To perform the task using MapReduce in MongoDB on the zip.json collection, follow the steps outlined below:

**Step 1: Import zip.json into MongoDB**

1. **Make sure the zip.json file is located on your system.** For this example, let’s assume it’s located at C:\data\zip.json on Windows or /path/to/your/data/zip.json on macOS/Linux.
2. **Use the mongoimport command** to import the file:
   * **On Windows**:

Bash code:

mongoimport --db library --collection zip --file C:\data\zip.json --jsonArray

* + **Explanation**: This command imports the zip.json file into a zip collection within the library database.

**Step 2: Use MapReduce to Calculate the Total Population in Each State**

1. **Open the MongoDB Shell**:

Bash code

mongosh

1. **Switch to the library Database**:

Code:

use library

1. **Define the Map Function**:

Code:

var mapFunction = function() {

emit(this.state, this.population);

};

* + **Explanation**: The mapFunction emits the state as the key and the population as the value.

1. **Define the Reduce Function**:

Code:

var reduceFunction = function(state, populations) {

return Array.sum(populations);

};

* + **Explanation**: The reduceFunction takes an array of population numbers for each state and sums them up using Array.sum().

1. **Run the MapReduce Operation**:

Code:

db.zip.mapReduce(

mapFunction,

reduceFunction,

{ out: "total\_population\_by\_state" }

);

* + **Explanation**: The mapReduce method processes the data using the map and reduce functions and outputs the results into a new collection named total\_population\_by\_state.

1. **View the Results**:

Code:

db.total\_population\_by\_state.find().pretty()

* + **Explanation**: This command retrieves and displays the total population for each state from the total\_population\_by\_state collection.

**Summary of MapReduce Process**

* **Map Function**: Extracts each document's state and population values and emits them.
* **Reduce Function**: Aggregates populations for each state by summing them up.
* **Output**: The total population per state is stored in a new collection, total\_population\_by\_state.

**Notes**

* **Performance**: MapReduce is powerful but can be slower than using MongoDB's aggregation framework, especially for large datasets. If performance is critical, consider using the aggregation framework as an alternative.
* **Debugging**: If any issues occur during the import or execution, double-check file paths and ensure the MongoDB server is running.

This guide will help you calculate the total population for each state using MongoDB's MapReduce functionality.