Experiment No. -9

Name: Sourav Shailesh Toshniwal
Class: TY CSE-8 AIEC-1
Roll no.: 2213047
Batch: A

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.

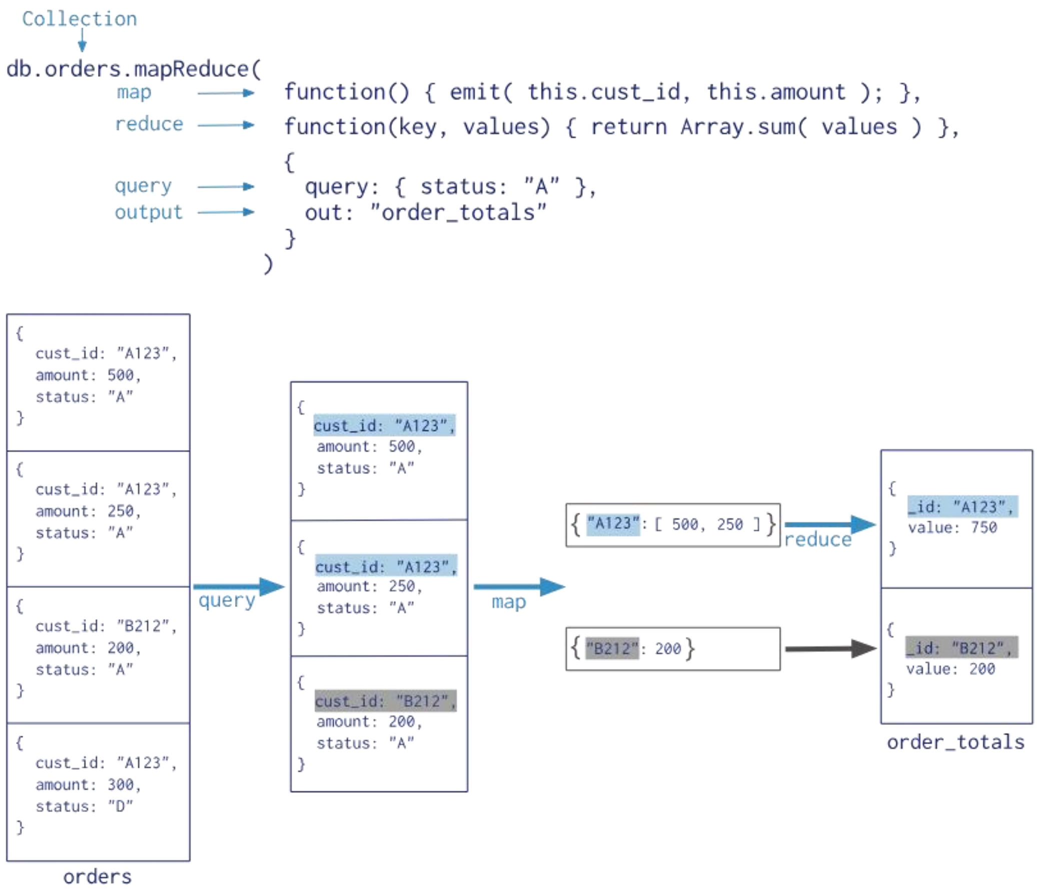
• Out: – It is used to specify the location of the map-reduce query output

. • 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.

Consider the following map-reduce operation:



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

Collection
map
reduce
query
output
{ cust_id: "A123", amount: 500, status: "A" }
{ cust_id: "A123", amount: 250, status: "A" }
{ cust_id: "B212", amount: 200, status: "A" }
{ cust_id: "A123", amount: 300, status: "D" }
orders
query
{ cust_id: "A123", amount: 500, status: "A" }
{ cust_id: "A123", amount: 250, status: "A" }
{ cust_id: "B212", amount: 200, status: "A" }
map
{ "A123": [500, 250] }
{ "B212": 200 }
reduce
{ _id: "A123", value: 750 }
{ _id: "B212", value: 200 }
order_totals

In this map-reduce operation, MongoDB applies the *map* phase to each input document (i.e. the documents in the collection that match the query condition). The map function emits key-value pairs. For those keys that have multiple values, MongoDB applies the *reduce* phase, which collects and condenses the aggregated data. MongoDB then stores the results in a collection. Optionally, the output of the reduce function may pass through a *finalize* function to further condense or process the results of the aggregation.

Consider the following document structure that stores book details author wise.

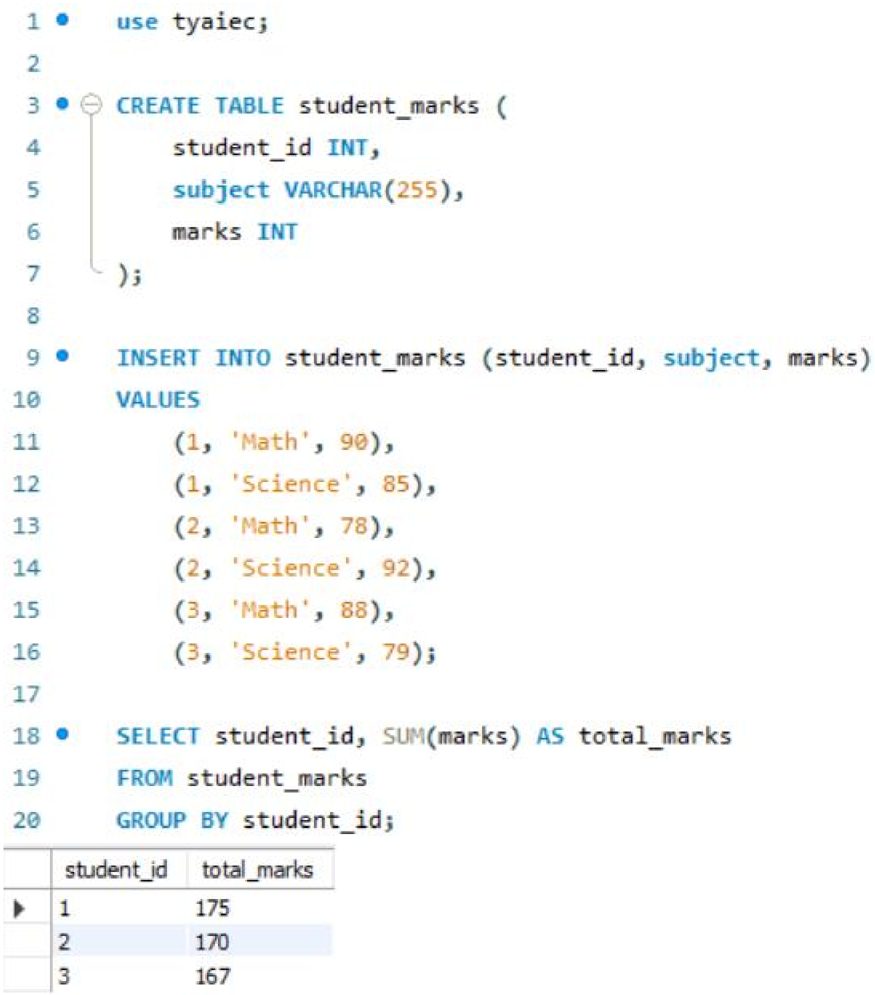
- > db.author.save({ "book_title" : "MongoDB Tutorial", "author_name" : "aparajita", "status" : "active", "publish_year": "2016" })
- > db.author.save({ "book_title" : "Software Testing Tutorial", "author_name" : "aparajita", "status" : "active", "publish_year": "2015" })
- > db.author.save({ "book_title" : "Node.js Tutorial", "author_name" : “Kritika", "status" : "active", "publish_year": "2016" })
- > db.author.save({ "book_title" : "PHP7 Tutorial", "author_name" : "aparajita", "status" : “passive", "publish_year": "2016" })

Perform Below Tasks using Mapreduce

1. To select all the active books
2. Group them together on the basis of author_name and Then count the number of books by each author

Let us consider school DB, where the student is a collection, and the collection contains documents, each of which includes a student’s name and the marks they received in a particular subject.

Write the Mapreduce Program to Calculate the Total Marks Secured by each student in all Subjects



```sql
use tyaiec;

CREATE TABLE student_marks (
    student_id INT,
    subject VARCHAR(255),
    marks INT
);

INSERT INTO student_marks (student_id, subject, marks)
VALUES
    (1, 'Math', 90),
    (1, 'Science', 85),
    (2, 'Math', 78),
    (2, 'Science', 92),
    (3, 'Math', 88),
    (3, 'Science', 79);

SELECT student_id, SUM(marks) AS total_marks
FROM student_marks
GROUP BY student_id;
```

student_id	total_marks
1	175
2	170
3	167

Example of Document:

```
db.students.find({}); { "_id" :
ObjectId("5a1f9ce431c157f3ec2aec39"), "name" : "Midhu",
"subject" : "science", "marks" : 68 } { "_id" :
ObjectId("5a1f9ce431c157f3ec2aec3a"), "name" : "Midhu",
"subject" : "maths", "marks" : 98 } { "_id" :
```

Questions:

1. MongoDB applies the _____ phase to each input document in mapreduce.

2. The map function emits —----------
3. MongoDB applies the _________ phase, which collects and condenses the aggregated data.
4. The output of the reduce function may pass through a ______ function to further condense or process the results of the aggregation.
5. _________ can return the results of a map-reduce operation as a document, or may write the results to collections.