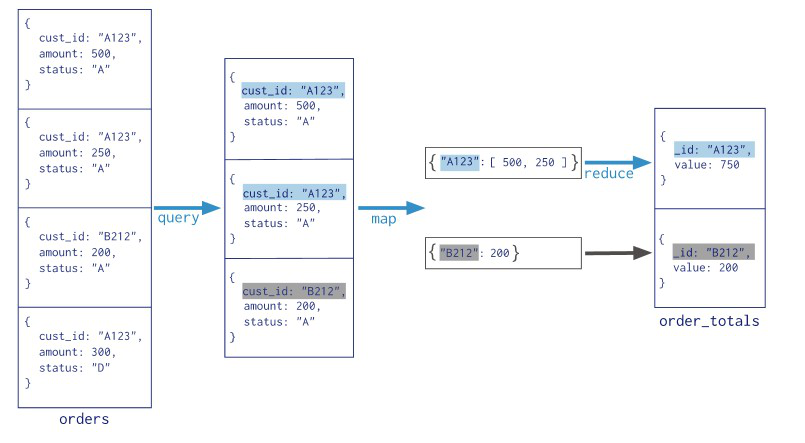
**Practical no-9-Program**

**Title :Implement Map reduces operation with suitable example using MongoDB**

# Map-Reduce

* **Map-reduce is a data processing paradigm for condensing large volumes of data into useful *aggregated* results. For map- reduce operations, MongoDB provides the** [**mapReduce**](https://docs.mongodb.com/manual/reference/command/mapReduce/dbcmd.mapReduce) **database command.**
* **Consider the following map-reduce operation:**

**Map-Reduce**

****

Steps in Map Reduce

* **The map takes data in the form of pairs and returns a list of <key, value> pairs. The keys will not be unique in this case.**
* **Using the output of Map, sort and shuffle are applied by the Hadoop architecture. This sort and shuffle acts on these list of <key, value> pairs and sends out unique keys and a list of values associated with this unique key <key, list(values)>.**
* **An output of sort and shuffle sent to the reducer phase. The reducer performs a defined function on a list of values for unique keys, and Final output <key, value> will be stored/displayed.**

## Sort and Shuffle

**The sort and shuffle occur on the output of Mapper and before the reducer. When the Mapper task is complete, the results are sorted by key, partitioned if there are multiple reducers, and then written to disk. Using the input from each Mapper <k2,v2>, we collect all the values for each unique key k2. This output from the shuffle phase in the form of <k2, list(v2)> is sent as input to reducer phase.**

## Usage of MapReduce

* **It can be used in various application like document clustering, distributed sorting, and web link-graph reversal.**
* **It can be used for distributed pattern-based searching.**
* **We can also use MapReduce in machine learning.**
* **It was used by Google to regenerate Google's index of the World Wide Web.**
* **It can be used in multiple computing environments such as multi-cluster, multi-core, and mobile environment.**

**Example:-**

**>db.collection.mapReduce(**

**function() {emit(key,value);}, //map function**

**function(key,values) {return reduceFunction}, { //reduce function**

**out: collection,**

**query: document,**

**sort: document,**

**limit: number**

**}**

**)**

**The map-reduce function first queries the collection, then maps the result documents to emit key-value pairs, which is then reduced based on the keys that have multiple values.**

**In the above syntax −**

* **map is a javascript function that maps a value with a key and emits a key-value pair**
* **reduce is a javascript function that reduces or groups all the documents having the same key**
* **out specifies the location of the map-reduce query result**
* **query specifies the optional selection criteria for selecting documents**
* **sort specifies the optional sort criteria**
* **limit specifies the optional maximum number of documents to be returned**

**Conclusion:-**