

## Assignment 2

### SQL, Python, SQLite

Last updated: **Monday 29th Mar 05:47pm** (most recent updates are in [...])

**Due : Friday 16th April 17:00**

## Aims

This assignment aims to give you practice in

- use of SQL in SQLite (i.e., sqlite3)
- writing scripts in Python that interact with a database
- populating a RDBMS with a larger dataset, and analysing the data;
- making test data for testing database applications;

Your task is to complete the functionality of some command-line tools via a combination of database code and Python code.

## Summary

**Submission:** Login to Course Web Site > Assignments > Assignment 2 > Assignment 2 Specification > Make Submission > upload required files > [Submit]

**Required Files (total 3 files):** msearch, toprank, shortest

You may also submit the assignment via the give command from CSE machines: `give cs3311 a2 msearch toprank shortest`

**Deadline:** Friday 16 April 2021 @ 17:00

**Late Penalty:** Late submissions will have marks deducted from the maximum achievable mark at the rate of 2% of the total mark per hour that they are late (i.e., 48% per day).

This assignment contributes **20 marks** toward your total mark for this course.

**Downloads:** [a2.tgz](#), [a2.zip](#)

Note that `a2.tgz` and `a2.zip` contain the same material.

Each archive contains the IMDB database dump `a2.db`, plus a sample Python code file called `sample`.

### What To Do Now:

- **read this specification carefully and completely**
- login to a CSE linux machine
- create a directory for this assignment
- unpack the supplied zip file into this directory
- get familiar with the schema and data by exploring and querying the provided database using the command: `sqlite3 a2.db`
- familiarise (read the code) yourself with the provided sample Python code file called `sample`
- try out the sample code by running: `./sample YEAR` where `YEAR` is a number representing a year, e.g., 1989
- make sure `sample` is executable (by `chmod u+x sample`) and run it
- complete the assignment tasks using `sample` as a reference template
- submit all these files (the 'Required Files') via WebCMS3 or give as described above

Details of the above steps are further elaborated below. You can edit and run the Python files on any CSE machines.

## Introduction

A successful movie (including TV show) not only entertains audience, but also enables film companies to gain tremendous profit. A lot of factors (such as good directors, experienced actors, etc) are important for creating good movies. Nevertheless, famous directors and actors usually bring an attractive box-office income, but they do not necessarily guarantee a highly rated imdb score. This assignment is based on an IMDB dataset to build several small Python commands to show interesting results.

The dataset itself contains around 5000 movies, spanning across 100 years in 66 countries. There are more than 2000 movie directors, and thousands of actors/actresses. It also contains the IMDB rating score, numbers of votes and various facebook likes. To give you a feel for the kind of data that you are dealing with, a (unordered) glimpse of the dataset is included below:

```
cs3311@wagner:~/sqlite/a2$ sqlite3 a2.db
SQLite version 3.27.2 2019-02-25 16:06:06
Enter ".help" for usage hints.
sqlite>
sqlite> select * from movie limit 10;
1|Avatar|2009|PG-13|178|English|USA|760505847|237000000|735
2|Pirates of the Caribbean: At World's End|2007|PG-13|169|English|USA|309404152|300000000|555
3|Spectre|2015|PG-13|148|English|UK|200074175|245000000|1776
4|The Dark Knight Rises|2012|PG-13|164|English|USA|448130642|250000000|1078
6|John Carter|2012|PG-13|132|English|USA|73058679|263700000|2217
7|Spider-Man 3|2007|PG-13|156|English|USA|336530303|258000000|625
8|Tangled|2010|PG|100|English|USA|200807262|260000000|922
9|Avengers: Age of Ultron|2015|PG-13|141|English|USA|458991599|250000000|1410
10|Harry Potter and the Half-Blood Prince|2009|PG|153|English|UK|301956980|250000000|1117
11|Batman v Superman: Dawn of Justice|2016|PG-13|183|English|USA|330249062|250000000|2180
sqlite>
sqlite> .headers on
sqlite>
sqlite> select * from movie limit 10;
id|title|year|content_rating|duration|lang|country|gross|budget|director_id
1|Avatar|2009|PG-13|178|English|USA|760505847|237000000|735
2|Pirates of the Caribbean: At World's End|2007|PG-13|169|English|USA|309404152|300000000|555
3|Spectre|2015|PG-13|148|English|UK|200074175|245000000|1776
4|The Dark Knight Rises|2012|PG-13|164|English|USA|448130642|250000000|1078
6|John Carter|2012|PG-13|132|English|USA|73058679|263700000|2217
7|Spider-Man 3|2007|PG-13|156|English|USA|336530303|258000000|625
8|Tangled|2010|PG|100|English|USA|200807262|260000000|922
9|Avengers: Age of Ultron|2015|PG-13|141|English|USA|458991599|250000000|1410
10|Harry Potter and the Half-Blood Prince|2009|PG|153|English|UK|301956980|250000000|1117
```

```

11|Batman v Superman: Dawn of Justice|2016|PG-13|183|English|USA|330249062|250000000|2180
sqlite>
sqlite> select * from rating limit 10;
movie_id|num_critic_for_reviews|num_user_for_reviews|num_voted_users|movie_facebook_likes|cast_total_facebook_likes|i
1|723|3054|886204|33000|4834|7.9
2|302|1238|471220|0|48350|7.1
3|602|994|275868|85000|11700|6.8
4|813|2701|1144337|164000|106759|8.5
6|462|738|212204|24000|1873|6.6
7|392|1902|383056|0|46055|6.2
8|324|387|294810|29000|2036|7.8
9|635|1117|462669|118000|92000|7.5
10|375|973|321795|10000|58753|7.5
11|673|3018|371639|197000|24450|6.9
sqlite>
sqlite> .mode column
sqlite>
sqlite> select * from director limit 10;
id          name          facebook_likes
-----
1           JK Youn         2
2           David S. W     42
3           James Fraw     21
4           Kar-Wai Wo     0
5           Brian Tren     53
6           Perry Lang     17
7           Jeff Burr      155
8           Al Sillima     0
9           Morten Tyl     77
10          Hue Rhodes     0
sqlite>
sqlite> select * from actor limit 10;
id          name          facebook_likes
-----
1           Maureen McCormick 458
2           Andrew Fiscella   137000
3           Brittany Daniel   861
4           Michael Smiley    177
5           Najarra Townsend  538
6           Gustaf Skarsgrd   908
7           Laila Haley       1000
8           Isaac C. Singleto 312
9           Veronica Ferres   30000
10          Chris 'Wonder' Sc 0
sqlite>
sqlite> select * from acting limit 10;
movie_id    actor_id
-----
407         2024
3699        1841
3016        11
2846        195
3421        738
3645        1186
2430        211
4823        1299
1737        786
2282        866
sqlite>
sqlite> select * from genre limit 10;
movie_id    genre
-----
407         Adventure
407         Comedy
407         Family
3699        Drama
3699        Horror
3699        Mystery
3699        Sci-Fi
3699        Thriller
3016        Drama
2846        Comedy
sqlite>
sqlite> select * from keyword limit 10;
movie_id    keyword
-----
407         dog
407         parole
407         parole off
407         prison
407         puppy
3699        alien
3699        bunker
3699        car crash
3699        kidnapping
3699        minimal ca
sqlite>
sqlite> .width 10 30
sqlite>
sqlite> select * from keyword limit 10;
movie_id    keyword
-----

```

```

407      dog
407      parole
407      parole officer
407      prison
407      puppy
3699     alien
3699     bunker
3699     car crash
3699     kidnapping
3699     minimal cast
sqlite>

```

You may also explore the schema and any table information using the sqlite system catalog table called `sqlite_master`, or the sqlite PRAGMA functions (please see the sqlite documentation for details), as shown in the examples below:

```

cs3311@wagner:~/sqlite/a2$ sqlite3 a2.db
SQLite version 3.27.2 2019-02-25 16:06:06
Enter ".help" for usage hints.
sqlite> .tables
acting  actor    director  genre    keyword  movie    rating
sqlite> pragma table_info(movie);
0|id|integer|1||1
1|title|character varying(256)|1||0
2|year|integer|0||0
3|content_rating|character varying(9)|0||0
4|duration|integer|0||0
5|lang|character varying(10)|0||0
6|country|character varying(20)|0||0
7|gross|bigint|0||0
8|budget|bigint|0||0
9|director_id|integer|0||0
sqlite>
sqlite> select * from sqlite_master where name="movie";
table|movie|movie|6|CREATE TABLE movie (
      id integer NOT NULL,
      title character varying(256) NOT NULL,
      year integer,
      content_rating character varying(9),
      duration integer,
      lang character varying(10),
      country character varying(20),
      gross bigint,
      budget bigint,
      director_id integer,
      PRIMARY KEY (id),
      FOREIGN KEY (director_id) REFERENCES director(id)
)
sqlite>

```

### The sample Python program

By following the steps above, you can experience the provided sample program and study its code as follows:

```

cs3311@wagner:~/sqlite/a2$ ./sample
Usage: ./sample YEAR
cs3311@wagner:~/sqlite/a2$ ./sample 1989
A Nightmare on Elm Street 5: The Dream Child (1989)
Back to the Future Part II (1989)
Batman (1989)
Bill & Ted's Excellent Adventure (1989)
Black Rain (1989)
Born on the Fourth of July (1989)
Dead Poets Society (1989)
Do the Right Thing (1989)
Driving Miss Daisy (1989)
Friday the 13th Part VIII: Jason Takes Manhattan (1989)
Glory (1989)
Halloween 5 (1989)
Henry V (1989)
Indiana Jones and the Last Crusade (1989)
Licence to Kill (1989)
Major League (1989)
New York Stories (1989)
Pet Sematary (1989)
Road House (1989)
Roger & Me (1989)
Sea of Love (1989)
Sex, Lies, and Videotape (1989)
Star Trek V: The Final Frontier (1989)
Tango & Cash (1989)
The Abyss (1989)
The Blood of Heroes (1989)
The Toxic Avenger Part II (1989)
Troop Beverly Hills (1989)
UHF (1989)
Warlock (1989)
We're No Angels (1989)
When Harry Met Sally... (1989)
cs3311@wagner:~/sqlite/a2$ ls
a2.db  sample
cs3311@wagner:~/sqlite/a2$
cs3311@wagner:~/sqlite/a2$ cat sample

```

```
#!/usr/bin/python3

"""
This is a sample minimal program for COMP3311 21T1 Assignment 2 to illustrate how to write an
executable python code, take in a command line argument, and connect to a sqlite3 db.
For simplicity (easier to read and follow), it does not include error checking,
exception handling and comments.
"""

import sqlite3,sys

if len(sys.argv) != 2 :
    print("Usage:",sys.argv[0],"YEAR")
    sys.exit(1)

year = sys.argv[1]

con = sqlite3.connect('a2.db')

cur = con.cursor()

cur.execute('SELECT title, year FROM movie WHERE year ={} ORDER BY title'.format(year))

while True:
    t = cur.fetchone()
    if t == None:
        break
    x,y = t
    print('{} {}'.format(x,y))

con.close()
cs3311@wagner:~/sqlite/a2$
```

## Submission and Testing

We will auto-mark your submission on wagner as follows:

- create a testing subdirectory
- place a modified a2.db file (with the same schema but probably slightly modified data) into that directory
- place your submitted files in that directory and make them executable
- run a series of tests using your submitted msearch, toprank and shortest scripts
- manually inspect your submitted Python code (if time allows)

**Time Requirement:** in order to avoid indefinitely running scripts that block the marking process, any scripts that run for more than **2 minutes** for a particular task on wagner will be terminated and considered failed for that test.

Your submitted code must be *complete* so that when we do the above, your Python will work just as it did in your assignment directory and with a database with the identical schema (with either the same or slightly modified dataset) to yours (a2.db). For submission simplicity, please only submit the 3 required Python files. In other words, do NOT break a Python program into multiple Python files (so each submitted file should be a self-contained Python program). If your code does not work when installed for testing on wagner, as described above (for example, it works on your home machine perfectly, but we need to edit one line in order to have it running properly on a CSE machine; OR it does not pass the auto-marking tests due to extra debugging messages, etc), you will be penalised by an additional 30% administrative penalty. A testing script for formatting sanity check (based on the examples below) will be available in Week 8.

Before you submit, you should test out whether the files you submit will work by following a similar sequence of steps to those noted above on wagner (wagner.cse.unsw.edu.au).

## Tasks

For each task, you are required to implement an executable Python program command that takes in commandline arguments and displays the result in a specified format (which we will use for auto-marking). We assume the sample output format for each task below to test your programs. If any field of your output is empty, you should hide that field and its related formatting text (for example, in ./msearch for "star war" below, a comma and a space will not be printed for year if it is empty). Of course, you are free to print debugging information temporarily, but please remember to eventually disable or remove them when your solution is submitted for marking. For all the tasks below, unless it is specified explicitly, we assume 'movies' will include any titles stored in the movie table (i.e., including movies and TV shows etc).

For each task below, output nothing if there is no result returned from the database. If the task does not specify particular output order, your program can output in any row order. If multiple columns are involved in your output, your output should follow the same column order as the sample output presented in each task below.

### Task A: List movie information by its title, actor or director substrings (6 marks)

The msearch (movie search) script lists the movie title, year, content rating, IMDB score and genres of those movies with the title, actor, or director matching the given substring (case insensitive), one per line. It accepts any number of given substrings (but at least one). If more than one substring are provided, they should be considered as a conjunctive selection query (i.e., via AND). For example, consider

```
msearch SUBSTRING-1 SUBSTRING-2 SUBSTRING-3 ...
```

A movie will be in its output if each of SUBSTRING-1, SUBSTRING-2, SUBSTRING-3 ... appears in its title, actor or director.

The output of multiple genres of a movie should be concatenated in one line delimited by a comma and sorted alphabetically in ascending order, as shown by an example below. The rows are ordered by year (in descending order, rows with empty year at the end), then by IMDB rating (descending, rows with empty rating at the end if any) and finally by title (ascending). The required output format is elaborated further by the examples below:

```
wagner$ ./msearch "james fRAnco"
1. Sausage Party (2016, R, 7.5) [Adventure,Animation,Comedy,Fantasy]
2. The Little Prince (2015, PG, 7.8) [Adventure,Animation,Drama,Family,Fantasy]
```

```

3. The Interview (2014, R, 6.6) [Comedy]
4. This Is the End (2013, R, 6.7) [Comedy,Fantasy]
5. Homefront (2013, R, 6.5) [Action,Crime,Drama,Thriller]
6. Oz the Great and Powerful (2013, PG, 6.4) [Adventure,Family,Fantasy]
7. Palo Alto (2013, R, 6.3) [Drama]
8. The Iceman (2012, R, 6.9) [Biography,Crime,Drama]
9. Spring Breakers (2012, R, 5.3) [Crime,Drama]
10. Rise of the Planet of the Apes (2011, PG-13, 7.6) [Action,Drama,Sci-Fi,Thriller]
11. Your Highness (2011, R, 5.6) [Adventure,Comedy,Fantasy]
12. 127 Hours (2010, R, 7.6) [Adventure,Biography,Drama,Thriller]
13. Date Night (2010, PG-13, 6.3) [Comedy,Crime,Romance,Thriller]
14. Eat Pray Love (2010, PG-13, 5.7) [Drama,Romance]
15. Milk (2008, R, 7.6) [Biography,Drama,History]
16. Pineapple Express (2008, R, 7.0) [Action,Comedy,Crime]
17. In the Valley of Elah (2007, R, 7.2) [Crime,Drama,Mystery,Thriller]
18. Spider-Man 3 (2007, PG-13, 6.2) [Action,Adventure,Romance]
19. The Dead Girl (2006, R, 6.7) [Crime,Drama,Mystery]
20. Flyboys (2006, PG-13, 6.6) [Action,Adventure,Drama,History,Romance,War]
21. The Great Raid (2005, R, 6.7) [Action,Drama,War]
22. Spider-Man 2 (2004, PG-13, 7.3) [Action,Adventure,Fantasy,Romance]
23. Spider-Man (2002, PG-13, 7.3) [Action,Adventure,Fantasy,Romance]
24. City by the Sea (2002, R, 6.2) [Crime,Drama,Mystery,Thriller]
25. Deuces Wild (2002, R, 5.6) [Action,Crime,Drama]
26. Whatever It Takes (2000, PG-13, 5.5) [Comedy,Drama,Romance]
wagner$ ./msearch "JOHN smith"
wagner$ ./msearch "star WAR"
1. Star Wars: Episode III - Revenge of the Sith (2005, PG-13, 7.6) [Action,Adventure,Fantasy,Sci-Fi]
2. Star Wars: Episode II - Attack of the Clones (2002, PG, 6.7) [Action,Adventure,Fantasy,Sci-Fi]
3. Star Wars: Episode I - The Phantom Menace (1999, PG, 6.5) [Action,Adventure,Fantasy,Sci-Fi]
4. Star Wars: Episode VI - Return of the Jedi (1983, PG, 8.4) [Action,Adventure,Fantasy,Sci-Fi]
5. Star Wars: Episode V - The Empire Strikes Back (1980, PG, 8.8) [Action,Adventure,Fantasy,Sci-Fi]
6. Star Wars: Episode IV - A New Hope (1977, PG, 8.7) [Action,Adventure,Fantasy,Sci-Fi]
7. Star Wars: The Clone Wars (TV-PG, 7.9) [Action,Adventure,Animation,Drama,Fantasy,Sci-Fi]
wagner$

```

Please note the output formatting of the last movie above (No. 7), i.e., it does not have year. For all the tasks in this assignment, as mentioned previously, your program shall output nothing if there are no results. If there are N results, it shall output exactly N lines of output without any extra lines or messages. Therefore, we may also use the line count to measure the size of an output to determine its preliminary correctness. For example:

```

wagner$ ./msearch "james franco"|wc -l
26
wagner$ ./msearch "john smith" |wc -l
0
wagner$

```

More examples:

```

wagner$ ./msearch "happy"
1. Happy Christmas (2014, R, 5.6) [Comedy,Drama]
2. Another Happy Day (2011, R, 6.0) [Comedy,Drama]
3. Happy Feet 2 (2011, PG, 5.9) [Animation,Comedy,Family,Musical]
4. The Pursuit of Happyness (2006, PG-13, 8.0) [Biography,Drama]
5. Happy Feet (2006, PG, 6.5) [Animation,Comedy,Family,Music,Romance]
6. Happy, Texas (1999, PG-13, 6.3) [Comedy,Crime,Romance]
7. Happy Gilmore (1996, PG-13, 7.0) [Comedy,Sport]
8. Happy Valley (TV-MA, 8.5) [Crime,Drama]
wagner$ ./msearch "ALBert"
1. The Secret Life of Pets (2016, PG, 6.8) [Animation,Comedy,Family]
2. The Big Short (2015, R, 7.8) [Biography,Comedy,Drama,History]
3. Concussion (2015, PG-13, 7.1) [Biography,Drama,Sport]
4. A Most Violent Year (2014, R, 7.0) [Action,Crime,Drama,Thriller]
5. Baggage Claim (2013, PG-13, 5.0) [Comedy]
6. Skyfall (2012, PG-13, 7.8) [Action,Adventure,Thriller]
7. Drive (2011, R, 7.8) [Crime,Drama]
8. Albert Nobbs (2011, R, 6.7) [Drama]
9. The Book of Eli (2010, R, 6.9) [Action,Adventure,Drama,Thriller]
10. Shine a Light (2008, PG-13, 7.2) [Biography,Documentary,Music]
11. The Bourne Ultimatum (2007, PG-13, 8.1) [Action,Mystery,Thriller]
12. The Simpsons Movie (2007, PG-13, 7.4) [Adventure,Animation,Comedy]
13. Once in a Lifetime: The Extraordinary Story of the New York Cosmos (2006, PG-13, 7.3) [Documentary,Sport]
14. A Good Year (2006, PG-13, 6.9) [Comedy,Drama,Romance]
15. Before Sunset (2004, R, 8.0) [Drama,Romance]
16. A Very Long Engagement (2004, R, 7.7) [Drama,Mystery,Romance,War]
17. Fat Albert (2004, PG, 4.3) [Comedy,Family,Fantasy,Romance]
18. Big Fish (2003, PG-13, 8.0) [Adventure,Drama,Fantasy]
19. Irreversible (2002, Not Rated, 7.4) [Crime,Drama,Mystery,Thriller]
20. From Hell (2001, R, 6.8) [Horror,Mystery,Thriller]
21. Erin Brockovich (2000, R, 7.3) [Biography,Drama]
22. Ready to Rumble (2000, PG-13, 5.3) [Comedy,Sport]
23. The Muse (1999, PG-13, 5.6) [Comedy]
24. Breakfast of Champions (1999, R, 4.6) [Comedy]
25. Out of Sight (1998, R, 7.0) [Crime,Drama,Romance,Thriller]
26. 54 (1998, R, 5.8) [Drama,Music]
27. Critical Care (1997, R, 6.0) [Comedy,Drama]
28. The Brothers McMullen (1995, R, 6.6) [Comedy,Drama,Romance]
29. Menace II Society (1993, R, 7.5) [Crime,Drama,Thriller]
30. Twilight Zone: The Movie (1983, PG, 6.5) [Fantasy,Horror,Sci-Fi]
31. Dragonslayer (1981, PG, 6.7) [Action,Adventure,Fantasy]
32. Private Benjamin (1980, R, 6.1) [Comedy,War]
33. Taxi Driver (1976, R, 8.3) [Crime,Drama]
34. Willy Wonka & the Chocolate Factory (1971, G, 7.8) [Family,Fantasy,Musical]
35. The Party's Over (1965, 7.3) [Drama]
36. Tom Jones (1963, Unrated, 6.8) [Adventure,Comedy,History]
37. The Longest Day (1962, G, 7.8) [Action,Drama,History,War]

```

```
wagner$ ./msearch "albert" "david"
1. A Most Violent Year (2014, R, 7.0) [Action,Crime,Drama,Thriller]
2. Baggage Claim (2013, PG-13, 5.0) [Comedy]
3. The Simpsons Movie (2007, PG-13, 7.4) [Adventure,Animation,Comedy]
wagner$ ./msearch "albert" "david" "simpson"
1. The Simpsons Movie (2007, PG-13, 7.4) [Adventure,Animation,Comedy]
wagner$ ./msearch "tom hanks" "spielberg" "ryan"
1. Bridge of Spies (2015, PG-13, 7.6) [Drama,History,Thriller]
2. Saving Private Ryan (1998, R, 8.6) [Action,Drama,War]
wagner$ ./msearch "star war" "Natalie PORTMAN" "Lucas"
1. Star Wars: Episode III - Revenge of the Sith (2005, PG-13, 7.6) [Action,Adventure,Fantasy,Sci-Fi]
2. Star Wars: Episode II - Attack of the Clones (2002, PG, 6.7) [Action,Adventure,Fantasy,Sci-Fi]
3. Star Wars: Episode I - The Phantom Menace (1999, PG, 6.5) [Action,Adventure,Fantasy,Sci-Fi]
wagner$ ./msearch "star war" "Natalie PORTMAN" "Lucas" "revenge"
1. Star Wars: Episode III - Revenge of the Sith (2005, PG-13, 7.6) [Action,Adventure,Fantasy,Sci-Fi]
wagner$
```

### Task B: Top ranked movies (6 marks)

The `toprank` script ranks movies based on their IMDB rating. It considers movies with given genres and the minimum acceptable IMDB score to be listed in the output. It takes in 2 commandline arguments:

```
./toprank Genres MinRating
```

where Genres is a list of genres (based on case insensitive matching) separated by '&', MinRating is the minimum acceptable IMDB score. The output is ranked by IMDB score and then by the number of votes (both in descending order). We interpret '&' as conjunction, i.e., the selected movies shall contain all the specified genres. When Genres is an empty string, perform the same ranking but on movies with any genres. The required output format is elaborated further by the examples below:

```
wagner$ ./toprank "Action&Sci-Fi&Adventure"
Usage: ./toprank Genres MinRating
wagner$ ./toprank "Action&Sci-Fi&Adventure" 8.5
1. Inception (2010, PG-13, English) [8.8, 1468200]
2. Star Wars: Episode V - The Empire Strikes Back (1980, PG, English) [8.8, 837759]
3. Daredevil (TV-MA, English) [8.8, 213483]
4. Star Wars: Episode IV - A New Hope (1977, PG, English) [8.7, 911097]
wagner$ ./toprank "Sci-Fi&Adventure&Action" 8
1. Inception (2010, PG-13, English) [8.8, 1468200]
2. Star Wars: Episode V - The Empire Strikes Back (1980, PG, English) [8.8, 837759]
3. Daredevil (TV-MA, English) [8.8, 213483]
4. Star Wars: Episode IV - A New Hope (1977, PG, English) [8.7, 911097]
5. Star Wars: Episode VI - Return of the Jedi (1983, PG, English) [8.4, 681857]
6. Aliens (1986, R, English) [8.4, 488537]
7. Stargate SG-1 (TV-14, English) [8.4, 63982]
8. Captain America: Civil War (2016, PG-13, English) [8.2, 272670]
9. Godzilla Resurgence (2016, Japanese) [8.2, 374]
10. The Avengers (2012, PG-13, English) [8.1, 995415]
11. Guardians of the Galaxy (2014, PG-13, English) [8.1, 682155]
12. Mad Max: Fury Road (2015, R, English) [8.1, 552503]
13. Deadpool (2016, R, English) [8.1, 479047]
14. Destiny (2014, English) [8.1, 3089]
15. X-Men: Days of Future Past (2014, PG-13, English) [8.0, 514125]
16. Star Trek (2009, PG-13, English) [8.0, 504419]
17. Serenity (2005, PG-13, English) [8.0, 242599]
18. The Iron Giant (1999, PG, English) [8.0, 128455]
wagner$ ./toprank "Sci-Fi&Adventure&Action" 8.175
1. Inception (2010, PG-13, English) [8.8, 1468200]
2. Star Wars: Episode V - The Empire Strikes Back (1980, PG, English) [8.8, 837759]
3. Daredevil (TV-MA, English) [8.8, 213483]
4. Star Wars: Episode IV - A New Hope (1977, PG, English) [8.7, 911097]
5. Star Wars: Episode VI - Return of the Jedi (1983, PG, English) [8.4, 681857]
6. Aliens (1986, R, English) [8.4, 488537]
7. Stargate SG-1 (TV-14, English) [8.4, 63982]
8. Captain America: Civil War (2016, PG-13, English) [8.2, 272670]
9. Godzilla Resurgence (2016, Japanese) [8.2, 374]
wagner$ ./toprank "" 8.8
1. Towering Inferno (English) [9.5, 10]
2. The Shawshank Redemption (1994, R, English) [9.3, 1689764]
3. The Godfather (1972, R, English) [9.2, 1155770]
4. Dekalog (TV-MA, Polish) [9.1, 12590]
5. The Dark Knight (2008, PG-13, English) [9.0, 1676169]
6. The Godfather: Part II (1974, R, English) [9.0, 790926]
7. Fargo (TV-MA, English) [9.0, 170055]
8. Pulp Fiction (1994, R, English) [8.9, 1324680]
9. The Lord of the Rings: The Return of the King (2003, PG-13, English) [8.9, 1215718]
10. Schindler's List (1993, R, English) [8.9, 865020]
11. The Good, the Bad and the Ugly (1966, Approved, Italian) [8.9, 503509]
12. 12 Angry Men (1957, Not Rated, English) [8.9, 447785]
13. Inception (2010, PG-13, English) [8.8, 1468200]
14. Fight Club (1999, R, English) [8.8, 1347461]
15. Forrest Gump (1994, PG-13, English) [8.8, 1251222]
16. The Lord of the Rings: The Fellowship of the Ring (2001, PG-13, English) [8.8, 1238746]
17. Star Wars: Episode V - The Empire Strikes Back (1980, PG, English) [8.8, 837759]
18. Daredevil (TV-MA, English) [8.8, 213483]
19. It's Always Sunny in Philadelphia (TV-MA, English) [8.8, 133415]
wagner$
```

### Task C: Six degrees of Kevin Bacon (8 marks)



This task is inspired by the Six degrees of Kevin Bacon. The shortest script takes in two actor names (with case insensitive matching), and lists the shortest path (up to Six Degrees of Separation) between two given actors. In other words, if two actors are not connected within six degrees, you can assume that they are not connected at all (i.e. output nothing). The output will be a list of actors, the movies and the years, as illustrated in the examples below. If there are more than one 'shortest paths', output all of them (all output lines are sorted ASCII-betically in ascending order). Reference (Wikipedia): [Six Degrees of Kevin Bacon](#)

```
wagner$ ./shortest "tom cruise" "Jeremy Renner"
1. Tom Cruise was in Mission: Impossible - Ghost Protocol (2011) with Jeremy Renner
2. Tom Cruise was in Mission: Impossible - Rogue Nation (2015) with Jeremy Renner
wagner$ ./shortest "chris evans" "Scarlett Johansson"
1. Chris Evans was in Captain America: Civil War (2016) with Scarlett Johansson
2. Chris Evans was in Captain America: The Winter Soldier (2014) with Scarlett Johansson
wagner$ ./shortest "tom cruise" "Robert Downey Jr."
1. Tom Cruise was in Days of Thunder (1990) with Robert Duvall; Robert Duvall was in Lucky You (2007) with Robert Downey Jr.
2. Tom Cruise was in Days of Thunder (1990) with Robert Duvall; Robert Duvall was in The Judge (2014) with Robert Downey Jr.
3. Tom Cruise was in Jack Reacher (2012) with Robert Duvall; Robert Duvall was in Lucky You (2007) with Robert Downey Jr.
4. Tom Cruise was in Jack Reacher (2012) with Robert Duvall; Robert Duvall was in The Judge (2014) with Robert Downey Jr.
5. Tom Cruise was in Mission: Impossible (1996) with Kristin Scott Thomas; Kristin Scott Thomas was in Richard III (1995) with Robert Downey Jr.
6. Tom Cruise was in Mission: Impossible III (2006) with Eddie Marsan; Eddie Marsan was in Sherlock Holmes (2009) with Robert Downey Jr.
7. Tom Cruise was in Mission: Impossible III (2006) with Eddie Marsan; Eddie Marsan was in Sherlock Holmes: A Game of Shadows (2011) with Robert Downey Jr.
8. Tom Cruise was in The Firm (1993) with Holly Hunter; Holly Hunter was in Home for the Holidays (1995) with Robert Downey Jr.
wagner$ ./shortest "brad pitt" "will smith"
1. Brad Pitt was in Burn After Reading (2008) with Kevin Sussman; Kevin Sussman was in Hitch (2005) with Will Smith
2. Brad Pitt was in Ocean's Thirteen (2007) with Matt Damon; Matt Damon was in The Legend of Bagdad Elad (2000) with Will Smith
3. Brad Pitt was in True Romance (1993) with Michael Rapaport; Michael Rapaport was in Hitch (2005) with Will Smith
wagner$ ./shortest "chris evans" "bill clinton"
1. Chris Evans was in Captain America: Civil War (2016) with Robert Downey Jr.; Robert Downey Jr. was in Kiss Kiss Bang Bang (2005) with Bill Clinton
2. Chris Evans was in Captain America: Civil War (2016) with Robert Downey Jr.; Robert Downey Jr. was in Kiss Kiss Bang Bang (2005) with Bill Clinton
3. Chris Evans was in Captain America: Civil War (2016) with Robert Downey Jr.; Robert Downey Jr. was in Lucky You (2007) with Bill Clinton
4. Chris Evans was in Captain America: Civil War (2016) with Robert Downey Jr.; Robert Downey Jr. was in Lucky You (2007) with Bill Clinton
5. Chris Evans was in Captain America: Civil War (2016) with Robert Downey Jr.; Robert Downey Jr. was in The Judge (2014) with Bill Clinton
6. Chris Evans was in Captain America: Civil War (2016) with Robert Downey Jr.; Robert Downey Jr. was in The Judge (2014) with Bill Clinton
7. Chris Evans was in Captain America: Civil War (2016) with Robert Downey Jr.; Robert Downey Jr. was in The Soloist (2009) with Bill Clinton
8. Chris Evans was in Captain America: Civil War (2016) with Robert Downey Jr.; Robert Downey Jr. was in The Soloist (2009) with Bill Clinton
...
45. Chris Evans was in The Iceman (2012) with James Franco; James Franco was in City by the Sea (2002) with Robert Downey Jr.
46. Chris Evans was in The Iceman (2012) with James Franco; James Franco was in City by the Sea (2002) with Robert Downey Jr.
47. Chris Evans was in The Iceman (2012) with James Franco; James Franco was in In the Valley of Elah (2007) with Channing Tatum
48. Chris Evans was in The Iceman (2012) with James Franco; James Franco was in In the Valley of Elah (2007) with Channing Tatum
49. Chris Evans was in The Losers (2010) with Jason Patric; Jason Patric was in The Lost Boys (1987) with Dianne Wiest
50. Chris Evans was in The Losers (2010) with Jason Patric; Jason Patric was in The Lost Boys (1987) with Dianne Wiest
```