

Title: Concept of pagination and looping

Aim: To apply pagination and looping on covid dataset

Name: Krish Agarwal

Register Number: 21112016

Class: 4BSc DS

NoSQL Lab5

03/03/2023

Importing the dataset

```
C:\Users\KRISH>mongoimport --db employee --collection covid_data --type=csv --headerline --file="D:\Z\Downloads\covid_worldwide.csv"
2023-03-03T13:48:34.051+0530    connected to: mongodb://localhost/
2023-03-03T13:48:34.085+0530    231 document(s) imported successfully. 0 document(s) failed to import.
```

```
employee> db.covid_data.find()
[
  {
    _id: ObjectId("6401ad5a22ed84f860cfed6b"),
    'Serial Number': 1,
    Country: 'USA',
    'Total Cases': '104,196,861',
    'Total Deaths': '1,132,935',
    'Total Recovered': '101,322,779',
    'Active Cases': '1,741,147',
    'Total Test': '1,159,832,679',
    Population: '334,805,269'
  },
  {
    _id: ObjectId("6401ad5a22ed84f860cfed6c"),
    'Serial Number': 2,
    Country: 'India',
    'Total Cases': '44,682,784',
    'Total Deaths': '530,740',
    'Total Recovered': '44,150,289',
    'Active Cases': '1,755',
    'Total Test': '915,265,788',
    Population: '1,406,631,776'
  },
  {
    _id: ObjectId("6401ad5a22ed84f860cfed6d"),
    'Serial Number': 4,
    Country: 'Germany',
    'Total Cases': '37,779,833',
    'Total Deaths': '165,711',
    'Total Recovered': '37,398,100',
    'Active Cases': '216,022',
    'Total Test': '122,332,384',
    Population: '83,883,596'
  },
  {
    _id: ObjectId("6401ad5a22ed84f860cfed6e"),
    'Serial Number': 5,
    Country: 'Brazil',
    'Total Cases': '36,824,580',
    'Total Deaths': '697,074',
    'Total Recovered': '35,919,372',
    'Active Cases': '208,134',
    'Total Test': '63,776,166',
    Population: '215,353,593'
  },
]
```

Q1) Display the top 10 countries name and Total_Death rate in highest to lowest order in page 1.

```
employee> var cur1 = db.covid_data.find({}, {'Country': 1, 'Total_Deaths': 1}).sort({'Total_Deaths': -1}).limit(10)
employee> while (cur1.hasNext())
... {
... print(cur1.next())
... }
{
  _id: ObjectId("6401ad5a22ed84f860cfedce"),
  Country: 'North Macedonia',
  'Total_Deaths': '9,638'
}
{
  _id: ObjectId("6401ad5a22ed84f860cfedb5"),
  Country: 'Saudi Arabia',
  'Total_Deaths': '9,574'
}
{
  _id: ObjectId("6401ad5a22ed84f860cfed9d"),
  Country: 'Lithuania',
  'Total_Deaths': '9,545'
}
{
  _id: ObjectId("6401ad5a22ed84f860cfeda5"),
  Country: 'Costa Rica',
  'Total_Deaths': '9,158'
}
{
  _id: ObjectId("6401ad5a22ed84f860cfedc3"),
  Country: 'Armenia',
  'Total_Deaths': '8,717'
}
{
  _id: ObjectId("6401ad5a22ed84f860cfed9b"),
  Country: 'Finland',
  'Total_Deaths': '8,639'
}
{
  _id: ObjectId("6401ad5a22ed84f860cfedb0"),
  Country: 'Panama',
  'Total_Deaths': '8,596'
}
{
  _id: ObjectId("6401ad5a22ed84f860cfeda6"),
  Country: 'Cuba',
  'Total_Deaths': '8,530'
}
```

Q2) Display the next 10 countries details in page 2.

```
employee> var cur2 = db.covid_data.find({}, {'Country': 1, 'Total_Deaths': 1}).sort({'Total_Deaths': -1}).limit(10).skip(10)
employee> while (cur2.hasNext()){
... print(cur2.next())
}
{
  _id: ObjectId("6401ad5a22ed84f860cfedea"),
  Country: 'Afghanistan',
  'Total_Deaths': '7,882'
}
{
  _id: ObjectId("6401ad5a22ed84f860cfedaf"),
  Country: 'Uruguay',
  'Total_Deaths': '7,609'
}
{
  _id: ObjectId("6401ad5a22ed84f860cfedd0"),
  Country: 'Ethiopia',
  'Total_Deaths': '7,572'
}
{
  _id: ObjectId("6401ad5a22ed84f860cfedb3"),
  Country: 'Belarus',
  'Total_Deaths': '7,118'
}
{
  _id: ObjectId("6401ad5a22ed84f860cfeda0"),
  Country: 'Slovenia',
  'Total_Deaths': '7,056'
}
{
  _id: ObjectId("6401ad5a22ed84f860cfedee"),
  Country: 'Brazil',
  'Total_Deaths': '697,074'
}
{
  _id: ObjectId("6401ad5a22ed84f860cfed6f"),
  Country: 'Japan',
  'Total_Deaths': '68,399'
}
{
  _id: ObjectId("6401ad5a22ed84f860cfed90"),
  Country: 'Romania',
  'Total_Deaths': '67,576'
}
{
  _id: ObjectId("6401ad5a22ed84f860cfed91"),
  Country: 'Philippines',
  'Total_Deaths': '65,802'
}
```

Q3) Likewise continue for next 2 pages.

```
employee> var cur3 = db.covid_data.find({}, {'Country': 1, 'Total Deaths': 1}).sort({'Total Deaths': -1}).limit(10).skip(20)
employee> while (cur3.hasNext()) {print(cur3.next())}
MongoCursorExhaustedError: Cursor is exhausted
employee> while (cur3.hasNext()) {print(cur3.next())}
{
  _id: ObjectId("6401ad5a22ed84f860cfedd2"),
  Country: 'Algeria',
  'Total Deaths': '6,881'
}

{
  _id: ObjectId("6401ad5a22ed84f860cfedca"),
  Country: 'Libya',
  'Total Deaths': '6,437'
}

{
  _id: ObjectId("6401ad5a22ed84f860cfedb4"),
  Country: 'Latvia',
  'Total Deaths': '6,218'
}

{
  _id: ObjectId("6401ad5a22ed84f860cfed6c"),
  Country: 'India',
  'Total Deaths': '530,740'
}

{
  _id: ObjectId("6401ad5a22ed84f860cfed8b"),
  Country: 'Canada',
  'Total Deaths': '50,380'
}

{
  _id: ObjectId("6401ad5a22ed84f860cfede3"),
  Country: 'Venezuela',
  'Total Deaths': '5,845'
}

{
  _id: ObjectId("6401ad5a22ed84f860cfedc8"),
  Country: 'Kenya',
  'Total Deaths': '5,688'
}

{
  _id: ObjectId("6401ad5a22ed84f860cfedd6"),
  Country: 'Zimbabwe',
  'Total Deaths': '5,652'
}

{
  _id: ObjectId("6401ad5a22ed84f860cfedb1"),
  Country: 'Palestine',
  'Total Deaths': '5,404'
}
```

```
employee> var cur4 = db.covid_data.find({}, {'Country': 1, 'Total Deaths': 1}).sort({'Total Deaths': -1}).limit(10).skip(40)
employee> while (cur4.hasNext()) {print(cur4.next())}
{
  _id: ObjectId("6401ad5a22ed84f860cfed9a"),
  Country: 'Norway',
  'Total Deaths': '5,112'
}

{
  _id: ObjectId("6401ad5a22ed84f860cfedfb"),
  Country: 'Sudan',
  'Total Deaths': '5,001'
}

{
  _id: ObjectId("6401ad5a22ed84f860cfedad"),
  Country: 'Hungary',
  'Total Deaths': '48,677'
}

{
  _id: ObjectId("6401ad5a22ed84f860cfed83"),
  Country: 'Vietnam',
  'Total Deaths': '43,186'
}

{
  _id: ObjectId("6401ad5a22ed84f860cfed89"),
  Country: 'Czechia',
  'Total Deaths': '42,312'
}

{
  _id: ObjectId("6401ad5a22ed84f860cfedc7"),
  Country: 'Oman',
  'Total Deaths': '4,628'
}

{
  _id: ObjectId("6401ad5a22ed84f860cfedbd"),
  Country: 'Dominican Republic',
  'Total Deaths': '4,384'
}

{
  _id: ObjectId("6401ad5a22ed84f860cfedee"),
  Country: 'Trinidad and Tobago',
  'Total Deaths': '4,326'
}

{
  _id: ObjectId("6401ad5a22ed84f860cfedeb"),
  Country: 'El Salvador',
  'Total Deaths': '4,230'
}
```

Q4) Display the count of active cases more than 68399.

```
employee> db.covid_data.find({"Active Cases": {$gt: '68399'}}).count()  
40
```

Q5) Display all the country names with their death rate in format “Country Name: Total_Death”.

```
employee> var cur5 = db.covid_data.find({}, {"Country": 1, "Total Deaths": 1}).forEach(function(myDoc) {print(myDoc.Country + ": " + myDoc["Total Deaths"])}))  
USA:1,132,935  
India:530,740  
Germany:165,711  
Brazil:697,074  
Japan:68,399  
S. Korea:33,486  
UK:204,171  
France:164,233  
Italy:186,833  
Turkey:101,492  
Russia:395,108  
Australia:18,615  
Spain:118,434  
Taiwan:16,356  
Argentina:130,421  
Netherlands:22,989  
Iran:144,749  
Mexico:332,198  
Indonesia:160,817  
Poland:118,736  
Austria:21,689  
Colombia:142,486  
Greece:35,630  
Portugal:26,022  
Vietnam:43,186  
Ukraine:111,020  
Chile:63,812  
Malaysia:36,942  
Israel:12,193  
Thailand:33,865  
Czechia:42,312  
Belgium:33,557  
Canada:50,380  
Peru:218,931  
Switzerland:14,452  
South Africa:102,595  
DPRK:74  
Romania:67,576  
Philippines:65,802  
Sweden:23,246
```

Q6) Check for null entry attempts made for a particular key and print the same.

There are no Null entries in the “Active Cases” column.s

```
employee> var cur5 = db.covid_data.find({"Active Cases": {$exists: false}})  
  
employee> while (cur5.hasNext()) {print(cur5.next())}
```