

Experiment 9

Name: Asad Shaikh

Branch: MTech CE

Registration Id: 242050023

Aim: Extend MongoDB functionality for MapReduce on a document collection

Theory:

What is NoSQL?

NoSQL stands for “**Not Only SQL**” and refers to a class of database systems that provide a mechanism for storage and retrieval of data modeled in ways other than the tabular relations used in relational databases (RDBMS).

Types of NoSQL Databases:

1. **Document-oriented** (e.g., MongoDB)
2. **Key-Value stores** (e.g., Redis)
3. **Column-oriented** (e.g., Apache Cassandra)
4. **Graph-based** (e.g., Neo4j)

Key Characteristics:

- **Schema-less data models**
- **Scalable** (horizontal scaling)
- **Flexible** data representation (JSON, BSON, XML)
- **High performance** for large volumes of unstructured or semi-structured data
- Designed for **distributed computing**

What is MongoDB?

MongoDB is a high-performance, open-source, document-oriented NoSQL database that stores data in BSON (Binary JSON) format. It's known for its flexibility, scalability, and ease of use.

MongoDB Data Structure:

MongoDB stores data in the following hierarchy:

- **Database** →
- **Collections** →
- **Documents** (equivalent to rows in SQL, but schema-less)

Example of a MongoDB document (BSON/JSON-like):

```
{
  "name": "Pizza Hut",
  "address": "Regent Street City Centre",
  "food": "Italian",
  "location": [52.20103, 0.126023],
  "priceRange": "cheap"
}
```

Key Features:

- **Schema-less:** Each document can have different fields.
- **Indexing:** Supports primary, secondary, text, geospatial indexes.
- **Aggregation Framework:** Powerful querying and data processing pipeline.
- **Replication:** High availability using replica sets.
- **Sharding:** Horizontal scaling to handle large data volumes.

What is MapReduce?

MapReduce is a programming model for processing large datasets with a distributed algorithm:

- **Map step:** Processes input data and produces intermediate key-value pairs.
- **Reduce step:** Merges intermediate values associated with the same key.

MongoDB includes a built-in `mapReduce()` function to perform similar computations.

MapReduce in MongoDB

MongoDB supports MapReduce operations directly on collections using JavaScript functions.

Code / Output:

Install MongoDB:

Command 1: `brew tap mongodb/brew`

Command 2: `brew install mongodb-community@7.0`

Start Services:

Command: `brew services start mongodb/brew/mongodb-community`

```
asadasifshaikh@ASADs-MacBook-Pro ~ % brew services start mongodb/brew/mongodb-community
⇒ Successfully started `mongodb-community` (label: homebrew.mxcl.mongodb-community)
asadasifshaikh@ASADs-MacBook-Pro ~ %
```

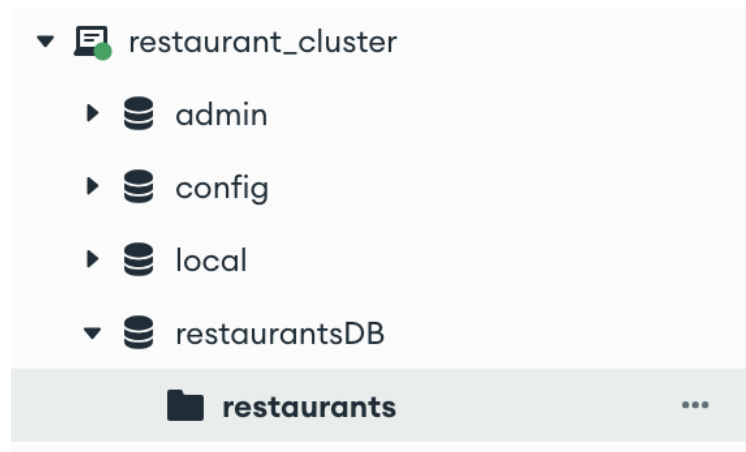
Check Installations:

Command: mongod --version

```
asadasifshaikh@ASADs-MacBook-Pro ~ % mongod --version
db version v8.0.8
Build Info: {
  "version": "8.0.8",
  "gitVersion": "7f52660c14217ed2c8d3240f823a2291a4fe6abd",
  "modules": [],
  "allocator": "system",
  "environment": {
    "distarch": "x86_64",
    "target_arch": "x86_64"
  }
}
asadasifshaikh@ASADs-MacBook-Pro ~ %
```

In MongoDB Compass:

Create New cluster, database and a collection:



Importing the Data:


ADD DATA


EXPORT DATA


UPDATE


DELETE

25

```
_id : ObjectId('681d9d2d2e4c958786636cd6')
address : "Regent Street City Centre"
area : "centre"
food : "italian"
id : "19210"
introduction : "Pizza hut is a large chain with restaurants nationwide offering conven..."
location : Array (2)
name : "pizza hut city centre"
phone : "01223323737"
postcode : "cb21ab"
pricerange : "cheap"
type : "restaurant"
```

```
_id: ObjectId('681d9d2d2e4c958786636cd7')  
address: "Finders Corner Newmarket Road"  
area: "east"  
food: "international"  
id: "30650"  
introduction: ""  
▶ location: Array (2)  
  name: "the missing sock"  
  phone: "01223812660"  
  postcode: "cb259aq"  
  pricerange: "cheap"  
  signature: "african babooti"  
  type: "restaurant"
```

CRUD Operations:

Insert Document: (CREATE)

```
db.restaurants.insertOne({
  "address": "High Street West",
  "area": "west",
  "food": "chinese",
  "id": "99999",
  "introduction": "New Chinese diner with fresh noodles",
  "location": [52.202, 0.125],
  "name": "Golden Wok",
  "phone": "01223399999",
  "postcode": "cb22xy",
  "pricerange": "moderate",
  "type": "restaurant"
});
```

```
>_MONGOSH
```

```
> db["restaurants"].insertOne({
  address: "High Street West",
  area: "west",
  food: "chinese",
  id: "99999",
  introduction: "New Chinese diner with fresh noodles",
  location: [52.202, 0.125],
  name: "Golden Wok",
  phone: "01223399999",
  postcode: "cb22xy",
  pricerange: "moderate",
  type: "restaurant"
});
< {
  acknowledged: true,
  insertedId: ObjectId('681dd624a7927cdd801e0aee')
}
restaurant_db>
```

Find Document: (READ)

db.restaurants.find({food: "italian"})

```
> db.restaurants.find({food: "italian"})
< {
  _id: ObjectId('681d9d2d2e4c958786636cd6'),
  address: 'Regent Street City Centre',
  area: 'centre',
  food: 'italian',
  id: '19210',
  introduction: 'Pizza hut is a large chain with restaurants nationwide offering convenience pizzas pasta and salads to eat in or take away'
  location: [
    52.20103,
    0.126023
  ],
  name: 'pizza hut city centre',
  phone: '01223323737',
  postcode: 'cb21ab',
  pricerange: 'cheap',
  type: 'restaurant'
}
{
  _id: ObjectId('681d9d2d2e4c958786636cda'),
  address: 'G4 Cambridge Leisure Park Clifton Way Cherry Hinton',
  area: 'south',
  food: 'italian',
  id: '19196',
  introduction: 'pizza hut is a large chain with restaurants nationwide offering convenience pizzas pasta and salads to eat in or take away'
  location: [
    52.190176,
    0.13699
  ],
  name: 'pizza hut city centre',
  phone: '01223323737',
  postcode: 'cb21ab',
  pricerange: 'cheap',
  type: 'restaurant'
}
```

Update Document: (UPDATE)

```
db.restaurants.updateOne(
  { name: "pizza hut city centre" },
  { $set: { phone: "01223330000" } } //DIFFERENT PHONE NUMBER
);
```

Before:

```
▶ {
  _id: ObjectId('681d9d2d2e4c958786636cd6'),
  address: "Regent Street City Centre",
  area: "centre",
  food: "italian",
  id: "19210",
  introduction: "Pizza hut is a large chain with restaurants nationwide offering conven...",
  location: Array (2)
    name: "pizza hut city centre"
    phone: "01223323737"
    postcode: "cb21ab"
    pricerange: "cheap"
    type: "restaurant"
  }
}
```

Update:

```
> db.restaurants.updateOne(
  { name: "pizza hut city centre" },
  { $set: { phone: "01223330000" } } //DIFFERENT PHONE NUMBER
);
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

After:

```
▶ {
  _id: ObjectId('681d9d2d2e4c958786636cd6'),
  address: "Regent Street City Centre",
  area: "centre",
  food: "italian",
  id: "19210",
  introduction: "Pizza hut is a large chain with restaurants nationwide offering conven...",
  location: Array (2)
    name: "pizza hut city centre"
    phone: "01223330000"
    postcode: "cb21ab"
    pricerange: "cheap"
    type: "restaurant"
  }
}
```

Delete Document: (DELETE)

```
db.restaurants.deleteOne({ id: "19210" });
```

```
> db.restaurants.deleteOne({ id: "19210" });  
  
< {  
  acknowledged: true,  
  deletedCount: 1  
}
```

MapReduce Extensions:

```
> var mapFunction = function () { emit(this.area, 1); };  
> var reduceFunction = function (key, values) { return Array.sum(values); };  
> db.restaurants.mapReduce(  
  mapFunction,  
  reduceFunction,  
  {  
    out: "area_counts"  
  }  
)  
  
< DeprecationWarning: Collection.mapReduce() is deprecated. Use an aggregation instead.  
  See https://docs.mongodb.com/manual/core/map-reduce for details.  
< { result: 'area_counts', ok: 1 }
```

Created collection 'area_counts' using MapReduce:

```
> db.area_counts.find().pretty();
< {
  _id: 'south',
  value: 9
}
{
  _id: 'east',
  value: 9
}
{
  _id: 'centre',
  value: 68
}
{
  _id: 'west',
  value: 15
}
{
  _id: 'north',
  value: 9
}
```

Collection:



Conclusion:

In this experiment, we learnt about NoSQL, MongoDB and its environment. We learnt about the advantages and disadvantages that come with using MongoDB and NoSQL. We also saw how to set up MongoDB in our system and how we can perform the CRUD operations in it. We also performed some MapReduce Extensions in the mongodb shell to create the 'area_counts' collection in the database