First take your student number and divide it by 3. Use the remainder value (modulus) to pick one of the following worksheets:

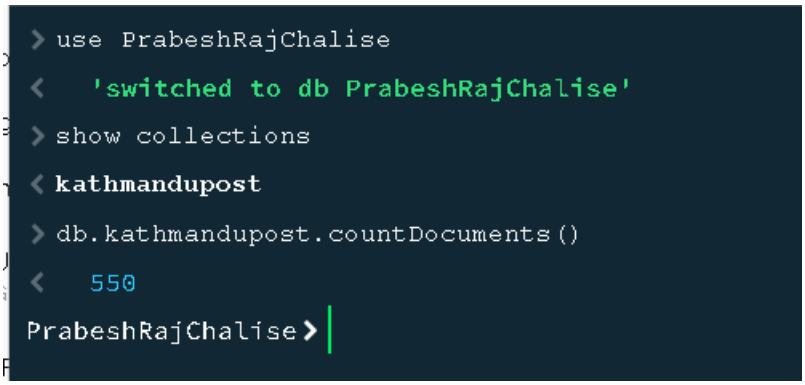
The output of my student number (2227938) modulus 3 was 0. So, the dataset I had to use was KathmanduPost.json

Import the data into your own MongoDB database:

-Show the command to do:

```
C:\Program Files\MongoDB\Tools\100\bin>mongoimport --host localhost --port 27017 --username prabesh --password prabesh --authenticationDatabase admin --db PrabeshRajChalise --collection kathmandupost --file "C:\Users\prabe\Downloads\InternalAssignment2_90195\Dataset - Assignment 2\kathmandupost.json"
```

-Write a command to show how many documents are in your collection:



Analyze the data:

Write a command to:

- Show one document:

```
> db.kathmandupost.findOne()
<   {
      _id: ObjectId("660542f8d457bdf16cda9787"),
      contributors: null,
      truncated: true,
      text: 'Source: @kathmandupost Says "Nepalese People from China are coming
      is_quote_status: false,
      in_reply_to_status_id: null,
      id: 1224637885917192200,
      favorite_count: 0,
      entities: {
        symbols: [],
        user_mentions: [
          {
            id: 625760052,
            indices: [
              8,
              22
            ],
            id_str: '625760052',
            screen_name: 'kathmandupost',
            name: 'The Kathmandu Post'
          },
          {
            id: 4642093093,
```

- Show the unique values in one field :

```
> db.kathmandupost.distinct("in_reply_to_user_id")
<   [
      null,                 15425377,
      29188119,             86678613,
      157682755,            223064868,
      256944885,            367658713,
      399923070,            625760052,
      1091554020,           1114801549,
      1356248622,           3348257833,
      837325668228730900,   1006750186985566200,
      1111659036255109100
    ]
```

- Shows a set of documents based on some criteria. Output just two fields from the document :

```
> db.kathmandupost.find({}, { text: 1, retweet_count: 1, _id: 0 })
<   {
      text: 'Source: @kathmandupost Says "Nepalese People from China are coming and going ho
      retweet_count: 0
    }
    {
      text: '@arjunpoudeltkp @kathmandupost 2/3 the location for the screening was before ar
      retweet_count: 0
    }
    {
      text: '@arjunpoudeltkp @kathmandupost 1/3 when the previous virus event took place a f
      retweet_count: 0
    }
    {
      text: '@anilkathmandu @sameAshis @RabindraMishra @sangamprasai @kathmandupost पढेर सुनाइ
      retweet_count: 0
    }
```

- Use a regular expression to search for some criteria. The search should be case insensitive :

```
> db.kathmandupost.find({ "text": { $regex: /Nepal/i }})
<   {
      _id: ObjectId("660542f8d457bdf16cda9787"),
      contributors: null,
      truncated: true,
      text: 'Source: @kathmandupost Says "Nepalese People from China are coming a
      is_quote_status: false,
      in_reply_to_status_id: null,
      id: 1224637885917192200,
      favorite_count: 0,
      entities: {
        symbols: [],
        user_mentions: [
          {
            id: 625760052,
            indices: [
              8,
              22
            ],
            id_str: '625760052',
            screen_name: 'kathmandupost'
```

Reshape the collection Write a command to:

- Update a field within the collection:

```
> db.kathmandupost.updateMany({},{ $set: { "sentiment": "positive" } })
<   {
      acknowledged: true,
      insertedId: null,
      matchedCount: 550,
      modifiedCount: 550,
      upsertedCount: 0
    }
PrabeshRajChalise>
```

- Create a new collection based on a subset of the dataset. Include a query to show a document from the new collection:

```
> db.kathmandupost.aggregate([{$match: { "text": /Nepal/i }},{$out: "new"}])
<
> db.new.findOne()
<   {
      _id: ObjectId("660542f8d457bdf16cda9787"),
      contributors: null,
      truncated: true,
      text: 'Source: @kathmandupost Says "Nepalese People from China are coming and
      is_quote_status: false,
      in_reply_to_status_id: null,
      id: 1224637885917192200,
      favorite_count: 0,
      entities: {
        symbols: [],
        user_mentions: [
          {
            id: 625760052,
            indices: [
              8,
```

Name one advantage to using this approach for handling Big Data and include a brief explanation of why you think this is an advantage.

One major advantage of using MongoDB for handling big data is its flexible schema design. MongoDB is a NoSQL database, which means it can handle semi structured data which does not have a fixed structure. In bigdata there may be many conditions where data are not structured, in such condition the flexible schema design of MongoDB can be very fruitful. By flexible schema design, it means that we should not predefine the structure of the data, we can operate the operations without stating the schema of the data. In context of relational and SQL database there is a fixed structure or schema which will not take the data out of that structure, or if the structure is defined for that field, then empty data will be stored there with null value which can take storage unnecessarily. Hence, for handling bigdata MongoDB can be a good approach.

Name one disadvantage to using this approach for handling Big Data and include a brief explanation of why you think this is a disadvantage.

One of the major disadvantages of using MongoDB for handling big data is limited scalability. Since MongoDB provides horizontal scaling of data, where data are distributed in different servers, it may face the problem in managing large datasets, in compared to other specialized distributed database like Apache Cassandra and Hadoop. Since these databases have major purpose to handle big data, such databases can be more better than MongoDB for handling the bigdata.