CSE 4410

DATABASE MANAGEMENT SYSTEMS II LAB

Notes On: MongoDB

PREPARED BY :

DR. ABU RAIHAN MOSTOFA KAMAL || PROFESSOR
raihan.kamal@iut-dhaka.edu
ZANNATUN NAIM SRISTY || LECTURER
zannatunnaim@iut-dhaka.edu
MD. RAFID HAQUE || LECTURER
rafidhaque@iut-dhaka.edu

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ISLAMIC UNIVERSITY OF TECHNOLOGY

MARCH 7, 2023

1 Getting Started with MongoDB

1.1 Basics

- To open mongo shell in command prompt

  ```
  > mongosh
  ```

- To see available operation

  ```
  > help
  ```

- To see available database

  ```
  > show dbs
  ```

- To switch into a a database for further operation

  ```
  > use <database_name>
  ```

- To see details of a database

  ```
  > db.stats()
  ```

- To see the collections of the database

  ```
  > show collection
  ```

- To insert a single data

  ```
  > db.<collection_name>.insertOne({<attribute_name>:<value>,......,
  <attribute_name>:<value>})
  ```

  we don't necessarily to have an existing collection to insert data. If there is no collection with the collection name it will create a new collection and insert the data there. So if we want create a new collection we can simply do this by inserting a new entry.

- To insert an entry with an array

  ```
  > db.<collection_name>.insertOne({<attribute_name>:<value>,......,
  <attribute_name>:[<value>,......,<value>]})
  ```

- To insert an entry with an array

  ```
  > db.<collection_name>.insertOne({<attribute_name>:<value>,......,
  <attribute_name>:[<value>,......,<value>]})
  ```

- To insert an entry with an array of object/ nested entry

```
> db.<collection_name>.insertOne({<attribute_name>:<value>,......,
<attribute_name>:[{<attribute_name>:<value>,......,<attribute_name>:
<value>},{<attribute_name>:<value>,......,<attribute_name>:<value>}]})
```

- To insert many data at a time

```
> db.<collection_name>.insertMany({<attribute_name>:<value>,......,
<attribute_name>:<value>},.....,{<attribute_name>:<value>,......,
<attribute_name>:<value>})
```

- To fetch the entries of a collection

```
> db.<collection_name>.find()
```

This command will show first 20 entries of the collection. To see 20 more entries type -> it.

- To fetch one entry from a collection

```
> db.<collection_name>.findOne()
```

- To fetch entries from a collection with condition

```
> db.<collection_name>.findOne()
```

- To fetch entries with certain attributes

```
> db.<collection_name>.find({<attribute_name>:<value>,......,
<attribute_name>:<value>}{<attribute_name>:1,...,<attribute_name>:1})
```

- To fetch entries with a certain value in the array attribute

```
> db.<collection_name>.find({<array_attribute_name>: <value>})
```

- To fetch entries with multiple certain values in the array attribute

```
> db.<collection_name>.find({<array_attribute_name>: {$all:[<value>,
...,<values>]}})
```

- To fetch entries with that contains only certain values in the array attribute

```
> db.<collection_name>.find({<array_attribute_name>: [<value>]})
```

- To fetch entries with a certain value in the nested attribute

```
> db.<collection_name>.find({"<outer_attribute_name>.<inner_attribute
_name>":<value>})
```

- To delete a document

```
> db.<collection_name>.deleteOne({<attribute_name>:<value>})
```

- To delete many document

```
> db.<collection_name>.deleteMany({<attribute_name>:<value>})
```

- To update a document

```
> db.<collection_name>.updateOne({<attribute_name>:<value>},{$set: {
<attribute_name>:<value>,..., <attribute_name>:<value>}})
```

- To update many document

```
> db.<collection_name>.updateMany({<attribute_name>:<value>},{$set: {
<attribute_name>:<value>,..., <attribute_name>:<value>}})
```

1.2 Method Chaining:

- To count how many entries we have got

```
> db.<collection_name>.find({<attribute_name>:<value>,......,
<attribute_name>:<value>}).count()
```

- To limit the entries we want to get

```
> db.<collection_name>.find({<attribute_name>:<value>,......,
<attribute_name>:<value>}).limit(<value>)
```

- To limit the entries we want to get

```
> db.<collection_name>.find().sort({<attribute_name><1/-1>,...,
<attribute_name>:<1/-1>})
```

to sort in ascending order use 1 and for descending order use -1

1.3 Special Query Operator: (begins with $)

- greater than operator, ($gt)

```
> db.<collection_name>.find({<attribute_name>: {$gt: <value>}})
```

- less than operator, ($lt)
- greater than or equal, ($gte)
- less than or equal, ($lte)

- or operator, ($or)

```
> db.<collection_name>.find({$or: [{<attribute_name>: <value>},
{<attribute_name>: <value>}]})
```

- and operator, ($and)
- in operator, ($in)

```
> db.<collection_name>.find({<attribute_name>:{$in: [<value>,...,
<value>]}})
```

- not in operator, ($nin)
- in operator, ($inc)

```
> db.<collection_name>.updateOne({<attribute_name>:<value>},{$inc: {
<attribute_name>:<value>}})
```

to increment assign a positive value and for decrement assign a negative value

- pull operator, ($pull)

```
> db.<collection_name>.updateOne({<attribute_name>:<value>},{$pull: {
<array_attribute_name>:<value>}})
```

- push operator, ($push)
- each operator, ($each)

```
> db.<collection_name>.updateOne({<attribute_name>:<value>},{$push: {
<array_attribute_name>:{$each: [<value>,...,<value>]}}})
```