**Assignment 2**

**MongoDB Querying**

Please work in **groups** to complete this assignment. This assignment is worth **15% of the total course grade** and will be evaluated through your written submission. Each day being late will result in 10% mark penalty.

Please submit the following files through Blackboard:

* Assign2.docx

1. Add this declaration on the top of your Assign2.docx file.

We, Mansoor Ahmad Zafar declare that the attached assignment is our own work in accordance with the Seneca Academic Policy. No part of this assignment has been copied manually or electronically from any other source (including web sites) **or distributed to other students.**

1. Specify what each member has done towards the completion of this work:

|  |  |  |
| --- | --- | --- |
|  | Name | Task(s) |
| 1 | Mansoor Ahmad Zafar | Everything |
| 2 |  |  |
| 3 |  |  |

# Importing the Database

You are to create a new mongodb database named “booksdb” by importing books.json into books collection.

1. Open Mongodb Compass and click the connect button.

A screenshot of a computer

Description automatically generated

1. Click on databases from the top menu, then create a database and name it booksdb

A close-up of a computer screen

Description automatically generated

1. Click on the database booksdb,

A screenshot of a computer

Description automatically generated

1. Click on “create collection” and name the collection “books”

A screenshot of a computer

Description automatically generated

1. Click on “ADD DATA”, and select “import file”

A screenshot of a computer

Description automatically generated

1. A dialogue box import to booksdb.books will show. Select the file from your local drive and hit Import.
2. Click on MONGOSH in the bottom of mongodb compass to type mongodb commands.

A screenshot of a computer

Description automatically generated

1. Make sure the database is created. type the following commands:

use booksdb

db.books.findOne()

A document will display in the output.

Instructions

Answer each of the following questions with a mongodb command, and the corresponding output.

# marking:

10 questions, 3pts each. Total: 30pts

# Part 1: Query

1. Find the number of books in the collection.

Code: (ran before adding the document from assignment 2 also id:12 doesn’t exist…)

db.books.countDocuments()

A blue background with white text

Description automatically generated

Output: 19

1. Find the books that have pagecount greater than or equal 600. Show only the \_id field.

Code:

db.books.find({"pageCount" : {"$gte" : 600}}, {"\_id" : 1})

A blue screen with white text

Description automatically generated

Output: \_id 5

1. Find the books that are authored by Robi Sen.

Code:

db.books.find({"authors" : {"$in" : ["Robi Sen"]}}, {"\_id" : 1})

A computer screen shot of a computer code

Description automatically generated

Output: \_id 1 and 2

1. Find the books that belong to Internet category.

Code: (if “belongs to” means has it then it should be : “categories” : {“$in”: [“internet”]})

db.books.find({"categories" : "Internet"}, {"\_id" : 1})

A screen shot of a computer program

Description automatically generated

Output: \_id 4, 5, 6, 8, 13

1. Find the books that have 3 authors. Use $size operator.

Code:

db.books.find({"authors" : {"$size" : 3}}, {"\_id" : 1})

A screen shot of a computer

Description automatically generated

Output: \_id 1 and 7

1. Find the books that have the word ‘Android’ in the title.

Code:

db.books.find({"title" : {"$regex" : ".\*Android.\*"}}, {"\_id" : 1})

A computer screen shot of a computer code

Description automatically generated

Output: \_id 1 and 2

**Reference**:

You can use $regex operator to check if a field contains a string in MongoDB. The syntax is as follows –

db.yourCollectionName.findOne({"yourFieldName":{$regex:".\*yourValue.\*"}});

<https://www.tutorialspoint.com/check-if-a-field-contains-a-string-in-mongodb>

1. Find the books that are missing the short description key. Use $exists operator.

Code: (added document from assignment 2 at this point onwards:)

db.books.find({"shortDescription" : {"$exists" : false}}, {"\_id" : 1});

A screen shot of a computer program

Description automatically generated

Output:

{ \_id: 3 },

{ \_id: 4 },

{ \_id: 5 },

{ \_id: 6 },

{ \_id: 15 },

{ \_id: 18 },

{ \_id: 20}

1. Find the books that Joe commented on. Show only the book isbn, title and the user name in the embedded document of the comments array.

Code:

db.books.find({"comments.Name" : {"$in" : ["Joe"]}}, {"isbn" : 1, "title" : 1, "comments.Name.$" : 1})

A computer screen shot of a program code

Description automatically generated

Output: \_id 19, 20

1. Find the books that have a rating less than or equal 3.

Code:

db.books.find({"comments.Rating" : {"$lte" : 3} }, {"\_id" : 1})

A blue screen with white text

Description automatically generated

Output: \_id 20

1. Find the books that are commented by Joe and have rating equal 3.

Code:

db.books.find({"comments.Name" : "Joe", "comments.Rating": 3}, {"\_id" : 1})

A blue screen with white text

Description automatically generated

Output: \_id 20