Q 25. Using MapReduce in mongodb solve following queries on given below collection.

```
{
“id” : 0,
“name” : “Leanne Flinn”,
“email” : “leanne.flinn@unilogic.com”,
“work” :”Unilogic” ,
“age” :27
“gender” :”Male”
“Salary” :16660
“hobbies”:”Acrobatics,Photography,Papier-Mache”
}
```

1.
2.

Get the count of Males and Females
Count the number of users in each hobby

```
test> db.users.insertMany([
...     {
...         "id": 0,
...         "name": "Leanne Flinn",
...         "email": "leanne.flinn@unilogic.com",
...         "work": "Unilogic",
...         "age": 27,
...         "gender": "Male",
...         "Salary": 16660,
...         "hobbies": "Acrobatics,Photography,Papier-Mache"
...     },
...     {
...         "id": 1,
...         "name": "John Doe",
...         "email": "john.doe@unilogic.com",
...         "work": "Unilogic",
...         "age": 30,
...         "gender": "Male",
...         "Salary": 18000,
...         "hobbies": "Photography,Cooking,Reading"
...     },
...     {
...         "id": 2,
...         "name": "Jane Smith",
...         "email": "jane.smith@unilogic.com",
...         "work": "Unilogic",
...         "age": 25,
...         "gender": "Female",
...         "Salary": 15000,
...         "hobbies": "Acrobatics,Cooking,Reading"
...     }
...     // Add more documents as required
... ]);
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('672ee973e9ec79521ec73bf8'),
    '1': ObjectId('672ee973e9ec79521ec73bf9'),
    '2': ObjectId('672ee973e9ec79521ec73bfa')
```

```
  }
}

------------------------------------------------------------------------------
----------------------------------

test> db.users.mapReduce(
...     function() { emit(this.gender, 1); },
...     function(key, values) { return Array.sum(values); },
...     {
...         out: "gender_counts"
...     }
... );
DeprecationWarning: Collection.mapReduce() is deprecated. Use an aggregation
instead.
See https://docs.mongodb.com/manual/core/map-reduce for details.
{ result: 'gender_counts', ok: 1 }
test> db.users.mapReduce(
...     function() {
...         var hobbiesArray = this.hobbies.split(",");
...         hobbiesArray.forEach(function(hobby) {
...             emit(hobby.trim(), 1);
...         });
...     },
...     function(key, values) { return Array.sum(values); },
...     {
...         out: "hobby_counts"
...     }
... );
{ result: 'hobby_counts', ok: 1 }
test> print("Gender Counts:");
Gender Counts:

------------------------------------------------------------------------------
----------------------------------

test> db.gender_counts.find().forEach(printjson);
{
  _id: 'Female',
  value: 1
}
{
  _id: 'Male',
  value: 2
}

test> print("Hobby Counts:");
Hobby Counts:

------------------------------------------------------------------------------
----------------------------------

test> db.hobby_counts.find().forEach(printjson);
{
  _id: 'Papier-Mache',
  value: 1
}
{
  _id: 'Reading',
```

```
  value: 2
}
{
  _id: 'Acrobatics',
  value: 2
}
{
  _id: 'Photography',
  value: 2
}
{
  _id: 'Cooking',
  value: 2
}
```

---

26. Using MapReduce in mongodb solve following queries on given below collection.
1. Import zip.json.
2. Find total population in each state.

Query 1: import the zip.json file

```
mongoimport --db your_database_name --collection zipcodes --file
/path/to/zip.json;
```

Query 2:

```
// Map Function
var mapFunction = function() {
emit(this.state, this.pop);  // Emit each state's population
};
```

Query 3:

```
// Reduce Function
var reduceFunction = function(key, values) {
    return Array.sum(values);  // Sum populations for each state
};
```

Query 4:

```
// Run MapReduce
db.zipcodes.mapReduce(
 mapFunction,
reduceFunction,
{
       out: "state_population_totals"  // Output collection
}
);
```

Query 5:

```
db.state_population_totals.find().pretty();
```

Q 27.Create a database called ‘library’, create a collection called ‘books’.find the number of books
having pages less 250 pages and consider ad small book and greater than 250 consider as Big
book using Map Reduce function.

```
test> use library;
switched to db library
library> db.books.insertMany([
...     { "title": "Book One", "author": "Author A", "pages": 200 },
...     { "title": "Book Two", "author": "Author B", "pages": 300 },
...     { "title": "Book Three", "author": "Author C", "pages": 150 },
...     { "title": "Book Four", "author": "Author D", "pages": 275 },
...     { "title": "Book Five", "author": "Author E", "pages": 220 }
...     // Add more documents as needed
... ]);
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('672ef99de9ec79521ec73bfb'),
    '1': ObjectId('672ef99de9ec79521ec73bfc'),
    '2': ObjectId('672ef99de9ec79521ec73bfd'),
    '3': ObjectId('672ef99de9ec79521ec73bfe'),
    '4': ObjectId('672ef99de9ec79521ec73bff')
  }
}

---------------------------------------------------------------------------------------
---------


library> var mapFunction = function() {
...     var category = this.pages < 250 ? "Small" : "Big";
...     emit(category, 1);
... };

---------------------------------------------------------------------------------------
--------


library> var reduceFunction = function(key, values) {
...     return Array.sum(values);
... };

---------------------------------------------------------------------------------------
--------


library> db.books.mapReduce(
...     mapFunction,
...     reduceFunction,
...     {
...         out: "book_size_counts"
...     }
... );
{ result: 'book_size_counts', ok: 1 }
library> print("Book Size Counts:");
```

Book Size Counts:

```
library> db.book_size_counts.find().forEach(printjson);
{
  _id: 'Small',
  value: 3
}
{
  _id: 'Big',
  value: 2
}
```