Name of the student:	Tanmay Prashant Rane	**Roll No.**	8031
Practical Number:	5	**Date of Practical:**	
Relevant CO's	**At the end of the course students will be able to use tools like hadoop and NoSQL to solve big data related problems.**		
Sign here to indicate that you have read all the relevant material provided before attempting this practical		**Sign:**	

Practical grading using Rubrics

Indicator	**Very Poor**	**Poor**	**Average**	**Good**	**Excellent**
Timeline (2)	More than a session late (0)	NA	NA	NA	Early or on time (2)
Code design (2)	N/A	Very poor code design with no comments and indentation(0.5)	Poor code design with very comments and indentation (1)	Design with good coding standards (1.5)	Accurate design with better coding satndards (2)
Performance (4)	Unable to perform the experiment (0)	Able to partially perform the experiment (1)	Able to perform the experiment for certain use cases (2)	Able to perform the experiment considering most of the use cases (3)	Able to perform the experiment considering all use cases (4)
Postlab (2)	No Execution(0)	N/A	Partially Executed (1)	N/A	Fully Executed (2)

Total Marks (10)	**Sign of instructor with date**

Practical

Course title: Big Data Analytics
Course term: 2019-2020

Problem Statement: Perform CRUD operations in MongoDB

Theory:Explain different CRUD Operations

Create Operations

Create or insert operations add new documents to a collection. If the collection does not currently exist, insert operations will create the collection.

MongoDB provides the following methods to insert documents into a collection:

db.collection.insertOne() New in version 3.2

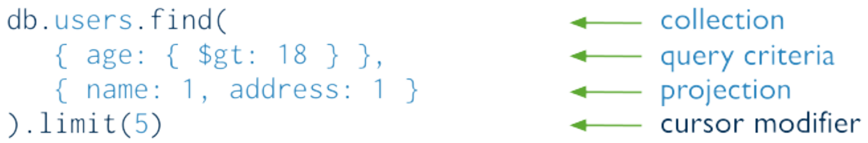
db.collection.insertMany() New in version 3.2

In MongoDB, insert operations target a single collection. All write operations in MongoDB are atomic on the level of a single document.

Read operations retrieves documents from a collection; i.e. queries a collection for documents. MongoDB provides the following methods to read documents from a collection:

db.collection.find()

You can specify query filters or criteria that identify the documents to return.



```
db.users.find(
   { age: { $gt: 18 } },
   { name: 1, address: 1 }
).limit(5)
```

Query Documents

Query on Embedded/Nested Documents

Query an Array

Query an Array of Embedded Documents

Update Operations: Update operations modify existing documents in a collection. MongoDB provides the following methods to update documents of a collection:

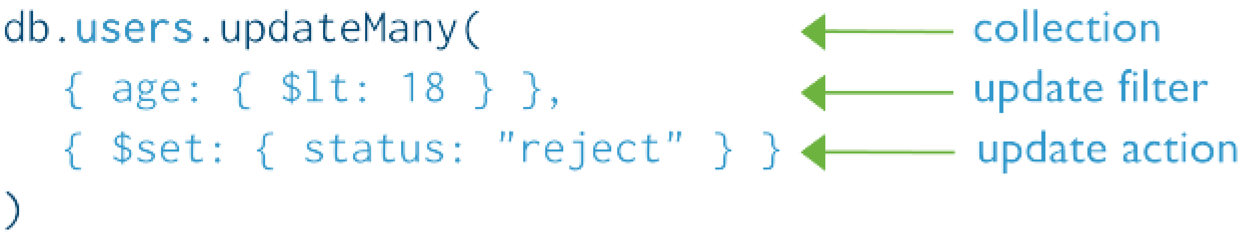
db.collection.updateOne() New in version 3.2

db.collection.updateMany() New in version 3.2

db.collection.replaceOne() New in version 3.2

In MongoDB, update operations target a single collection. All write operations in MongoDB are atomic on the level of a single document.

You can specify criteria, or filters, that identify the documents to update. These filters use the same syntax as read operations.



```
db.users.updateMany(
   { age: { $lt: 18 } },
   { $set: { status: "reject" } }
)
```

Delete Operations

Delete operations remove documents from a collection. MongoDB provides the following methods to delete documents of a collection:

db.collection.deleteOne()

db.collection.deleteMany()

In MongoDB, delete operations target a single collection. All write operations in MongoDB are atomic on the level of a single document.

You can specify criteria, or filters, that identify the documents to remove. These filters use the same syntax as read operations.

Code:

Code of CRUD operations in MongoDB

1. Create documents for following data in collection called media.

Type	Title	ISBN	Publisher	Author
Book	Def. guide	978-1-482-0	Apress	"Hows, David","Plugge,Eelco","Membrey, Peter","Hawkins, Tim"
Book	A text book on automata theory	978-2-482-0	Foundation books	"Nasir, S.F.B","Srimani,P.K"
Book	MongoDB in Action	978-3-482-0	Manning Publication	"Banker, Kyle"
Book	NoSQL for dummies	978-4-482-0	Wiley	"Fowler, Adam"
Book	Big Data Analytics	978-5-482-0	Wiley	"Shankarmani, Radha"

code for creating documents:

```
0) mongo

0) use mydb

0) db.createCollection("media")

1) db.media.insertMany([
... { Type: "Book", Title: "Def.guide", ISBN: "978-1-482-0", Publisher: "Apress", Author:
["Hows,David","Plugge,Eelco","Membrey,Peter","Hawkins,Tim"]},
... { Type: "Book", Title: "A text book on automata theory", ISBN: "978-2-482-0", Publisher: "Foundation books",
Author: ["Nasir,S.F.B","Srimani,P.K"]},
... { Type: "Book", Title: "MongodDB in Action", ISBN: "978-3-482-0", Publisher: "Manning Publication", Author:
["Banker,Kyle"]},
... { Type: "Book", Title: "NoSQL for dummies", ISBN: "978-4-482-0", Publisher: "Wiley", Author:
["Fowler,Adam"]},
... { Type: "Book", Title: "Big Data Analytics", ISBN: "978-5-482-0", Publisher: "Wiley", Author:
["Shankarmani,Radha"]}
... ])
```

2. Insert a document with type=CD, Artist=Nirvana, Title=Never Mind, Tracklist=[Track:1,Title:Smells Like Teen Spirit,length:5:02,Track:2,Title:In Bloom,length:4:15] in same collection named media.

Query Code

```
2) db.media.insertOne({ Type: "CD", Artist : "Nirvana", Title: "Never Mind", Tracklist : [{ Track: 1,Title:
"Smells like Teen Spirit", length: "5:02"}, { Track: 2, Title: "In Bloom", length: "4:15" }]})
```

3. Find all documents in collection named media.

Query Code

```
3)db.media.find( {} )
```

4. Find documents where publisher is Wiley

Query Code

```
4)db.media.find({Publisher:"Wiley"})
```

5. Find titles of CDs whose artist is Nirvana.

Query Code

```
5)db.media.find({Artist:"Nirvana",Type:"CD"},{Title:1,_id:0})
```

6. Find all documents sorted in descending order.

Query Code

```
6)db.media.find({}).sort({Title:-1})
```

7. Find only 3 documents of book type.

Query Code

```
7)db.media.find({Type:"Book"}).limit(3)
```

8. Find last 3 documents from collection named media

Query Code

```
8)db.media.find().skip(db.media.count() - 3)
```

PostLab:

Compute customerwise total amount on given dataset using map reduce for the customers with status as "A" and store this result in a document called Order_total.

code for mapreduce function of MongoDB

Calculate number of times the site has been visited using mapreduce in MongoDB on given dataset.

Code for the question

```
1) mongoimport --jsonArray --db mydb --collection postlab --file
"/media/tanmay/Data/SEM-8/BDA/EXP5/MongoMapreducedata"

2)MongoDB Enterprise > var f1 = function(){
... emit(this.cust_id,this.price);
... };

3)MongoDB Enterprise > var reduce = function(CustId, Prices){
... return Array.sum(Prices);
... };

4)MongoDB Enterprise > db.postlab.mapReduce(
... f1,
... reduce,
... {query:{status:"A"},out:"map_reduce_postlab"}
... )

5)db.map_reduce_postlab.find({})
o/p:
{ "_id" : "A123", "value" : 750 }
{ "_id" : "B212", "value" : 200 }
```

```
1)MongoDB Enterprise > db.postlab2.insertMany([
... {"url" : "www.google.com","date" : ISODate("2016-02-17T05:33:17.073Z"),"trash_data" : 5},
... {"url" : "www.frcrce.ac.in","date" : ISODate("2016-02-17T05:33:17.075Z"),"trash_data" : 13},
... {"url" : "www.google.com","date" : ISODate("2016-02-17T05:33:17.075Z"),"trash_data" : 1},
... {"url" : "www.frcrce.ac.in","date" : ISODate("2016-02-17T05:33:17.075Z"),"trash_data" : 69},
... {"url" : "www.fragnel.edu.in","date" : ISODate("2016-02-17T05:33:17.076Z"),"trash_data" : 256}
... ])

2)MongoDB Enterprise > var mapf = function(){ emit(this.url,1); };

3)MongoDB Enterprise > var redf = function(URL, Visited){ return Array.sum(Visited); };

4)MongoDB Enterprise > db.postlab2.mapReduce( mapf, redf, {out:"map_reduce_postlab2"} )

5)MongoDB Enterprise > db.map_reduce_postlab2.find({})

O/P:
{ "_id" : "www.fragnel.edu.in", "value" : 1 }
{ "_id" : "www.frcrce.ac.in", "value" : 2 }
{ "_id" : "www.google.com", "value" : 2 }
```