|  |
| --- |
| 12)Create the collection Books having the following fields TITLE, DESCRIPTION, BY, URL, TAGS AND LIKES. (MONGO)  **Implement the following Aggregation and Indexing Queries**    1. Find the number of books published by john.  2. Find books which have minimum likes and maximum likes published by john.  3. Find the average number of likes of the books published by john.  4. Find the first and last book published by john..  5. Create an index on author name.  Display the books published by john and check if it uses the index which we have created |

> use tebA

switched to db teA

#DISPLAY ALL COLLECTIONS

> db.getCollectionNames()

[ "Books", "Student" ]

#DISPLAY ALL THE RECORDS

> db.Books.find().pretty()

{

"\_id" : ObjectId("5b935d685d7169c11982a10e"),

"Title" : "SQL",

"Description" : "User manual for SQL",

"URL" : "https://mailchimp.com",

"Likes" : 34,

"Tags" : 13,

"Author" : "John Harrrison"

}

{

"\_id" : ObjectId("5b935e315d7169c11982a10f"),

"Title" : "MongoDB Atlas",

"Description" : "Guidance for Mongodb",

"URL" : "https://www.mongodb.com/",

"Likes" : 123,

"Tags" : 45,

"Author" : "Mark S"

}

{

"\_id" : ObjectId("5b935eb85d7169c11982a110"),

"Title" : "NOSQL overview",

"Description" : "Guidance for NOsql",

"URL" : "https://en.wikipedia.org/wiki/NoSQL",

"Likes" : 8,

"Tags" : 15,

"Author" : "Steve K"

}

{

"\_id" : ObjectId("5b935f115d7169c11982a111"),

"Title" : "Database Concepts",

"Description" : "Guidance for Databases",

"URL" : "https://en.wikipedia.org",

"Likes" : 83,

"Tags" : 68,

"Author" : "Korth"

}

{

"\_id" : ObjectId("5b9361415d7169c11982a114"),

"Title" : "Oracle Concepts",

"Description" : "Guidance for Databases",

"URL" : "https://en.wikipedia.org",

"Likes" : 23,

"Tags" : 67,

"Author" : "Ron K L",

"Username" : "Ron\_K",

"Comments" : 8

}

{

"\_id" : ObjectId("5b9362195d7169c11982a115"),

"Title" : "Mongodb",

"Description" : "User manual for Mongodb server",

"URL" : "https://www.tutorialspoint.com/mongodb",

"Likes" : 178,

"Tags" : 34,

"Author" : "John Harrrison"

}

{

"\_id" : ObjectId("5b9362365d7169c11982a116"),

"Title" : "Mongodb",

"Description" : "User manual for Mongodb server",

"URL" : "https://www.tutoria/mongodb",

"Likes" : 118,

"Tags" : 64,

"Author" : "Kevin"

}

{

"\_id" : ObjectId("5b9c94fc9220914280322648"),

"Title" : "Mongodb2",

"Description" : "Educational",

"Author" : "John Harrrison",

"URL" : "www.dsafssaffdf",

"Tags" : 234,

"Likes" : 1230

}

{

"\_id" : ObjectId("5b9c95319220914280322649"),

"Title" : "Mongodb3",

"Description" : "Reference",

"Author" : "John Harrrison",

"URL" : "www.dsafssaffdf",

"Tags" : 34,

"Likes" : 150

}

{

"\_id" : ObjectId("5b9c953c922091428032264a"),

"Title" : "Mongodb4",

"Description" : "Reference",

"Author" : "John Harrrison",

"URL" : "www.dsafssaffdf",

"Tags" : 34,

"Likes" : 10

}

{

"\_id" : ObjectId("5b9c954a922091428032264b"),

"Title" : "Mongodb5",

"Description" : "Reference",

"Author" : "John Harrrison",

"URL" : "www.dsafssaffdf",

"Tags" : 34,

"Likes" : 1567

}

======================================================================================

#DISPLAYING ALL BOOKS PUBLISHED BY JOHN

--------------------------------------------------------------------------------------> db.Books.aggregate([{$match:{Author:"John Harrrison"}}]).pretty()

{

"\_id" : ObjectId("5b935d685d7169c11982a10e"),

"Title" : "SQL",

"Description" : "User manual for SQL",

"URL" : "https://mailchimp.com",

"Likes" : 34,

"Tags" : 13,

"Author" : "John Harrrison"

}

{

"\_id" : ObjectId("5b9362195d7169c11982a115"),

"Title" : "Mongodb",

"Description" : "User manual for Mongodb server",

"URL" : "https://www.tutorialspoint.com/mongodb",

"Likes" : 178,

"Tags" : 34,

"Author" : "John Harrrison"

}

{

"\_id" : ObjectId("5b9c94fc9220914280322648"),

"Title" : "Mongodb2",

"Description" : "Educational",

"Author" : "John Harrrison",

"URL" : "www.dsafssaffdf",

"Tags" : 234,

"Likes" : 1230

}

{

"\_id" : ObjectId("5b9c95319220914280322649"),

"Title" : "Mongodb3",

"Description" : "Reference",

"Author" : "John Harrrison",

"URL" : "www.dsafssaffdf",

"Tags" : 34,

"Likes" : 150

}

{

"\_id" : ObjectId("5b9c953c922091428032264a"),

"Title" : "Mongodb4",

"Description" : "Reference",

"Author" : "John Harrrison",

"URL" : "www.dsafssaffdf",

"Tags" : 34,

"Likes" : 10

}

{

"\_id" : ObjectId("5b9c954a922091428032264b"),

"Title" : "Mongodb5",

"Description" : "Reference",

"Author" : "John Harrrison",

"URL" : "www.dsafssaffdf",

"Tags" : 34,

"Likes" : 1567

}

------------------------------------------------------------------------------------- 1. Find the number of books published by john.

--------------------------------------------------------------------------------------

> db.Books.aggregate([{$match:{Author:"John Harrrison"}},{$group:{\_id:null,Total\_no\_of\_books:{$sum:1}}}]).pretty()

{ "\_id" : null, "Total\_no\_of\_books" : 6 }

--------------------------------------------------------------------------------------

2. Find books which have minimum likes and maximum likes published by john

--------------------------------------------------------------------------------------

> db.Books.aggregate([{$match:{Author:"John Harrrison"}},{$group:{\_id:"Likes",Minimum\_likes:{$min:"$Likes"}}}]).pretty()

{ "\_id" : "Likes", "Minimum\_likes" : 10 }

>

>

> db.Books.aggregate([{$match:{Author:"John Harrrison"}},{$group:{\_id:"Likes",Maximum\_likes:{$max:"$Likes"}}}]).pretty()

{ "\_id" : "Likes", "Maximum\_likes" : 1567 }

>

>>

> db.Books.aggregate([{$match:{Author:"John Harrrison"}},{$group:{\_id:"John Harrrison",Maximum\_likes:{$max:"$Likes"},Minimum\_Likes:{$min:"$Likes"}}}]).pretty()

{ "\_id" : "John Harrrison", "Maximum\_likes" : 1567, "Minimum\_Likes" : 10 }

>

>

> db.Books.aggregate([{$match:{Author:"John Harrrison"}},{$group:{\_id:"$Title",Maximum\_likes:{$max:"$Likes"}}}]).pretty()

{ "\_id" : "Mongodb5", "Maximum\_likes" : 1567 }

{ "\_id" : "Mongodb4", "Maximum\_likes" : 10 }

{ "\_id" : "SQL", "Maximum\_likes" : 34 }

{ "\_id" : "Mongodb2", "Maximum\_likes" : 1230 }

{ "\_id" : "Mongodb", "Maximum\_likes" : 178 }

{ "\_id" : "Mongodb3", "Maximum\_likes" : 150 }

--------------------------------------------------------------------------------------

3. Find the average number of likes of the books published by john.

--------------------------------------------------------------------------------------

db.Books.aggregate([{$match:{Author:"John Harrrison"}},{$group:{\_id:"Likes",Average\_likes:{$avg:"$Likes"}}}]).pretty()

{ "\_id" : "Likes", "Average\_likes" : 528.1666666666666 }

--------------------------------------------------------------------------------------

4. Find the first and last book published by john..

------------------------------------------------------------------------------------

> db.Books.aggregate([{$match:{Author:"John Harrrison"}},{$group:{\_id:null,First\_book:{$first:"$Title"}}}]).pretty()

{ "\_id" : null, "First\_book" : "SQL" }

> db.Books.aggregate([{$match:{Author:"John Harrrison"}},{$group:{\_id:null,Last\_book:{$last:"$Title"}}}]).pretty()

{ "\_id" : null, "Last\_book" : "Mongodb5" }

--------------------------------------------------------------------------------------

5. Create an index on author name.

--------------------------------------------------------------------------------------

> db.Books.createIndex({Author:1})

{

"createdCollectionAutomatically" : false,

"numIndexesBefore" : 2,

"numIndexesAfter" : 2,

"note" : "all indexes already exist",

"ok" : 1

}

--------------------------------------------------------------------------------------

Display the books published by john and check if it uses the index which we have created.

--------------------------------------------------------------------------------------

> db.Books.getIndexes()

[

{

"v" : 2,

"key" : {

"\_id" : 1

},

"name" : "\_id\_",

"ns" : "teb\_70.Books"

},

{

"v" : 2,

"key" : {

"Author" : 1

},

"name" : "Author\_1",

"ns" : "teb\_70.Books"

}

]

>

======================================================================================