**Step 1: Create the Collection and Insert the Sample Document**

1. **Switch to your database**:

Code:

use dataBaseName

1. **Create the users collection and insert the sample document**:

code:

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"

},

// Add more documents as needed to test MapReduce

])

**Step 2: Use MapReduce to Get the Count of Males and Females**

1. **Map function**: This function emits gender as the key and 1 as the value.

Code:

var mapGender = function() {

emit(this.gender, 1);

};

1. **Reduce function**: This function sums the values for each gender.

Code:

var reduceGender = function(key, values) {

return Array.sum(values);

};

1. **Run MapReduce**:

Code:

db.users.mapReduce(

mapGender,

reduceGender,

{ out: "gender\_count" }

)

1. **Retrieve the results**:

Code:

db.gender\_count.find().pretty()

* + **Explanation**: This MapReduce operation counts the number of male and female users and stores the results in the gender\_count collection.

**Step 3: Use MapReduce to Count the Number of Users in Each Hobby**

1. **Map function**: This function splits the hobbies string into individual hobbies and emits each hobby as a key with a value of 1.

Code:

var mapHobbies = function() {

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

hobbiesArray.forEach(function(hobby) {

emit(hobby.trim(), 1);

});

};

1. **Reduce function**: This function sums the values for each hobby.

Code:

var reduceHobbies = function(key, values) {

return Array.sum(values);

};

1. **Run MapReduce**:

Code:

db.users.mapReduce(

mapHobbies,

reduceHobbies,

{ out: "hobby\_count" }

)

1. **Retrieve the results**:

Code:

db.hobby\_count.find().pretty()

* + **Explanation**: This MapReduce operation counts how many users have each hobby and stores the results in the hobby\_count collection.

**Summary**

1. **MapReduce for Gender Count**: Emitted each gender and counted occurrences using Array.sum.
2. **MapReduce for Hobby Count**: Split the hobbies field, emitted each hobby, and used Array.sum to count occurrences.
3. **Results Storage**: The results are stored in gender\_count and hobby\_count collections, which can be queried as needed.

This approach effectively uses MapReduce in MongoDB to perform data aggregation based on gender and hobbies.

Top of Form

Bottom of Form