**ASSIGNMENT 5**

|  |  |  |
| --- | --- | --- |
| **Student ID** | **Name** | **Email** |
| 8800060 | Zarana Gohil | Zgohil0060@conestogac.on.ca |
| 8800149 | Basavraj Jaliminche | Bjaliminche0149@econestogac.on.ca |
| 8807575 | Dharti Patel | Dpatel757@conestogac.on.ca |

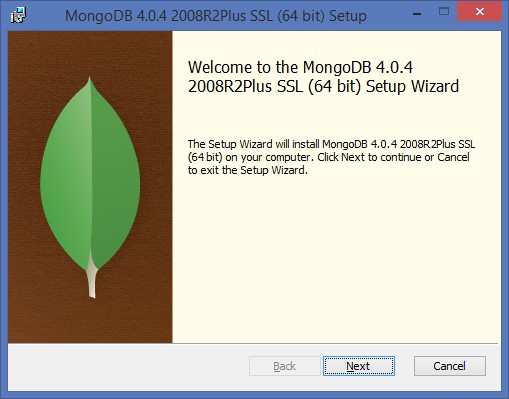
**PROG8600 Laboratory Exercise 7**

In the previous laboratory exercise, we downloaded and installed MongoDB Community Server and worked with data using MongoDB Shell. In this exercise, we will build on the previous work and further operate on data in MongoDB. Navigate to https://www.codewall.co.uk/mongodb-beginner-tutorial-with-compass-gui-the-mongo-shell-cli/ and carefully read and complete the tutorial. As you complete the tutorial, write your own notes on the tasks completed, observations made, difficulties encountered, and any questions. Save your completed work and notes, as they may be required to be formally submitted as part of a future assignment

**Solution :**

**Step 1 :**

* Firstly we have downloaded the latest version of MongoDB from  <https://www.mongodb.com/download-center/community>.
* Then follow the steps as per instruction and installed the downloaded file.



* after installation of mongo dB we can check it using Command prompt.
* Enter the following command – cd C:\Program Files\MongoDB\Server\5.0\bin
* Then enter the following command – mongo.

Text

Description automatically generatedcoon

**Step 2 :**

* Go to the following download page URL for Compass <https://www.mongodb.com/download-center/compass?jmp=docs>
* Pick the **Community Edition Stable** version from the drop-down.
* Select Windows 64-bit (7+) from the platforms dropdown.
* Click the download button.
* Wait for the download to complete.

**Installing Compass for MongoDB**

* Locate and launch the MongoDB Compass installer that was downloaded earlier.
* Read the License Agreement and click agree if you are happy with the conditions.
* Navigate through the series of how-to panes until you reach the privacy setting window and tick/untick to your preference.
* Lastly, click Start Using Compass.
* Now create a new connection
* Hostname: localhost or Remote Server IP
* Port: 27017
* Leave the rest of the fields as their default and press connect.

Now, generate a CSV file from the website [https://generatedata.com](https://www.generatedata.com/). The data consists of people

* Open up MongoDB Compass Community that was installed earlier.
* Connect to the MongoDB Server by specifying parameters as below
  1. Hostname: localhost
  2. Port: 27017
* Click connect.
* From the loaded dashboard click the Create database button.
* The created database helper will be loaded.
* Specify the Database Name as peopledb
* Specify the Collection Name as people
* Click the Create Database button.

### Creating documents (data)

* Locate the People Database under the MyCluster Pane located on the left-hand pane and click it.
* After that click on ADD DATA button where we can find two options (Import file , Import document)
* Then we have imported the already downloaded CSV file.
* to view data need just click on database from side panel and then click on collection.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generated

**Update:**

* To update data need click on button as shown in below image.
* Double click the value field of the data you want to update. We can update single or multiple elements values.
* You will notice a Document Modified notification will appear, click the update button on this notification to save changes

Graphical user interface, application

Description automatically generated

**Delete :**

* Locate the document you want to update and hover over it until the tooltip pops up.
* Click the delete Trash Can icon as shown in Figure.
* Graphical user interface, text, application, email

  Description automatically generatedA notification will appear stating the document has been flagged for deletion in red, click the delete button as shown in Figure and the document will be deleted.

**View data after update and delete command :**

Graphical user interface, text, application, email

Description automatically generated

###### **Create the DB using cmd command**

* Type use peopledb. (This command **creates** the peopledb database)
* Press enter, and it should be displayed as below image

Text

Description automatically generated

**Inserting one document**

* Make sure you are using the peopledb database
* Using the command db.people.insertOne() will create the People collection automatically if it does not exist. Between the parenthesis is where the JSON data should go.
* Use the *CodeBlock 1* below to insert a row of the CSV file dataset.

db.people.insertOne({"Name":"Grace","phone":"120451545","email":"ch@hjnnkk.ca","postalzip":"78684w})

However, this command gives error something like this as shown below figure.

Text

Description automatically generated

### Viewing documents (data) :

* Make sure you are using the peopledb database
* Type the command db.people.find() and press enter. The documents in the collection will be returned to the console. See Figure .

A picture containing text

Description automatically generated

**Update operation:**

db.people.update(

{" **name**" : “Wayne Lara”},

{$set: { " country" : "Canada"}})

Text

Description automatically generated

#### **With Compass**

Using the Compass query bar, we can specify projection fields easily. Following the steps below to perform a projection query on the collection.

* Locate the database and collection of documents you want to query in the left-hand pane and click it.
* After the view is loaded locate the query bar and press the options button as shown in Figure and click it.
* Four options will now appear on the left-hand side, one of these is Project.
* Fill out fields that you want to return in the result set in JSON format where the first value in the field and the second is a Boolean 1 or 0 to show the field or not. For example {email : 1}
* After filling out the JSON within the project option, click the find button as shown in Figure
* The result set returned will now be displayed in the list view below the query bar. See Figure for an example.

**Query bar options button :**

Graphical user interface, application, email

Description automatically generated

**Find button & Projected result set :**

**Graphical user interface, text, application, email

Description automatically generated**

#### **With MongoDB CLI :**

#### See the following steps –

1. Connect to MongoDB instance with the CLI.
2. Type use peopledb
3. Type db.people.find({}, {email : 1})
4. Notice the first empty object parameter in the method above, as we do not want to specify any criteria, in particular, just simply ask for all results instead.
5. Also, notice the default behavior of the output, the \_id field is also included without request.
6. See Figure for the output.

Text

Description automatically generated

#### **Sort With Compass :**

Sorting of data (documents) can be utilized by using the Sort option which takes a field name and the ascending or descending parameter in the format of a number. 1 for Ascending and -1 for descending.

* Locate the database and collection of documents you want to query in the left-hand pane and click it.
* After the view is loaded locate the query bar and press the options button as shown in Figure 21 and click it.
* Four options will now appear on the left-hand side, one of these is Sort.
* In the sort option bar enters the parameters in the format of JSON. For example {name: -1}
* Press the find button and you will see the document view now be ordered by length descending. See figure 31 for the result view.

**Graphical user interface, text, application, email

Description automatically generated-1 for descending**

**1 for Ascending :**

**Graphical user interface, text, application, email

Description automatically generated**

#### **With MongoDB CLI :**

#### See the following steps –

* Connect to MongoDB instance with the CLI.
* Type use peopledb
* Type db.people.find({}, {email : 1}).sort({email : -1})for descending result.
* Type db.people.find({}, {email : 1}).sort({email : 1})for ascending result.
* See Figure for results sorted by age.

**Text

Description automatically generated**

### Skip

Skip serves the purpose of skipping x number of documents during a query search, by specifying an integer, the MongoDB will skip this amount and return the results next in the collection.

#### **With Compass**

To skip documents when executing a query, use the following steps for Compass.

* Locate the database and collection of documents you want to query in the left-hand pane and click it.
* Click the options button on the query bar and look to the bottom right of the dropped-down box, parallel to collation. Here you will find the Skip option. See Figure.
* Enter an integer, for this tutorial the number 1 will be used.
* **Graphical user interface, text, application

  Description automatically generated**Click the find button and you will see the result set with the first two collections missing. See Figure.

#### **With MongoDB CLI**

Again, this method can be chained to existing methods already in the command line, for this example we will use the commands for both the Find and Sort tutorials.

* Connect to MongoDB instance with the CLI.
* Type use peopledb
* Type db.people.find({}, {email : 1}).sort({email : -1}).skip(1)
* for descending result that skips the first document, see Figure for an example.
* Alternatively use people.find().skip(1) to request all documents, skipping the first document.

**Text

Description automatically generated**

### Limit

Limit is a filter to limit the number of results returned, this can be any number you desire and can serve really good purposes in some use cases.

#### **With Compass**

To skip documents when executing a query, use the following steps for Compass.

* Locate the database and collection of documents you want to query in the left-hand pane and click it.
* Click the options button on the query bar and look to the bottom right of the dropped-down box, parallel to collation. Here you will find the limit option. See Figure.
* Enter an integer, for this tutorial the number 1 will be used.
* Click the find button and you will only see one document in the result view. See Figure.

Graphical user interface, text, application, email

Description automatically generated

#### **With MongoDB CLI**

The Limit method is highly similar to the Skip method, it requires a single integer and again, it can be chained, see the following steps.

* Connect to MongoDB instance with the CLI.
* Type use peopledb
* Type db.people.find({}, {email : 1}).sort({email : -1}).skip(1).limit(1) See Figure for an example
* Alternatively, you can use skip without chaining to all the methods above. Use the command people.find().limit(1) to retrieve the first document in the collection.

Text

Description automatically generated

### Indexing

Just like in SQL server, MongoDB can place indexes on specific fields too. This helps speed up queries but at the same time can add more hard-disk usage with the more indexes utilized.

#### With Compass

By using Compass GUI, indexes can be added to fields in the database’s collections. See the following steps to utilize indexing.

1. Locate the database and collection you want to place an index on and click it.
2. From the initial collection dashboard look to the top of the view and find the Indexes tab as shown in Figure 39 and click it.
3. Next, click the green Create Index button, as shown in Figure.
4. From the modal popup, enter an index name, something of descriptive purpose to the field you want to index. For this example, ageIndex.
5. Select the field you want to index, for this example, age was selected.
6. Choose ASC from the index type dropdown**. (MongoDB, Index Types, 2018).**
7. Click the create button, see Figure.
8. You will now be able to see the index within the indexes tab view. See Figure

Graphical user interface, text, application, email

Description automatically generated

**Index creation :**

**Graphical user interface, application

Description automatically generated**

**Create Field Index Button :**

**Graphical user interface, application

Description automatically generated**

#### **With MongoDB CLI**

Creating an index is executed with the following method createIndex(), it takes two parameters, the first being specifying the fields that you want to index and the second is specifically for index options which you can see more detail in the documentation.

* Connect to MongoDB instance with the CLI.
* Type use peopledb
* Type people.createIndex({name: 1}) to place an index on the age field.
* See Figure 43 for the successful index creation response.