MongoDB Indexing



Estimated time needed: **30** minutes

Objectives

After completing this lab you will be able to:

- Measure the time it takes to execute a query with the explain function
- Describe the process of creating, listing and deleting indexes
- Evaluate the effectiveness of an index

Prerequisites

Before starting this lab, it'll be helpful to have knowledge about accessing and performing basic operations with MongoDB. If you're unfamiliar with MongoDB, feel free to take a look at the Getting Started with MongoDB and MongoDB CRUD labs!

About Skills Network Cloud IDE

Skills Network Cloud IDE (based on Theia and Docker) provides an environment for hands on labs for course and project related labs. Theia is an open source IDE (Integrated Development Environment), that can be run on desktop or on the cloud. to complete this lab, we will be using the Cloud IDE based on Theia and MongoDB provided by Skills Network.

Important Notice about this lab environment

Please be aware that sessions for this lab environment are not persisted. Every time you connect to this lab, a new environment is created for you. Any data you may have saved in the earlier session would get lost. Plan to complete these labs in a single session, to avoid losing your data.

Exercise 1 – Get the environment ready

Open the MongoDB database page by clicking the button below:

Open MongoDB Page in IDE

On that page, click the `Create` button to create a MongoDB database.

After creation has finished, click the `MongoDB CLI` button (below the `Create` button from the previous step) to connect to the database using the mongosh CLI.

1. Problem:

   *Select the **training** database.*

   ▼ Click here for Hint

   Refer to the Mongodb Getting Started lab.

   ▼ Click here for Solution

   ```
   use training
   ```

2. Problem:

    *Create a collection named **bigdata**.*

    ▼ Click here for Hint

    Refer to the Mongodb Getting Started lab.

    ▼ Click here for Solution

    ```
    db.createCollection("bigdata")
    ```

Exercise 2 – Insert documents

- Let us insert a lot of documents into the newly created collection.
- This should take around 3 minutes, so please be patient.
- The code given below will insert 200000 documents into the 'bigdata' collection.
- Each document would have a field named **account_no** which is a simple auto increment number.
- And a field named **balance** which is a randomly generated number, to simulate the bank balance for the account.

Copy the below code and paste it on the mongo client.

```
use training
for (i=1;i<=200000;i++){print(i);db.bigdata.insert({"account_no":i,"balance":Math.round(Math.random()*1000000)})}
```

Verify that 200000 documents got inserted by running the below command.

```
db.bigdata.countDocuments()
```

Exercise 3 – Measure the time taken by a query

Let us run a query and find out how much time it takes to complete.

Let us query for the details of account number 58982.

We will make use of the explain function to find the time taken to run the query in milliseconds.

Run the below command.

```
db.bigdata.find({"account_no":58982}).explain("executionStats").executionStats.executionTimeMillis
```

Make a note of the milliseconds it took to run the query. We will need this at a later stage.

Exercise 4 – Working with indexes

Before you create an index, choose the field you wish to create an index on. It is usually the field that you query most.

Run the below command to create an index on the field **account_no**.

```
db.bigdata.createIndex({"account_no":1})
```

Run the below command to get a list of indexes on the 'bigdata' collection.

```
db.bigdata.getIndexes()
```

You should see an index named `account_no_1`

Exercise 5 – Find out how effective an index is

Let us run a query and find out how much time it takes to complete, using an index.

Let us query for the details of account number 69271.

Run the below command.

```
db.bigdata.find({"account_no": 69271}).explain("executionStats").executionStats.executionTimeMillis
```

Make a note of the milliseconds it took to run the query. Compare this with the time you recored during Exercise–3. This should be a lot less than that.

Exercise 6 – Delete an index

Use the below command to delete the index we created earlier.

```
db.bigdata.dropIndex({"account_no":1})
```

Practice exercises

1. Problem:

   *Create an index on the **balance** field.*

▼ Click here for Hint

use the command db.collection.createIndex()

▼ Click here for Solution

On the mongo client run the below commands.

```
db.bigdata.createIndex({"balance":1})
```

2. Problem:

   Query for documents with a balance of 10000 and record the time taken.

▼ Click here for Hint

use the command db.collection.find().explain()

▼ Click here for Solution

```
db.bigdata.find({"balance":10000}).explain("executionStats").executionStats.executionTimeMillis
```

3. Problem:

   Drop the index you have created.

▼ Click here for Hint

use the command db.collection.dropIndex()

▼ Click here for Solution

```
db.bigdata.dropIndex({"balance":1})
```

4. Problem:

   Query for documents with a balance of 10000 and record the time taken, and compare it with the previously recorded time.

▼ Click here for Hint

use the command db.collection.find().explain()

▼ Click here for Solution

```
db.bigdata.find({"balance": 10000}).explain("executionStats").executionStats.executionTimeMillis
```

5. Problem:

   Disconnect from the mongodb server.

▼ Click here for Hint

Refer to the Mongodb Getting Started lab.

▼ Click here for Solution

Run the below command on the terminal.

```
exit
```

Authors

Ramesh Sannareddy

Other Contributors

Rav Ahuja

© IBM Corporation. All rights reserved.