**Report Lab Assignment 2 – MongoDB**

**Name: Marco Farfan Quiroz**

**Course: Big Data, algorithms, techniques and Platforms**

**Exercise 1.**

Q1. The release year of the movie “Le parrain III”

```
db.movies.find({title: "Le parrain III"},{year: 1, _id: 0})
```

We first filter the movie "Le parrain III" and then only show the year of our result.

Q2. The title of the movies released between 1980 and 1990.

```
db.movies.find({"year" : {"$gte" : 1980, "$lte" : 1990}}, {"title" : 1, "_id" : 0})
```

We apply a filter to the year so that we keep movies with a released year greater than or equal to 1980 and less than or equal to 1990. Finally, we show only the title of our results.

Q3. Same query as b. with the titles must be sorted by alphabetical order.

```
db.movies.find({"year" : {"$gte" : 1980, "$lte" : 1990}}, {"title" : 1, "_id" : 0}).sort({"title": 1})
```

Same explanation as query 2. In addition, we added a sort operator to show the results in alphabetical order (ascending).

Q4. The titles of the french movies.

```
db.movies.find({country: "FR"}, {title: 1, _id: 0})
```

We first filter the French movies. Then, we only show the title of our results.

Q5. The title of the “crime” or “drama” movies.

```
db.movies.find({"genre" : {"$in" : ["crime", "drama"]}},{"title" : 1, "_id" : 0})
```

We use the “in” operator to implement the OR condition. So, we filter the movies of genre either crime or drama. Finally, we only show the title of our results.

Q6. The names and birth dates of the directors of french movies.

```
db.movies.find({country: "FR"}, {"director.last_name": 1,"director.first_name": 1,"director.birth_date": 1, _id: 0})
```

First, we filter the French movies. Then we display the names (first and last names) and the birth date of the directors of such movies.

Q7. The title of the movies in which Sofia Coppola played.

```
db.movies.find({"actors.first_name": "Sofia", "actors.last_name": "Coppola"}, {"title" : 1, "_id" : 0})
```

In this case, we filter the actor first name as Sofia and last name as Coppola to have all the documents in which Sofia Coppola played as an actress. Finally, we only show the title of our results.

Q8. The title and the genre of the movies of which Hitchcock is director.

```
db.movies.find({"director.last_name": "Hitchcock"}, {"title" : 1, "genre":1, "_id" : 0})
```

First, we filter by the director last name as "Hitchcock" in order to get all the documents in which Hitchcock is director. Then, we only show the title and genre of such results.

**Exercise 2.**

Q1. The number of movies by country. Show by decreasing number.

```
db.movies.aggregate({$group: {"_id" : {"country":"$country"},nb_movies : {$sum : 1}}},{$sort: {"nb_movies" : -1}})
```

We need to group by country and do a sum operation to get the total number of movies by country. Finally, we do a “sort” operation with a -1 value in order to get it in decreasing order.

Q2. The name of the actor in the role “Mary Corleone” in the movie “Le parrain III”.

```
db.movies.aggregate({$match: {"title" : "Le parrain III"}},{$unwind : "$actors"},{$match: {"actors.role": "Mary Corleone"}},{"$project":{"full_name" :{$concat: ["$actors.first_name", " ","$actors.last_name"]},"_id":0}})
```

In this case, we had to do 2 matches and 1 unwind in between because when doing the first filter, "title": "Le parrain III", our result has a list of actors and we cannot select only one element. We, therefore, do an unwind to divide this list into all its constituent elements. Then we apply the filter based on the actor role. Finally, we can display the full name as a concatenation of the first and last names.

Q3. The number of actors by movie. Sort by decreasing number.

```
db.movies.aggregate({$unwind : "$actors"},{$group: {"_id" : {"movie": "$title"}, nb_actors : {$sum : 1}}},{$sort: {"nb_actors" : -1}})
```

We first need to unwind the actors in order to have them separated as different elements. We are then able to group by movie and sum the number of occurrences, which is going to be the number of actors by movie. Finally, we do a “sort” operation with a -1 value in order to get it in decreasing order.

Q4. The average number of actors in a film.

```
db.movies.aggregate({$unwind : "$actors"},{$group: {"_id" : {"movie": "$title"}, nb_actors : {$sum : 1}}},{$group: {"_id":null, average:{$avg:"$nb_actors"}}},{"$project":{"_id":0}})
```

In this case, we first unwind actors so that we have them in separate documents. Then, we group by movie and sum so that we get the number of actors by movie. After this we need another group by because we want an aggregated value for all movies. So, we apply another group by and do not specify

an id in order to apply the aggregation function (average) to all documents and, therefore, obtain the average number of actors in a film. Finally, we drop the id column to only have the average value displayed.

**Exercise 3.** By using the operator $lookup on the collections movies and movies_boffice, find the box office of the movie “Le parrain III”.

```
db.movies.aggregate({$match: {"title" : "Le parrain III"}}, {$lookup:{from: "movies_boffice", localField: "_id", foreignField: "_id", as: "box_office"}})
```

For this exercise, we had to use the lookup operation on the collections “movies” and “movies_boffice”. First, we filter the movie “Le parrain III” by using the match operator as we only want to see results for that specific movie. Then, we use the lookup operator in order to join both collections and get the box office information from the collection “movies_boffice” based on the common key "_id".