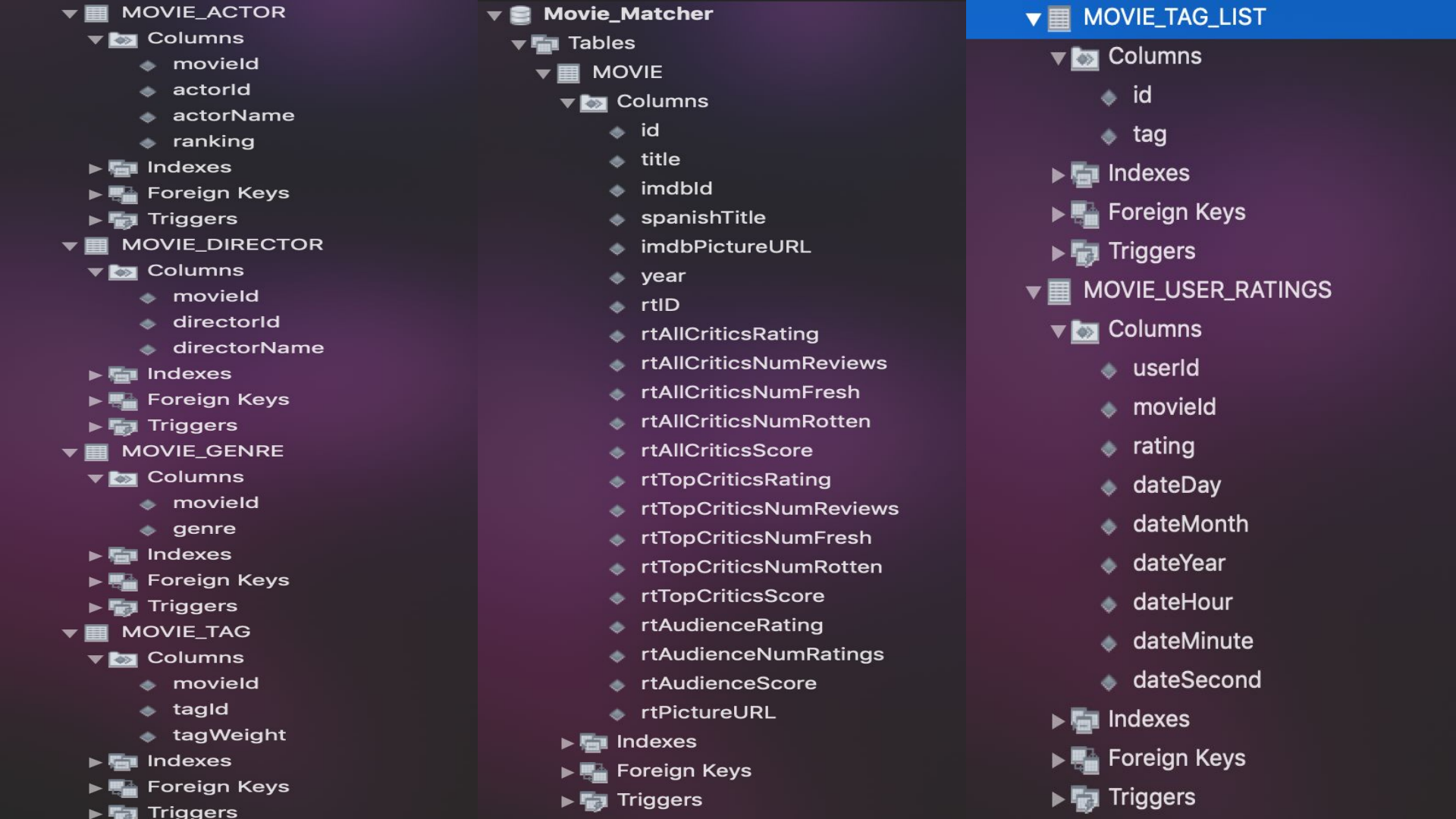
Movie Matcher

Andrew Russell, Sean Passmore,
Sean Northcutt, Caleb Pierce


MOVIE_ACTOR
Columns
movieId
actorId
actorName
ranking
Indexes
Foreign Keys
Triggers
MOVIE_DIRECTOR
Columns
movieId
directorId
directorName
Indexes
Foreign Keys
Triggers
MOVIE_GENRE
Columns
movieId
genre
Indexes
Foreign Keys
Triggers
MOVIE_TAG
Columns
movieId
tagId
tagWeight
Indexes
Foreign Keys
Triggers
Movie_Matcher
Tables
MOVIE
Columns
id
title
imdbId
spanishTitle
imdbPictureURL
year
rtID
rtAllCriticsRating
rtAllCriticsNumReviews
rtAllCriticsNumFresh
rtAllCriticsNumRotten
rtAllCriticsScore
rtTopCriticsRating
rtTopCriticsNumReviews
rtTopCriticsNumFresh
rtTopCriticsNumRotten
rtTopCriticsScore
rtAudienceRating
rtAudienceNumRatings
rtAudienceScore
rtPictureURL
Indexes
Foreign Keys
Triggers
MOVIE_TAG_LIST
Columns
id
tag
Indexes
Foreign Keys
Triggers
MOVIE_USER_RATINGS
Columns
userId
movieId
rating
dateDay
dateMonth
dateYear
dateHour
dateMinute
dateSecond
Indexes
Foreign Keys
Triggers

Query 1: See Top popular K movies
 This was a simple query that let selected the title, score, year, and ID for a movie and ordered it by the score and by the number of reviews.
It used the slider to get the K for the query defaulted to 10.

```
SELECT id, title, year, rtAllCriticsNumReviews, rtAllCriticsScore FROM movie_matcher.movie
      group by title
      order by rtAllCriticsScore DESC, rtAllCriticsNumReviews DESC
      LIMIT K
```

A)
```
SELECT id, title, year, rtAllCriticsNumReviews, rtAllCriticsScore FROM movie_matcher.movie
      group by title
      order by rtAllCriticsScore DESC, rtAllCriticsNumReviews DESC
      LIMIT 15
```
Shows the top 15 movies

B)
```
SELECT id, title, year, rtAllCriticsNumReviews, rtAllCriticsScore FROM movie_matcher.movie
      group by title
      order by rtAllCriticsScore DESC, rtAllCriticsNumReviews DESC
      LIMIT 100
```
Shows the top 100 movies

Query 2: Search by movie title

A simple search query where we innerjoined the tag list sp that you could view the tags associated with the movie. It used substring pattern matching to show all movies that contained or was the movie in the search box.

```
SELECT movie.title,year,rtAudienceScore, GROUP_CONCAT(movie_tag_list.tag) AS Tags
                FROM movie
                INNER JOIN movie_tag
                ON movie.id = movie_tag.movieId
                INNER JOIN movie_tag_list
                ON movie_tag.tagId = movie_tag_list.id
                WHERE movie.title LIKE ?
                GROUP BY movie.title
                order by rtAllCriticsScore
```

A) Shows all movies with Alien in the title

```
SELECT movie.title,year,rtAudienceScore, GROUP_CONCAT(movie_tag_list.tag) AS Tags
                FROM movie
                INNER JOIN movie_tag
                ON movie.id = movie_tag.movieId
                INNER JOIN movie_tag_list
                ON movie_tag.tagId = movie_tag_list.id
                WHERE movie.title LIKE "%Alien%"
                GROUP BY movie.title
                order by rtAllCriticsScore
```

B)Shows all movies with cowboy in the title

```
SELECT movie.title,year,rtAudienceScore, GROUP_CONCAT(movie_tag_list.tag) AS Tags
                FROM movie
                INNER JOIN movie_tag
                ON movie.id = movie_tag.movieId
                INNER JOIN movie_tag_list
                ON movie_tag.tagId = movie_tag_list.id
                WHERE movie.title LIKE "%Cowboy%"
                GROUP BY movie.title
                order by rtAllCriticsScore
```

Query 3: Search by genre
This query selected the top K movies in the specified genre X.
It Inner joined the movie genre table to make the genres visible

```
SELECT id, title,year, rtAllCriticsScore,rtAudienceScore
        FROM movie_matcher.movie
        INNER JOIN movie_matcher.movie_genre
        ON movie.id = movie_genre.movieID
        WHERE movie_genre.genre = X
        GROUP BY movie.title
        ORDER BY rtAllCriticsScore DESC, rtAllCriticsNumReviews DESC
        LIMIT K
```

A) Shows the top 25 Adventure movies

```
SELECT id, title,year, rtAllCriticsScore,rtAudienceScore
        FROM movie_matcher.movie
        INNER JOIN movie_matcher.movie_genre
        ON movie.id = movie_genre.movieID
        WHERE movie_genre.genre = "Adventure"
        GROUP BY movie.title
        ORDER BY rtAllCriticsScore DESC, rtAllCriticsNumReviews DESC
        LIMIT 25
```

B)Shows the top 50 horror movies

```
SELECT id, title,year, rtAllCriticsScore,rtAudienceScore
        FROM movie_matcher.movie
        INNER JOIN movie_matcher.movie_genre
        ON movie.id = movie_genre.movieID
        WHERE movie_genre.genre = "Horror"
        GROUP BY movie.title
        ORDER BY rtAllCriticsScore DESC, rtAllCriticsNumReviews DESC
        LIMIT 50
```

Query 4: Search by Director
Using substring pattern matching it shows the directors that match what was in the search query

```
SELECT directorName, movie.title,year,rtAudienceScore
                FROM movie_director
                INNER JOIN movie
                ON movie.id = movie_director.movieId
                WHERE directorName LIKE K
```

A)

```
SELECT directorName, movie.title,year,rtAudienceScore
                FROM movie_director
                INNER JOIN movie
                ON movie.id = movie_director.movieId
                WHERE directorName LIKE "%Martin%"
```

Shows the directors with Martin in their name

B)

```
SELECT directorName, movie.title,year,rtAudienceScore
                FROM movie_director
                INNER JOIN movie
                ON movie.id = movie_director.movieId
                WHERE directorName LIKE "%James Cameron%"
```

Shows the directors with James Cameron in their name

```
Query 5: Search by Actor
Searches by actor using substring pattern matching and shows their name and what movie they were in
SELECT actorName, movie.title,year,rtAudienceScore
        FROM movie_actor
        INNER JOIN movie
        ON movie.id = movie_actor.movieId
        WHERE actorName LIKE K

A)SHows actors with Martin in their name
SELECT actorName, movie.title,year,rtAudienceScore
        FROM movie_actor
        INNER JOIN movie
        ON movie.id = movie_actor.movieId
        WHERE actorName LIKE "%Martin%"

B) Shows actors that have angelina in their name
SELECT actorName, movie.title,year,rtAudienceScore
        FROM movie_actor
        INNER JOIN movie
        ON movie.id = movie_actor.movieId
        WHERE actorName LIKE "%Angelina%"
```

Query 6: Search by Tag
This query inner joined the tag and tag list tables to show movies that had that tag and used sub string pattern matching to show ones that contained it too.

```
SELECT movie_tag_list.tag, movie.title,year,rtAudienceScore
               FROM movie
               INNER JOIN movie_tag
               ON movie.id = movie_tag.movieId
               INNER JOIN movie_tag_list
               ON movie_tag.tagId = movie_tag_list.id
               WHERE tag LIKE K
               order by rtAllCriticsScore

A) shows all films that contain the nudity tag
SELECT movie_tag_list.tag, movie.title,year,rtAudienceScore
               FROM movie
               INNER JOIN movie_tag
               ON movie.id = movie_tag.movieId
               INNER JOIN movie_tag_list
               ON movie_tag.tagId = movie_tag_list.id
               WHERE tag LIKE "%nudity%"
               order by rtAllCriticsScore
B) Shows all films that contain the 70mm tag
SELECT movie_tag_list.tag, movie.title,year,rtAudienceScore
               FROM movie
               INNER JOIN movie_tag
               ON movie.id = movie_tag.movieId
               INNER JOIN movie_tag_list
               ON movie_tag.tagId = movie_tag_list.id
               WHERE tag LIKE "%70mm%"
               order by rtAllCriticsScore
```

Query 7: See Top 10 Directors in K movies
This query inner joined the movie table so that we could sort the results by highest audience score.
But essentially it takes the slider value and compares it to the count of movies that the director directed
and if it is greater than K it is added to the query with a hard Limit of 10 directors.

```
SELECT count(movieId)as Num_Movies, directorName as Name, rtAudienceScore
        from movie_matcher.movie_director
        INNER JOIN movie
        ON movie.id = movie_director.movieID
        group by directorID
        having count(movie_director.movieId) >= K
        ORDER BY rtAudienceScore DESC, rtAudienceNumRatings DESC, count(movieID) DESC
        LIMIT 10;
```

A)Shows the top 10 directors that directed 25 movies

```
SELECT count(movieId)as Num_Movies, directorName as Name, rtAudienceScore
        from movie_matcher.movie_director
        INNER JOIN movie
        ON movie.id = movie_director.movieID
        group by directorID
        having count(movie_director.movieId) >= 25
        ORDER BY rtAudienceScore DESC, rtAudienceNumRatings DESC, count(movieID) DESC
        LIMIT 10;
```

B) Shows the top 10 directors that directed 15 movies

```
SELECT count(movieId)as Num_Movies, directorName as Name, rtAudienceScore
        from movie_matcher.movie_director
        INNER JOIN movie
        ON movie.id = movie_director.movieID
        group by directorID
        having count(movie_director.movieId) >= 15
        ORDER BY rtAudienceScore DESC, rtAudienceNumRatings DESC, count(movieID) DESC
        LIMIT 10;
```

Query 8: See Top 10 Actors in K movies
This query inner joined the movie table so that we could sort the results by highest audience score.
But essentially it takes the slider value and compares it to the count of movies that the actor is in
and if it is greater than K it is added to the query with a hard Limit of 10 actors.

```
        SELECT count(movieId)as Num_Movies, actorName as Name, rtAudienceScore
        from movie_matcher.movie_actor
        INNER JOIN movie
        ON movie.id = movie_actor.movieID
        group by actorID
        having count(movie_actor.movieId) >= K
        ORDER BY rtAudienceScore DESC, rtAudienceNumRatings DESC, count(movieID) DESC
        LIMIT 10;
```

A)

```
SELECT count(movieId)as Num_Movies, actorName as Name, rtAudienceScore
        from movie_matcher.movie_actor
        INNER JOIN movie
        ON movie.id = movie_actor.movieID
        group by actorID
        having count(movie_actor.movieId) >= 20
        ORDER BY rtAudienceScore DESC, rtAudienceNumRatings DESC, count(movieID) DESC
        LIMIT 10;
```

SHows the top 10 actors that appeared in 20 movies

B)

```
SELECT count(movieId)as Num_Movies, actorName as Name, rtAudienceScore
        from movie_matcher.movie_actor
        INNER JOIN movie
        ON movie.id = movie_actor.movieID
        group by actorID
        having count(movie_actor.movieId) >= 50
        ORDER BY rtAudienceScore DESC, rtAudienceNumRatings DESC, count(movieID) DESC
        LIMIT 10;
```

SHows the top 10 actors that appeared in 50 movies

Query 9: See time line and graph for a user
	Using the user ID provided in the search box this one does two queries one for the graph and one for the timeline.
The timeline one is a simple list of all movies they have reviewed and orders it by Date nothing too special about it.
The Pie Chart one was a bit more involved. It selects the distinct genres so there is only 1 result per genre and it inner joins a subquery that takes the count of each genre and limits it by the user ID.

```
Select concat(dateMonth,'/', dateDay,'/', dateYear,'  at  ', dateHour,':', dateMinute,':', dateSecond) as Date, movie.title, rating
	From movie_user_ratings
	inner join movie
	on movie.id = movie_user_ratings.movieId
	where userId = K
	group by title
	order by Date ASC

SELECT distinct S.genre, C.cnt
	FROM movie_genre  S
	INNER JOIN (SELECT genre, count(genre) as cnt
	FROM movie_genre as C
	INNER JOIN movie_user_ratings
	ON c.movieID = movie_user_ratings.movieId
	where userId = ? GROUP BY genre) C ON S.genre = C.genre
```

A) Shows the timeline of movie reviews for USer ID 75

```
Select concat(dateMonth,'/', dateDay,'/', dateYear,'  at  ', dateHour,':', dateMinute,':', dateSecond) as Date, movie.title, rating
	From movie_user_ratings
	inner join movie
	on movie.id = movie_user_ratings.movieId
	where userId = 75
	group by title
	order by Date ASC
```

B) Shows the timeline of user reviews for user ID 170

```
Select concat(dateMonth,'/', dateDay,'/', dateYear,'  at  ', dateHour,':', dateMinute,':', dateSecond) as Date, movie.title, rating
	From movie_user_ratings
	inner join movie
	on movie.id = movie_user_ratings.movieId
	where userId = 170
	group by title
	order by Date ASC
```

C) Shows the Genre and the amount watched for user ID 75

```
SELECT distinct S.genre, C.cnt
	FROM movie_genre  S
	INNER JOIN (SELECT genre, count(genre) as cnt
	FROM movie_genre as C
	INNER JOIN movie_user_ratings
	ON c.movieID = movie_user_ratings.movieId
	where userId = 75 GROUP BY genre) C ON S.genre = C.genre
```

D)Shows the Genre and the amount watched for user ID 170

```
SELECT distinct S.genre, C.cnt
	FROM movie_genre  S
	INNER JOIN (SELECT genre, count(genre) as cnt
	FROM movie_genre as C
	INNER JOIN movie_user_ratings
	ON c.movieID = movie_user_ratings.movieId
	where userId = 170 GROUP BY genre) C ON S.genre = C.genre
```

Query 10: See movies containing the same tags
This one was hard and we had to limit it to EXACT movie titles. This takes the movie in the search box and shows all of its tags ordered alphabetically just like the tag query. It limits it by the subquery where all movies that contain any of the specified movies tags. It goes 1 step further and from the tags that those movies have in common it only shows the ones that have at least all the tags in common.

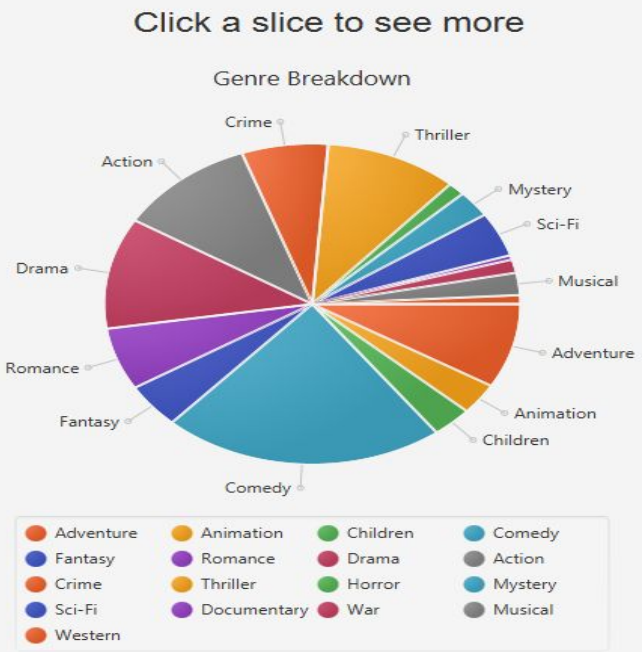
```
SELECT movieId,title, GROUP_CONCAT(movie_tag_list.tag ORDER BY tag ASC separator ', ') as Tags
       FROM MOVIE_TAG
       INNER JOIN Movie
       ON movie.id = movie_tag.movieId
       INNER JOIN movie_tag_list
       ON movie_tag_list.id = tagId
       WHERE movie_tag.tagId in ( SELECT tagId from MOVIE_TAG
       INNER JOIN Movie
       ON movie.id = movie_tag.movieId
       WHERE movie.title = ?)
       GROUP BY movieId
       having count(distinct MOVIE_TAG.tagId) >= (SELECT COUNT(tagId) from MOVIE_TAG
              INNER JOIN Movie
       ON movie.id = movie_tag.movieId
       WHERE movie.title = ?)

A) Shows all movies that have all the tags of Hellraiser: Bloodline
SELECT movieId,title, GROUP_CONCAT(movie_tag_list.tag ORDER BY tag ASC separator ', ') as Tags
       FROM MOVIE_TAG
       INNER JOIN Movie
       ON movie.id = movie_tag.movieId
       INNER JOIN movie_tag_list
       ON movie_tag_list.id = tagId
       WHERE movie_tag.tagId in ( SELECT tagId from MOVIE_TAG
       INNER JOIN Movie
       ON movie.id = movie_tag.movieId
       WHERE movie.title = "Hellraiser: Bloodline")
       GROUP BY movieId
       having count(distinct MOVIE_TAG.tagId) >= (SELECT COUNT(tagId) from MOVIE_TAG
              INNER JOIN Movie
       ON movie.id = movie_tag.movieId
       WHERE movie.title = "Hellraiser: Bloodline")

B)Shows all the movies that have all the tags of Aliens
SELECT movieId,title, GROUP_CONCAT(movie_tag_list.tag ORDER BY tag ASC separator ', ') as Tags
       FROM MOVIE_TAG
       INNER JOIN Movie
       ON movie.id = movie_tag.movieId
       INNER JOIN movie_tag_list
       ON movie_tag_list.id = tagId
       WHERE movie_tag.tagId in ( SELECT tagId from MOVIE_TAG
       INNER JOIN Movie
       ON movie.id = movie_tag.movieId
       WHERE movie.title = "Aliens")
       GROUP BY movieId
       having count(distinct MOVIE_TAG.tagId) >= (SELECT COUNT(tagId) from MOVIE_TAG
              INNER JOIN Movie
       ON movie.id = movie_tag.movieId
       WHERE movie.title = "Aliens")
```

GUI

Movie Matcher

Home | Movie | User

User ID: 170



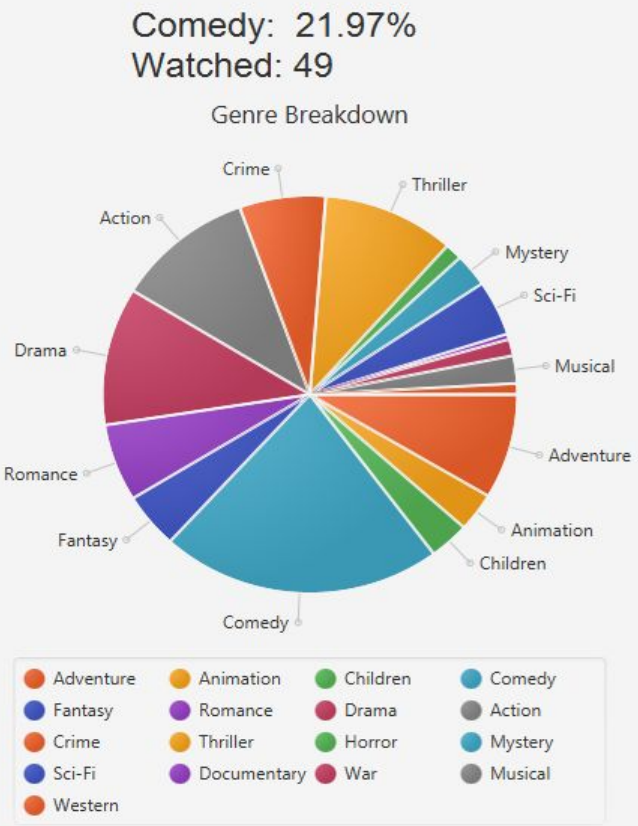
Date	title	rating
10/30/2006 at 12:24:18	Top Gun	2.0
10/30/2006 at 12:24:28	Back to the Future Part III	3.0
10/30/2006 at 12:24:32	Beetle Juice	0.5
10/30/2006 at 12:24:55	Happy Gilmore	4.0
10/30/2006 at 12:24:59	Life of Brian	4.0
10/30/2006 at 12:24:6	The Blues Brothers	3.0
10/30/2006 at 12:25:26	Beverly Hills Cop III	2.5
10/30/2006 at 12:25:42	Almost Famous	2.0
10/30/2006 at 12:25:49	Liar Liar	2.0
10/30/2006 at 12:25:54	South Park: Bigger Longer & Uncut	3.5
10/30/2006 at 12:25:8	What's Eating Gilbert Grape	3.5
10/30/2006 at 12:26:26	City Slickers II: The Legend of Curly's Gold	2.5
10/30/2006 at 12:26:29	Grease	2.0
10/30/2006 at 12:26:37	Mission: Impossible	2.5
10/30/2006 at 12:27:24	Chasing Amy	3.5
10/30/2006 at 12:29:21	American History X	4.5
10/30/2006 at 12:30:31	American Beauty	3.5
10/30/2006 at 12:31:27	Fight Club	5.0
10/30/2006 at 12:31:41	The Great Escape	1.0
10/30/2006 at 12:31:7	Donnie Darko	5.0
10/30/2006 at 12:33:19	Memento	4.0
10/30/2006 at 12:33:30	Monty Python and the Holy Grail	3.5
10/30/2006 at 12:35:21	Pulp Fiction	4.5
10/30/2006 at 12:35:28	Forrest Gump	3.0

Movie Matcher

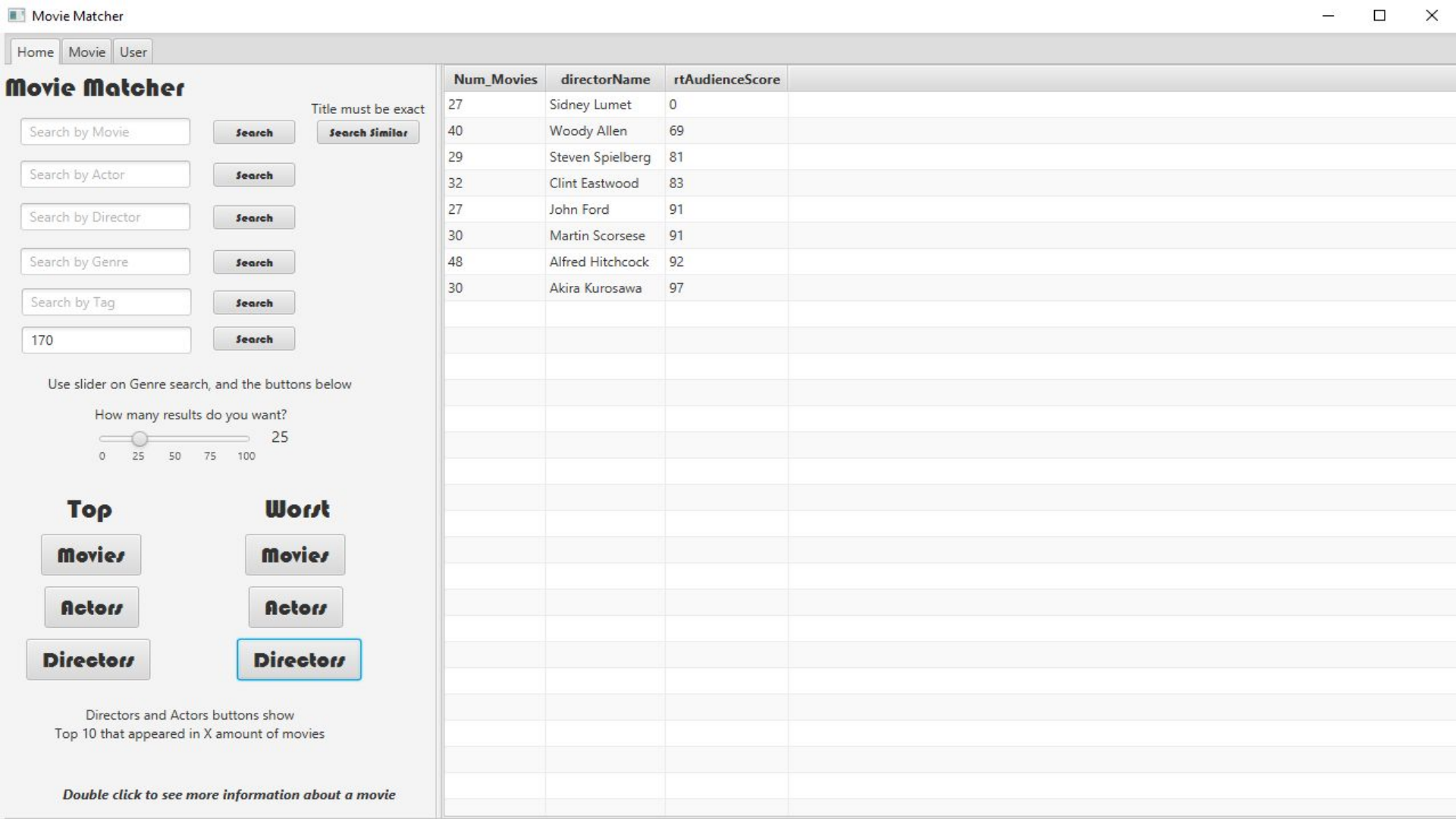
Home | Movie | User

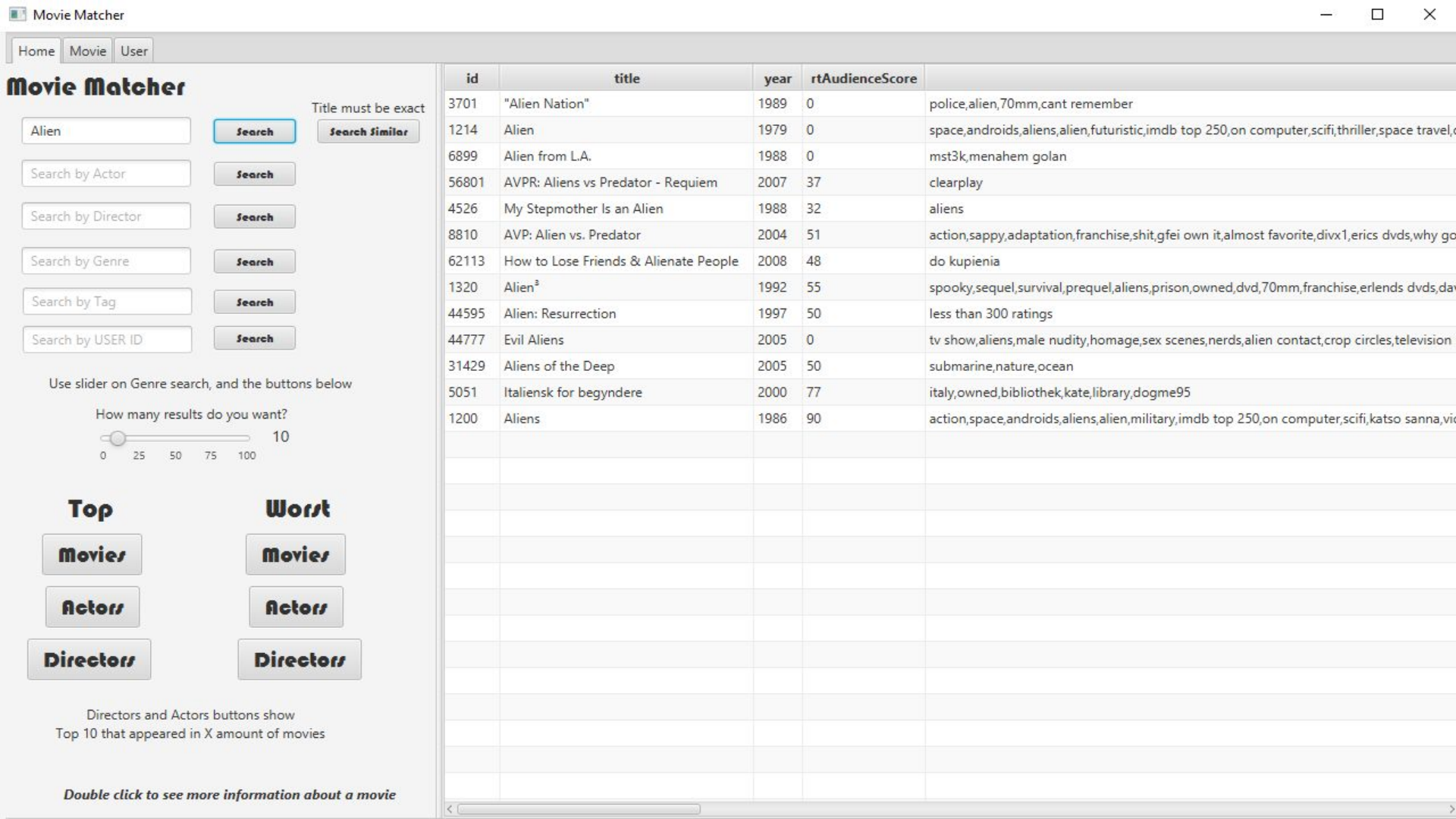
User ID: 170

Comedy: 21.97%
Watched: 49



Date	title	rating
10/30/2006 at 12:24:18	Top Gun	2.0
10/30/2006 at 12:24:28	Back to the Future Part III	3.0
10/30/2006 at 12:24:32	Beetle Juice	0.5
10/30/2006 at 12:24:55	Happy Gilmore	4.0
10/30/2006 at 12:24:59	Life of Brian	4.0
10/30/2006 at 12:24:6	The Blues Brothers	3.0
10/30/2006 at 12:25:26	Beverly Hills Cop III	2.5
10/30/2006 at 12:25:42	Almost Famous	2.0
10/30/2006 at 12:25:49	Liar Liar	2.0
10/30/2006 at 12:25:54	South Park: Bigger Longer & Uncut	3.5
10/30/2006 at 12:25:8	What's Eating Gilbert Grape	3.5
10/30/2006 at 12:26:26	City Slickers II: The Legend of Curly's Gold	2.5
10/30/2006 at 12:26:29	Grease	2.0
10/30/2006 at 12:26:37	Mission: Impossible	2.5
10/30/2006 at 12:27:24	Chasing Amy	3.5
10/30/2006 at 12:29:21	American History X	4.5
10/30/2006 at 12:30:31	American Beauty	3.5
10/30/2006 at 12:31:27	Fight Club	5.0
10/30/2006 at 12:31:41	The Great Escape	1.0
10/30/2006 at 12:31:7	Donnie Darko	5.0
10/30/2006 at 12:33:19	Memento	4.0
10/30/2006 at 12:33:30	Monty Python and the Holy Grail	3.5
10/30/2006 at 12:35:21	Pulp Fiction	4.5
10/30/2006 at 12:35:28	Forrest Gump	3.0


Movie Matcher
Home
Movie
User
Movie Matcher
Title must be exact
Search by Movie
Search
Search Similar
Search by Actor
Search
Search by Director
Search
Search by Genre
Search
Search by Tag
Search
170
Search
Use slider on Genre search, and the buttons below
How many results do you want?
25
0
25
50
75
100
Top
Movies
Actors
Directors
Worst
Movies
Actors
Directors
Directors and Actors buttons show
Top 10 that appeared in X amount of movies
Double click to see more information about a movie
Num_Movies
directorName
rtAudienceScore
27 Sidney Lumet 0
40 Woody Allen 69
29 Steven Spielberg 81
32 Clint Eastwood 83
27 John Ford 91
30 Martin Scorsese 91
48 Alfred Hitchcock 92
30 Akira Kurosawa 97


Movie Matcher
Home
Movie
User
Movie Matcher
Title must be exact
Alien
Search
Search Similar
Search by Actor
Search
Search by Director
Search
Search by Genre
Search
Search by Tag
Search
Search by USER ID
Search
Use slider on Genre search, and the buttons below
How many results do you want?
10
0
25
50
75
100
Top
Movies
Actors
Directors
Worst
Movies
Actors
Directors
Directors and Actors buttons show
Top 10 that appeared in X amount of movies
Double click to see more information about a movie
id
title
year
rtAudienceScore
3701 "Alien Nation" 1989 0 police,alien,70mm,cant remember
1214 Alien 1979 0 space,androids,aliens,alien,futuristic,imdb top 250,on computer,scifi,thriller,space travel,
6899 Alien from L.A. 1988 0 mst3k,menahem golan
56801 AVPR: Aliens vs Predator - Requiem 2007 37 clearplay
4526 My Stepmother Is an Alien 1988 32 aliens
8810 AVP: Alien vs. Predator 2004 51 action,sappy,adaptation,franchise,shit,gfei own it,almost favorite,divx1,erics dvds,why go
62113 How to Lose Friends & Alienate People 2008 48 do kupienia
1320 Alien³ 1992 55 spooky,sequel,survival,prequel,aliens,prison,owned,dvd,70mm,franchise,erlends dvds,da
44595 Alien: Resurrection 1997 50 less than 300 ratings
44777 Evil Aliens 2005 0 tv show,aliens,male nudity,homage,sex scenes,nerds,alien contact,crop circles,television
31429 Aliens of the Deep 2005 50 submarine,nature,ocean
5051 Italiensk for begyndere 2000 77 italy,owned,bibliothek,kate,library,dogme95
1200 Aliens 1986 90 action,space,androids,aliens,alien,military,imdb top 250,on computer,scifi,katso sanna,vi

Workload

Andrew Russell - Responsible for the GUI and SQL Statement 9

Sean Passmore - Responsible for Big Data import/processing and SQL Statement 10

Sean Northcutt - Responsible for statements 1 - 4

Caleb Pierce - Responsible for statements 5 - 8

Learning Outcome

We all learned a great deal about manipulating and querying data with SQL during this project. It was much easier to jump right in and have things come together in a project rather than small demonstrations.

We also all learned more about Git and Github together as well as learning how to be a contributing member of a remote team.