

Q1. What are different Aggregation commands?

Ans. 1. Aggregate:

- Performs aggregation tasks such as group using the aggregation framework.

2. Count:

- Counts the number of documents in a collection.

3. Distinct:

- Displays the distinct values found for a specified key on a collection.

4. Group:

- Groups documents in a collection by the specified key and performs simple aggregation.

5. mapReduce:

- Performs map-reduce to the aggregation pipeline.

Q2. What is map and reduce phase?

Ans. 2. Map-reduce is a data processing paradigm for condensing large volumes of data ~~and~~ into useful aggregated results.

- For map-reduce operations, MongoDB provides the mapReduce database command.

→ Syntax:

`db.COLLECTION_NAME.mapReduce()`

- In general, map-reduce operations have two phases: a map stage that processes each document and emits one or more objects for each input document, and a reduce phase that combines the

the output of the map operation.

- Optionally, map-reduce can have finalize stage to make final modifications to the result.
- Like other aggregation operations, map-reduce can specify a query condition to select the input documents as well as sort and limit the results.

Q3. Explain map Reduce concurrency.

- Ans.
- Map-reduce concurrency is the process of managing locks during a map-reduce operation.
 - Map-reduce is a computer science technique that uses two steps, map and reduce, to process large amounts of data efficiently.
 - Here are some of the locks that are taken during a map-reduce operation:

1. Read phase:

- Takes a read lock and yields every 100 documents.

2. Insert into temporary collection:

- Takes a write lock for a single write.

3. Create output collection:

- Takes a write lock if the output collection doesn't exist.

4. Output actions:

- Takes a global write lock that blocks all operations on the MongoDB instance if the output collection exists.

Q4. Write steps for mapReduce operation with example.

Ans. The steps for mapReduce operation are:

1. Define the map function
- Process each statement from the collection by defining a map function.

2. Create ~~a~~ the reduce function:
• Removes a single item from the values returned from the mapReduce process by creating a reduce function.

3. Perform the map-reduce process:
• Use the map and reduce functions to perform the map-reduce process.

4. Check the results:
• Use the necessary command to check the result ~~of~~ of the mapReduce command.

Q5. what is map-reduce JavaScript function?

Ans. The map() method is used for creating new array from an existing one, applying a function to each one of the elements of the first array.

→ Syntax:

```
var new_array = arr.map(function call back element,
                        index, array) {
```

Return value for new_array

{ , thisArg }

- The reduce() method reduces an array of values

down to just one value.

- To get the output value, it runs a reducer function on each element of the array.
- eg:

```
const numbers = [1, 2, 3, 4]
```

```
const sum = numbers.reduce(function (result, item) {  
    return result + item  
}, 0);
```