Student name: Pothumulla Kankanamge Mewan Madhusha

MongoDB NoSQL Assignment

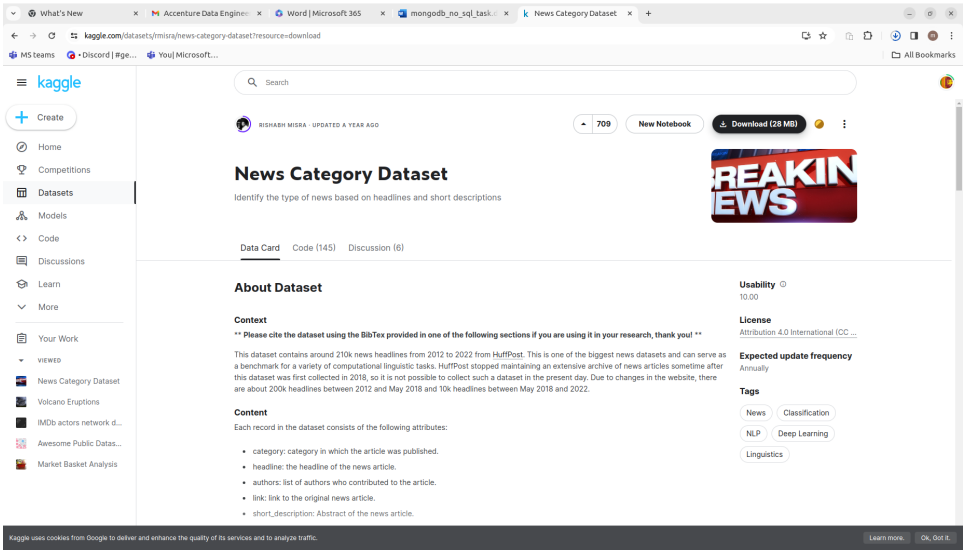
Task 1.

Download dataset from Kaggle*:

News Category Dataset

https://www.kaggle.com/datasets/rmisra/news-category-dataset

** for that you will need to create free account on kaggle.com. Go ahead.*

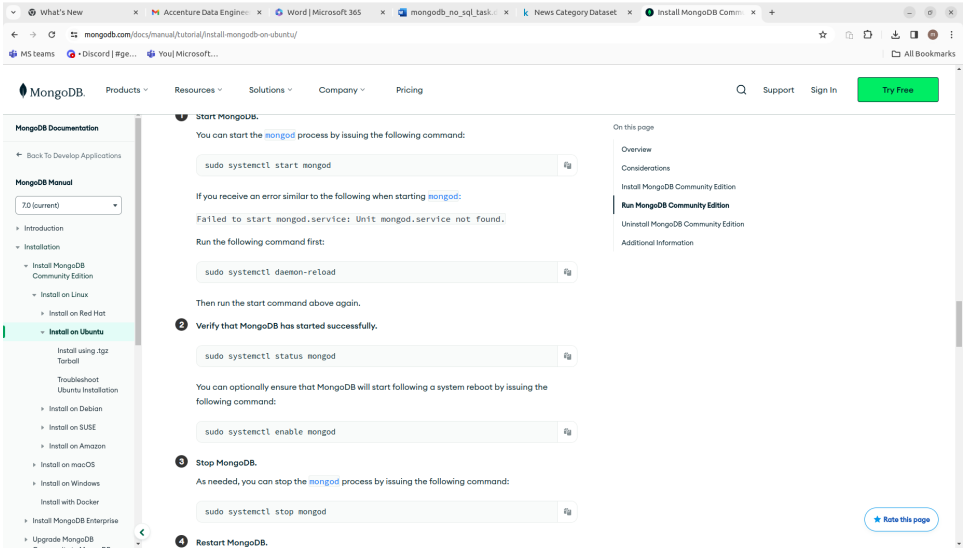


Create new database "news_db" in MongoDB on your local machine.

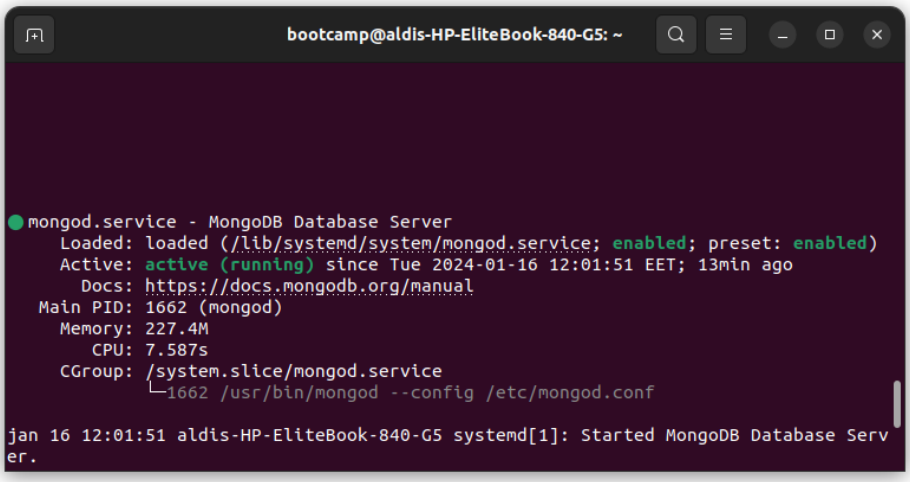
Provide command of newly created DB.

Follow the below steps to install the MongoDB community server on ubuntu

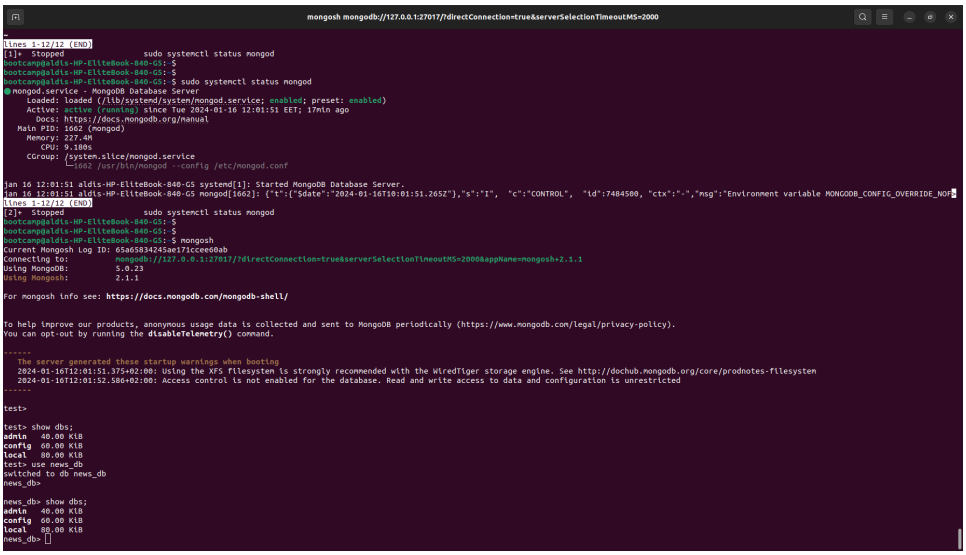
https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-ubuntu/



Installing DB



Create new database mongosh command - use news_db

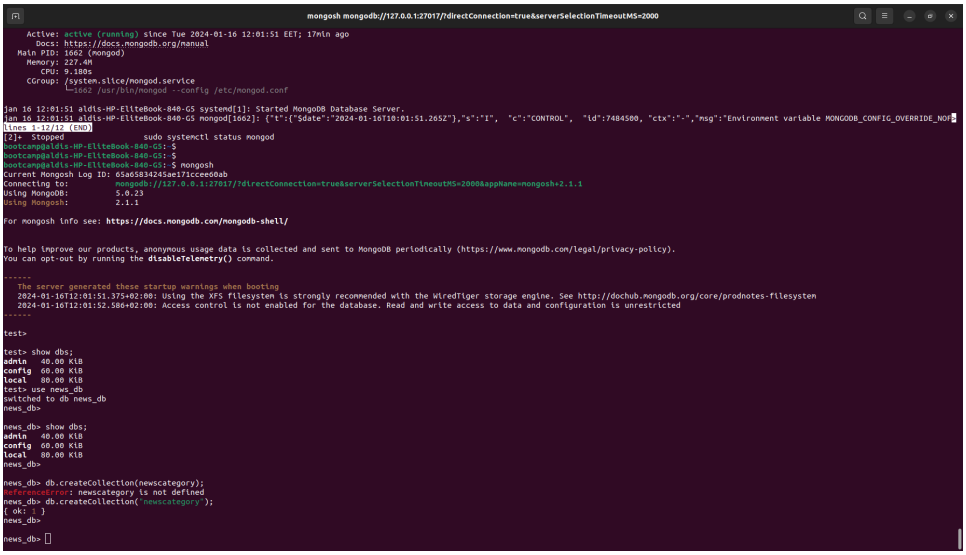


Task 2.

Load data from news dataset to news_db.

Explain how you solved the task.

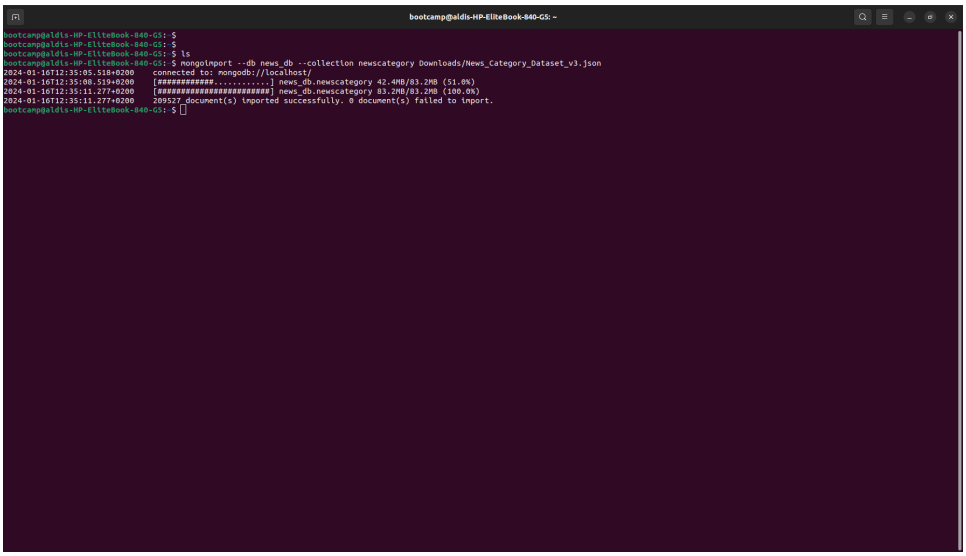
First create collection inside the news_db - db.createCollection("newscategory");



Use mongoimport functionality to load data into the collection

mongoimport --db news_db --collection newscategory Downloads/News_Category_Dataset_v3.json

First give the DB name then provide collection name and then give the dataset location in the local system, I have provided downloads file location since my file was in there.



Let’s check data set got imported

Task 3.

Describe the dataset loaded to news_db

The dataset loaded to news_db appears to be in JSON format and contains information about a news article. Here's a breakdown of the key-value pairs:

category: Category article belongs to

headline: Headline of the article

authors: Person authored the article

link: Link to the post

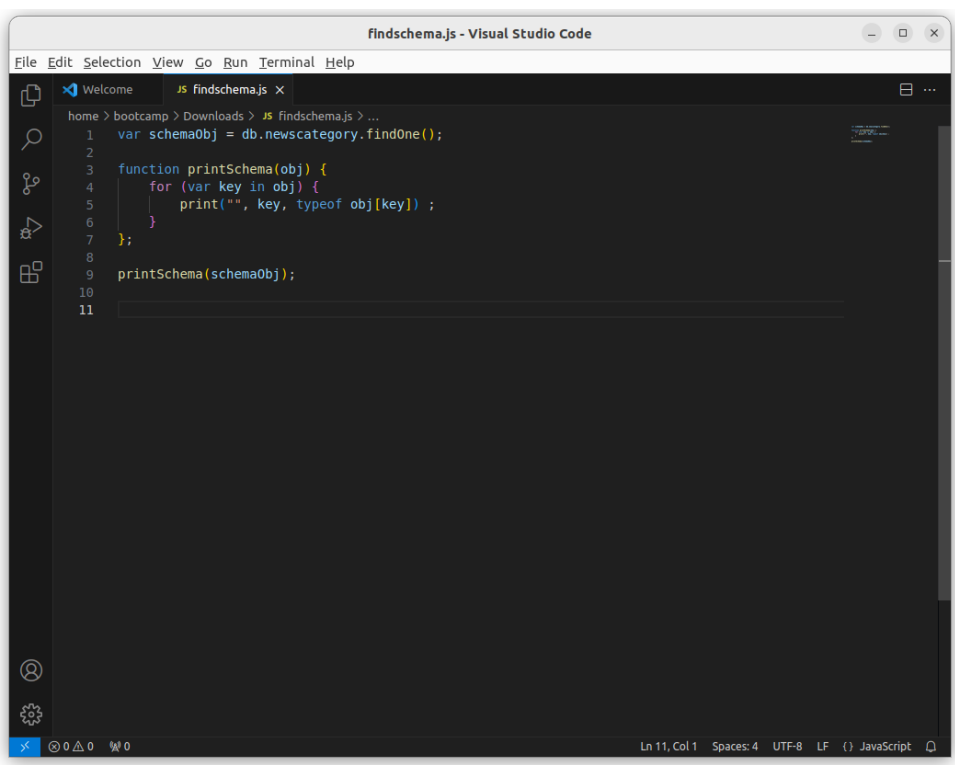
short_description: Short description of the article

date: Date the article was published

The data structure uses key-value pairs, which is common in JSON formats. This structure is database-friendly as it allows for easy retrieval and querying of specific information such as the headline, author, or publication date. The use of strings and basic data types makes it suitable for storage and retrieval in various database systems.

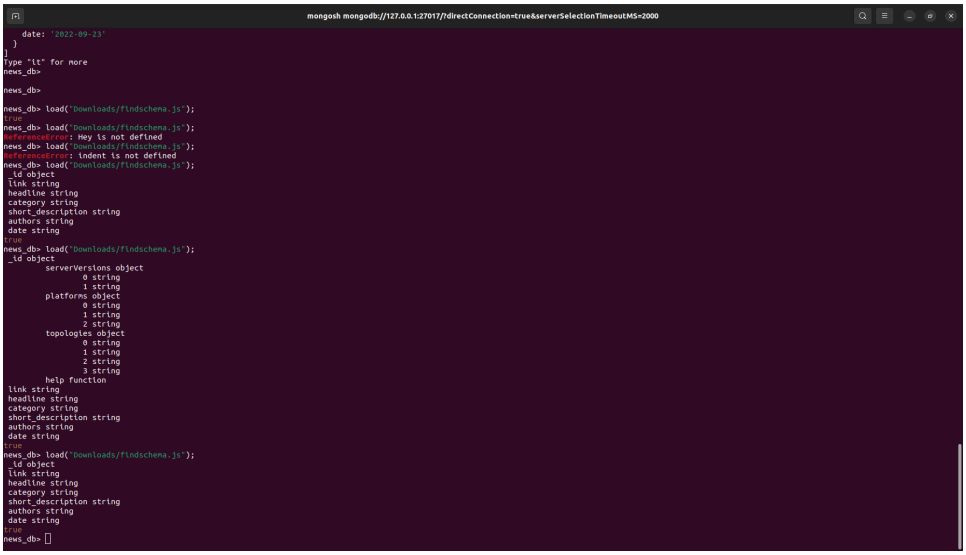
Print news_db schema.

```
var schemaObj = db.newscategory.findOne();

function printSchema(obj) {

for (var key in obj) {

print("", key, typeof obj[key]) ;}};

printSchema(schemaObj);
```



Load into mongosh and execute the above JS file

load("Downloads/findschema.js");



Explain how you solved the task.

var schemaObj = db.newscategory.findOne();Retrieves a single document from the newscategory collection in the current database (news_db).

This document is used to analyze the structure

function printSchema(obj): Defines a function named printSchema that takes an object (obj) as a parameter. Inside the function, it iterates through the keys of the object and prints each key along with its corresponding data type using the print function.

while Iterates through the keys of the object (obj) and prints each key and its data type. The typeof operator is used to determine the type of the value associated with each key.

To execute this code, I have saved it to a JavaScript file (printSchema.js) and then load it into the mongosh shell

Task 4.

Add new collections link_db, headline_db, category_db, short_description_db, suthors_db, date_db, link_db.

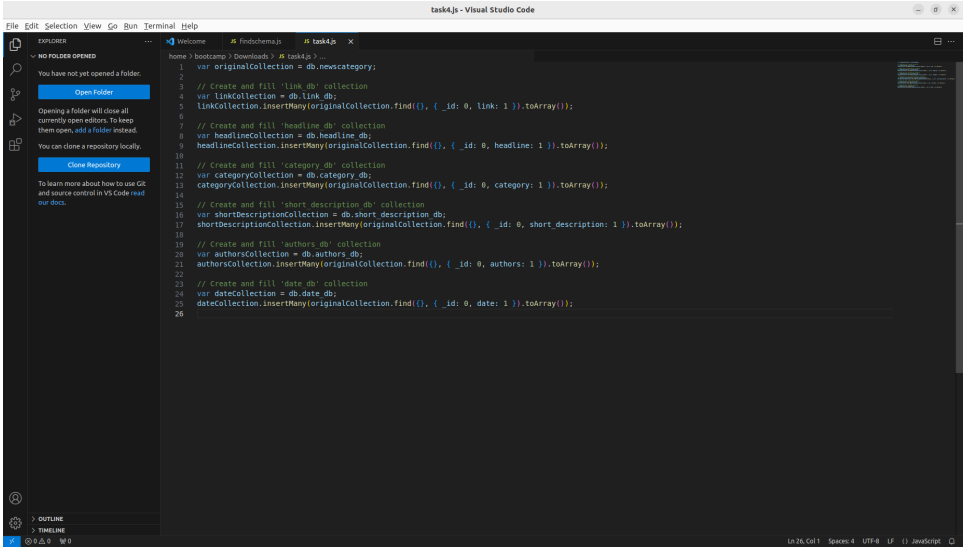
Fill in each collection from news_db corresponding value.

Explain how you solved the task.

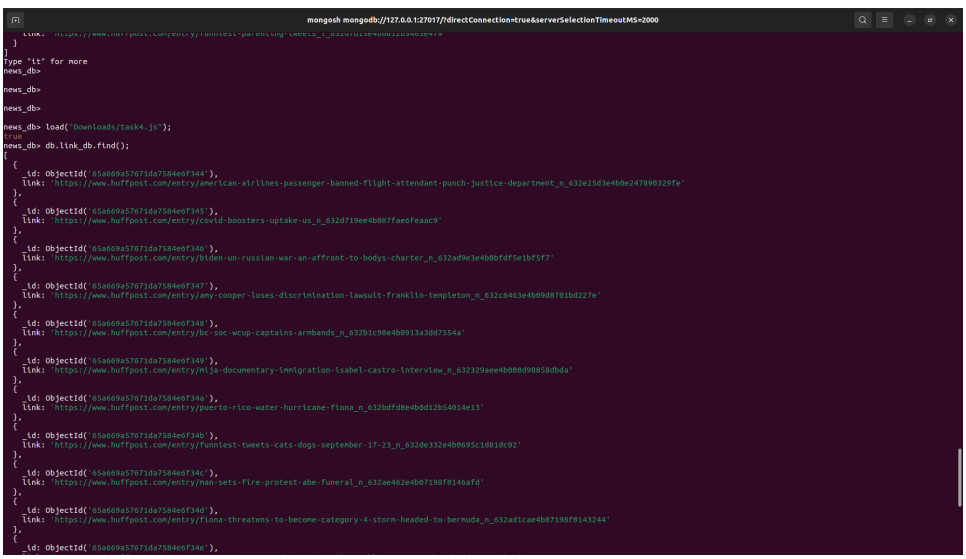
Code:

```
var originalCollection = db.newscategory;

// Create and fill 'link_db' collection
var linkCollection = db.link_db;
linkCollection.insertMany(originalCollection.find({}, { _id: 0, link: 1 }).toArray());

// Create and fill 'headline_db' collection
var headlineCollection = db.headline_db;
headlineCollection.insertMany(originalCollection.find({}, { _id: 0, headline: 1 }).toArray());

// Create and fill 'category_db' collection
var categoryCollection = db.category_db;
categoryCollection.insertMany(originalCollection.find({}, { _id: 0, category: 1 }).toArray());

// Create and fill 'short_description_db' collection
var shortDescriptionCollection = db.short_description_db;
shortDescriptionCollection.insertMany(originalCollection.find({}, { _id: 0, short_description: 1
}).toArray());

// Create and fill 'authors_db' collection
var authorsCollection = db.authors_db;
authorsCollection.insertMany(originalCollection.find({}, { _id: 0, authors: 1 }).toArray());

// Create and fill 'date_db' collection
var dateCollection = db.date_db;
```

dateCollection.insertMany(originalCollection.find({}, { _id: 0, date: 1 }).toArray());



Saved into JavaScript file and then load into mongosh



Below I have checked all the newly added collection in news_db.

db.link_db.find();

db.headline_db.find();

db.category_db.find();

db. short_description_db.find();

db. authors_db.find();

db. date_db.find();

Task 5.

Remove all records from news_db, which have at least one empty or NULL value in object.

How many records are left in news_db?156859

Task 6.

Explain how you solved the task and provide screenshots.

```
var newsCollection = db.newscategory;

// Remove records with at least one empty or null value

var result = newsCollection.deleteMany({

$or: [

{ link: { $eq: null } },

{ link: { $eq: "" } },

{ headline: { $eq: null } },

{ headline: { $eq: "" } },

{ category: { $eq: null } },

{ category: { $eq: "" } },
```

```
{ short_description: { $eq: null } },
{ short_description: { $eq: "" } },
{ authors: { $eq: null } },
{ authors: { $eq: "" } },
{ date: { $eq: null } },
{ date: { $eq: "" } }
]
});
```

```
var remainingCount = newsCollection.countDocuments();
print("Number of records remaining in 'news_db':", remainingCount);
```

Save the code in JS file and then load into mongsh after run the command

Based on the above code uses the deleteMany method to remove records that have at least one field with an empty or null value. After the deletion, it prints the number of records remaining in the news_db collection.

How many categories are in news_db? 42

```
db.newscategory.distinct("category").length;
```

Retrieves the distinct values for the "category" field in the newscategory collection.

How many news count is for every category?

```
db.newscategory.aggregate([{ $group: { _id: "$category", count: { $sum: 1 } } }]);
[
  { _id: 'PARENTS', count: 3491 },
  { _id: 'COMEDY', count: 3934 },
  { _id: 'BLACK VOICES', count: 3313 },
  { _id: 'HOME & LIVING', count: 3523 },
  { _id: 'MONEY', count: 1539 },
  { _id: 'STYLE & BEAUTY', count: 7275 },
  { _id: 'IMPACT', count: 2945 },
  { _id: 'ENTERTAINMENT', count: 13463 },
  { _id: 'U.S. NEWS', count: 1093 },
  { _id: 'FIFTY', count: 1042 },
  { _id: 'GREEN', count: 1682 },
  { _id: 'MEDIA', count: 2105 },
  { _id: 'POLITICS', count: 29685 },
  { _id: 'CULTURE & ARTS', count: 693 },
  { _id: 'HEALTHY LIVING', count: 5072 },
  { _id: 'COLLEGE', count: 860 },
  { _id: 'QUEER VOICES', count: 4700 },
  { _id: 'ARTS', count: 863 },
  { _id: 'ENVIRONMENT', count: 778 },
  { _id: 'FOOD & DRINK', count: 4527 }
]
```

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
    date: '2022-09-20'
  },
  {
    _id: ObjectId('65a65bd97ad05540e85f59ca'),
    link: 'https://www.huffpost.com/entry/funniest-parenting-tweets_l_632d7d15e4b0d12b5403e479',
    headline: 'The Funniest Tweets From Parents This Week (Sept. 17-23)',
    category: 'PARENTING',
    short_description: '"Accidentally put grown-up toothpaste on my toddler's toothbrush and he screamed like I was cleaning his teeth with a Carolina Reaper dipped in Tabasco sauce."',
    authors: 'Caroline Bologna',
    date: '2022-09-23'
  },
  {
    _id: ObjectId('65a65bd97ad05540e85f59cb'),
    link: 'https://www.huffpost.com/entry/ukraine-festival_n_6327b4a0e4b082746beb52c7',
    headline: ''Beautiful And Sad At The Same Time': Ukrainian Cultural Festival Takes On A Deeper Meaning This Year',
    category: 'POLITICS',
    short_description: 'An annual celebration took on a different feel as Russia's invasion dragged into Day 206.',
    authors: 'Jonathan Nicholson',
    date: '2022-09-19'
  }
]
Type "it" for more
news_db> db.newscategory().distinct("category").length();
TypeError: db.newscategory is not a function
news_db> db.newscategory(distinct("category")).length();
ReferenceError: distinct is not defined
news_db> db.newscategory.distinct("category").length;
34
news_db> db.newscategory.aggregate([{ $group: { _id: "$category", count: { $sum: 1 } } }]);
[
  { _id: 'PARENTS', count: 3491 },
  { _id: 'COMEDY', count: 3934 },
  { _id: 'BLACK VOICES', count: 3313 },
  { _id: 'HOME & LIVING', count: 3523 },
  { _id: 'MONEY', count: 1539 },
  { _id: 'STYLE & BEAUTY', count: 7275 },
  { _id: 'IMPACT', count: 2045 },
  { _id: 'ENTERTAINMENT', count: 13463 },
  { _id: 'U.S. NEWS', count: 1893 },
  { _id: 'FIFTY', count: 1842 },
  { _id: 'GREEN', count: 1682 },
  { _id: 'MEDIA', count: 2105 },
  { _id: 'POLITICS', count: 20605 },
  { _id: 'CULTURE & ARTS', count: 693 },
  { _id: 'HEALTHY LIVING', count: 5072 },
  { _id: 'COLLEGE', count: 868 },
  { _id: 'QUEER VOICES', count: 4708 },
  { _id: 'ARTS', count: 663 },
  { _id: 'ENVIRONMENT', count: 778 },
  { _id: 'FOOD & DRINK', count: 4527 }
]
Type "it" for more
news_db>
(To exit, press Ctrl+C again or Ctrl+D or type .exit)
news_db>
```

Explain how you solved the task.

This aggregation pipeline uses the “$group” stage to group documents by the "category" field and then uses the $sum accumulator to count the number of documents in each group. The result will display each category along with its corresponding news count.

Task 7.

How many news are created in 2016?

db.newscategory.countDocuments({ date: { $gte: "2016-01-01", $lt: "2017-01-01" } })

In previous code I have just defined a query that selects documents where the "date" field is greater than or equal to January 1, 2016, and less than January 1, 2017.

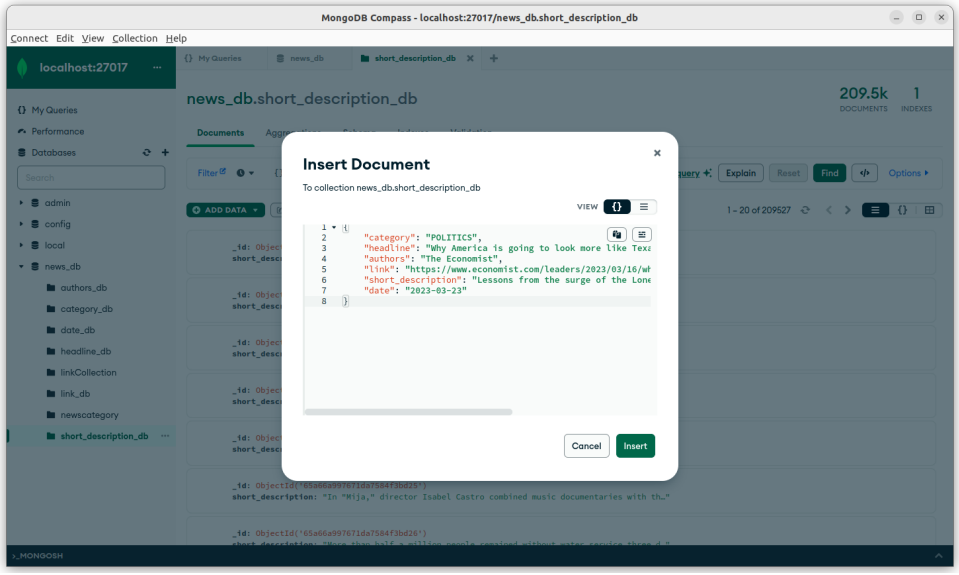
Explain how you solved the task and add the following records to the DB:

{"category": "POLITICS", "headline": "Why America is going to look more like Texas", "authors": "The Economist", "link": "https://www.economist.com/leaders/2023/03/16/why-america-is-going-to-look-more-like-texas", "short_description": "Lessons from the surge of the Lone Star State", "date": "2023-03-23"}

{"category": "POLITICS", "headline": "The Federal Reserve must choose between inflation and market chaos", "authors": "The Economist", "link": "https://www.economist.com/finance-and-economics/2023/03/19/the-federal-reserve-must-choose-between-inflation-and-market-chaos", "short_description": "Will policymakers raise interest rates as planned?", "date": "2023-03-23"}

Explain how you solved the task.

Since most of the things i have don through the mongosh , here I have decided to use MongoDB compass tool to connect with my local server, after that I have uploaded each file into newscategory collection.



Task 8.

Can you categorize news articles based on their headlines and short descriptions?

Let’s try to categories political news based on heading and short description and also using most popular words in political news such as politics, election, government

In the code below I'm searching for keywords related to politics in both the "headline" and "short_description" fields. If a document contains any of these keywords, we set its "category" to "Politics."

```
db.newscategory.aggregate([
{
$match: {
$or: [
{ headline: { $regex: /politics|election|government/i } },
{ short_description: { $regex: /politics|election|government/i } }
]
}
```

},

{

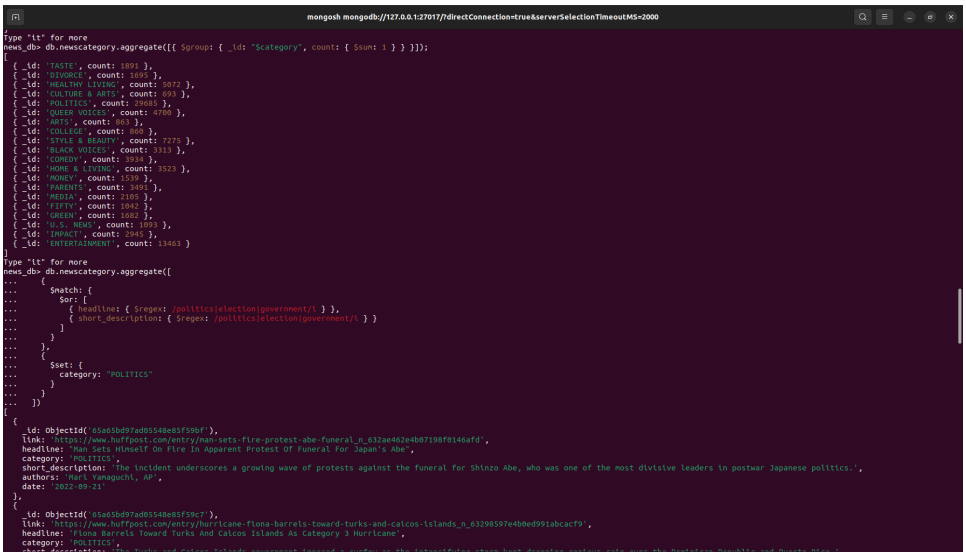
$set: {

category: "POLITICS"

}

}

])



Do news articles from different categories have different writing styles?

Seems yes because categories before and after values were same in the POLITICS category

Explain your answer.

Justification

Before

```
news_db> db.newscategory.aggregate([{ $group: { _id: "$category", count: { $sum: 1 } } }]);
[
  { _id: 'TASTE', count: 1891 },
  { _id: 'DIVORCE', count: 1695 },
  { _id: 'HEALTHY LIVING', count: 5072 },
  { _id: 'CULTURE & ARTS', count: 693 },
  { _id: 'POLITICS', count: 29685 },
  { _id: 'QUEER VOICES', count: 4700 },
  { _id: 'ARTS', count: 863 },
  { _id: 'COLLEGE', count: 860 },
  { _id: 'STYLE & BEAUTY', count: 7275 },
  { _id: 'BLACK VOICES', count: 3313 },
  { _id: 'COMEDY', count: 3934 },
  { _id: 'HOME & LIVING', count: 3523 },
  { _id: 'MONEY', count: 1539 },
  { _id: 'PARENTS', count: 3491 },
  { _id: 'MEDIA', count: 2105 },
  { _id: 'FIFTY', count: 1042 },
  { _id: 'GREEN', count: 1682 },
  { _id: 'U.S. NEWS', count: 1093 },
  { _id: 'IMPACT', count: 2945 },
  { _id: 'ENTERTAINMENT', count: 13463 }
]
Type "it" for more
```

After

```
    category: 'POLITICS',
    short_description: 'Five voters from Greene's district sought to have her removed from the ballot, saying t
    authors: 'Kate Brumback, AP',
    date: '2022-07-26'
  },
  {
    _id: ObjectId('65a65bd97ad05548e85f5b1d'),
    link: 'https://www.huffpost.com/entry/trial-expected-to-begin-for-ex-trump-adviser-steve-bannon_n_62d51ff6e
    headline: 'Jury Selection Begins In Trial Of Ex-Trump Adviser Steve Bannon',
    category: 'POLITICS',
    short_description: 'Bannon is charged in Washington's federal court with defying a subpoena from the Jan. 6
    authors: 'Gary Fields, Ashraf Khalil, AP',
    date: '2022-07-18'
  }
]
Type "it" for more
news_db> db.newscategory.aggregate([{ $group: { _id: "$category", count: { $sum: 1 } } }]);
[
  { _id: 'ENTERTAINMENT', count: 13463 },
  { _id: 'IMPACT', count: 2945 },
  { _id: 'U.S. NEWS', count: 1093 },
  { _id: 'GREEN', count: 1682 },
  { _id: 'MEDIA', count: 2105 },
  { _id: 'PARENTS', count: 3491 },
  { _id: 'MONEY', count: 1539 },
  { _id: 'COMEDY', count: 3934 },
  { _id: 'HOME & LIVING', count: 3523 },
  { _id: 'BLACK VOICES', count: 3313 },
  { _id: 'STYLE & BEAUTY', count: 7275 },
  { _id: 'FIFTY', count: 1042 },
  { _id: 'COLLEGE', count: 860 },
  { _id: 'QUEER VOICES', count: 4700 },
  { _id: 'ARTS', count: 863 },
  { _id: 'POLITICS', count: 29685 },
  { _id: 'CULTURE & ARTS', count: 693 },
  { _id: 'HEALTHY LIVING', count: 5072 },
  { _id: 'DIVORCE', count: 1695 },
  { _id: 'EDUCATION', count: 893 }
]
Type "it" for more
```

Task 9.

How many news are about Turkey?

db.newscategory.countDocuments({

$or: [

{ headline: { $regex: /turkey/i } },

{ short_description: { $regex: /turkey/i } }

]

});

```
news_db> ;

news_db>   db.newscategory.countDocuments({
...     $or: [
...       { headline: { $regex: /turkey/i } },
...       { short_description: { $regex: /turkey/i } }
...     ]
...   });
398
news_db>
```

Create a view turkey_news containing news only about Turkey.

Explain your answer.

```
news_db>  db.createView("turkey_news", "newscategory", [
...     {
...       $match: {
...         $or: [
...           { headline: { $regex: /turkey/i } },
...           { short_description: { $regex: /turkey/i } }
...         ]
...       }
...     }
...   ]);
{ ok: 1 }
news_db>
```

db.createView("turkey_news", "newscategory", [

{

$match: {

$or: [

{ headline: { $regex: /turkey/i } },

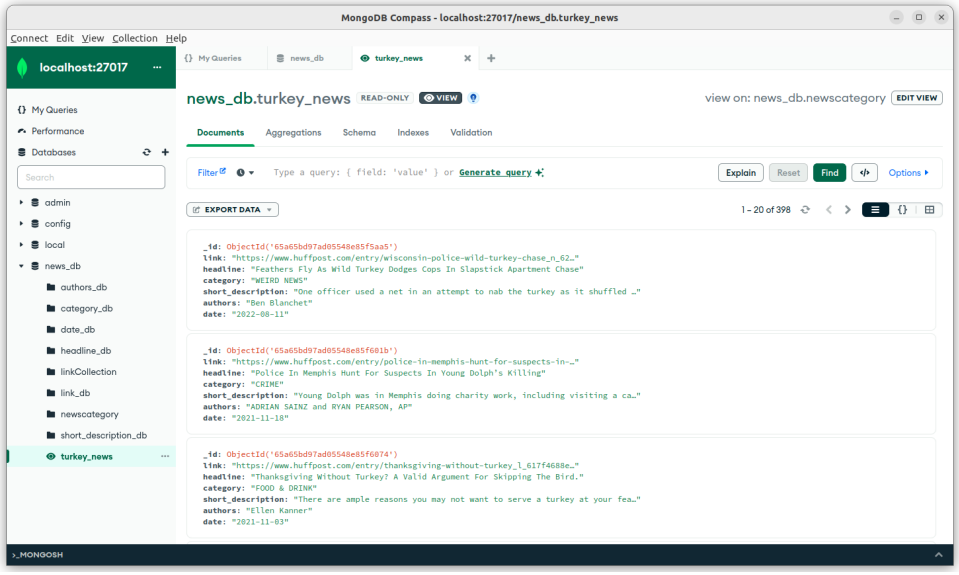
{ short_description: { $regex: /turkey/i } }

]

}

}

]);



The regular expression “/turkey/i “ is used for case-insensitive matching. This ensures that variations like "Turkey," "turkey," or "TURKEY" are all considered when regex matching.

The $or operator is used to match documents where the keyword "Turkey" is present in either the headline or short description.

Creating a View: The createView method is used to create a view named turkey_news. The view is based on the original newscategory collection, and the documents in the view are filtered based on the specified criteria.

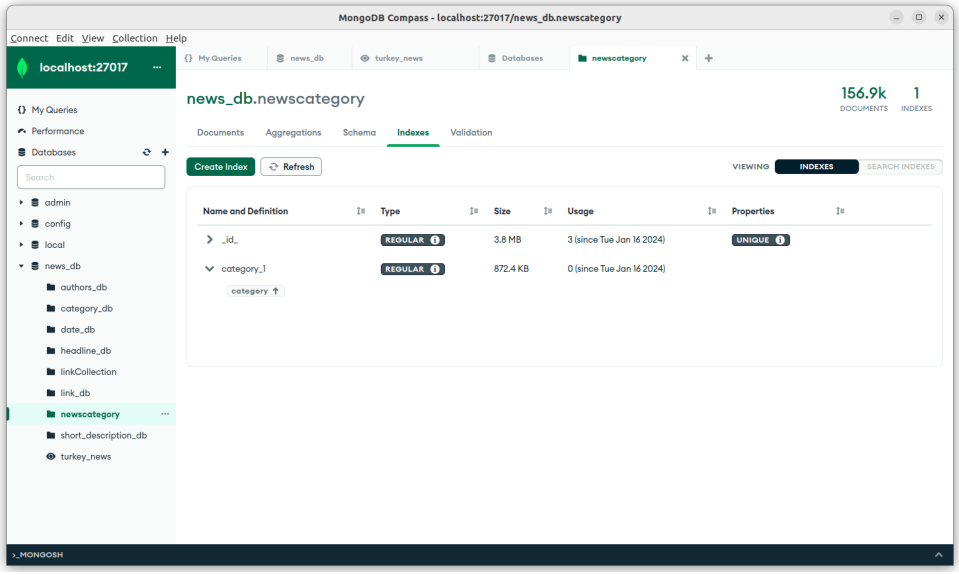
Task 10.

Add indexes to news_db based on news categories.

Explain how you solved the task.

```
db.newscategory.createIndex({ category: 1 });
```

To enhance query performance, I added an index to the "category" field in the newscategory collection using MongoDB's createIndex method. This index enables efficient retrieval of documents based on their categories, optimizing queries involving category-based filtering, sorting, or aggregation. Indexing is a crucial optimization strategy, and it's essential to consider query patterns and strike a balance between improved read performance and potential write operation overhead.



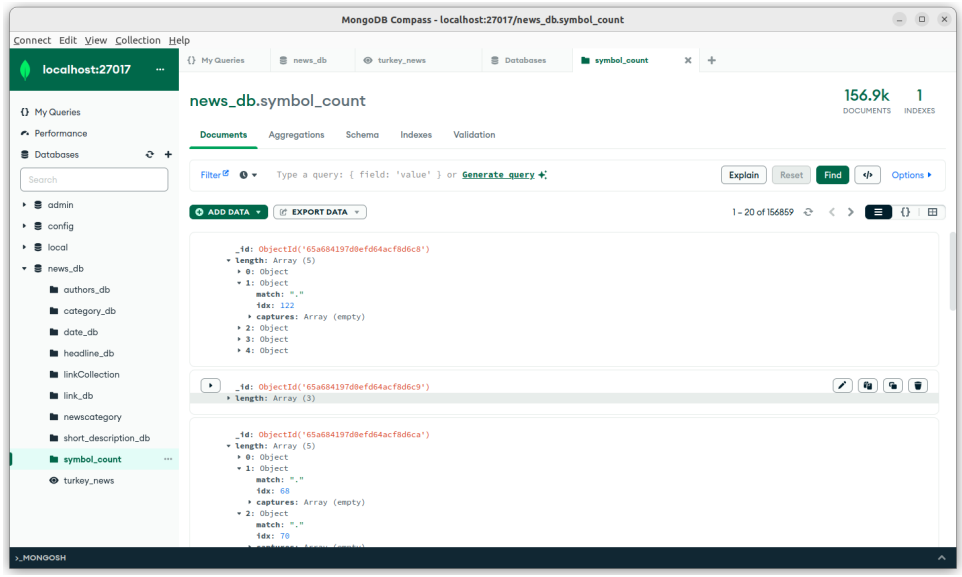
Task 11.

Add new collection that contains number of symbols in short_description.

Explain how you solved the task.

```
db.createCollection("symbol_count");
db.symbol_count.insertMany(
db.newscategory.aggregate([
{
$project: {
_id: 0,
length: { $regexFindAll: {
input: "$short_description",
regex: /[^a-zA-Z0-9\s]/g
} }
}
}
]).toArray()
);
```

```
db.symbol_count.insertMany(
  db.newscategory.aggregate([
    {
      $project: {
        _id: 0,
        length: {  $regexFindAll: {
            input: "$short_description",
            regex: /[^a-zA-Z0-9\s]/g
          } }
      }
    }
  ]).toArray()
);
```



The createCollection method establishes a new 'symbol_count' collection. Using the aggregation framework, the $project stage calculates the length of each short description using $regexFindAll, ensuring accurate Unicode code point counting based on given regex filtering. The results are inserted into 'symbol_count'.

Task 12.

Remove obsolete records that are older than 1 Jan 2016 from news db. How many records left in the database?

Explain how you solved the task.

Initial data count - db.newscategory.countDocuments(); - 156859

```
news_db> db.newscategory.deleteMany({ date: { $lt: Date("2016-01-01T00:00:00Z") } });
{ acknowledged: true, deletedCount: 156859 }
news_db> db.newscategory.deleteMany({ date: { $lt: "2016-01-01T00:00:00Z" } });
{ acknowledged: true, deletedCount: 130357 }
news_db> db.newscategory.countDocuments();
79170
news_db>
```

Deleted document count - db.newscategory.deleteMany({ date: { $lt: "2016-01-01T00:00:00Z" } }); - 130357

Remaining count - 79170

Task 13.

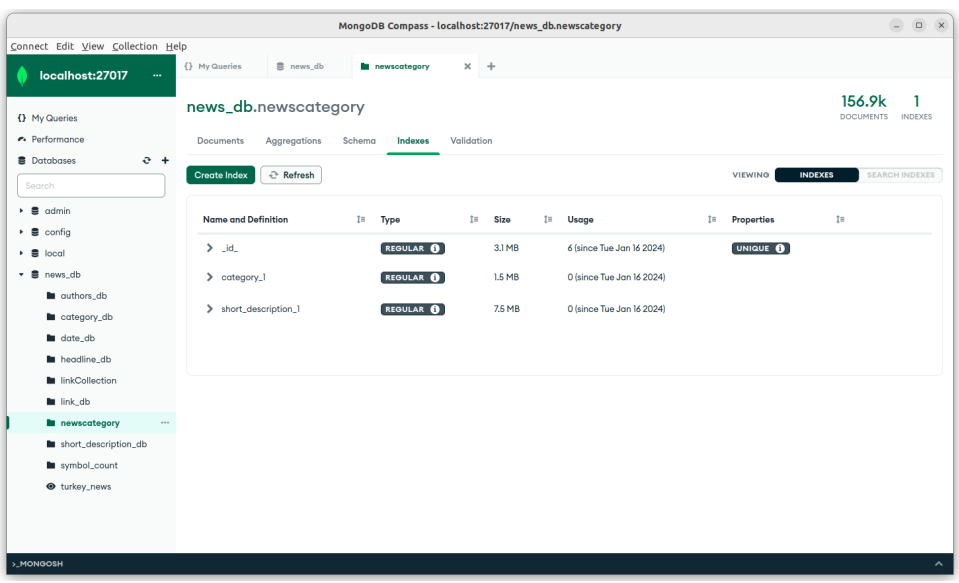
Apply aggregated function to news_db that will sort dataset ascending based on indexed categories and length of short_description. (You worked on them in previous tasks).

Explain how you solved the task.

First create index in both categories in ascending order and length of short_description

db.newscategory.createIndex({ category: 1 });

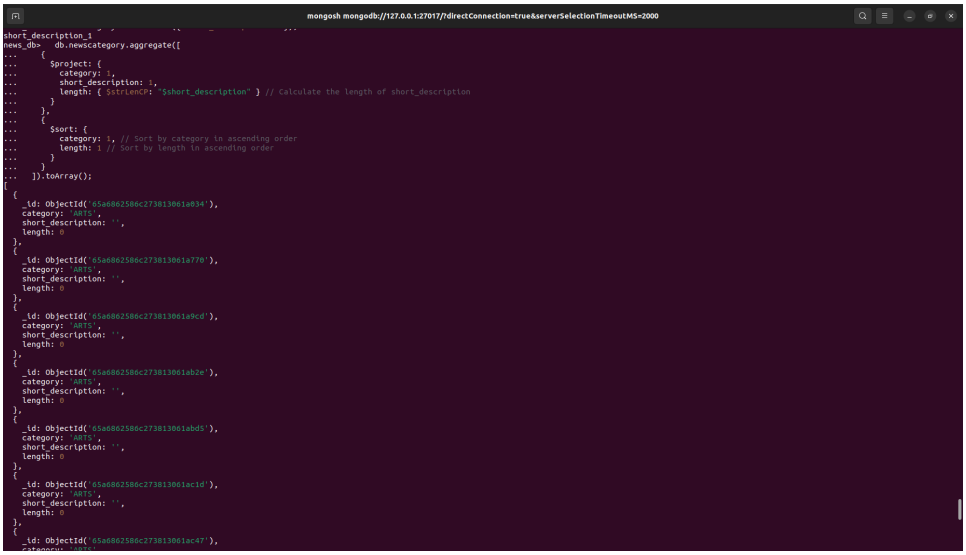
db.newscategory.createIndex({ short_description: 1 });



Use the $sort stage in the aggregation pipeline to sort the dataset based on the "category" field in ascending order. Additionally, sort based on the assumed "length" field, which should contain the length of the short_description.

db.newscategory.aggregate([

```
{
$project: {
category: 1,
short_description: 1,
length: { $strLenCP: "$short_description" } // Calculate the length of short_description
}
},
{
$sort: {
category: 1, // Sort by category in ascending order
length: 1 // Sort by length in ascending order
}
}
]).toArray();
```



In the above MongoDB aggregation pipeline, the createIndex method is initially used to ensure indexes on the "category" and "short_description" fields for optimized sorting. The aggregation pipeline then

employs the $project stage to calculate the length of the "short_description" using the $strLenCP operator, creating a new field called "length." Subsequently, the $sort stage arranges the dataset in ascending order based on both the "category" and "length" fields. The resulting sorted dataset is converted to an array using toArray(). This comprehensive process facilitates efficient sorting of the news_db collection, incorporating both category and short_description length considerations.