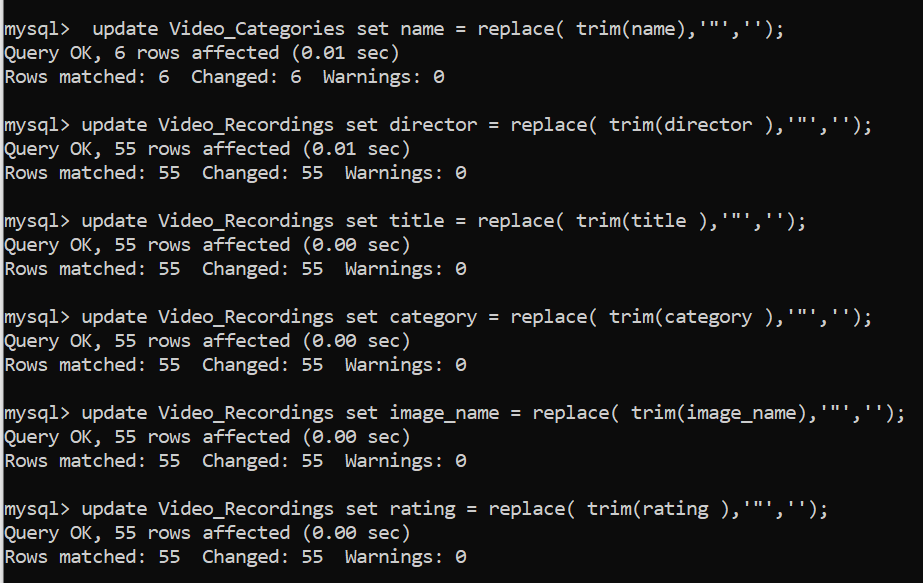
Sset SQL\_SAFE\_UPDATES = 0;

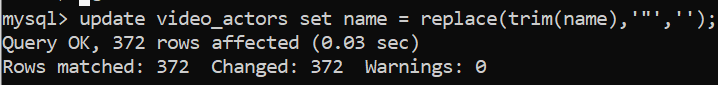
4.



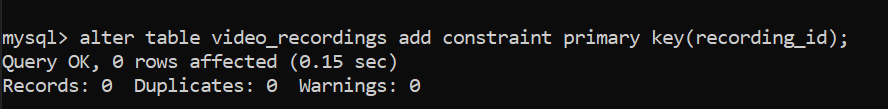


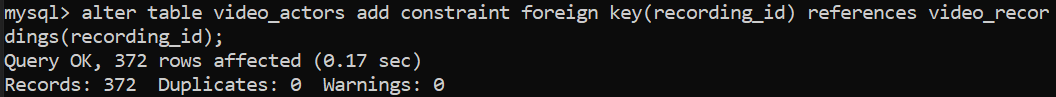


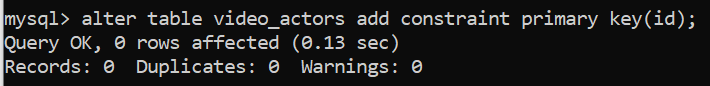


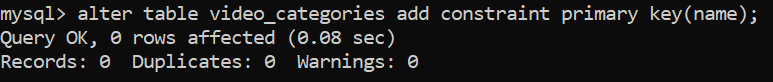


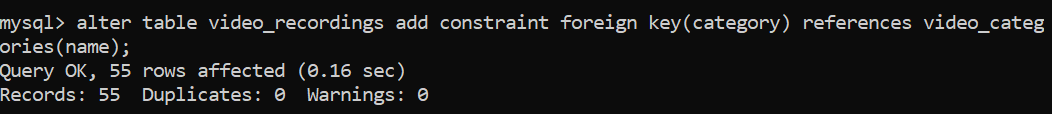
* Creating new tables
* Setting primary keys



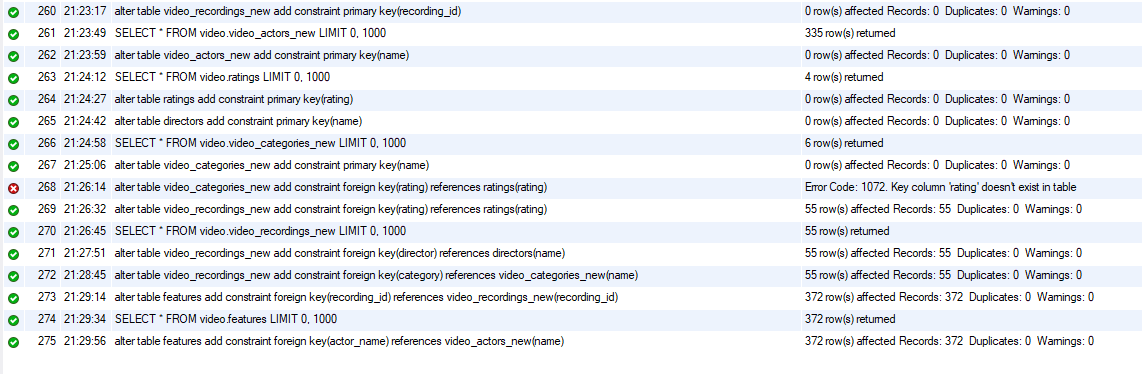


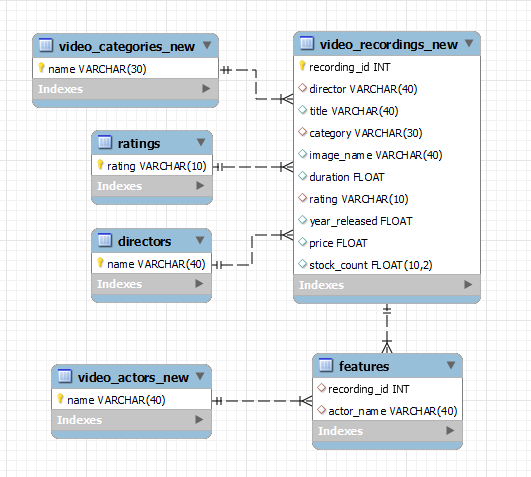






Part 1 revisited:



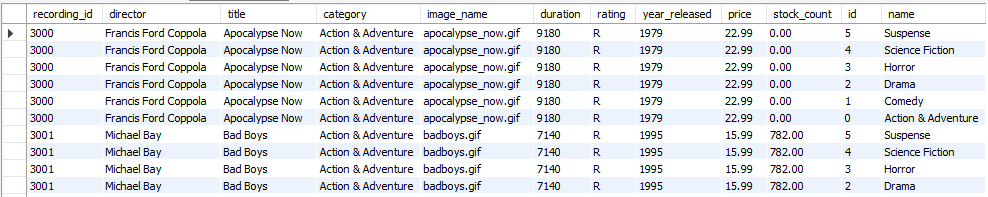


PART TWO:

1. Execute select \* from Video\_Recordings, Video\_Categories. Note the cross-product effect of joining two tables. Record the number of rows generated. Do all permutations of Video\_Recordings X Video\_Categories make sense? Explain.
   1. Query:



* 1. Results:



* 1. Analysis
     1. This query returned 330 rows for this given request. However, all of the returned permutations do not necessarily make much sense. The reason for this is that the query returned every possible combination of each video categories and each video recordings, resulting in video recordings being concatenated with mismatched video categories.

1. Execute select \* from Video\_Recordings vr, Video\_Categories vc where vr.category=vc.name. Note the cross-product effect of joining two tables when restricted on the appropriate keys. Record the number of rows generated. Explain the purpose of the join.
   1. Query:

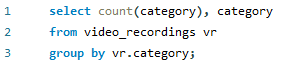


* 1. Results:

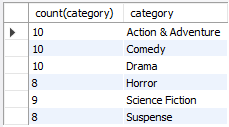


* 1. Analysis
     1. This query returned 55 rows for this given request. The purpose of this “join” is to ensure that the only combinations of video recordings and video categories appear where the video recording’s category matches the video category’s name. This corrects the randomness of the permutations combined by the last query, and ensures only matching permutations are returned.

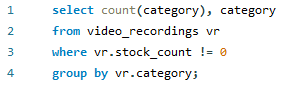
1. List the number of videos for each video category.
   1. Query:



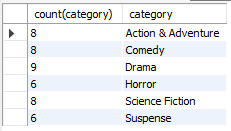
* 1. Results:



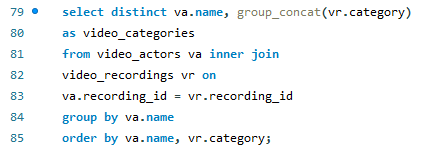
1. List the number of videos for each video category where the inventory is non-zero.
   1. Query:



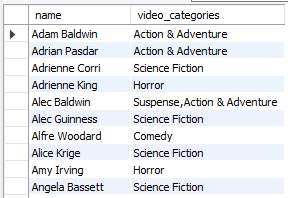
* 1. Results:



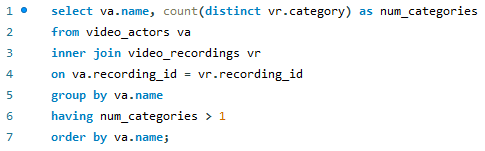
1. For each actor, list the video categories.
   1. Query:



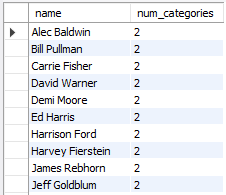
* 1. Results: **335 rows**



1. Which actors have appeared in movies in different video categories?
   1. Query:



* 1. Results: **25 rows**



1. Which actors have not appeared in a comedy?
   1. Query:



* 1. Results: **265 rows**



1. Which actors have appeared in comedy and action adventure movies?
   1. Query:



* 1. Results:

