Work Integrated Learning Programmes

BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Computer Science & Information Systems

Big Data System – MongoDB Lab Sheet:2

Data Manipulations in MongoDB

1. Objective

Students should be able to

A. Insert documents into the collection
B. Update the document
C. Delete the document

2. Steps to be performed

A. Insert documents into the collection

A1. The "insert" command

"insert" command is used to add a new document into the collection.
For example,

Basic syntax of **INSERT** statement is as follows −

```
>db.collection_name.insert({//document})
```

Example, let's add a new document with attributes like id, title, reviews into books collection.

```
> db.books.insert({"id":1, "title":"Cloud computing",
"tags":["cloud", "prog"], "reviews":5})
```

Example

A2. The "find" command

"find" command is used to retrieve the documents from the collection.

Example

```
> db.books.find({})

{"_id" : ObjectId("5dc1a8e9c85994f17d1b97fe"), "id" : 1, "title" :
"Cloud computing", "tags" : [ "cloud", "prog" ], "reviews" : 5 }
```

A3. The "multi insert" command

The [ ] operator can be used to insert the multiple documents into the collection.

Example

```
> db.books.insert([{"id":2, "title":"IDS", "tags":["DataScience"],
"reviews":3},
{"id":3, "title":"Data Analytics", "reviews":5},
{"id":4, "title":"Big Data Systems", "reviews":5}])
```

A4. The "pretty" command

The documents retrieved by using "find" command can be shown properly formatted using "pretty" command.

Example

```
> db.books.find().pretty()
{
        "_id" : ObjectId("5dc1a8e9c85994f17d1b97fe"),
        "id" : 1,
        "title" : "Cloud computing",
        "tags" : [
                "cloud",
                "prog"
        ],
        "reviews" : 5
}
{
        "_id" : ObjectId("5dc1a989c85994f17d1b97ff"),
        "id" : 2,
        "title" : "IDS",
        "tags" : [
                "DataScience"
        ],
        "reviews" : 3
}
{
        "_id" : ObjectId("5dc1a989c85994f17d1b9800"),
        "id" : 3,
        "title" : "Data Analytics",
        "reviews" : 5
}
{
        "_id" : ObjectId("5dc1a989c85994f17d1b9801"),
        "id" : 4,
        "title" : "Big Data Systems",
        "reviews" : 5
}
```

A4. The "save" command

"Save" command can also be used to insert the document into the collection.

Example

```
> db.books.save({"id":5, "title":"Big Data", "reviews":3})
WriteResult({ "nInserted" : 1 })
```

You can verify the inserted document using find command

```
> db.books.find()
{ "_id" : ObjectId("5dc1a8e9c85994f17d1b97fe"), "id" : 1, "title" : "Cloud
computing", "tags" : [ "cloud", "prog" ], "reviews" : 5 }
{ "_id" : ObjectId("5dc1a989c85994f17d1b97ff"), "id" : 2, "title" : "IDS",
"tags" : [ "DataScience" ], "reviews" : 3 }
{ "_id" : ObjectId("5dc1a989c85994f17d1b9800"), "id" : 3, "title" : "Data
Analytics", "reviews" : 5 }
{ "_id" : ObjectId("5dc1a989c85994f17d1b9801"), "id" : 4, "title" : "Big
Data Systems", "reviews" : 5 }
{ "_id" : ObjectId("5dc1aba2baa5bd2fd56dc9ed"), "id" : 5, "title" : "Big
Data", "reviews" : 3 }
```

B. Update the document

B1. Add new field in the existing document

"Update" command can be used to add a new field into document matching the given criteria.

Example

```
> db.books.update({"id":5}, {$set:{"author":"Celin"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

You can verify the inserted document using find command

```
> db.books.find()
{ "_id" : ObjectId("5dc1a8e9c85994f17d1b97fe"), "id" : 1, "title" : "Cloud
computing", "tags" : [ "cloud", "prog" ], "reviews" : 5 }
{ "_id" : ObjectId("5dc1a989c85994f17d1b97ff"), "id" : 2, "title" : "IDS",
"tags" : [ "DataScience" ], "reviews" : 3 }
{ "_id" : ObjectId("5dc1a989c85994f17d1b9800"), "id" : 3, "title" : "Data
Analytics", "reviews" : 5 }
{ "_id" : ObjectId("5dc1a989c85994f17d1b9801"), "id" : 4, "title" : "Big
Data Systems", "reviews" : 5 }
{ "_id" : ObjectId("5dc1aba2baa5bd2fd56dc9ed"), "id" : 5, "title" : "Big
Data", "reviews" : 3, "author" : "Celin" }
```

B2. Update the field in the existing document

"Update" command can be used to update a field from document as follows

Example

```
> db.books.update({"id":5}, {$set:{"title":"New Big Data"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

You can verify the inserted document using find command

```
> db.books.find()
{ "_id" : ObjectId("5dc1a8e9c85994f17d1b97fe"), "id" : 1, "title" : "Cloud
computing", "tags" : [ "cloud", "prog" ], "reviews" : 5 }
{ "_id" : ObjectId("5dc1a989c85994f17d1b97ff"), "id" : 2, "title" : "IDS",
"tags" : [ "DataScience" ], "reviews" : 3 }
{ "_id" : ObjectId("5dc1a989c85994f17d1b9800"), "id" : 3, "title" : "Data
Analytics", "reviews" : 5 }
{ "_id" : ObjectId("5dc1a989c85994f17d1b9801"), "id" : 4, "title" : "Big
Data Systems", "reviews" : 5 }
{ "_id" : ObjectId("5dc1aba2baa5bd2fd56dc9ed"), "id" : 5, "title" : "New
Big Data", "reviews" : 3 }
```

B3. Remove field from the existing document

"Update" command can be used to delete a field into document as follows

Example

```
> db.books.update({"id":5}, {$unset:{"author":""}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

You can verify the inserted document using find command

```
> db.books.find()
{ "_id" : ObjectId("5dc1a8e9c85994f17d1b97fe"), "id" : 1, "title" : "Cloud
computing", "tags" : [ "cloud", "prog" ], "reviews" : 5 }
{ "_id" : ObjectId("5dc1a989c85994f17d1b97ff"), "id" : 2, "title" : "IDS",
"tags" : [ "DataScience" ], "reviews" : 3 }
{ "_id" : ObjectId("5dc1a989c85994f17d1b9800"), "id" : 3, "title" : "Data
Analytics", "reviews" : 5 }
{ "_id" : ObjectId("5dc1a989c85994f17d1b9801"), "id" : 4, "title" : "Big
Data Systems", "reviews" : 5 }
{ "_id" : ObjectId("5dc1aba2baa5bd2fd56dc9ed"), "id" : 5, "title" : "Big
Data", "reviews" : 3 }
```

C. Delete the document

C1. The "remove" command

"remove" command is used to remove the document matching the criteria.

Example

```
> db.books.remove({"id":5})
WriteResult({ "nRemoved" : 1 })
```

You can verify the inserted document using find command

```
> db.books.find()
{ "_id" : ObjectId("5dc1a8e9c85994f17d1b97fe"), "id" : 1, "title" : "Cloud
computing", "tags" : [ "cloud", "prog" ], "reviews" : 5 }
{ "_id" : ObjectId("5dc1a989c85994f17d1b97ff"), "id" : 2, "title" : "IDS",
"tags" : [ "DataScience" ], "reviews" : 3 }
{ "_id" : ObjectId("5dc1a989c85994f17d1b9800"), "id" : 3, "title" : "Data
Analytics", "reviews" : 5 }
{ "_id" : ObjectId("5dc1a989c85994f17d1b9801"), "id" : 4, "title" : "Big
Data Systems", "reviews" : 5 }
```

C2. The "remove" command with "justOnce" option

"remove" command supports a flag "justOnce" which just deletes the first document matching the criteria.

Example

```
> db.books.remove({"reviews":5}, {"justOne":true})
WriteResult({ "nRemoved" : 1 })
```

You can verify the inserted document using find command

```
> db.books.find()
{ "_id" : ObjectId("5dc1a989c85994f17d1b97ff"), "id" : 2, "title" : "IDS",
"tags" : [ "DataScience" ], "reviews" : 3 }
{ "_id" : ObjectId("5dc1a989c85994f17d1b9800"), "id" : 3, "title" : "Data
Analytics", "reviews" : 5 }
{ "_id" : ObjectId("5dc1a989c85994f17d1b9801"), "id" : 4, "title" : "Big
Data Systems", "reviews" : 5 }
```

3. Outputs/Results

Students should be able to appreciate the usage of following commands

- Insert
- Update
- Remove

4. Observations

Students should carefully observe the syntax of insert, update and remove commands

5. References

- MongoDB documentation