Project 3 - The Mongo Mash

CECS 323 - Sec 05

Lorenzo Murillo IV, #028112355

Joakim Eckerman, #028731311

California State University, Long Beach

College of Engineering

Schema's:

Note: Theater Schema was not changed and will not be included.

Showings:

```
{
  $jsonSchema: {
    title: 'showing',
    description: 'A showing of a film at a theater.',
    bsonType: 'object',
    required: [
      'theaterId',
      'filmId',
      'showTime',
      'format',
      'tickets',
      'starring'
    ],
    properties: {
      _id: {
        bsonType: 'objectId'
      },
      theaterId: {
        bsonType: 'objectId'
      },
      roomTitle: {
        bsonType: 'string'
      },
      filmId: {
        bsonType: 'objectId'
      },
      filmTitle: {
        bsonType: 'string'
      },
      filmRating: {
        bsonType: 'string',
        'enum': [
          'NR',
          'G',
          'PG',
          'PG-13',
          'R',
          'NC-17'
        ]
      },
      showTime: {
        bsonType: 'date'
      },
```

```
      format: {
        bsonType: 'string'
      },
      tickets: {
        bsonType: 'array',
        items: {
          title: 'ticket',
          description: 'a ticket sold for a showing',
          bsonType: 'object',
          required: [
            '_id',
            'price',
            'seatNumber'
          ],
          properties: {
            _id: {
              bsonType: 'objectId'
            },
            price: {
              bsonType: 'int'
            },
            seatNumber: {
              bsonType: 'string'
            }
          }
        }
      },
      starring: {
        bsonType: 'array',
        minItems: 1,
        maxItems: 3,
        items: {
          bsonType: 'string'
        }
      }
    }
  }
}
```

Films:

```
{
  $jsonSchema: {
    title: 'film',
    description: 'A film shown at theaters.',
    bsonType: 'object',
    required: [
      'title',
```

```
      'duration',
      'releaseDate',
      'rating',
      'genres'
    ],
    properties: {
      _id: {
        bsonType: 'objectId'
      },
      title: {
        bsonType: 'string'
      },
      duration: {
        bsonType: 'int',
        minimum: 1
      },
      releaseDate: {
        bsonType: 'date'
      },
      rating: {
        bsonType: 'string',
        'enum': [
          'NR',
          'G',
          'PG',
          'PG-13',
          'R',
          'NC-17'
        ]
      },
      genres: {
        bsonType: 'array',
        minItems: 1,
        items: {
          bsonType: 'string'
        }
      },
      actors: {
        bsonType: 'array',
        minItems: 1,
        items: {
          bsonType: 'string'
        }
      }
    }
  }
}
```

Actors:

```
{
  $jsonSchema: {
    title: 'actors',
    description: 'A star in a film',
    bsonType: 'object',
    required: [
      'firstName',
      'lastName',
      'birthDate',
      'url',
      'films'
    ],
    properties: {
      _id: {
        bsonType: 'objectId'
      },
      firstName: {
        bsonType: 'string'
      },
      lastName: {
        bsonType: 'string'
      },
      birthDate: {
        bsonType: 'date'
      },
      url: {
        bsonType: 'string'
      },
      films: {
        bsonType: 'array',
        items: {
          bsonType: 'string'
        }
      }
    }
  }
}
```

Written Justification:

While a film may have a multitude of actors and actresses, sometimes they will need to be considered on an individual level. There is going to be some finite amount or actors listed, for example, a film cannot have every single known actor and actress. The edge case is too broad and not realistic. Additionally, actors can have a range of films that they have starred in from one to many, but when considering many it is still some countable value. After researching a star's appearance in films, it seems the most is around 1000. This is an extreme edge case and while this is a lot of films to star in we would associate a films array within the actor collection. This would allow for a database like IMDB to access information based on a given actor or actress. Utilizing this method we can find the actor or film and their associations appropriately.

We will also need to implement an array of Actors within the Films collection in order to properly capture the enterprise. For example, if we want to inquire about actors within a film we would want to denaturalize the actors of this film into the film collection. In terms of showings we can also include some type of denaturalization for actors, but it can be smaller than the other two previously described. If we were to store actors in showings we can store the leads of the films. This would allow us to keep a smaller array of data but still introduce some redundancy, but only enough for a rational purpose.

Storing all of the data of actors within all collections except for theaters allows for the enterprise to have access to actors data without needing to use joins. We are using denaturalization within reason for a faster querying process. Ultimately it depends on the design of the enterprise and in my instance I'm emulating something similar to IMDB, however similar movie sites use similar methods. For example, Fandango allows you to search by actor or actress, if you wish, and can then see their filmography and then search for theaters and showings near you. However that is a roundabout way, it should still be considered.

JSON Documents:

Theaters:


```
[{
  "_id": {
    "$oid": "63891758aa3bcdcc2f7f6289"
  },
  "name": "Regal Edwards Long Beach",
  "address": "7501 E. Carson St.",
  "city": "Long Beach",
  "state": "California",
  "zipcode": "90808",
  "rooms": [
    {
      "title": "Screen 1",
      "capacity": 5,
      "formats": [
        "Standard",
        "IMAX"
      ],
      "seats": [
        {
          "number": "1",
          "labels": [
            "reclining"
          ]
        },
        {
          "number": "2",
          "labels": [
            "reclining"
          ]
        },
        {
          "number": "3",
          "labels": [
            "reclining"
          ]
        },
        {
          "number": "4",
          "labels": [
            "reclining"
          ]
        },
        {
          "number": "5",
          "labels": [
```

```
            "accessible seating",
            "non-reclining"
          ]
        }
      ],
      "description": "Screen 1 seating 5 guests"
    },
    {
      "title": "Screen 2",
      "capacity": 1,
      "formats": [
        "IMAX",
        "ScreenX"
      ],
      "seats": [
        {
          "number": "1"
        }
      ],
      "description": "Screen 2 seating 1 guest"
    }
  ]
}]
```


Showings:


```
[{
  "_id": {
    "$oid": "638da7cc6cdb3591cea96917"
  },
  "theaterId": {
    "$oid": "63891758aa3bcdcc2f7f6289"
  },
  "filmId": {
    "$oid": "63891c65aa3bcdcc2f7f628e"
  },
  "showTime": {
    "$date": {
      "$numberLong": "1668095100000"
    }
  },
  "format": "Standard",
  "tickets": [
    {
      "_id": {
        "$oid": "3c4f626a656374496428293e"
      },
      "price": 18,
```

```
      "seatNumber": "1"
    }
  ],
  "starring": [
    "Letitia Wright",
    "Lupita Nyong'o",
    "Danai Gurira"
  ]
},{
  "_id": {
    "$oid": "638da82d6cdb3591cea96918"
  },
  "theaterId": {
    "$oid": "63891758aa3bcdcc2f7f6289"
  },
  "filmId": {
    "$oid": "63891c65aa3bcdcc2f7f628e"
  },
  "showTime": {
    "$date": {
      "$numberLong": "1668106800000"
    }
  },
  "format": "IMAX",
  "tickets": [
    {
      "_id": {
        "$oid": "3c4f626a656374496428293e"
      },
      "price": 22,
      "seatNumber": "1"
    },
    {
      "_id": {
        "$oid": "3c4f626a656374496428293e"
      },
      "price": 15,
      "seatNumber": "5"
    }
  ],
  "starring": [
    "Letitia Wright",
    "Lupita Nyong'o",
    "Danai Gurira"
  ]
},{
  "_id": {
```

```
    "$oid": "638da9266cdb3591cea96919"
  },
  "theaterId": {
    "$oid": "63891758aa3bcdcc2f7f6289"
  },
  "filmId": {
    "$oid": "63891c65aa3bcdcc2f7f628e"
  },
  "showTime": {
    "$date": {
      "$numberLong": "1668258000000"
    }
  },
  "format": "Standard",
  "tickets": [],
  "starring": [
    "Letitia Wright",
    "Lupita Nyong'o",
    "Danai Gurira"
  ]
}]
```


Films:


```
[{
  "_id": {
    "$oid": "63891703aa3bcdcc2f7f6287"
  },
  "title": "No Time To Die",
  "duration": 163,
  "releaseDate": {
    "$date": {
      "$numberLong": "1633676400000"
    }
  },
  "rating": "PG-13",
  "genres": [
    "Action",
    "Adventure",
    "Thriller"
  ],
  "actors": [
    "Daniel Craig",
    "Ana de Armas",
    "Rami Malek"
  ]
},{
```

```
  "_id": {
    "$oid": "63891c30aa3bcdcc2f7f628d"
  },
  "title": "Akeelah and The Bee",
  "duration": 112,
  "releaseDate": {
    "$date": {
      "$numberLong": "1142496000000"
    }
  },
  "rating": "PG",
  "genres": [
    "Family",
    "Drama"
  ],
  "actors": [
    "Angela Bassett",
    "Laurence Fishburne",
    "Keke Palmer"
  ]
},{
  "_id": {
    "$oid": "63891c65aa3bcdcc2f7f628e"
  },
  "title": "Black Panther: Wakanda Forever",
  "duration": 161,
  "releaseDate": {
    "$date": {
      "$numberLong": "1668067200000"
    }
  },
  "rating": "PG-13",
  "genres": [
    "Action",
    "Adventure",
    "Superhero"
  ],
  "artists": [
    "Letitia Wright",
    "Lupita Nyong'o",
    "Danai Gurira",
    "Winston Duke",
    "Angela Bassett"
  ]
}]
```

Actors:


```
[{
  "_id": {
    "$oid": "63915f186cdb3591cea96928"
  },
  "firstName": "Letitia",
  "lastName": "Wright",
  "birthDate": {
    "$date": {
      "$numberLong": "752050800000"
    }
  },
  "url":
"https://www.imdb.com/name/nm4004793/?ref_=nv_sr_srsg_0",
  "films": [
    "Black Panther",
    "Black Panther: Wakanda Forever",
    "Small Axe",
    "The Silent Twins",
    "Aisha"
  ]
},{
  "_id": {
    "$oid": "639161026cdb3591cea96929"
  },
  "firstName": "Lupita",
  "lastName": "Nyong'o",
  "birthDate": {
    "$date": {
      "$numberLong": "415353600000"
    }
  },
  "url":
"https://www.imdb.com/name/nm2143282/?ref_=nv_sr_srsg_0",
  "films": [
    "Black Panther",
    "Black Panther: Wakanda Forever",
    "12 Years a Slave",
    "Us",
    "Star Wars: Episode VII - The Force Awakens"
  ]
},{
  "_id": {
    "$oid": "639161f96cdb3591cea9692a"
  },
  "firstName": "Danai",
  "lastName": "Gurira",
```

```
  "birthDate": {
    "$date": {
      "$numberLong": "256291200000"
    }
  },
  "url": "https://www.imdb.com/name/nm1775091/?ref_=fn_al_nm_1",
  "films": [
    "Black Panther",
    "Black Panther: Wakanda Forever",
    "Avengers: Infinity War",
    "The Visitor",
    "Ghost Town"
  ]
},{
  "_id": {
    "$oid": "639165646cdb3591cea9692b"
  },
  "firstName": "Angela",
  "lastName": "Bassett",
  "birthDate": {
    "$date": {
      "$numberLong": "-359053200000"
    }
  },
  "url": "https://www.imdb.com/name/nm0000291/?ref_=fn_al_nm_1",
  "films": [
    "Black Panther",
    "Black Panther: Wakanda Forever",
    "Strange Days",
    "Contact",
    "Olympus has Fallen",
    "Akeelah and the Bee"
  ]
}]
```


Queries:

```
1. [
  {
    '$match': {
      'genres': 'Action'
    }
  }, {
    '$sort': {
      'title': 1
    }
  }]
```

2.
```
[
    {
        '$match': {
            'filmId': ObjectId('63891c65aa3bcdcc2f7f628e')
        }
    }, {
        '$match': {
            'showTime': {
                '$gt': datetime(2022, 11, 11, 8, 0, 0,
tzinfo=timezone.utc),
                '$lt': datetime(2022, 11, 12, 8, 0, 0,
tzinfo=timezone.utc)
            }
        }
    }
]
```

3.
```
[
  {
    '$lookup': {
      'from': 'showings',
      'localField': '_id',
      'foreignField': 'theaterId',
      'as': 'theaterShowings'
    }
  }, {
    '$project': {
      'name': 1,
      'theaterShowings': {
        'filmId': 1,
        'showTime': 1
      }
    }
  }
]
```