Date

Assignment - B2

Title :- Design & develope mongoDB 'querie-s using CRUD operations.

Problem Statement :-

Design & develope mongo-DB. Querries using CRUD operations (use CRUD operation, SAVE method, logical operations, comparative operations & embedded doccements)

Objective :- To understand & implement CRUD operations in mongoDB.

S/W & H/W :- MongoDB, Fedora OS.

Outcome :- • Implement the commands on two tier.

• Implement the database in mongoDB

Theory :-

CRUD operations.

c - create - mongo store the data in form of JSON objects. so every

record for a collection, in mongo is called a document. If the collection does not currently exist, insert operations will create the collection we can ~~inter~~ insert documents into collection in 3 ways.

i) insert_one()
ii) insert_many()
iii) insert()

R - Read -

We can retrive the document from collection using 2 methods.

- find()
- Find_one()

Find() functional will return all the documents in that collection by default it returns a cursor objects.

find_one(): returns the first document in the collection.

D - Delete -

We can delete the document in that collection using following methods.

delete_one()
delete_many()

Both of these methods will return a DeleteResult object the general syntax As above methods

<method.name> (condition).

U - Update -

We can update the documents from the collection with the following methods.

update(). updateOne().
updateMany(). ~~update~~
replace-one().

The general syntax for all the above method is

<methodname> (condition, update, upsert = false, bypass_document_validation = false).

Logical query operators

$OR :- Joins query clauses with a logical OR returns all documents that match the condition of either clause.

{ $or [{ <expression> }, { <expression> }, ... }

$and - Joins query clauses with a logical AND returns all documents that match the conditions of both clauses

{ $and : [{ <expression> }, { <expression> }, ...] }

$not - Inverts the effect of a query expression & returns documents that do not match the query expression.

{ field : { $not : { <operator-expression> } } }

$nor - Joins query clauses with a logical NOR return all documents that fail to match both clauses.

{ $nor : [{ <expression> }, { <expression> }, ...] }

Conclusion :-

Implemented CRUD operations successfully using logical, comparator operators.

```
> use b2
switched to db b2
> db.student.insert({rollno : 201 , subject : "Physics" , marks : 40 })
WriteResult({ "nInserted" : 1 })
> db.student.insertMany([
... { rollno : 202 , subject : "Physics" , marks : 44 } ,
... { rollno : 201 , subject : "Chemistry" , marks : 40 } ,
... { rollno : 202 , subject : "Chemistry" , marks : 38 } ,
... { rollno : 201 , subject : "Math" , marks : 92 } ,
... { rollno : 202 , subject : "Math" , marks : 88 } ,
... { rollno : 203 , subject : "Physics" ,marks : 45 } ,
... { rollno : 203 , subject : "Chemistry" , marks :41 } ,
... { rollno : 203 , subject : "Math" , marks : 94 } ,
... { rollno : 204 , subject : "Physics" , marks : 32 } ,
... { rollno : 204 , subject : "Chemistry" , marks : 40 } ] )
{
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("5fa7eb298e612b224bb37dcf"),
                ObjectId("5fa7eb298e612b224bb37dd0"),
                ObjectId("5fa7eb298e612b224bb37dd1"),
                ObjectId("5fa7eb298e612b224bb37dd2"),
                ObjectId("5fa7eb298e612b224bb37dd3"),
                ObjectId("5fa7eb298e612b224bb37dd4"),
                ObjectId("5fa7eb298e612b224bb37dd5"),
                ObjectId("5fa7eb298e612b224bb37dd6"),
                ObjectId("5fa7eb298e612b224bb37dd7"),
                ObjectId("5fa7eb298e612b224bb37dd8")
        ]
}
>
>
> var mapFunction = function(){
... var key = this.rollno ;
... var value = { total_marks : this.marks , count : 1 , percentage : 0};
... emit (key,value);
... };
>
>
> var reduceFunction = function(key , values ) {
... var reducedObject = { total_marks : 0 ,count : 0 ,percentage : 0 } ;
... values.forEach( function(value) {
...     reducedObject.total_marks += value.total_marks;
...     reducedObject.count += value.count ;
... });
...
... return reducedObject ;
... };
>
>
> var finalizeFunction = function ( key,reducedValue ) {
...
... if(reducedValue.count > 0 )
```

```
...    reducedValue.percentage = reducedValue.total_marks / reducedValue.count ;
...
... return reducedValue;
... };
>
>
>
>
> db.student.mapReduce(    mapFunction ,   reduceFunction ,   {     out : "result",     finalize : finalizeFunction    } )
{
        "result" : "result",
        "timeMillis" : 172,
        "counts" : {
                "input" : 11,
                "emit" : 11,
                "reduce" : 4,
                "output" : 4
        },
        "ok" : 1
}
>
>
> db.result.find({}).pretty()
{
        "_id" : 201,
        "value" : {
                "total_marks" : 172,
                "count" : 3,
                "percentage" : 57.333333333333336
        }
}
{
        "_id" : 202,
        "value" : {
                "total_marks" : 170,
                "count" : 3,
                "percentage" : 56.666666666666664
        }
}
{
        "_id" : 203,
        "value" : {
                "total_marks" : 180,
                "count" : 3,
                "percentage" : 60
        }
}
{
        "_id" : 204,
        "value" : {
                "total_marks" : 72,
                "count" : 2,
                "percentage" : 36
        }
```

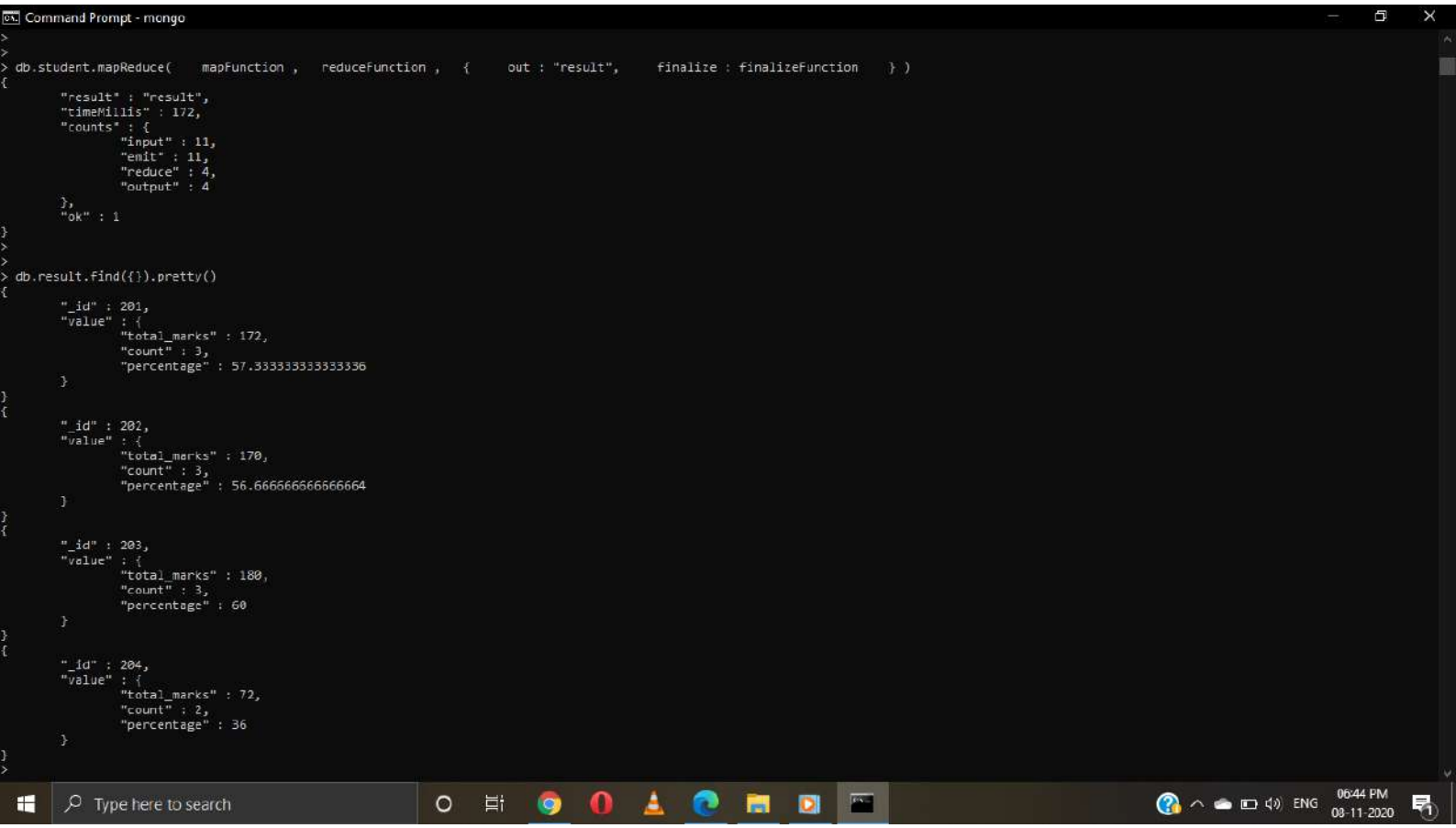
}
>

Command Prompt - mongo

```
> use b2
switched to db b2
> db.student.insert([rollno : 201 , subject : "Physics" , marks : 40 })
WriteResult({ "nInserted" : 1 })
> db.student.insertMany([
... { rollno : 202 , subject : "Physics" , marks : 44 } ,
... { rollno : 201 , subject : "Chemistry" , marks : 40 } ,
... { rollno : 202 , subject : "Chemistry" , marks : 38 } ,
... { rollno : 201 , subject : "Math" , marks : 92 } ,
... { rollno : 202 , subject : "Math" , marks : 88 } ,
... { rollno : 203 , subject : "Physics" ,marks : 45 } ,
... { rollno : 203 , subject : "Chemistry" , marks :41 } ,
... { rollno : 203 , subject : "Math" , marks : 94 } ,
... { rollno : 204 , subject : "Physics" , marks : 32 } ,
... { rollno : 204 , subject : "Chemistry" , marks : 40 } ] )
{
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("5fa7eb298e612b224bb37dcf"),
                ObjectId("5fa7eb298e612b224bb37dd0"),
                ObjectId("5fa7eb298e612b224bb37dd1"),
                ObjectId("5fa7eb298e612b224bb37dd2"),
                ObjectId("5fa7eb298e612b224bb37dd3"),
                ObjectId("5fa7eb298e612b224bb37dd4"),
                ObjectId("5fa7eb298e612b224bb37dd5"),
                ObjectId("5fa7eb298e612b224bb37dd6"),
                ObjectId("5fa7eb298e612b224bb37dd7"),
                ObjectId("5fa7eb298e612b224bb37dd8")
        ]
}
>
>
> var mapFunction = function(){
... var key = this.rollno ;
... var value = { total_marks : this.marks , count : 1 , percentage : 0};
... emit (key,value);
... };
>
>
> var reduceFunction = function(key , values ) {
... var reducedObject = { total_marks : 0 ,count : 0 ,percentage : 0 } ;
... values.forEach( function(value) {
...     reducedObject.total_marks += value.total_marks;
...     reducedObject.count += value.count ;
```

Command Prompt - mongo

```
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("5fa7eb298e612b224bb37dcf"),
                ObjectId("5fa7eb298e612b224bb37dd0"),
                ObjectId("5fa7eb298e612b224bb37dd1"),
                ObjectId("5fa7eb298e612b224bb37dd2"),
                ObjectId("5fa7eb298e612b224bb37dd3"),
                ObjectId("5fa7eb298e612b224bb37dd4"),
                ObjectId("5fa7eb298e612b224bb37dd5"),
                ObjectId("5fa7eb298e612b224bb37dd6"),
                ObjectId("5fa7eb298e612b224bb37dd7"),
                ObjectId("5fa7eb298e612b224bb37dd8")
        ]
}
>
>
> var mapFunction = function(){
... var key = this.rollno ;
... var value = { total_marks : this.marks , count : 1 , percentage : 0};
... emit (key,value);
... };
>
>
> var reduceFunction = function(key , values ) {
... var reducedObject = { total_marks : 0 ,count : 0 ,percentage : 0 } ;
... values.forEach( function(value) {
...     reducedObject.total_marks += value.total_marks;
...     reducedObject.count += value.count ;
... });
...
... return reducedObject ;
... };
>
>
> var finalizeFunction = function ( key,reducedValue ) {
...
... if(reducedValue.count > 0 )
...     reducedValue.percentage = reducedValue.total_marks / reducedValue.count ;
...
... return reducedValue;
... };
>
>
>
```


Command Prompt - mongo
> db.student.mapReduce(mapFunction , reduceFunction , { out : "result", finalize : finalizeFunction })
{
"result" : "result",
"timeMillis" : 172,
"counts" : {
"input" : 11,
"emit" : 11,
"reduce" : 4,
"output" : 4
},
"ok" : 1
}
> db.result.find({}).pretty()
{
"_id" : 201,
"value" : {
"total_marks" : 172,
"count" : 3,
"percentage" : 57.333333333333336
}
}
{
"_id" : 202,
"value" : {
"total_marks" : 170,
"count" : 3,
"percentage" : 56.666666666666664
}
}
{
"_id" : 203,
"value" : {
"total_marks" : 180,
"count" : 3,
"percentage" : 60
}
}
{
"_id" : 204,
"value" : {
"total_marks" : 72,
"count" : 2,
"percentage" : 36
}
}
Type here to search
ENG
06:44 PM
08-11-2020