Create index and fire queries with MongoDB

* 1. Import zip.json.
  2. Create single field, composite and multikey indexes.
  3. Fire queries given below again and write your analysis.
     1. Display all cities having population above 1600.
     2. Display all cities in state “KS”.
     3. Display location of city "TIMKEN"

To perform the requested tasks using MongoDB, you can follow these steps:

1. Import zip.json data:

Assuming you have the "zip.json" file in your working directory, you can import it into your MongoDB database with the following command:

```bash

mongoimport --db test --collection zips --file zip.json

```

This command assumes you have a MongoDB database named "test" and a collection named "zips."

2. Create single field, composite, and multikey indexes:

After importing the data, you can create indexes as follows:

a. Create a single-field index on "pop" for population:

```javascript

db.zips.createIndex({ pop: 1 })

```

b. Create a compound index on "state" and "city" fields:

```javascript

db.zips.createIndex({ state: 1, city: 1 })

```

c. Create a multikey index on "loc" field as a 2dsphere index:

```javascript

db.zips.createIndex({ loc: "2dsphere" })

```

3. Fire queries and provide analysis:

Query 1: Display all cities having a population above 1,600:

```javascript

db.zips.find({ pop: { $gt: 1600 } }, { city: 1, pop: 1, \_id: 0 })

```

Analysis: This query uses the single-field index on "pop" to efficiently filter cities with a population above 1,600.

Query 2: Display all cities in the state "KS":

```javascript

db.zips.find({ state: "KS" }, { city: 1, \_id: 0 })

```

Analysis: This query benefits from the compound index on "state" and "city" to filter cities in the state of "KS."

Query 3: Display the location of the city "TIMKEN":

```javascript

db.zips.find({ city: "TIMKEN" }, { loc: 1, \_id: 0 })

```

Analysis: This query leverages the multikey index on the "loc" field (2dsphere index) to efficiently retrieve the location of the city "TIMKEN."

By using appropriate indexes, these queries should perform efficiently and improve query execution time. The actual results will depend on the data in your "zip.json" file.

Please make sure to replace "test" and "zips" with the actual database name and collection name you are using in your MongoDB setup.