**MongoDB**

MongoDB is a cross-platform, document oriented database that provides, high performance, high availability, and easy scalability. MongoDB works on concept of collection and document.

The following table shows the relationship of RDBMS terminology with MongoDB.

|  |  |
| --- | --- |
| **RDBMS** | **MongoDB** |
| Database | Database |
| Table | Collection |
| Tuple/Row | Document |
| column | Field |
| Table Join | Embedded Documents |
| Primary Key | Primary Key (Default key \_id provided by MongoDB itself) |
| **Database Server and Client** | |
| mysqld/Oracle | mongod |
| mysql/sqlplus | mongo |

**\_id** is a 12 bytes hexadecimal number which assures the uniqueness of every document. You can provide \_id while inserting the document. If you don’t provide then MongoDB provides a unique id for every document. These 12 bytes first 4 bytes for the current timestamp, next 3 bytes for machine id, next 2 bytes for process id of MongoDB server and remaining 3 bytes are simple incremental VALUE.

Any relational database has a typical schema design that shows number of tables and the relationship between these tables. While in MongoDB, there is no concept of relationship.

Advantages of MongoDB over RDBMS

* **Schema less** − MongoDB is a document database in which one collection holds different documents.
* Structure of a single object is clear.
* No complex joins.
* Deep query-ability. MongoDB supports dynamic queries on documents.
* Tuning.
* **Ease of scale-out** − MongoDB is easy to scale.
* Conversion/mapping of application objects to database objects not needed.
* Uses internal memory for storing the (windowed) working set, enabling faster access of data.
* **Document Oriented Storage** − Data is stored in the form of JSON style documents.

## Install MongoDB On Windows

**STEP 1:** To install MongoDB on Windows, first download the latest release of MongoDB from <https://www.mongodb.com/download-center>.

**STEP 2**: In the command prompt, navigate to the bin directory current in the MongoDB installation folder. Suppose my installation folder is **C:\Program Files\MongoDB**

C:\Users\XYZ>d:cd C:\Program Files\MongoDB\Server\4.2\bin

C:\Program Files\MongoDB\Server\4.2\bin>mongod.exe

**STEP 3:** Now to run the MongoDB, you need to open another command prompt and issue the following command.

C:\Program Files\MongoDB\Server\4.2\bin>mongo.exe

MongoDB shell version v4.2.1

connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb

Implicit session: session { "id" : UUID("4260beda-f662-4cbe-9bc7-5c1f2242663c") }

MongoDB server version: 4.2.1

>

**STEP 4:** This will show that MongoDB is installed and run successfully. Next time when you run MongoDB, you need to issue only commands.

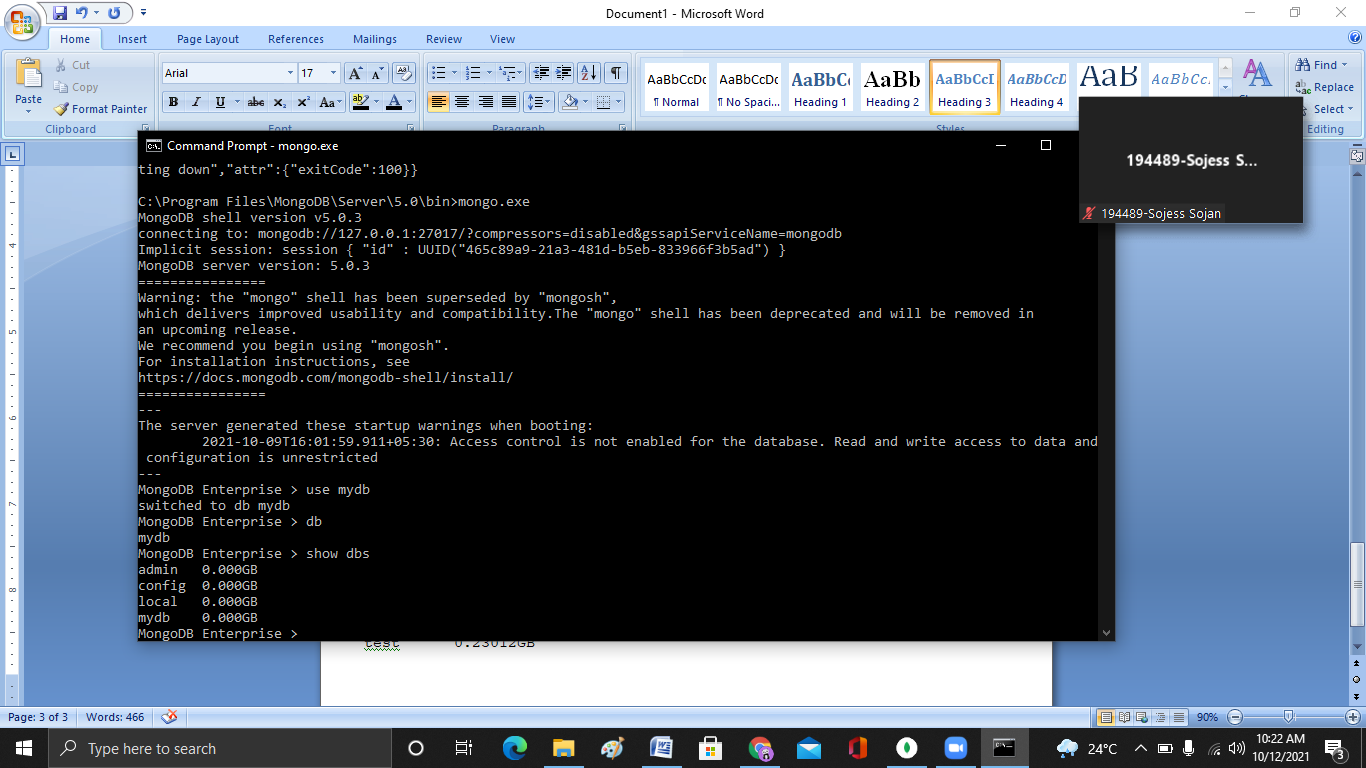
C:\Program Files\MongoDB\Server\4.2\bin>mongod.exe

C:\Program Files\MongoDB\Server\4.2\bin>mongo.exe

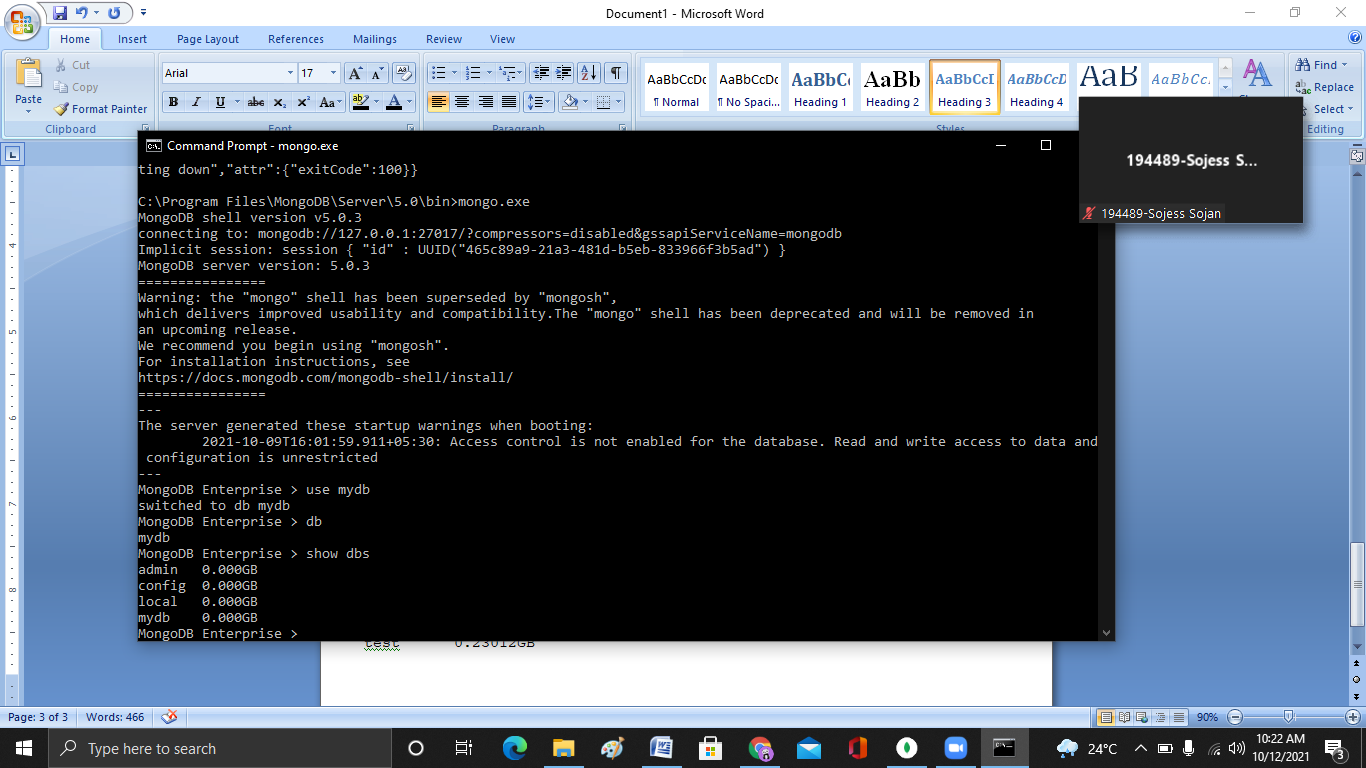
## The use Command

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.

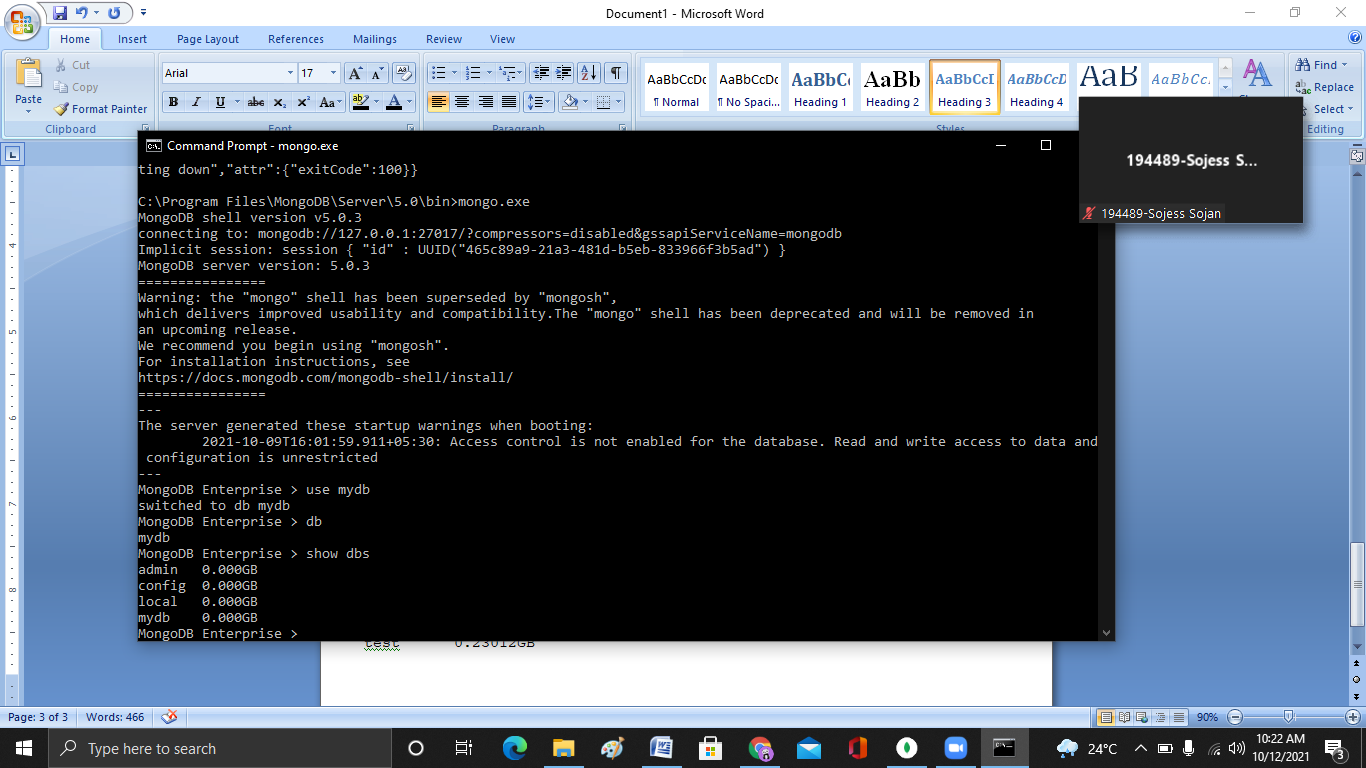
### Syntax: use DATABASE\_NAME



To check your currently selected database, use the command **db**



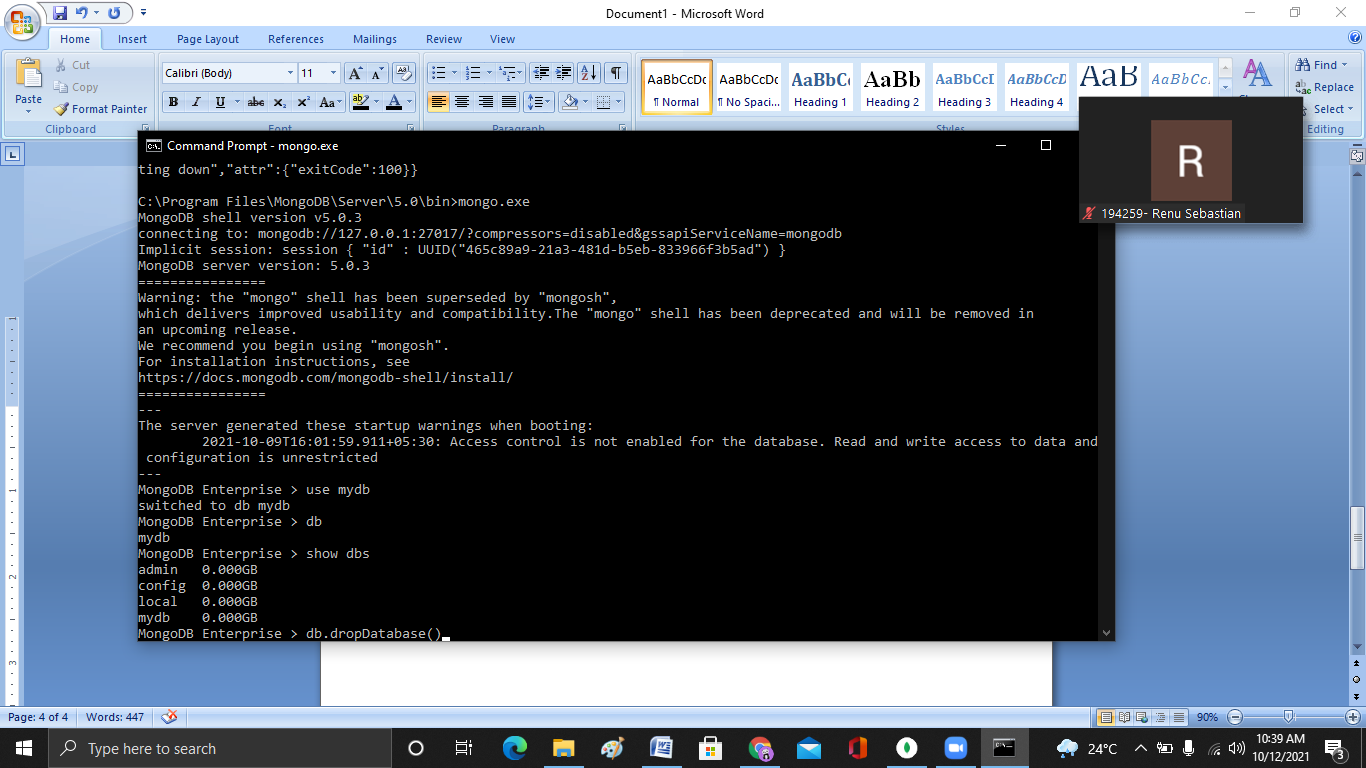
If you want to check your databases list, use the command **show dbs**.



## The dropDatabase() Method

MongoDB **db.dropDatabase()** command is used to drop a existing database.

### Syntax: db.dropDatabase()



MongoDB supports many datatypes. Some of them are −

* **String** − This is the most commonly used datatype to store the data. String in MongoDB must be UTF-8 valid.
* **Integer** − This type is used to store a numerical value. Integer can be 32 bit or 64 bit depending upon your server.
* **Boolean** − This type is used to store a boolean (true/ false) value.
* **Double** − This type is used to store floating point values.
* **Min/ Max keys** − This type is used to compare a value against the lowest and highest BSON elements.
* **Arrays** − This type is used to store arrays or list or multiple values into one key.
* **Timestamp** − ctimestamp. This can be handy for recording when a document has been modified or added.
* **Object** − This datatype is used for embedded documents.
* **Null** − This type is used to store a Null value.
* **Symbol** − This datatype is used identically to a string; however, it's generally reserved for languages that use a specific symbol type.
* **Date**− This datatype is used to store the current date or time in UNIX time format. You can specify your own date time by creating object of Date and passing day, month, year into it.
* **Object ID** − This datatype is used to store the document’s ID.
* **Binary data** − This datatype is used to store binary data.
* **Code** − This datatype is used to store JavaScript code into the document.
* **Regular expression** − This datatype is used to store regular expression.

## The insert() Method

To insert data into MongoDB collection, you need to use MongoDB's **insert()** or **save()** method.

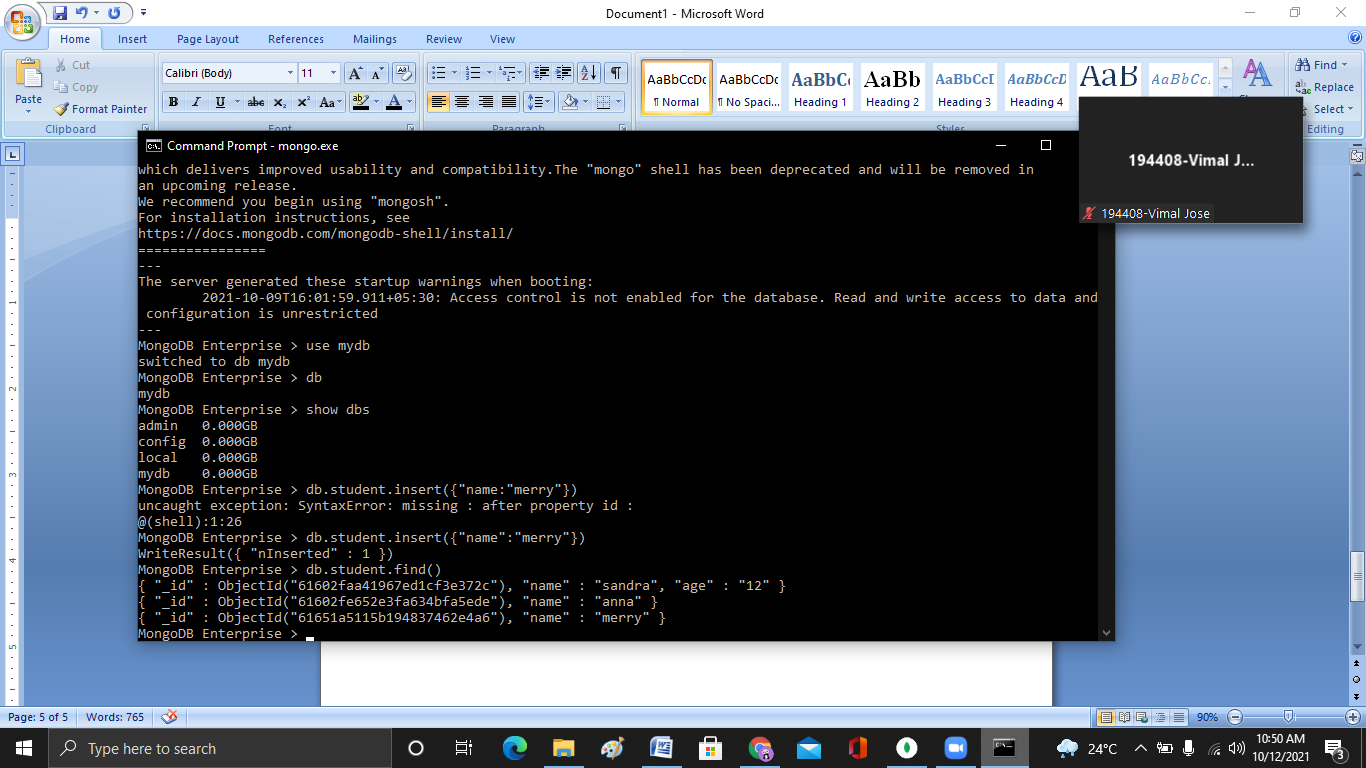
**Syntax: db.COLLECTION\_NAME.insert(document)**

### 

## The find() Method

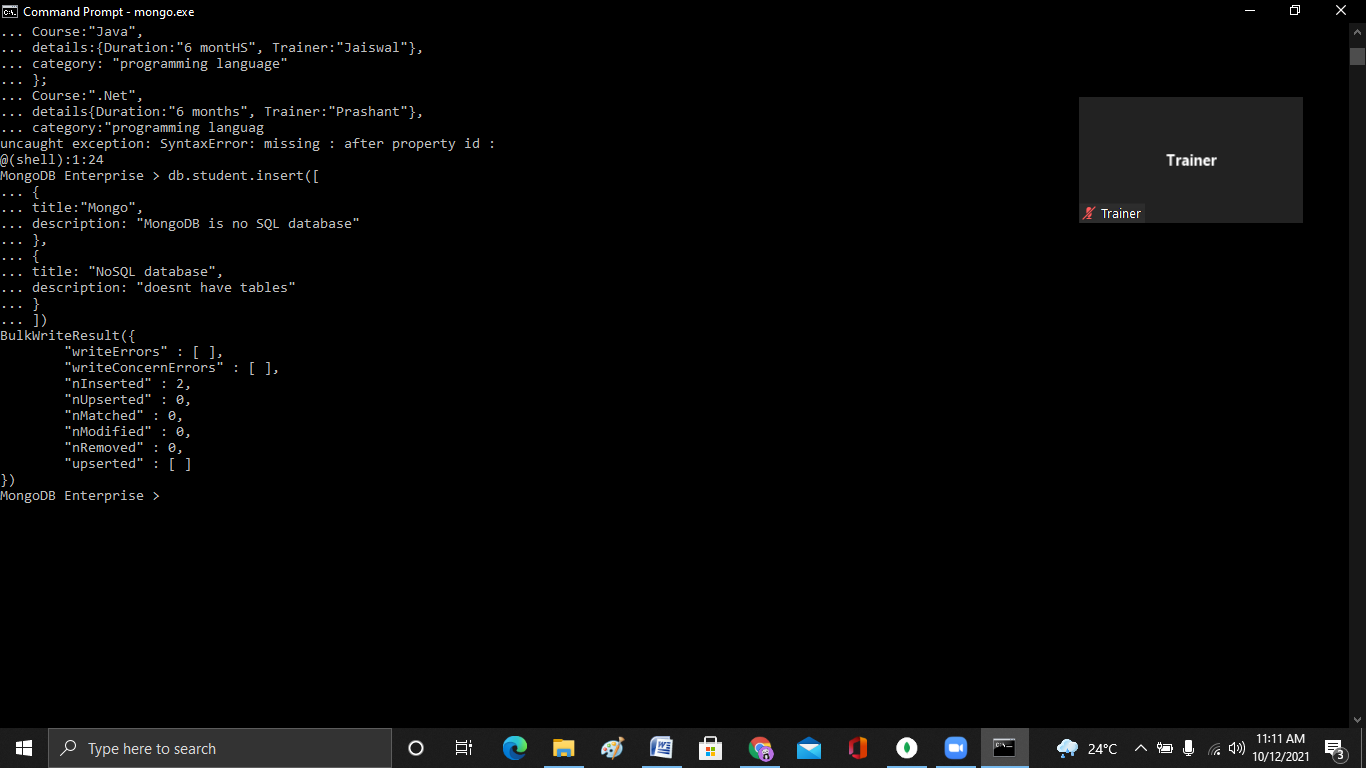
To query data from MongoDB collection, you need to use MongoDB's **find()** method. **find()** method will display all the documents in a non-structured way.

### Syntax: >db.COLLECTION\_NAME.find()



## MongoDB insert multiple documents

If you want to insert multiple documents in a collection, you have to pass an array of documents to the db.collection.insert() method.



## The aggregate() Method

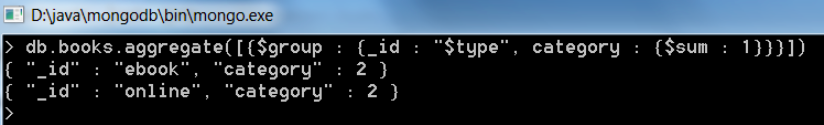
For the aggregation in MongoDB, you should use **aggregate()** method.

Aggregations operations process data records and return computed results. Aggregation operations group values from multiple documents together, and can perform a variety of operations on the grouped data to return a single result. In SQL count(\*) and with group by is an equivalent of MongoDB aggregation.

### Syntax

>db.COLLECTION\_NAME.aggregate(AGGREGATE\_OPERATION)

db.books.aggregate([{$group : {\_id: "$type", category: {$sum : 1}}}])



|  |  |
| --- | --- |
| **Expression** | **Description** |
| $sum | Summates the defined values from all the documents in a collection |
| $avg | Calculates the average values from all the documents in a collection |
| $min | Return the minimum of all values of documents in a collection |
| $max | Return the maximum of all values of documents in a collection |
| $addToSet | Inserts values to an array but no duplicates in the resulting document |
| $push | Inserts values to an array in the resulting document |
| $first | Returns the first document from the source document |
| $last | Returns the last document from the source document |