

Practical No . 1

Aim : Practical of Data collection, Data curation and management for Unstructured data(NoSQL/MongoDB) .

Theory :

- **MongoDB –**

- 10gen software company began developing MongoDB in 2007 as a component of a planned [platform as a service](#) product. In 2009, the company shifted to an open-source development model, with the company offering commercial support and other services. In 2013, 10gen changed its name to MongoDB Inc.
- MongoDB can be used as a [file system](#), called [GridFS](#), with load balancing and data replication features over multiple machines for storing files.
- MongoDB provides three ways to perform aggregation: the aggregation pipeline, the map-reduce function, and single-purpose aggregation methods.
- MongoDB provides high availability with replica sets. A replica set consists of two or more copies of the data. Each replica-set member may act in the role of primary or secondary replica at any time. All writes and reads are done on the primary replica by default.

- **NoSQL –**

- A NoSQL (originally referring to "non-[SQL](#)" or "non-relational") [database](#) provides a mechanism for [storage](#) and [retrieval](#) of data that is modeled in means other than the tabular relations used in [relational databases](#).
- The data structures used by NoSQL databases (e.g. [key-value pair](#), [wide column](#), [graph](#), or [document](#)) are different from those used by default in relational databases, making some operations faster in NoSQL.
- The particular suitability of a given NoSQL database depends on the problem it must solve. Sometimes the data structures used by NoSQL databases are also viewed as "more flexible" than relational database tables.

- **Data Collection –**

- **Data collection** is the process of gathering and [measuring information](#) on targeted variables in an established system, which then enables one to answer relevant questions and evaluate outcomes.
- **Data** collection is a [research](#) component in all study fields, including [physical](#) and [social sciences](#), [humanities](#), and [business](#).

- The goal for all data collection is to capture quality evidence that allows analysis to lead to the formulation of convincing and credible answers to the questions that have been posed.

- **Data Curation –**

- **Data curation** is the organization and integration of [data](#) collected from various sources.
- It involves annotation, publication and presentation of the data such that the value of the data is maintained over time, and the data remains available for reuse and preservation.
- Data curation includes "all the processes needed for principled and [controlled data](#) creation, maintenance, and [management](#), together with the capacity to add value to data".

- **Data Management –**

- **Data management** is the practice of collecting, keeping, and using data securely, efficiently, and cost-effectively.
- The goal of data management is to help people, organizations, and connected things optimize the use of data within the bounds of policy and regulation so that they can make decisions and take actions that maximize the benefit to the organization.
- A robust data management strategy is becoming more important than ever as organizations increasingly rely on intangible assets to create value.

- **Unstructured Data -**

- **Unstructured data** (or **unstructured information**) is information that either does not have a pre-defined [data model](#) or is not organized in a pre-defined manner.
- Unstructured information is typically [text](#)-heavy, but may contain data such as dates, numbers, and facts as well.
- This results in irregularities and [ambiguities](#) that make it difficult to understand using traditional programs as compared to data stored in fielded form in databases or [annotated](#) ([semantically tagged](#)) in documents.

Excel Sheet :

Serial_code	Mobile_Model	Operating_System	Internal Storage	Ram	Rear Camera Pixel
7575	Samsung M31	Android 10	64 GB	3 GB	20 MP
4234	Apple iPhone 12	iOS 14	512 GB	4GB	24 MP
1639	Moto G5	Android 8	64 GB	4 GB	26 MP
7364	Samsung Galaxy S21	Android 10	128 GB	6 GB	64 MP
8648	One Plus 8	Android 11	64 GB	8 GB	32 MP
1289	Apple iPhone 6	iOS 9	128 GB	2 GB	8 MP
1209	Redmi Note 9	Android 10	128 GB	6 GB	13 MP
6324	RealMe 3	Android 10	128 GB	3 GB	20 MP
8644	Apple iPhone XR	iOS 12	64 GB	2 GB	12 MP
8430	Asus ROG 2	Android 10	128 GB	8 GB	20 MP

❖ ***Database in MongoDB :***

1. Database creation

Syntax : **use DataBase_Name**

```

> use Pract1DS
switched to db Pract1DS
>

```

2. Current Database

Syntax : **db**

```

> db
Pract1DS
>

```

3. Available Databases**Syntax : show dbs**

```
> show dbs
admin      0.000GB
config     0.000GB
local      0.000GB
pract1DS   0.000GB
rc         0.000GB
test       0.000GB
>
```

4. Collection creation and Document insertion in Database**Syntax : db.collection_name.insert({key : value})**

```
> use pract1DS
switched to db pract1DS
> db.Student.insert({Roll_no:53 ,name : "Jay",marks:70})
WriteResult({ "nInserted" : 1 })
> db.Student.insert({Roll_no:20 ,name : "mrunal",marks:89})
WriteResult({ "nInserted" : 1 })
> db.Student.insert({Roll_no:31 ,name : "yash",marks:79})
WriteResult({ "nInserted" : 1 })
```

5. Drop database**Syntax : db.dropDatabase()**

```
> db.dropDatabase()
{ "dropped" : "pract1DS", "ok" : 1 }
```

```
> show dbs
admin      0.000GB
config     0.000GB
local      0.000GB
rc         0.000GB
test       0.000GB
>
```

❖ Collections in MongoDB :

1. Creation of Collection (Method 1 for creating collection inserting document directly):

Syntax : **db.collection_name.insert({key : value})**

```
> use mobile_info
switched to db mobile_info
> db.mob_info.find()
{ "_id" : ObjectId("602ac4894962019b8d25569b"), "serial_code" : 7575, "Mobile_ModelName" : "Samsung N31", "Operating_System" : "Android 10", "Internal_Storage" : "64 GB", "RAM" : "3 GB", "RearCamera" : "20 MP" }
{ "_id" : ObjectId("602ac5de4962019b8d25569c"), "serial_code" : 4234, "Mobile_ModelName" : "Apple iPhone 12", "Operating_System" : "iOS 14", "Internal_Storage" : "512 GB", "RAM" : "4 GB", "RearCamera" : "24 MP" }
{ "_id" : ObjectId("602ac5eb4962019b8d25569d"), "serial_code" : 1639, "Mobile_ModelName" : "Moto G5", "Operating_System" : "Android 8", "Internal_Storage" : "64 GB", "RAM" : "4 GB", "RearCamera" : "26 MP" }
{ "_id" : ObjectId("602ac5ee4962019b8d25569e"), "serial_code" : 7364, "Mobile_ModelName" : "Samsung Galaxy S21", "Operating_System" : "Android 10", "Internal_Storage" : "128 GB", "RAM" : "6 GB", "RearCamera" : "64 MP" }
{ "_id" : ObjectId("602ac5fa4962019b8d25569f"), "serial_code" : 8648, "Mobile_ModelName" : "One Plus 8", "Operating_System" : "Android 11", "Internal_Storage" : "64 GB", "RAM" : "8 GB", "RearCamera" : "32 MP" }
{ "_id" : ObjectId("602ac5fc4962019b8d2556a0"), "serial_code" : 1289, "Mobile_ModelName" : "Apple iPhone 6", "Operating_System" : "iOS 9", "Internal_Storage" : "128 GB", "RAM" : "2 GB", "RearCamera" : "8 MP" }
{ "_id" : ObjectId("602ac5fd4962019b8d2556a1"), "serial_code" : 1209, "Mobile_ModelName" : "Redmi Note 9", "Operating_System" : "Android 10", "Internal_Storage" : "128 GB", "RAM" : "6 GB", "RearCamera" : "13 MP" }
{ "_id" : ObjectId("602ac6004962019b8d2556a2"), "serial_code" : 6324, "Mobile_ModelName" : "Realme 3", "Operating_System" : "Android 10", "Internal_Storage" : "128 GB", "RAM" : "3 GB", "RearCamera" : "20 MP" }
{ "_id" : ObjectId("602ac6024962019b8d2556a3"), "serial_code" : 8644, "Mobile_ModelName" : "Apple iPhone XR", "Operating_System" : "iOS 12", "Internal_Storage" : "64 GB", "RAM" : "2 GB", "RearCamera" : "12 MP" }
{ "_id" : ObjectId("602ac6034962019b8d2556a4"), "serial_code" : 8430, "Mobile_ModelName" : "Asus ROG 2", "Operating_System" : "Android 10", "Internal_Storage" : "128 GB", "RAM" : "8 GB", "RearCamera" : "20 MP" }
```

2. Available Collections in the current Database :

Syntax : **show collections**

```
> show collections
mob_info
>
```

3. Create collection before inserting document (Method 2 for creating collection):

Syntax: **db.createCollection("collection_name")**

```
> db
mobile_info
> show collections
mob_info
> db.createCollection("Mob_moreInfo")
{ "ok" : 1 }
> show collections
Mob_moreInfo
mob_info
>
```

4.Drop collection

Syntax : **db.collection_name.drop()**

```
> show collections
Mob_moreInfo
mob_info
> db.Mob_moreInfo.drop()
true
> show collections
mob_info
>
```

❖ *Documents in MongoDB :*

1. MongoDB Projection :

Syntax : **db.collection_name.find({}, {KEY:1})**

```
> db.mob_info.find().pretty()
{
  "_id" : ObjectId("602ac4894962019b8d25569b"),
  "serial_code" : 7575,
  "Mobile_ModelName" : "Samsung MB1",
  "Operating_System" : "Android 10",
  "Internal_Storage" : "64 GB",
  "RAM" : "3 GB",
  "RearCamera" : "20 MP"
}
{
  "_id" : ObjectId("602ac5de4962019b8d25569c"),
  "serial_code" : 4234,
  "Mobile_ModelName" : "Apple iPhone 12",
```

2 . Remove one document :

Syntax : **db.collection_name.remove(deletion_criteria)**

```
> db.mob_info.remove({"serial_code":6324},1)
WriteResult({ "nRemoved" : 1 })
```

3 . Sorting of documents:

Syntax : **db.collection_name.find().sort({KEY : 1})**

(Ascending -> 1 , Decending -> -1)

```
> db.mob_info.find({},{"serial_code":1,_id:0}).sort({"serial_code":1})
{ "serial_code" : 1209 }
{ "serial_code" : 1289 }
{ "serial_code" : 1639 }
{ "serial_code" : 4234 }
{ "serial_code" : 7364 }
{ "serial_code" : 7575 }
{ "serial_code" : 8430 }
{ "serial_code" : 8644 }
{ "serial_code" : 8648 }
> db.mob_info.find({},{"serial_code":1,_id:0}).sort({"serial_code":-1})
{ "serial_code" : 8648 }
{ "serial_code" : 8644 }
{ "serial_code" : 8430 }
{ "serial_code" : 7575 }
{ "serial_code" : 7364 }
{ "serial_code" : 4234 }
{ "serial_code" : 1639 }
{ "serial_code" : 1289 }
{ "serial_code" : 1209 }
```

```
> db.mob_info.find({},{"serial_code":0,_id:0}).sort({"serial_code":1})
{ "Mobile_ModelName" : "Redmi Note 9", "Operating_System" : "Android 10", "Internal_Storage" : "128 GB", "RAM" : "6 GB", "RearCamera" : "13 MP" }
{ "Mobile_ModelName" : "Apple iPhone 6", "Operating_System" : "iOS 9", "Internal_Storage" : "128 GB", "RAM" : "2 GB", "RearCamera" : "8 MP" }
{ "Mobile_ModelName" : "Moto G5", "Operating_System" : "Android 8", "Internal_Storage" : "64 GB", "RAM" : "4 GB", "RearCamera" : "26 MP" }
{ "Mobile_ModelName" : "Apple iPhone 12", "Operating_System" : "iOS 14", "Internal_Storage" : "512 GB", "RAM" : "4 GB", "RearCamera" : "24 MP" }
{ "Mobile_ModelName" : "Samsung Galaxy S21", "Operating_System" : "Android 10", "Internal_Storage" : "128 GB", "RAM" : "6 GB", "RearCamera" : "64 MP" }
{ "Mobile_ModelName" : "Samsung M31", "Operating_System" : "Android 10", "Internal_Storage" : "64 GB", "RAM" : "3 GB", "RearCamera" : "20 MP" }
{ "Mobile_ModelName" : "Asus ROG 2", "Operating_System" : "Android 10", "Internal_Storage" : "128 GB", "RAM" : "8 GB", "RearCamera" : "20 MP" }
{ "Mobile_ModelName" : "Apple iPhone XR", "Operating_System" : "iOS 12", "Internal_Storage" : "64 GB", "RAM" : "2 GB", "RearCamera" : "12 MP" }
{ "Mobile_ModelName" : "One Plus 8", "Operating_System" : "Android 11", "Internal_Storage" : "64 GB", "RAM" : "8 GB", "RearCamera" : "32 MP" }
```

4 . Update document :

- Syntax :

db.collection_name.update(selection_criteria,update_criteria)

```
> db.mob_info.update({"serial_code" : 8644},{ $set : {"Internal_Storage":"128 GB"}, $currentDate:{lastModified:true}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.mob_info.find({"serial_code" : 8644}).pretty()
{
  "_id" : ObjectId("602ac6024962019b8d2556a3"),
  "serial_code" : 8644,
  "Mobile_ModelName" : "Apple iPhone XR",
  "Operating_System" : "iOS 12",
  "Internal_Storage" : "128 GB",
  "RAM" : "2 GB",
  "RearCamera" : "12 MP",
  "lastModified" : ISODate("2021-03-03T20:09:40.747Z")
}
```

- Syntax :

db.collection_name.updateMany(selection_criteria,update_criteria)

```
> db.mob_info.updateMany({"Operating_System":"Android 10"},{$set : {"Internal_Storage":"128 GB"},$currentDate:{lastModified:true}})
{ "acknowledged" : true, "matchedCount" : 4, "modifiedCount" : 4 }
> db.mob_info.find({"Operating_System":"Android 10"}).pretty()
{
  "_id" : ObjectId("602ac4894962019b8d25569b"),
  "serial_code" : 7575,
  "Mobile_ModelName" : "Samsung MB1",
  "Operating_System" : "Android 10",
  "Internal_Storage" : "128 GB",
  "RAM" : "3 GB",
  "RearCamera" : "20 MP",
  "lastModified" : ISODate("2021-03-03T20:27:35.753Z")
}
{
  "_id" : ObjectId("602ac5ee4962019b8d25569e"),
  "serial_code" : 7364,
  "Mobile_ModelName" : "Samsung Galaxy S21",
  "Operating_System" : "Android 10",
  "Internal_Storage" : "128 GB",
  "RAM" : "6 GB",
  "RearCamera" : "64 MP",
  "lastModified" : ISODate("2021-03-03T20:27:35.753Z")
}
{
  "_id" : ObjectId("602ac5fd4962019b8d2556a1"),
  "serial_code" : 1209,
  "Mobile_ModelName" : "Redmi Note 9",
  "Operating_System" : "Android 10",
  "Internal_Storage" : "128 GB",
}
```

4. Skip document :

Syntax : **db.collection_name.find().skip(number)**

```
> db.mob_info.find({"serial_code":{"$gt:8430"}}).limit(1).skip(1).pretty()
{
  "_id" : ObjectId("602ac6024962019b8d2556a3"),
  "serial_code" : 8644,
  "Mobile_ModelName" : "Apple iPhone XR",
  "Operating_System" : "iOS 12",
  "Internal_Storage" : "64 GB",
  "RAM" : "2 GB",
  "RearCamera" : "12 MP"
}
>
```


5. Limit method :

Syntax : **db.collection_name.find().limit(number)**

```
> db.mob_info.find({"serial_code":{"$gt:8430}}).pretty()
{
  "_id" : ObjectId("602ac5fa4962019b8d25569f"),
  "serial_code" : 8648,
  "Mobile_ModelName" : "One Plus 8",
  "Operating_System" : "Android 11",
  "Internal_Storage" : "64 GB",
  "RAM" : "8 GB",
  "RearCamera" : "32 MP"
}
{
  "_id" : ObjectId("602ac6024962019b8d2556a3"),
  "serial_code" : 8644,
  "Mobile_ModelName" : "Apple iPhone XR",
  "Operating_System" : "iOS 12",
  "Internal_Storage" : "64 GB",
  "RAM" : "2 GB",
  "RearCamera" : "12 MP"
}
> db.mob_info.find({"serial_code":{"$gt:8430}}).limit(1).pretty()
{
  "_id" : ObjectId("602ac5fa4962019b8d25569f"),
  "serial_code" : 8648,
  "Mobile_ModelName" : "One Plus 8",
  "Operating_System" : "Android 11",
  "Internal_Storage" : "64 GB",
  "RAM" : "8 GB",
  "RearCamera" : "32 MP"
}
```

6. Index Method :

Syntax : **db.collection_name.createIndex({KEY :1})**

```
> db.mob_info.createIndex({Mobile_ModelName:1})
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
```

7. Finding indexes :

Syntax : **db.collection_name.getIndexes()**

```
> db.mob_info.getIndexes()
[
  {
    "v" : 2,
    "key" : {
      "_id" : 1
    },
    "name" : "_id_",
    "ns" : "mobile_info.mob_info"
  },
  {
    "v" : 2,
    "key" : {
      "Mobile_ModelName" : 1
    },
    "name" : "Mobile_ModelName_1",
    "ns" : "mobile_info.mob_info"
  }
]
```

8. Drop index :

Syntax : **db.collections_name.dropIndex({KEY :1})**

```
> db.mob_info.dropIndex({Mobile_ModelName:1})
{ "nIndexesWas" : 2, "ok" : 1 }
>
```

9. Drop indexes:

Syntax : **db.collection_name.dropIndexes()**

```
> db.mob_info.dropIndexes()
{
  "nIndexesWas" : 1,
  "msg" : "non-_id indexes dropped for collection",
  "ok" : 1
}
```