S. B. JAIN INSTITUTE OF TECHNOLOGY, MANAGEMENT & RESEARCH, NAGPUR.

Practical No. 4

Aim: To install MongoDB in windows O.S and use of basic commands though MongoDB command line interface.

Name of Student: Shrutika Pradeep Bagdi

Roll No.: CS22130

Semester/Year: 7th / 4th

Academic Session: 2025-2026

Date of Performance: __________

Date of Submission: __________

AIM: To install MongoDB in windows O.S and use of basic commands though MongoDB command line interface.

OBJECTIVE/EXPECTED LEARNING OUTCOME:
The objectives and expected learning outcome of this practical are:

- The primary objective of MongoDB is to provide a highly flexible, scalable, and efficient NoSQL database solution that can effectively manage and store large volumes of structured and semi-structured data for various applications.
- Agility: MongoDB enables rapid application development and iteration due to its flexible schema and easy integration with popular programming languages.
- Scalability: Applications can handle growth in data volume and user traffic by scaling out MongoDB clusters horizontally.
- Performance: MongoDB's indexing and data retrieval capabilities enhance query performance, ensuring fast response times for applications.
- Reduced Development Effort: Developers can focus more on application logic and less on database management tasks, resulting in reduced development effort and faster time-to-market.

HARDWARE AND SOFTWARE REQUIRMENTS:

Hardware Requirement: High Configuration computer

Software Requirement: MongoDB-7.0

THEORY:

What is MongoDB?

MongoDB is a well-known open-source NoSQL database built on the C++ programming language. MongoDB is a document-oriented database that uses JSON-like documents and a Dynamic Schema to store information. It means that while saving your data, you won't have to worry about the Data Structure, the number of fields or the types of fields used to store values. JSON objects are identical to MongoDB Documents.

Database development of modern software that utilizes transactional processing involves making some tradeoffs in terms of scalability and performance on one end and integrity on the other. These two things are peculiar to NoSQL and SQL databases, respectively. However, MongoDB, being a NoSQL database, provides both Graphical and Command-line interfaces (Windows MongoDB Shell) for interacting with your database.

MongoDB is a document-oriented database created by MongoDB Inc to provide software database services. It is classified as a NoSQL database due to its document-oriented nature. Meanwhile, MongoDB documents are represented in a JSON-like structure.

MongoDB provides shell interactions via tools like Mongo and **mongosh**. **MongoDB shell** or **mongosh** is a superset of mongo providing improved syntax highlighting, command history, and improved logging. Mongosh works typically like mongo so a user doesn't have to learn new syntax.

Key Features of MongoDB

In comparison to other traditional databases, **MongoDB** has a number of unique features that make it a superior choice. The following are some of these attributes:

Index-based Document: In a MongoDB database, every field in the Document is indexed with Primary and Secondary Indices, making it easier to get data from the pool.

Horizontal Scalability: It is possible with MongoDB's sharding. The practice of distributing data over numerous servers is known as sharding. The Shard Key is used to partition a huge quantity of data into data chunks, which are then uniformly dispersed among Shards that span several Physical Servers.

Schema-Less Database: This type of Database stores many sorts of documents in a single collection (the equivalent of a table). To put it another way, a single collection in the MongoDB database can hold numerous documents, each having its own set of **Fields, Content,** and **Size**. It is not required that one document be comparable to another, as it is with Relational Databases. MongoDB provides a lot of freedom to consumers because of this capability.

Replication: MongoDB provides high availability of data by generating several copies of the data and sending these copies to a separate Server, allowing the data to be retrieved even if one server fails

Installing MongoDB

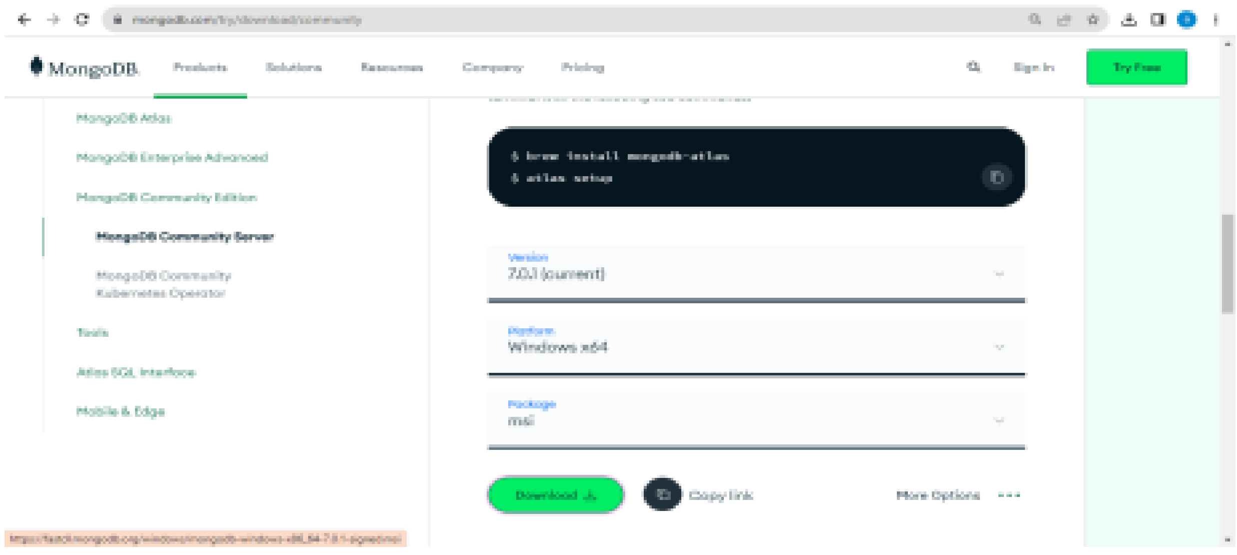
To start with Windows MongoDB Shell installation you need to have MongoDB installed in the first place. If you don't already have MongoDB installed on your computer, the first section will put you through just before moving on to the installation of the MongoDB shell

Step 1: Download The Installer

Go to the download page at https://www.mongodb.com/try/download/community

Choose your OS and your desired MongoDB version.

Click Download.



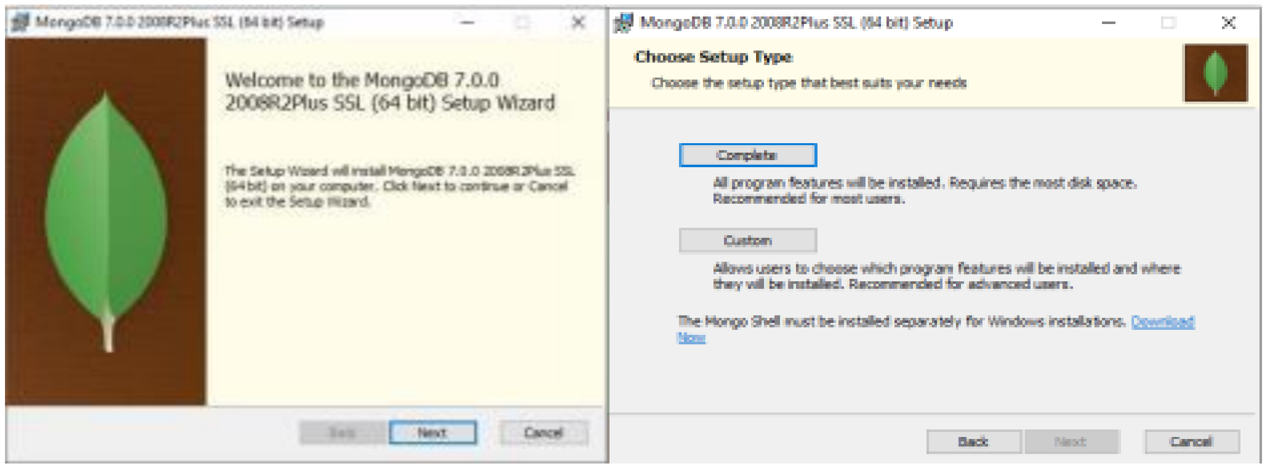
```
$ brew install mongodb-atlas
$ atlas setup
```

Version: 7.0.1 (current)
Platform: Windows x64
Package: msi

https://fastdl.mongodb.org/windows/mongodb-windows-x86_64-7.0.1-signed.msi

Step 2: Run The MongoDB Installer(a .msi file)

- Go to your 'Downloads' folder.
- Click on the installer.
- Follow the instructions.



Welcome to the MongoDB 7.0.0 2008R2Plus SSL (64 bit) Setup Wizard

The Setup Wizard will install MongoDB 7.0.0 2008R2Plus SSL (64 bit) on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

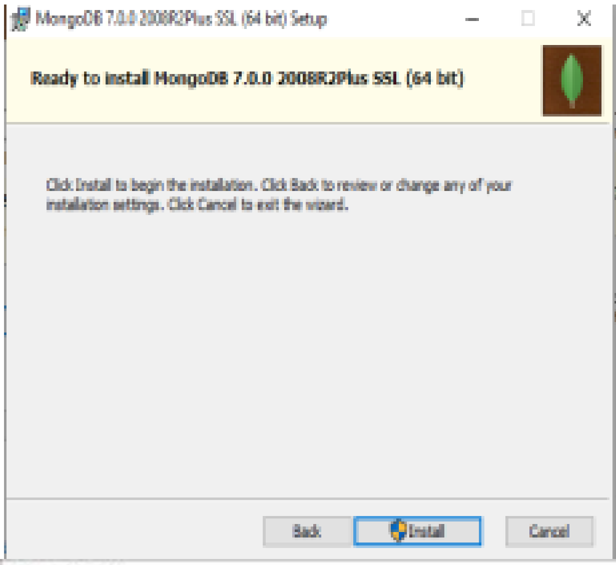
Choose Setup Type

Choose the setup type that best suits your needs

Complete: All program features will be installed. Requires the most disk space. Recommended for most users.

Custom: Allows users to choose which program features will be installed and where they will be installed. Recommended for advanced users.

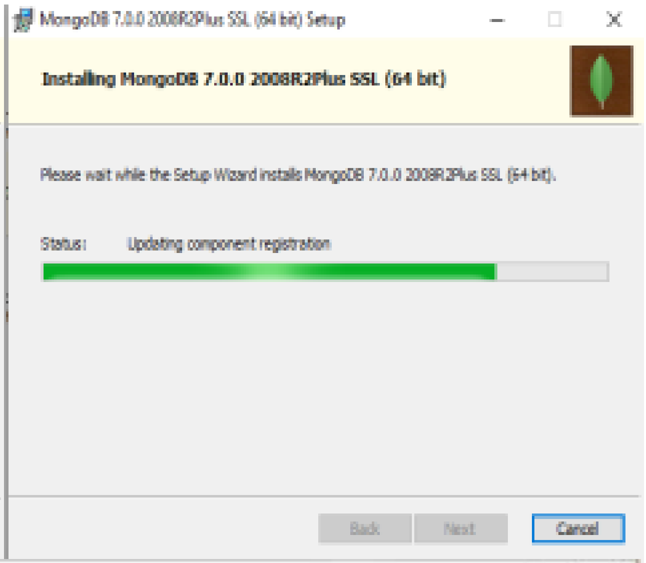
The Mongo Shell must be installed separately for Windows installations. Download Now



MongoDB 7.0.0 2008R2Plus SSL (64 bit) Setup

Ready to install MongoDB 7.0.0 2008R2Plus SSL (64 bit)

Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.

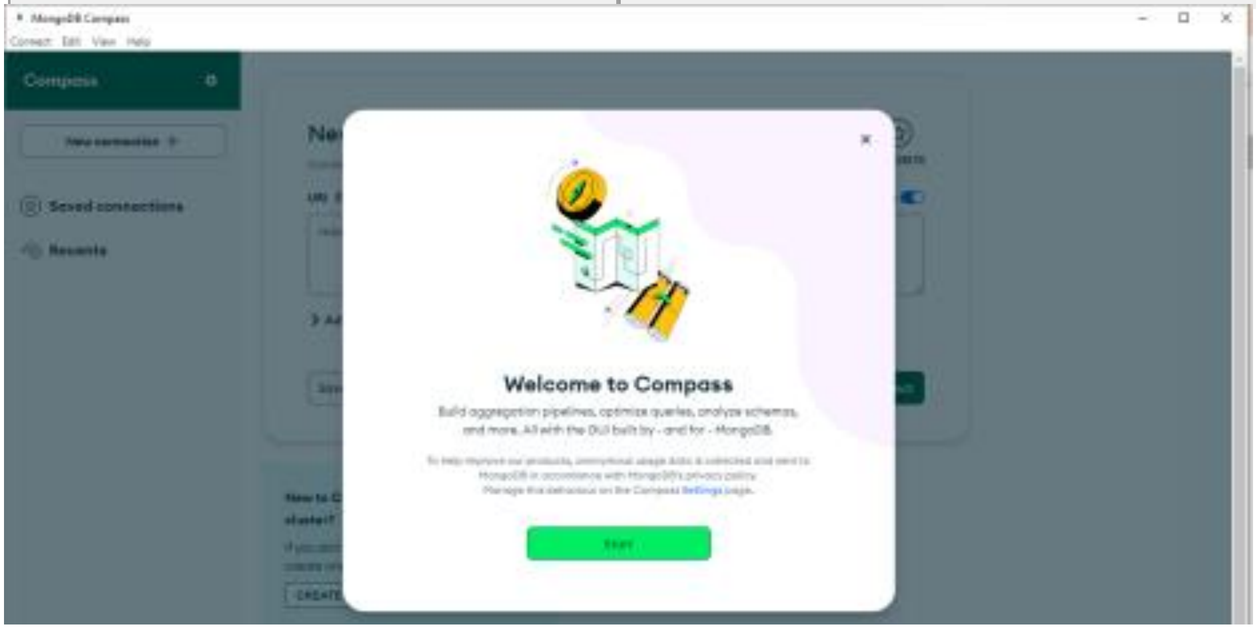


MongoDB 7.0.0 2008R2Plus SSL (64 bit) Setup

Installing MongoDB 7.0.0 2008R2Plus SSL (64 bit)

Please wait while the Setup Wizard installs MongoDB 7.0.0 2008R2Plus SSL (64 bit).

Status: Updating component registration



Welcome to Compass

Build aggregation pipelines, optimize queries, analyze schemas, and more. All with the GUI built by - and for - MongoDB.

After completing the installation process, you will find MongoDB software in your C drive. To view it, go to **C:Program FilesMongoDBServer{version}bin.**

However, you would notice the presence of any executable alongside mongo – **mongod**. Mongod is a daemon process that runs in the background. It handles database processes like accessing, retrieval, and updates.

If you try using MongoDB right away, you might have to specify a directory structure every time you need to use it. To avoid this you need to specify an environment variable for MongoDB, which leads us to step 3.

Step 3: Specify An Environment Variable

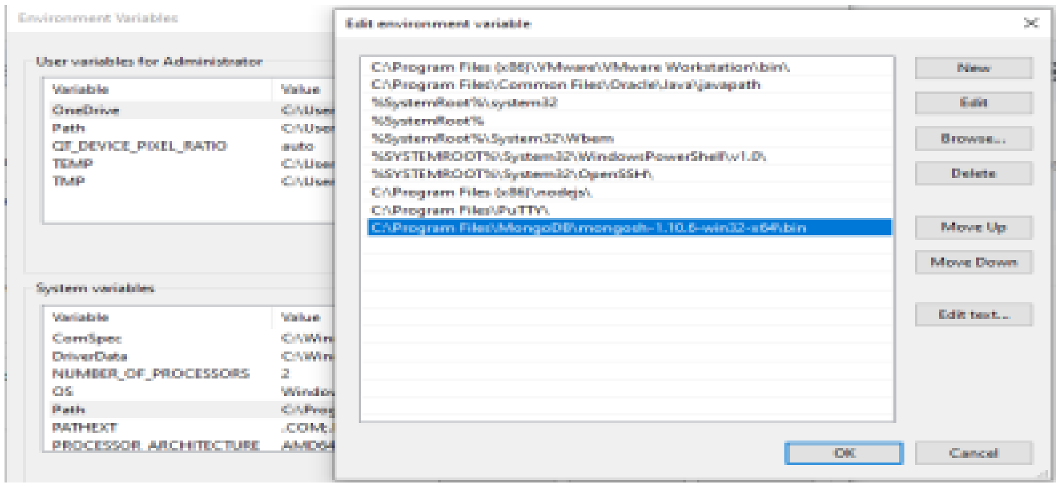
• Go to your system settings or type "Advanced system settings" in the search bar. • Click 'Environment Variables' under 'Advanced'.

• Click 'New' to create an environment variable.

• Select the 'Path' variable and click on 'Edit'.

• Change the environment variable to C:Program FilesMongoDBServer{version}bin. Where version is the MongoDB version you downloaded.

• Click Save.



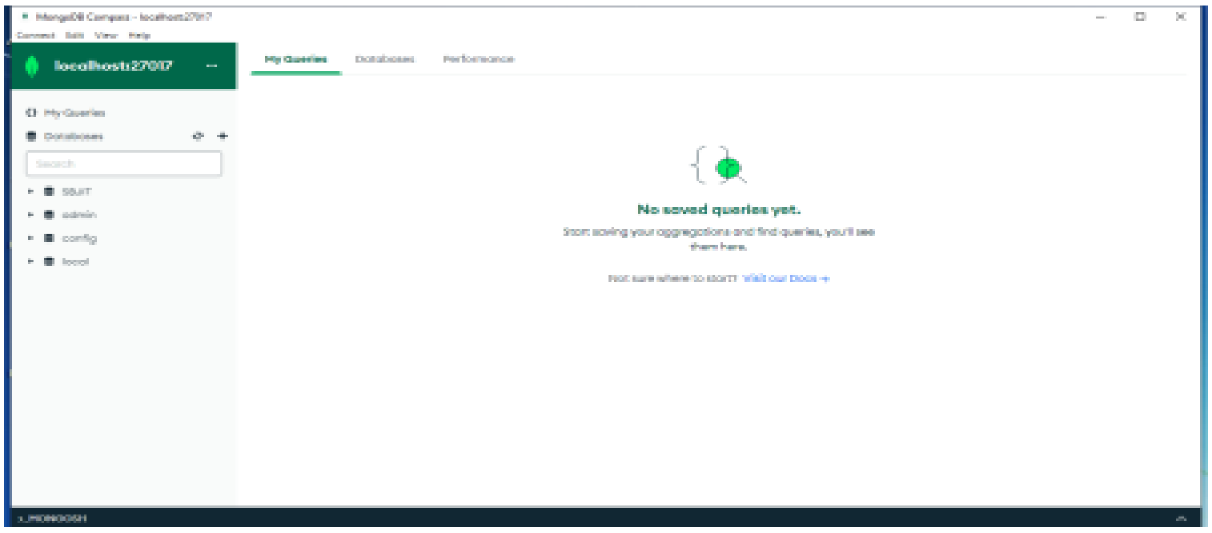
Variable	Value
OneDrive	C:\User
Path	C:\User
QT_DEVICE_PIXEL_RATIO	auto
TEMP	C:\User
TMP	C:\User

Variable	Value
ComSpec	C:\Win
DriverData	C:\Win
NUMBER_OF_PROCESSORS	2
OS	Windo
Path	C:\Prog
PATHEXT	.COM;
PROCESSOR_ARCHITECTURE	AMD64

```
C:\Program Files (x86)\VMware\VMware Workstation\bin\
C:\Program Files\Common Files\Oracle\Java\javapath
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\Program Files (x86)\nodejs\
C:\Program Files\PuTTY\
C:\Program Files\MongoDB\mongosh-1.10.6-win32-x64\bin
```

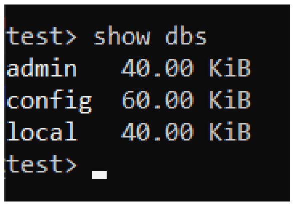
Step 4: Test Your Installation

To confirm your installation, run **'mongo'** in the terminal. You should see a command shell similar to the command prompt that allows you to enter commands. Type '**show dbs**' to see the list of the existing databases.


MongoDB Compass - localhost:27017
Connect Edit View Help
localhost:27017
My Queries
Databases
Performance
My Queries
Databases
Search
SBJIT
admin
config
local
No saved queries yet.
Start saving your aggregations and find queries, you'll see them here.
Not sure where to start? Visit our Docs →
>_MONGOSH

INPUT / OUTPUT (SCREENSHOTS):

1)If you want to check your databases list, use the command **show dbs.**



```
test> show dbs
admin   40.00 KiB
config  60.00 KiB
local   40.00 KiB
test> _
```

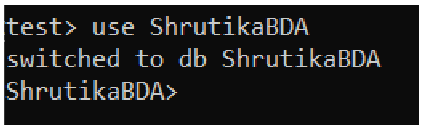
2)The use Command

MongoDB use DATABASE_NAME is used to create database. The command will create a new database if it doesn't exist, otherwise it will return the existing database.

Syntax

Basic syntax of use DATABASE statement is as follows with following

screenshot use DATABASE_NAME



```
test> use ShrutikaBDA
switched to db ShrutikaBDA
ShrutikaBDA>
```

In MongoDB default database is test. If you didn't create any database, then collections will be stored in test database.

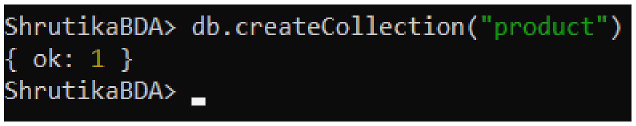
3)The createCollection() Method
MongoDB db.createCollection(name, options) is used to create collection.

Syntax

Basic syntax of createCollection() command is as follows −

db.createCollection(name, options)

In the command, **name** is name of collection to be created. **Options** is a document and is used to specify configuration of collection.

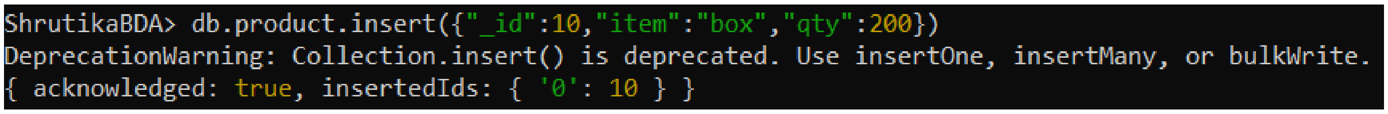


```
ShrutikaBDA> db.createCollection("product")
{ ok: 1 }
ShrutikaBDA> _
```

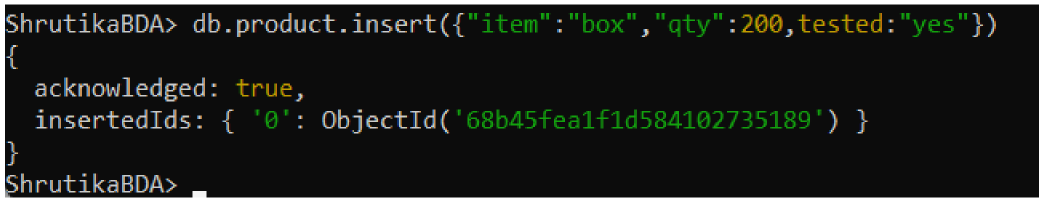
4)The insert() Method
To insert data into MongoDB collection, you need to use MongoDB's insert() method.

In the inserted document, if we don't specify the _id parameter, then MongoDB assigns a unique ObjectId for this document.

_id is 12 bytes hexadecimal number unique for every document in a collection



```
ShrutikaBDA> db.product.insert({"_id":10,"item":"box","qty":200})
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{ acknowledged: true, insertedIds: { '0': 10 } }
```



```
ShrutikaBDA> db.product.insert({"item":"box","qty":200,tested:"yes"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('68b45fea1f1d584102735189') }
}
ShrutikaBDA> _
```

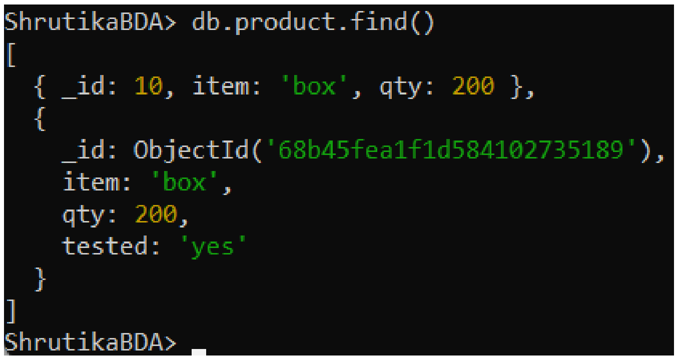
5) The find() Method

To query data from MongoDB collection, you need to use MongoDB's find() method. Syntax

The basic syntax of find() method is as follows –

>db.COLLECTION_NAME.find()

find() method will display all the documents in a non-structured way.



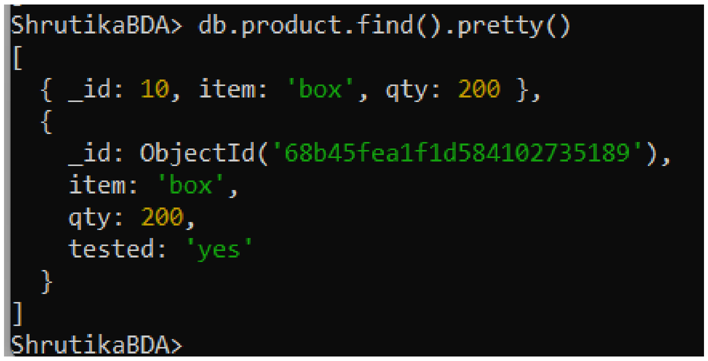
```
ShrutikaBDA> db.product.find()
[
  { _id: 10, item: 'box', qty: 200 },
  {
    _id: ObjectId('68b45fea1f1d584102735189'),
    item: 'box',
    qty: 200,
    tested: 'yes'
  }
]
ShrutikaBDA> _
```

6) The pretty() Method

To display the results in a formatted way, you can use pretty() method.

Syntax

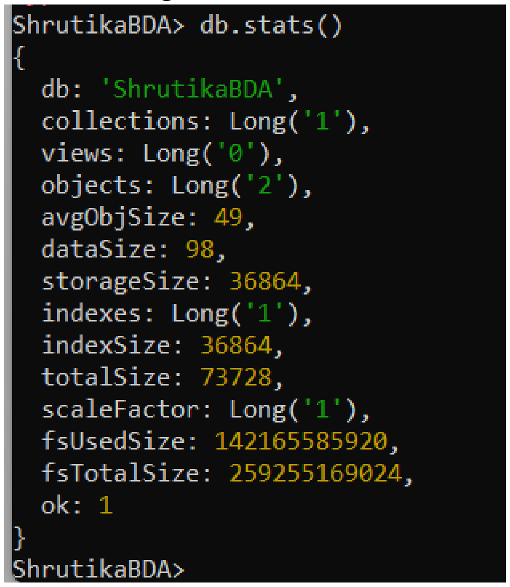
>db.COLLECTION_NAME.find().pretty()



```
ShrutikaBDA> db.product.find().pretty()
[
  { _id: 10, item: 'box', qty: 200 },
  {
    _id: ObjectId('68b45fea1f1d584102735189'),
    item: 'box',
    qty: 200,
    tested: 'yes'
  }
]
ShrutikaBDA>
```

7) MongoDB Statistics

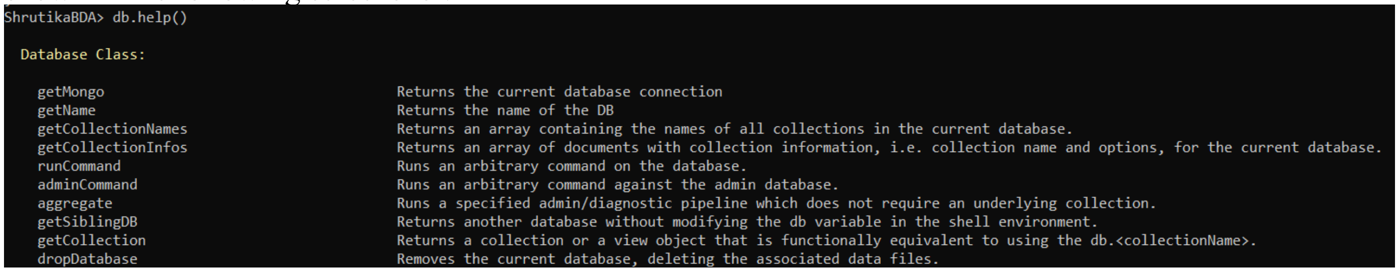
To get stats about MongoDB server, type the command db.stats() in MongoDB client. This will show the database name, number of collection and documents in the database. Output of the command is shown in the following screenshot.



```
ShrutikaBDA> db.stats()
{
  db: 'ShrutikaBDA',
  collections: Long('1'),
  views: Long('0'),
  objects: Long('2'),
  avgObjSize: 49,
  dataSize: 98,
  storageSize: 36864,
  indexes: Long('1'),
  indexSize: 36864,
  totalSize: 73728,
  scaleFactor: Long('1'),
  fsUsedSize: 142165585920,
  fsTotalSize: 259255169024,
  ok: 1
}
ShrutikaBDA>
```

8) MongoDB Help

To get a list of commands, type db.help() in MongoDB client. This will give you a list of commands as shown in the following screenshot.



```
ShrutikaBDA> db.help()

  Database Class:

    getMongo                                   Returns the current database connection
    getName                                    Returns the name of the DB
    getCollectionNames                         Returns an array containing the names of all collections in the current database.
    getCollectionInfos                         Returns an array of documents with collection information, i.e. collection name and options, for the current database.
    runCommand                                 Runs an arbitrary command on the database.
    adminCommand                               Runs an arbitrary command against the admin database.
    aggregate                                  Runs a specified admin/diagnostic pipeline which does not require an underlying collection.
    getSiblingDB                               Returns another database without modifying the db variable in the shell environment.
    getCollection                              Returns a collection or a view object that is functionally equivalent to using the db.<collectionName>.
    dropDatabase                               Removes the current database, deleting the associated data files.
```

CONCLUSION:

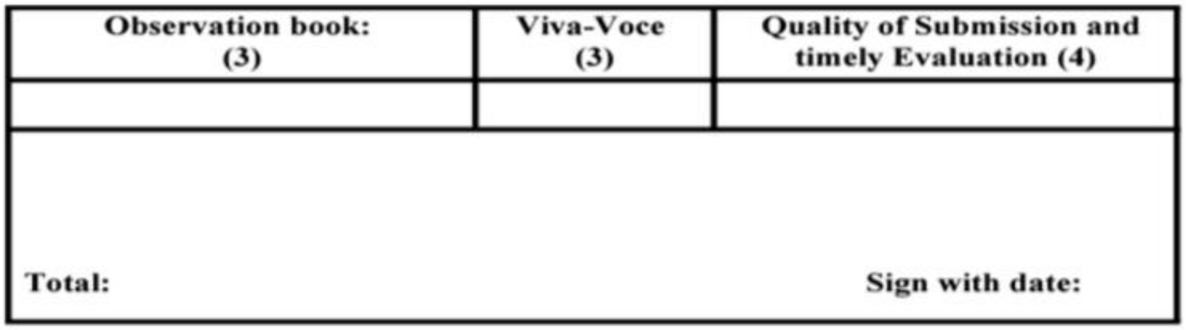
DISCUSSION AND VIVA VOCE:

- What is MongoDB, and how does it differ from traditional relational databases?
- Explain the concept of a document in MongoDB.
- What is BSON, and why is it used in MongoDB?
- Describe the primary data model used in MongoDB.

- What is a MongoDB collection, and how is it different from a MongoDB document?
- What is the primary key in MongoDB, and how does it work?

REFERENCE:

- https://www.scribd.com/presentation/590599731/MongoDB-Lab
- https://hevodata.com/learn/windows-mongodb-shell/#wm



Observation book: (3)	Viva-Voce (3)	Quality of Submission and timely Evaluation (4)
Total:		Sign with date: