Name of the student:	Yash Sankpal	**Roll No.**	8695
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 Exe- cuted (1)	N/A	Fully Executed (2)

Total Marks (10)	Sign of instructor with date

Practical

COURSE TITLE: BIG DATA
ANALYTICS COURSE TERM: 2021-
2022

Problem Statement: Perform CRUD operations in MongoDB

Theory:Explain different CRUD Operations

CRUD operations describe the conventions of a user-interface that let users view, search, and modify parts of the database.

MongoDB documents are modified by connecting to a server, querying the proper documents, and then changing the setting properties before sending the data back to the database to be updated. CRUD is data-oriented, and it's standardized according to HTTP action verbs.

When it comes to the individual CRUD operations:

- The Create operation is used to insert new documents in the MongoDB database.
- The Read operation is used to query a document in the database.
- The Update operation is used to modify existing documents in the database.
- The Delete operation is used to remove documents in the database.

For MongoDB CRUD, if the specified collection doesn't exist, the **create** operation will create the collection when it's executed. Create operations in MongoDB target a single collection, not multiple collections. Insert operations in MongoDB are atomic on a single document level.
MongoDB provides two different create operations that you can use to insert documents into a collection:
db.collection.insertOne()
db.collection.insertMany()

The **read** operations allow you to supply special query filters and criteria that let you specify which documents you want. The MongoDB documentation contains more information on the available query filters. Query modifiers may also be used to change how many results are returned.

MongoDB has two methods of reading documents from a collection:
db.collection.find()
db.collection.findOne()

Like create operations, **update** operations operate on a single collection, and they are atomic at a single document level. An update operation takes filters and criteria to select the documents you want to update.

You should be careful when updating documents, as updates are permanent and can't be rolled back. This applies to delete operations as well.

For MongoDB CRUD, there are three different methods of updating documents:

db.collection.updateOne()
db.collection.updateMany()
db.collection.replaceOne()

Delete operations operate on a single collection, like update and create operations. Delete operations are also atomic for a single document. You can provide delete operations with filters and criteria in order to specify which documents you would like to delete from a collection. The filter options rely on the same syntax that read operations utilize.

MongoDB has two different methods of deleting records from a collection:

db.collection.deleteOne()
db.collection.deleteMany()

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 Publica-tion	"Banker, Kyle"
Book	NoSQL for dummies	978-4-482-0	Wiley	"Fowler, Adam"

code for creating documents:

```
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: 'MongoDB in Action',
        ISBN: '978-3-482-0',
        Publisher: 'Manning Pubication',
```

```
        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.

code for creating document:

```
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:

```
db.media.find();
```

4. Find documents where publisher is Wiley

Query Code:

```
db.media.find({ Publisher: 'Wiley' });
```

5. Find titles of CDs whose artist is Nirvana.

Query Code:

```
db.media.find({ Artist: /Nirvana/ });
```

6. Find all documents sorted in descending order.

Query Code:

```
db.media.find().sort({ $natural: -1 });
```

7. Find only 3 documents of book type.

Query Code:

```
db.media.find({ Type: 'Book' }).limit(3);
```

8. Find last 3 documents from collection named media.

Query Code:

```
db.media.find().limit(3).sort({ $natural: -1 });
```

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

```
db.customer.mapReduce(function(){emit(this.cur_id,this.amount)},function(key,values){return Array.sum(values)},{query:{status:"A
},out:"output_file")
```

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

Code for the question

```
db.site.mapReduce(function(){emit(this.url,1)},function(key,values){return Array.sum(values)},{query:{},out:"output"})
```