Page No.
Date

Assignment - B4

* Date of Completion: 28/10/2020
* Date of Submission: 04/11/2020

* Title: Map Reduce

* Problem Statement: Write and implement example of MapReduce operation with suitable collection using MongoDB

* Learning Objectives:
  - To understand concept of Map-Reduce as data processing paradigm
  - To compare mapreduce and aggregation pipeline in MongoDB

* Learning Outcomes:
  - Learn and implement Map-Reduce for condensing large volumes of data into useful aggregated results.
  - To know the difference between the use of Mapreduce & aggregation pipeline in MongoDB

* Hardware & Software Requirements:
  - MongoDB Version Server, Windows 10 64 bit, Intel i5.

* Theory:
MapReduce is a data processing paradigm for condensing large volumes of data into useful aggregated results. It is a powerful & flexible

Teacher's Sign.: _____

tool for aggregating tools data. It can solve some problems that are too complex to express using the aggregation frameworks query language, & uses Java Script as its 'query language'

Syntax:

```
db.collections.mapReduce (
    mapFunction () { emit (<key>, <value>) },
  reduceFunction (key, value) { //operation & return },
    {
      query: { < query document > }
      out: < collection name >
    }
)
```

Map & Reduce Phases:

- MongoDB applies the "map phase" to each input document (i.e; the document in the collection that match the query condition). The map emits key-value pairs.
- For those keys that have multiple values, MongoDB applies the "reduce phase", which collects & condenses the aggregated data. optionally the output of reduced data function may pass through a "finalize" function to further condense or process the result of the aggregation.

MapReduce Results:

the map reduce operation can write results to a collection on return the results in life time, if you

Teacher's Sign : _____

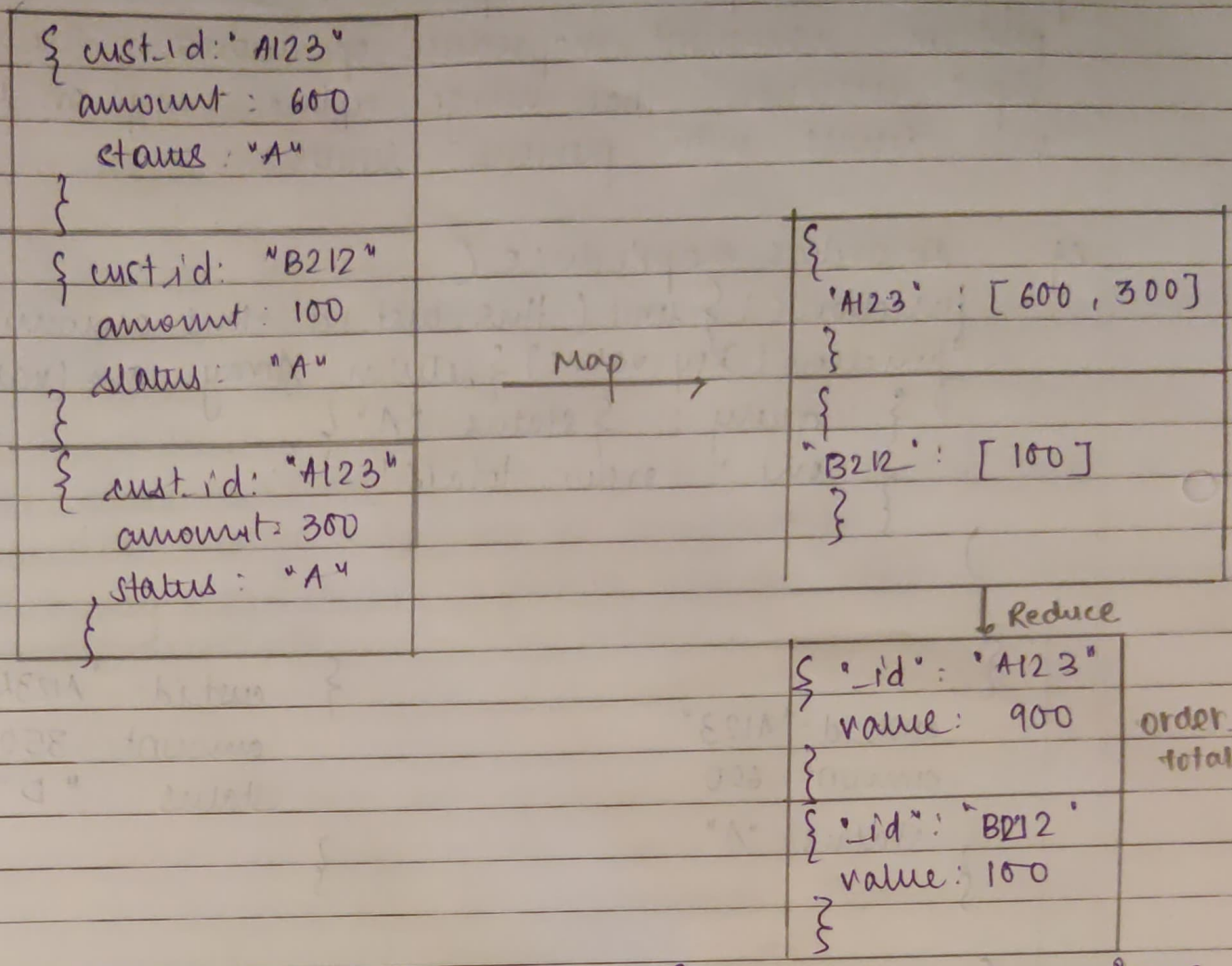
if you write map reduce output to a collection then you can perform subsequent mapreduce operations on the same input collection that merge replace, merge or reduce new results with previous results.

eg:
```
db.orders.mapReduce(
    function() { emit(this.cust_id, this.amount); },
    function(key, value) { return Array.sum(values) },
    { query: { status: "A" },
      out: "order_totals"
    }
)
```

```
{
  cust_id: "A123"
  amount: 600
  status: "A"
}

{
  cust_id: "B212"
  amount: 100
  status: "A"
}

{ cust_id: 'A1234'
  amount: 350
  status: "D"
}

{
  cust_id: "A123"
  amount: 300
  status: "A"
}
```

Orders

Teacher's Sign.: _____



- MongoDB: Aggregation framework & MapReduce comparisons
  - MongoDB MapReduce can perform complex and flexible aggregations that can't be done with aggregation framework query language
  - average CPU utilization incase of mapreduce is higher incomparison of aggregation
  - Use of aggregation pipe line is faster as compared to mapreduce
  - Aggregation framework returns in line result. Thus cannot be used for further processing unlike in MapReduce result can be stored to a collection.

Teacher's Sign.: _____

- Output from the single pipeline cannot exceed the BSON document size (ie; 16 MB) whereas in mapReduce single emit can hold half of BSON size limit (ie; 8MB)

Use cases of mapReduce & Aggregation:

- Queries that are very complex are difficult to handle in Aggregation Framework can be helped out with map reduce
- the small datasets will take considerable time to load in mapReduce, be it a small data set or big both will take some amount of time to execute. So it's better to handle large data sets in mapReduce
- Parallely Aggregation pipeline can be used for handling small data set.

Conclusion:

We studied the use of mapReduce in MongoDB and implemented examples of the same and studied the difference between mapReduce and aggregation Pipeline and their usecases with comparisons.

Teacher's Sign.: _____