Peter Tadrous

CUS 725 - Advanced Database Systems

<u>Week 8 - Homework 7</u>

MongoDB

1. Returning a list of all genre names.
    a. `db.movies.distinct('genres')`

```
> db.movies.distinct("genres")
< [ 'Action',
    'Adventure',
    'Animation',
    'Biography',
    'Comedy',
    'Crime',
    'Documentary',
    'Drama',
    'Family',
    'Fantasy',
    'Film-Noir',
    'History',
    'Horror',
    'Music',
    'Musical',
    'Mystery',
    'News',
    'Romance',
    'Sci-Fi',
    'Short',
    'Sport',
    'Talk-Show',
    'Thriller',
    'War',
    'Western' ]
```

2. Text index to find movies that contain the word “Kentucky” in either the title, full plot, or cast.
   a. `db.movies.createIndex({title: "text", fullplot: "text", cast: "text", genres: "text"})`
   b. `db.movies.find({$text: {$search: "Kentucky"}}, {'_id':0,'title':1, 'cast':1, 'fullplot':1})`

```
> db.movies.createIndex({
    title: "text",
    fullplot: "text",
    cast: "text",
    genres: "text"})
< { createdCollectionAutomatically: false,
    numIndexesBefore: 1,
    numIndexesAfter: 2,
    ok: 1 }
> db.movies.find({$text: {$search: "Kentucky"}}, {'_id':0,'title':1, 'cast':1, 'fullplot':1})
< { fullplot: 'As youths in Azusa, Vinnie, Carter, and Rosie pull off a racing scam, substituting winners
  ey set him up for blackmail. Jump ahead twenty years, Carter and Rosie are married, successful racers i
  innie decides to make a play for Rosie, lures Carter to California, steals his wallet and heads for Ken
  k named Cecilia, to follow Vinnie and get the stuff back that he has in a box. Will she succeed?',
    title: 'Simpatico',
    cast:
     [ 'Nick Nolte',
       'Jeff Bridges',
       'Sharon Stone',
       'Catherine Keener' ] }
  { cast:
     [ 'Bruce Greenwood',
       'Hayden Panettiere',
       'Caspar Poyck',
       'Gary Bullock' ],
    title: 'Racing Stripes',
    fullplot: 'In the middle of a raging thunderstorm, a traveling circus accidentally leaves behind some
  lan Walsh, who takes him home to his young daughter Channing. Once a champion thoroughbred trainer, Wal
```

3. Return the title, year, and directors fields for movies directed by anyone whose first name is "John".
    a. `db.movies.find({'directors': /^John/},{'_id':0, 'title': 1, 'directors':1, 'year':1})`

```
> db.movies.find({'directors': /^John/},{'_id':0, 'title': 1, 'directors':1, 'year':1})
< { title: 'Wild and Woolly',
    directors: [ 'John Emerson' ],
    year: 1917 }
  { title: 'Our Hospitality',
    directors: [ 'John G. Blystone', 'Buster Keaton' ],
    year: 1923 }
  { title: 'The Iron Horse',
    directors: [ 'John Ford' ],
    year: 1924 }
  { title: 'Upstream', directors: [ 'John Ford' ], year: 1927 }
  { title: 'Four Sons', directors: [ 'John Ford' ], year: 1928 }
  { title: 'King of Jazz',
    directors: [ 'John Murray Anderson', 'Pèl Fejès' ],
    year: 1930 }
  { title: 'Men Without Women',
    directors: [ 'John Ford' ],
    year: 1930 }
  { title: 'Imitation of Life',
```

4. There were 969 movies made in 2008.
    a. `db.movies.count({'year':2008})`

```
> db.movies.count({'year':2008})
< 969
```

5. Return the average number of mflix comments, grouped by year.
   a. `db.movies.aggregate([{$group: {_id: '$year', avg_mflix_comments: {$avg: '$num_mflix_comments' }}}, {$sort: {_id:-1}}])`

```
> db.movies.aggregate(
    [
      {$group: {_id: '$year', avg_mflix_comments: { $avg: '$num_mflix_comments' } } },
      {$sort: {_id: -1}}
    ] )
< [ { _id: '2015è', avg_mflix_comments: 5 },
    { _id: '2014è', avg_mflix_comments: 1 },
    { _id: '2012è', avg_mflix_comments: 1.5 },
    { _id: '2011è', avg_mflix_comments: 1 },
    { _id: '2010è', avg_mflix_comments: 1 },
    { _id: '2009è', avg_mflix_comments: 1 },
    { _id: '2007è', avg_mflix_comments: 3 },
    { _id: '2006è2012', avg_mflix_comments: 2 },
    { _id: '2006è2007', avg_mflix_comments: 2 },
    { _id: '2006è', avg_mflix_comments: 1 },
    { _id: '2005è', avg_mflix_comments: 1.5 },
    { _id: '2003è', avg_mflix_comments: 2 },
    { _id: '2002è', avg_mflix_comments: 1 },
    { _id: '2000è', avg_mflix_comments: 1 },
    { _id: '1999è', avg_mflix_comments: 2 },
    { _id: '1997è', avg_mflix_comments: 1.5 },
    { _id: '1996è', avg_mflix_comments: 2 },
    { _id: '1995è', avg_mflix_comments: 1 },
    { _id: '1994è1998', avg_mflix_comments: null },
    { _id: '1988è', avg_mflix_comments: 2 } ]
> it
< { _id: '1987è', avg_mflix_comments: 1 }
  { _id: '1986è', avg_mflix_comments: 1 }
  { _id: '1981è', avg_mflix_comments: 1 }
  { _id: 2016, avg_mflix_comments: 5 }
```

6. A pipeline on the movies collection that will return movies made after 2010 and their comments.
    a. `var pipeline = [ {$match: {'year': {$gt: 2010}}}, {$lookup: { from: 'comments', localField: '_id', foreignField: 'movie_id', as: 'comments' }}]`
    b. `db.movies.aggregate(pipeline)`

```
> var pipeline = [
    {$match: {'year': {$gt: 2010}}},
    {$lookup: {
      from: 'comments',
      localField: '_id',
      foreignField: 'movie_id',
      as: 'comments' }}
    ]
> db.movies.aggregate(pipeline)
< [ { _id: ObjectId("573a13abf29313caabd23f34"),
      fullplot: 'Shaken by the death of his father and discouraged by his stalled career, writer Sal Paradise goes on a road trip hopin
  arismatic and fearless Dean Moriarty and Moriarty\'s free-spirited and seductive young wife, Marylou. Traveling across the American s
  d and search the unknown, and their decisions change the very course of their lives.',
      imdb: { rating: 6.1, votes: 30483, id: 337692 },
      year: 2012,
      plot: 'Young writer Sal Paradise has his life shaken by the arrival of free-spirited Dean Moriarty and his girl, Marylou. As they
  who each impact their journey indelibly.',
      genres: [ 'Adventure', 'Drama' ],
      rated: 'R',
      title: 'On the Road',
      lastupdated: '2015-08-18 00:09:34.297000000',
      languages: [ 'English', 'French' ],
      writers:
       [ 'Jack Kerouac (based on the novel by)',
```

```
      directors: [ 'Walter Salles' ],
      runtime: 124,
      comments:
       [ { _id: ObjectId("5a9427658b0beebeb696de30"),
           name: 'Justin Williams',
           email: 'justin_williams@fakegmail.com',
           movie_id: ObjectId("573a13abf29313caabd23f34"),
           text: 'Asperiores hic vel totam quia. Occaecati voluptates labore deserunt simil
           date: 2004-08-11T03:06:55.000Z } ] },
```

7. Adding a project to the previous pipeline so that the results just contain the movie title and the array of comments.
   a. `var pipeline = [ {$match: {'year': {$gt: 2010}}}, {$lookup: { from: 'comments', localField: '_id', foreignField: 'movie_id', as: 'comments' }} , {$project: {'_id':0,'title':1,'comments':1}} ]`
   b. `db.movies.aggregate(pipeline)`

```
> var pipeline = [
    {$match: {'year': {$gt: 2010}}},
    {$lookup: {
      from: 'comments',
      localField: '_id',
      foreignField: 'movie_id',
      as: 'comments' }},
    {$project: {'_id':0,'title':1,'comments':1}}
    ]
> db.movies.aggregate(pipeline)
< [ { title: 'On the Road',
      comments:
       [ { _id: ObjectId("5a9427658b0beebeb696de30"),
           name: 'Justin Williams',
           email: 'justin_williams@fakegmail.com',
           movie_id: ObjectId("573a13abf29313caabd23f34"),
           text: 'Asperiores hic vel totam quia. Occaecati voluptates labore
           date: 2004-08-11T03:06:55.000Z } ] },
    { title: 'The Secret Life of Walter Mitty', comments: [] },
    { title: 'Jurassic World', comments: [] },
    { title: 'The Rum Diary', comments: [] },
    { title: 'Gnomeo & Juliet', comments: [] },
    { title: 'The Three Stooges', comments: [] },
    { title: 'The Crimson Petal and the White',
      comments:
       [ { _id: ObjectId("5a9427658b0beebeb696f80f"),
           name: 'Grey Worm',
           email: 'jacob_anderson@gameofthron.es',
           movie_id: ObjectId("573a13aef29313caabd2edcd"),
           text: 'Quisquam dignissimos maxime eaque iusto maiores. Omnis rem
```

8. Adding a second match stage to the previous pipeline so that all results have comments.
   a. `var pipeline = [ {$match: {'year': {$gt: 2010}}}, {$lookup: { from: 'comments', localField: '_id', foreignField: 'movie_id', as: 'comments' }} , {$project: {'_id':0,'title':1,'comments':1}}, {$match: {'comments': {$not: {$size: 0}}}}]`
   b. `db.movies.aggregate(pipeline)`

```
> var pipeline = [
    {$match: {'year': {$gt: 2010}}},
    {$lookup: {
      from: 'comments',
      localField: '_id',
      foreignField: 'movie_id',
      as: 'comments' }},
    {$project: {'_id':0,'title':1,'comments':1}},
    {$match: {'comments': {$not: {$size: 0}}}}
    ]
> db.movies.aggregate(pipeline)
< [ { title: 'On the Road',
      comments:
       [ { _id: ObjectId("5a9427658b0beebeb696de30"),
           name: 'Justin Williams',
           email: 'justin_williams@fakegmail.com',
           movie_id: ObjectId("573a13abf29313caabd23f34"),
           text: 'Asperiores hic vel totam quia. Occaecati voluptates lab
           date: 2004-08-11T03:06:55.000Z } ] },
    { title: 'The Crimson Petal and the White',
      comments:
       [ { _id: ObjectId("5a9427658b0beebeb696f80f"),
           name: 'Grey Worm',
           email: 'jacob_anderson@gameofthron.es',
           movie_id: ObjectId("573a13aef29313caabd2edcd"),
```

9. Connect MongoDB to python.

mongotest.py

Provided Files > mongotest.py > MongoClient

```python
from pymongo import MongoClient
client = MongoClient(host="localhost", port=27017)

print("The databases are")
print(client.list_database_names())

db = client.week7
print("The collections are")
print(db.list_collection_names())
coll = db.inventory2
for doc in coll.find():
    print(doc)

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\peter\Documents\Programming\Spring2021\CUS 725 - Advanced Database\08 - Week 8\Homework 7> conda activa
PS C:\Users\peter\Documents\Programming\Spring2021\CUS 725 - Advanced Database\08 - Week 8\Homework 7> & C:/Users/p
/Documents/Programming/Spring2021/CUS 725 - Advanced Database/08 - Week 8/Homework 7/Provided Files/mongotest.py"
The databases are
['admin', 'config', 'local', 'week7', 'week8']
The collections are
['zips']
PS C:\Users\peter\Documents\Programming\Spring2021\CUS 725 - Advanced Database\08 - Week 8\Homework 7>
```