Prof R. Madana Mohana

BIG DATA ANALYTICS

MongoDB

Query Document

https://www.youtube.com/c/RASINENIMADANAMOHANA

MongoDB: Query Document

MongoDB: *Query Document*

The find() Method:

- To query data from **MongoDB** collection, we need to use MongoDB's **find()** method.

Syntax:

The basic syntax of **find()** method is as follows:

```
>db.COLLECTION_NAME.find()
```

find() method will display all the documents in a non-structured way.

MongoDB: *Query Document*

The find() Method: *Example*

Consider previously created a collection named **Employee**:

```
> show collections
Employee
bdacollection
blogspost
courses
mongodbcollection
mycollection
mycollection1
>
```

MongoDB: *Query Document*

The find() Method: *Example*

And inserted documents into **Employee** collection using the **insert()** method as discussed in *previous lecture*.

MongoDB: *Query Document*

The find() Method: *Example*

Following method retrieves all the documents in the collection:

```
> db.Employee.find()
```

MongoDB: *Query Document*

The find() Method: *Example*

```
> db.Employee.find()
{ "_id" : ObjectId("634bfe4112c650f5e845ddc9"),
"First_Name" : "R Madana", "Last_Name" : "Mohana",
"Date_Of_Birth" : "1981-06-15", "e_mail" :
"rmmnaidu@gmail.com", "phone" : "0123456789" }
{ "_id" : ObjectId("634c03ef12c650f5e845ddca"),
"First_Name" : "R Madana", "Last_Name" : "Mohana",
"Date_Of_Birth" : "1981-06-15", "e_mail" :
"rmmnaidu@gmail.com", "phone" : "0123456789" }
```

MongoDB: *Query Document*

The find() Method: *Example*

```
{ "_id" : ObjectId("634c03ef12c650f5e845ddcb"),
"First_Name" : "Madan", "Last_Name" : "Mohan",
"Date_Of_Birth" : "1980-12-26", "e_mail" :
"r2431663@yahoo.com", "phone" : "9876543210" }
{ "_id" : ObjectId("634c03ef12c650f5e845ddcc"),
"First_Name" : "Madanamohana", "Last_Name" : "Rasineni",
"Date_Of_Birth" : "1982-06-20", "e_mail" :
"madanmohanar@gmail.com", "phone" : "9440793154" }
>
```

MongoDB: *Query Document*

The find() Method: *Example*

Command Prompt - mongo

```
> show collections
Employee
bdacollection
blogspost
courses
mongodbcollection
mycollection
mycollection1
> db.Employee.find()
{ "_id" : ObjectId("634bfe4112c650f5e845ddc9"), "First_Name" : "R Madana", "Last_Name" : "Mohana", "Date_Of_Birth" : "
1981-06-15", "e_mail" : "rmmnaidu@gmail.com", "phone" : "0123456789" }
{ "_id" : ObjectId("634c03ef12c650f5e845ddca"), "First_Name" : "R Madana", "Last_Name" : "Mohana", "Date_Of_Birth" : "
1981-06-15", "e_mail" : "rmmnaidu@gmail.com", "phone" : "0123456789" }
{ "_id" : ObjectId("634c03ef12c650f5e845ddcb"), "First_Name" : "Madan", "Last_Name" : "Mohan", "Date_Of_Birth" : "1980
-12-26", "e_mail" : "r2431663@yahoo.com", "phone" : "9876543210" }
{ "_id" : ObjectId("634c03ef12c650f5e845ddcc"), "First_Name" : "Madanamohana", "Last_Name" : "Rasineni", "Date_Of_Birt
h" : "1982-06-20", "e_mail" : "madanmohanar@gmail.com", "phone" : "9440793154" }
>
```

MongoDB: *Query Document*

The pretty() Method:

- To display the results in a formatted way, we can use **pretty()** method.

Syntax:

The basic syntax of **pretty()** method is as follows:

```
>db.COLLECTION_NAME.find().pretty()
```

MongoDB: *Query Document*

The pretty() Method: *Example*

- *Following example* retrieves all the documents from the collection named **Employee** and arranges them in an easy-to-read format.

```
> db.Employee.find().pretty()
```

MongoDB: *Query Document*

The pretty() Method: *Example*

```
> db.Employee.find().pretty()
{
        "_id" : ObjectId("634bfe4112c650f5e845ddc9"),
        "First_Name" : "R Madana",
        "Last_Name" : "Mohana",
        "Date_Of_Birth" : "1981-06-15",
        "e_mail" : "rmmnaidu@gmail.com",
        "phone" : "0123456789"
}
```

MongoDB: *Query Document*

The pretty() Method: *Example*

```
{
        "_id" : ObjectId("634c03ef12c650f5e845ddca"),
        "First_Name" : "R Madana",
        "Last_Name" : "Mohana",
        "Date_Of_Birth" : "1981-06-15",
        "e_mail" : "rmmnaidu@gmail.com",
        "phone" : "0123456789"
}
```

MongoDB: *Query Document*

The pretty() Method: *Example*

```
{
        "_id" : ObjectId("634c03ef12c650f5e845ddcb"),
        "First_Name" : "Madan",
        "Last_Name" : "Mohan",
        "Date_Of_Birth" : "1980-12-26",
        "e_mail" : "r2431663@yahoo.com",
        "phone" : "9876543210"
}
```

MongoDB: *Query Document*

The pretty() Method: *Example*

```
{
        "_id" : ObjectId("634c03ef12c650f5e845ddcc"),
        "First_Name" : "Madanamohana",
        "Last_Name" : "Rasineni",
        "Date_Of_Birth" : "1982-06-20",
        "e_mail" : "madanmohanar@gmail.com",
        "phone" : "9440793154"
}
```

MongoDB: *Query Document*

The findOne() Method:

- Apart from the **find()** method, there is **findOne()** method, that returns only one document.

Syntax:

The basic syntax of **pretty()** method is as follows:

```
>db.COLLECTION_NAME.findOne()
```

MongoDB: *Query Document*

The findOne() Method: *Example*

- *Following example* retrieves the document with **First_Name: "Madanamohana"** from **Employee** collection.

MongoDB: *Query Document*

The findOne() Method: *Example*

```
> db.Employee.findOne({First_Name: "Madanamohana"})
{
        "_id" : ObjectId("634c03ef12c650f5e845ddcc"),
        "First_Name" : "Madanamohana",
        "Last_Name" : "Rasineni",
        "Date_Of_Birth" : "1982-06-20",
        "e_mail" : "madanmohanar@gmail.com",
        "phone" : "9440793154"
}
>
```

MongoDB: *Query Document*

RDBMS *Where* Clause Equivalents in MongoDB:

To **query** the document on the basis of some condition, we can use following operations:

Operation	Syntax	Example	RDBMS Equivalent
Equality	`{<key>:{$eg;<value>}}`	`db.Employee.find({"Date_Of_Birth":"1982-06-20"}).pretty()`	`where Date_Of_Birth = '1982-06-20'`
Less Than	`{<key>:{$lt:<value>}}`	`db.Employee.find({"age":{$lt:40}}).pretty()`	`where age < 40`
Less Than Equals	`{<key>:{$lte:<value>}}`	`db.Employee.find({"age":{$lte:40}}).pretty()`	`where age <= 40`

MongoDB: *Query Document*

RDBMS *Where* Clause Equivalents in MongoDB:

Operation	Syntax	Example	RDBMS Equivalent
Greater Than	{<key>:{$gt:<value>}}	db.Employee.find({"age": {$gt:40}}).pretty()	where age > 40
Greater Than Equals	{<key>:{$gte:<value>}}	db.Employee.find({"age": {$gte:40}}).pretty()	where age >= 40
Not Equals	{<key>:{$ne:<value>}}	db.Employee.find({"age": {$ne:40}}).pretty()	where age != 40

MongoDB: *Query Document*

RDBMS *Where* Clause Equivalents in MongoDB:

Operation	Syntax	Example	RDBMS Equivalent
Values in an array	`{<key>:{$in:[<value1>, <value2>,……<valueN>]}}`	`db.Employee.find({"First_Name":{$in:["R Madana", "Madan"]}}).pretty()`	`Where First_Name matches any of the value in :[:["R Madana", "Madan"]`
Values not in an array	`{<key>:{$nin:[<value1>, <value2>,……<valueN>]}}`	`db.Employee.find({"First_Name":{$nin:["R Madana", "Madan"]}}).pretty()`	`Where First_Name values is not in the array:[:["R Madana", "Madan"] or, doesn't exist at all`

MongoDB: *Query Document*

AND in MongoDB:

- To query documents based on the **AND** condition, we need to use **$and** keyword.

AND in MongoDB: *Syntax*

Following is the basic syntax of **AND**:

```
>db.Collection_Name.find({ $and: [
{<key1>:<value1>}, { <key2>:<value2>} ] })
```

MongoDB: *Query Document*

AND in MongoDB: *Example*

Following example will show all the **Date_Of_Birth**s from '**Employee**' collection and whose **First_Name**s is '**R Madana**'.

```
> db.Employee.find({$and:
[{"First_Name":"Madan"},{"Date_Of_Birth":
"1980-12-26"}]}).pretty()
```

MongoDB: *Query Document*

AND in MongoDB: *Example*

Output:

```
{
        "_id" : ObjectId("634c03ef12c650f5e845ddcb"),
        "First_Name" : "Madan",
        "Last_Name" : "Mohan",
        "Date_Of_Birth" : "1980-12-26",
        "e_mail" : "r2431663@yahoo.com",
        "phone" : "9876543210"
}
>
```

MongoDB: *Query Document*

AND in MongoDB: *Example*

Command Prompt - mongo

```
> db.Employee.find({$and:[{"First_Name":"Madan"},{"Date_Of_Birth": "1980-12-26"}]}).pretty()
{
        "_id" : ObjectId("634c03ef12c650f5e845ddcb"),
        "First_Name" : "Madan",
        "Last_Name" : "Mohan",
        "Date_Of_Birth" : "1980-12-26",
        "e_mail" : "r2431663@yahoo.com",
        "phone" : "9876543210"
}
>
```

MongoDB: *Query Document*

AND in MongoDB: *Example*

For the *above given example*, equivalent **where** clause will be '

```
where First_Name = "Madan" AND Date_Of_Birth
='1980-12-26'
```

We can pass any number of key, value pairs in **find** clause.

MongoDB: *Query Document*

OR in MongoDB:

- To query documents based on the **OR** condition,we need to use **$or** keyword.

OR in MongoDB: *Syntax*

Following is the basic syntax of **OR**:

```
>db.Collection_Name.find({$or: [
        {key1: value1}, {key2:value2}
    ]
}
).pretty()
```

MongoDB: *Query Document*

OR in MongoDB: *Example*

Following example will show all the details from '**Employee**' collection with '**First_Name**' is '**Madanamohana**' **OR** '**Date_Of_Birth**' is '**1980-12-26**'

```
> db.Employee.find({$or:
[{"First_Name":"Madanamohana"},
{"Date_Of_Birth": "1980-12-26"}]}).pretty()
```

MongoDB: *Query Document*

OR in MongoDB: *Example*

Output:

```
{
        "_id" : ObjectId("634c03ef12c650f5e845ddcb"),
        "First_Name" : "Madan",
        "Last_Name" : "Mohan",
        "Date_Of_Birth" : "1980-12-26",
        "e_mail" : "r2431663@yahoo.com",
        "phone" : "9876543210"
}
```

MongoDB: *Query Document*

OR in MongoDB: *Example*

Output:

```
{
        "_id" : ObjectId("634c03ef12c650f5e845ddcc"),
        "First_Name" : "Madanamohana",
        "Last_Name" : "Rasineni",
        "Date_Of_Birth" : "1982-06-20",
        "e_mail" : "madanmohanar@gmail.com",
        "phone" : "9440793154"
}
>
```

MongoDB: *Query Document*

Using AND and OR Together:

- The *following example* will show the documents that have **age** greater than 40 and whose **First_Name** is either 'Madanamohana' or '**Date_Of_Birth**' is '1980-12-26'.
- Equivalent **SQL where** clause is

```
where age>40 AND (First_Name = 'Madanamohana'
OR Date_Of_Birth = '1980-12-26')'
```

MongoDB: *Query Document*

Using AND and OR Together:

```
> db.Employee.find({"age": {$gt:40},$or:
[{"First_Name":"Madanamohana"},
{"Date_Of_Birth": "1980-12-26"}]}).pretty()
```

MongoDB: *Query Document*

NOR in MongoDB:

- To query documents based on the **NOR** condition, we need to use **$nor** keyword.

NOR in MongoDB: *Syntax*

Following is the basic syntax of **NOR**:

```
>db.Collection_Name.find(
{
$nor: [
          {key1: value1}, {key2:value2}
       ]
   }
).pretty()
```

MongoDB: *Query Document*

NOR in MongoDB: *Example*

Consider the collection **EmployeeDetails** with the following documents:

```
> db.EmployeeDetails.find().pretty()
```

MongoDB: *Query Document*

NOR in MongoDB: *Example*

Command Prompt - mongo

```
> db.EmployeeDetails.find().pretty()
{
        "_id" : ObjectId("634c366412c650f5e845ddcd"),
        "First_Name" : "Rasineni Madana",
        "Last_Name" : "Mohana",
        "Age" : "42",
        "e_mail" : "rmmnaidu@gmail.com",
        "phone" : "0123456789"
}
{
        "_id" : ObjectId("634c366412c650f5e845ddce"),
        "First_Name" : "R Madana",
        "Last_Name" : "Mohana",
        "Age" : "46",
        "e_mail" : "r2431663@yahoo.com",
        "phone" : "9876543210"
}
{
        "_id" : ObjectId("634c366412c650f5e845ddcf"),
        "First_Name" : "Madana Mohana",
        "Last_Name" : "Rasineni",
        "Age" : "47",
        "e_mail" : "madanmohanar@gmail.com",
        "phone" : "9440793154"
}
>
```

MongoDB: *Query Document*

NOR in MongoDB: *Example*

Following example will retrieve the document(s) whose "**First_Name**" is not "Rasineni Madana" and "**Last_Name**" is not "Mohana"

```
> db.EmployeeDetails.find({$nor:
[{"First_Name": "Rasineni Madana"},
{"Last_Name": "Mohana"}]}).pretty()
```

MongoDB: *Query Document*

NOR in MongoDB: *Example*

Output:

```
{
        "_id" : ObjectId("634c366412c650f5e845ddcf"),
        "First_Name" : "Madana Mohana",
        "Last_Name" : "Rasineni",
        "Age" : "47",
        "e_mail" : "madanmohanar@gmail.com",
        "phone" : "9440793154"
}
>
```

MongoDB: *Query Document*

NOT in MongoDB:

- To query documents based on the **NOT** condition, we need to use **$not** keyword .

NOT in MongoDB: *Syntax*

Following is the basic syntax of **NOT**:

```
>db.Collection_Name.find(
{
$not: [
          {key1: value1}, {key2:value2}
       ]
   }
).pretty()
```

MongoDB: *Query Document*

NOT in MongoDB: *Example*

- *Following example* will retrieve the document(s) whose **Age** is not greater than 46

```
> db.EmployeeDetails.find( { "Age": { $not: { $gt: "46" } } } ).pretty()
```

MongoDB: *Query Document*

NOT in MongoDB: *Example*

Output:

```
{
        "_id" : ObjectId("634c366412c650f5e845ddcd"),
        "First_Name" : "Rasineni Madana",
        "Last_Name" : "Mohana",
        "Age" : "42",
        "e_mail" : "rmmnaidu@gmail.com",
        "phone" : "0123456789"
}
```

MongoDB: *Query Document*

NOT in MongoDB: *Example*

Output:

```
{
        "_id" : ObjectId("634c366412c650f5e845ddce"),
        "First_Name" : "R Madana",
        "Last_Name" : "Mohana",
        "Age" : "46",
        "e_mail" : "r2431663@yahoo.com",
        "phone" : "9876543210"
}
```

MongoDB: *Query Document*

NOT in MongoDB: *Example*

Command Prompt - mongo

```
> db.EmployeeDetails.find( { "Age": { $not: { $gt: "46" } } } ).pretty()
{
        "_id" : ObjectId("634c366412c650f5e845ddcd"),
        "First_Name" : "Rasineni Madana",
        "Last_Name" : "Mohana",
        "Age" : "42",
        "e_mail" : "rmmnaidu@gmail.com",
        "phone" : "0123456789"
}
{
        "_id" : ObjectId("634c366412c650f5e845ddce"),
        "First_Name" : "R Madana",
        "Last_Name" : "Mohana",
        "Age" : "46",
        "e_mail" : "r2431663@yahoo.com",
        "phone" : "9876543210"
}
>
```