My Redis database involved three keys with a hash and three keys with a list. The three hashes were a hash for each actor and their information, where each key was the “Actor\_[id]” and their details in the hash, each movie and its information, where the key is “Film\_[id]” and their details in the hash, and each category and its information, where each key was the “Category\_[id]” and their details in the hash. The three lists included lists of what movies each actor and the movies they are associated with, where the key is “actor\_[id]” and the movies in the list, each movie and what actors they are associated with, where the key is “film\_[id]” and the actors in the list, and each category and what movies they are associated with, where the key is “category\_[id]” and the movies in the list. I did this so that way when the functions ask for the details of an actor or film and asks for the movies or actors associated with them, I would be able to make a dictionary of the details of the actor or movie and then tag on a list of movies or actors associated with them.

There are several pros and cons to using Redis instead of a relational database. One of the pros is that because it is shallow and has only one layer of nesting, it is a lot quicker than a relational database, which can have a lot of hierarchy. This is good for small databases. The cons are that it seems that it can only handle a small amount of data before it starts to get messy. There isn’t a good equivalent of an associative entity in Redis. From what I’ve learned in this assignment is that Redis is good for something small, but relational databases are much better in my opinion because they have more functionality and can do more things and for me personally I can visualize the structure better than key-value databases like Redis.