1. Using MapReduce in mongodb solve following queries on given below collection.

{

“id” : 0,

“name” : “Leanne Flinn”,

“email” : “[leanne.flinn@unilogic.com”,](mailto:leanne.flinn@unilogic.com) “work” :”Unilogic” ,

“age” :27 “gender” :”Male” “Salary” :16660

“hobbies”:”Acrobatics,Photography,Papier-Mache”

}

* 1. Get the count of Males and Females
  2. Count the number of users in each hobby

To solve the given queries using MapReduce in MongoDB, you can create map and reduce functions and run them on the collection. Here's how you can do it:

1. Get the count of Males and Females:

Map function:

```javascript

var mapFunction = function() {

emit(this.gender, 1);

};

```

Reduce function:

```javascript

var reduceFunction = function(key, values) {

return Array.sum(values);

};

```

Run the MapReduce operation:

```javascript

db.collectionName.mapReduce(mapFunction, reduceFunction, { out: "genderCount" });

```

After running this, you can query the "genderCount" collection to get the count of males and females.

2. Count the number of users in each hobby:

Map function:

```javascript

var mapFunction = function() {

var hobbies = this.hobbies.split(",");

for (var i = 0; i < hobbies.length; i++) {

emit(hobbies[i], 1);

}

};

```

Reduce function:

```javascript

var reduceFunction = function(key, values) {

return Array.sum(values);

};

```

Run the MapReduce operation:

```javascript

db.collectionName.mapReduce(mapFunction, reduceFunction, { out: "hobbyCount" });

```

After running this, you can query the "hobbyCount" collection to get the count of users in each hobby.

Make sure to replace `collectionName` with the actual name of your collection in MongoDB.