Name : Bhavani Rajpurohit

Class : AIA-3 Subject: DBMS LAB

Roll No : 2213688 Batch : B

ASSIGNMENT NO: 09

Aim –

Implement MapReduce example in MongoDB with suitable dataset.

A. Create a sample collection order with 10 documents.

B. Perform the map-reduce operation on the orders collection to group by the cust_id, and calculate the sum of the price for each cust_id.

Software Required – MongoDB

Theory :-

Map-reduce is a data processing paradigm for condensing large volumes of data into useful aggregated results. To perform map-reduce operations, MongoDB provides the mapReduce database command.

Map-Reduce Syntax

```
db.collection.mapReduce( function() {emit(key, value);},
//Define map function
function(key,values) {return reduceFunction}, {
//Define reduce function
out: collection,
query: document,
sort: document,
limit: number
}
)
```

Map-Reduce Syntax Explanation •

The above map-reduce function will query the collection, and then map the output documents to the emit key-value pairs.

After this, it is reduced based on the keys that have multiple values. Here, we have used the following functions and parameters.

• Map: – It is a JavaScript function. It is used to map a value with a key and produces a key-value pair

. • Reduce: – It is a JavaScript function. It is used to reduce or group together all the documents which have the same key.

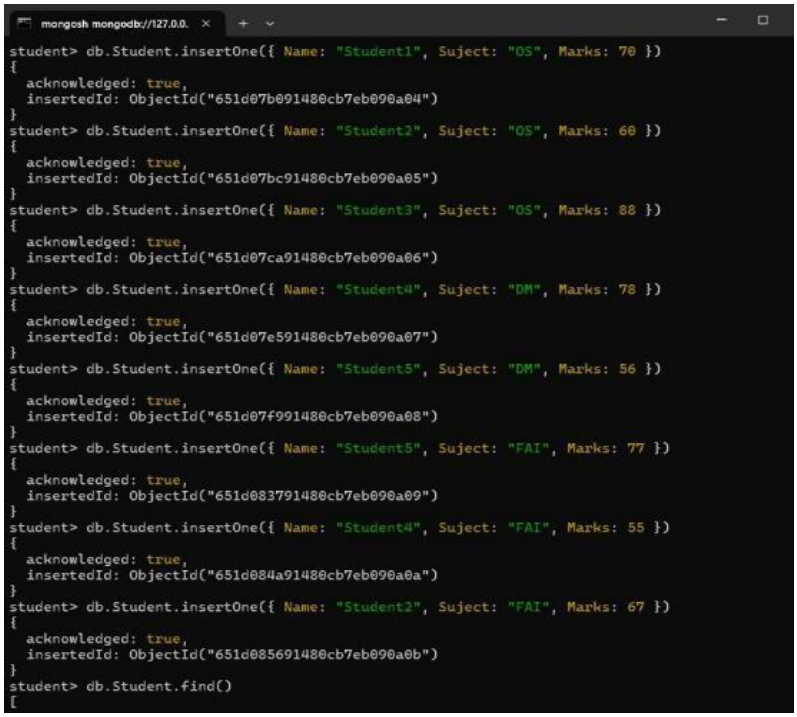
• Out: – It is used to specify the location of the map-reduce query output

. • Query: – It is used to specify the optional selection criteria for selecting documents.

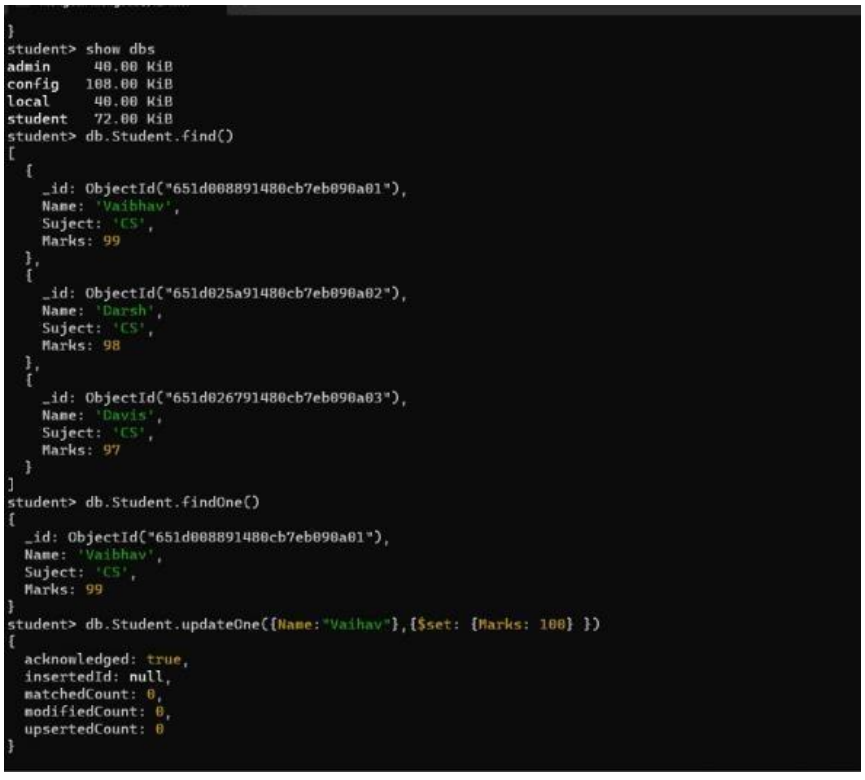
• Sort: – It is used to specify the optional sort criteria.

• Limit: – It is used to specify the optional maximum number of documents which are desired to be returned.

Code:



```
mongosh mongodb://127.0.0.
}
student> db.Student.findOne()
{
  _id: ObjectId("651d008891480cb7eb090a01"),
  Name: 'Vaibhav',
  Suject: 'CS',
  Marks: 99
}
student> db.Student.updateOne({Name:"Vaibhav"},{$set: {Marks: 100} })
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
student> db.Student.findOne()
{
  _id: ObjectId("651d008891480cb7eb090a01"),
  Name: 'Vaibhav',
  Suject: 'CS',
  Marks: 100
}
student> db.Student.find()
[
  {
    _id: ObjectId("651d008891480cb7eb090a01"),
    Name: 'Vaibhav',
    Suject: 'CS',
    Marks: 100
  },
  {
    _id: ObjectId("651d025a91480cb7eb090a02"),
    Name: 'Darsh',
    Suject: 'CS',
    Marks: 98
  },
  {
    _id: ObjectId("651d026791480cb7eb090a03"),
    Name: 'Davis',
    Suject: 'CS',
    Marks: 97
```



```
}
student> show dbs
admin     40.00 KiB
config   108.00 KiB
local     40.00 KiB
student   72.00 KiB
student> db.Student.find()
[
  {
    _id: ObjectId("651d008891480cb7eb090a01"),
    Name: 'Vaibhav',
    Suject: 'CS',
    Marks: 99
  },
  {
    _id: ObjectId("651d025a91480cb7eb090a02"),
    Name: 'Darsh',
    Suject: 'CS',
    Marks: 98
  },
  {
    _id: ObjectId("651d026791480cb7eb090a03"),
    Name: 'Davis',
    Suject: 'CS',
    Marks: 97
  }
]
student> db.Student.findOne()
{
  _id: ObjectId("651d008891480cb7eb090a01"),
  Name: 'Vaibhav',
  Suject: 'CS',
  Marks: 99
}
student> db.Student.updateOne({Name:"Vaihav"},{$set: {Marks: 100} })
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
```

```
    _id: ObjectId("651d025a91480cb7eb090a02"),
    Name: 'Darsh',
    Suject: 'CS',
    Marks: 98
  },
  {
    _id: ObjectId("651d026791480cb7eb090a03"),
    Name: 'Davis',
    Suject: 'CS',
    Marks: 97
  }
]
student> db.Student.deleteOne({Name:"Davis"})
{ acknowledged: true, deletedCount: 1 }
student> db.Student.find()
[
  {
    _id: ObjectId("651d008891480cb7eb090a01"),
    Name: 'Vaibhav',
    Suject: 'CS',
    Marks: 100
  },
  {
    _id: ObjectId("651d025a91480cb7eb090a02"),
    Name: 'Darsh',
    Suject: 'CS',
    Marks: 98
  }
]
student>
```

```
Microsoft Windows [Version 10.0.22621.2283]
(c) Microsoft Corporation. All rights reserved.

C:\Users\vaibh>mongosh
Current Mongosh Log ID: 651cfcf491480cb7eb090a00
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.0.1
Using MongoDB:          7.0.2
Using Mongosh:          2.0.1

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

------
   The server generated these startup warnings when booting
   2023-10-03T23:05:25.802+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
------

test> use student
switched to db student
student> show dbs
admin   40.00 KiB
config  72.00 KiB
local   40.00 KiB
student> db.createCollection("Student")
{ ok: 1 }
student> db.Student.insertOne({
... Name: "Vaibhav",
... Suject: "CS",
... Marks: 99,
... })
{
  acknowledged: true,
  insertedId: ObjectId("651d008891480cb7eb090a01")
}
student> db.Student.insertOne({ Name: "Darsh", Suject: "CS", Marks: 98 })
{
  acknowledged: true,
  insertedId: ObjectId("651d025a91480cb7eb090a02")
}
student> db.Student.insertOne({ Name: "Davis", Suject: "CS", Marks: 97 })
{
  acknowledged: true,
  insertedId: ObjectId("651d026791480cb7eb090a03")
```

```
}
student> db.Student.find()
[
  {
    _id: ObjectId("651d008891480cb7eb090a01"),
    Name: 'Vaibhav',
    Suject: 'CS',
    Marks: 100
  },
  {
    _id: ObjectId("651d025a91480cb7eb090a02"),
    Name: 'Darsh',
    Suject: 'CS',
    Marks: 98
  },
  {
    _id: ObjectId("651d07b091480cb7eb090a04"),
    Name: 'Student1',
    Suject: 'OS',
    Marks: 70
  },
  {
    _id: ObjectId("651d07bc91480cb7eb090a05"),
    Name: 'Student2',
    Suject: 'OS',
    Marks: 60
  },
  {
    _id: ObjectId("651d07ca91480cb7eb090a06"),
    Name: 'Student3',
    Suject: 'OS',
    Marks: 88
  },
  {
    _id: ObjectId("651d07e591480cb7eb090a07"),
    Name: 'Student4',
    Suject: 'DM',
    Marks: 78
  },
  {
    _id: ObjectId("651d07f991480cb7eb090a08"),
    Name: 'Student5',
```

```
    Marks: 78
  },
  {
    _id: ObjectId("651d07f991480cb7eb090a08"),
    Name: 'Student5',
    Suject: 'DM',
    Marks: 56
  },
  {
    _id: ObjectId("651d083791480cb7eb090a09"),
    Name: 'Student5',
    Suject: 'FAI',
    Marks: 77
  },
  {
    _id: ObjectId("651d084a91480cb7eb090a0a"),
    Name: 'Student4',
    Suject: 'FAI',
    Marks: 55
  },
  {
    _id: ObjectId("651d085691480cb7eb090a0b"),
    Name: 'Student2',
    Suject: 'FAI',
    Marks: 67
  }
]
student> var map1 = function(){emit(this.Subject,this.Marks)};

student> var reduce1 = function(Subject,Marks){return Array.sum(Marks);};

student> db.Student.mapReduce(map1,reduce1,{out:"Result"});
{ result: 'Result', ok: 1 }
student> db.Result.find()
[ { _id: null, value: 749 } ]
student> var map1 = function(){emit(this.Suject,this.Marks)};

student> var reduce1 = function(Suject,Marks){return Array.sum(Marks);};

student> db.Student.mapReduce(map1,reduce1,{out:"Result"});
{ result: 'Result', ok: 1 }
student> db.Result.find()
```