**National University of Ho Chi Minh City UNIVERSITY OF INFORMATION TECHNOLOGY**



**FINAL PROJECT REPORT**

**TOPIC: MONGO DB**

**Student: Nguyen Thi Cam Van**

ID: 20522145

Lecturer: Nguyễn Thanh Bình

Trương Thu Thủy

Class: IS201.M21.HTCL

**Ho Chi Minh city, June 2022**

MONGODB

[I INTRODUCTION MONGODB 5](#_Toc106636697)

[**1.** **What is MongoDB?** 5](#_Toc106636698)

[**2.** **Pros and cons of MongoDB** 8](#_Toc106636699)

[**3.** **Data Model** 10](#_Toc106636700)

[**3.1 Embedded data model:** 10](#_Toc106636701)

[**3.2 Normalized Data Model** 10](#_Toc106636702)

[**4.** **Data types in MongoDB** 11](#_Toc106636703)

[II OPERATIONS IN MONGODB 12](#_Toc106636704)

[**1.** **Create Database in MongoDB** 12](#_Toc106636705)

[**1.1** **Command use** 12](#_Toc106636706)

[**1.2** **View the database in use** 13](#_Toc106636707)

[**1.3** **View all databases in the system** 13](#_Toc106636708)

[**2.** **Delete Database in Mongo** 14](#_Toc106636709)

[**3.** **Create Collection in MongoDB** 14](#_Toc106636710)

[**4.** **Delete Collection in MongoDB** 15](#_Toc106636711)

[**5.** **Insert Document in MongoDB** 15](#_Toc106636712)

[**5.1 Method Insert** 16](#_Toc106636713)

[**5.2 Method insertOne()** 18](#_Toc106636714)

[**5.3 Method insertMany()** 18](#_Toc106636715)

[**6.** **Query Document** 19](#_Toc106636716)

[**6.1 Method Find()** 19](#_Toc106636717)

[**6.2 Method Pretty()** 20](#_Toc106636718)

[**6.3 Method FindOne()** 20](#_Toc106636719)

[**6.4 AND in MongoDB** 21](#_Toc106636720)

[**6.5 OR in MongoDB** 21](#_Toc106636721)

[**6.6 NOR in MongoDB** 21](#_Toc106636722)

[**7.** **Update Document** 21](#_Toc106636723)

[**7.1 Method Update** 21](#_Toc106636724)

[**7.2 Method Save** 22](#_Toc106636725)

[**7.3 FindOneAndUpdate** 22](#_Toc106636726)

[**7.4 UpdateOne** 22](#_Toc106636727)

[**7.5 UpdateMany** 22](#_Toc106636728)

[**8.** **Delete Document** 22](#_Toc106636729)

[III MONGODB ATLAS 23](#_Toc106636730)

# **I INTRODUCTION MONGODB**

1. **What is MongoDB?**

**MongoDb** **is a** **cross-platform database**, working on **Collection** and **Document** concepts, it provides high performance, high availability, easy scalability.

MongoDB will avoid the table-based structure of a relational database to accommodate documents like JSON that have a very flexible schema called BSON.

“*BSON is the binary encoding of JSON-like documents that MongoDB uses when storing documents in collections. It adds support for data types like Date and binary that are not supported in JSON*.”

***MongoDB uses data storage in the form of Document JSON***, so each collection will have different sizes and documents. The data is stored in a JSON document so the query will be very fast.

MongoDB was first born by MongoDB Inc., at 10th generation, in October 2007, it is part of a PaaS (Platform as a Service) product similar to **Windows Azure and Google App Engine**. It was later made open source in 2009.

As one of the most prominent NoSQL databases, used as the back-end for many websites such as eBay, SourceForge and The New York Times.

|  |  |
| --- | --- |
| RDBMS | MongoDB |
| Database | Database |
| Table | Collection |
| Row | Document |
| Column | Field |
| Table Join | Embedded Documents |
| Primary Key | Primary Key (Mặc định là \_ID) |

* **Collection:** Is a group of Documents in MongoDB. It is equivalent to a table in an RDBMS. Thus, a Collection exists inside a single database. Collections are not constrained Relationship like different admin database, very fast access, so each Collection can contain many different types unlike tables in mysql system admin are fixed fields. Documents inside a Collection can have many different fields. In particular, all Documents in a Collection are similar or with a related purpose.
* **Document:** Document is a document stored in the form of JSON. It is equivalent to the row in the table of the RDBMS but the data in the Document is quite flexible, it is not fixed how many fields, the data in each field must be fixed.
* **Embedded Documents**: If in RDBMS, we have a type of joining tables, for example the person table joins with the address table to know what addresses a person has, then in MongoDB, the person's document will contain an array of document addresses.
* **Primary Key:** The default Primary key in MongoDB is the \_ID field, when inserting if you do not specify the \_ID field, it will automatically generate the value for \_ID.
* **Simple structure of Document**

Text

Description automatically generated

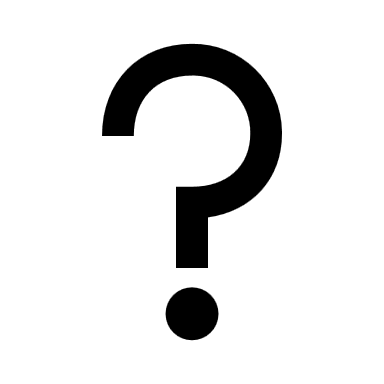
Here**, \_id is a 12byte hexadecimal number to ensure the uniqueness of each Document.** You can provide \_id while inserting into Document. If you do not, then MongoDB will provide a unique id for each Document.

Of these 12 bytes, **the first 4 bytes are for the current Timestamp, the next 3 bytes for the device ID, the next 2 bytes are the process id of the MongoDB server, and the remaining 3 bytes are the incrementable value.**

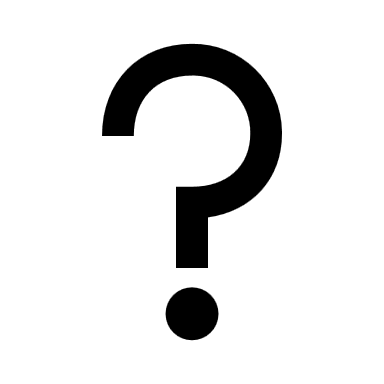
* **The features of MongoDB include:**
* **Ad hoc queries:** Support search by field, normal searches, regular expression searches, range queries.
* **Indexing:** Any field in the BSON document can be indexed.
* **Replication:** There is a version identical to the existing version, in use. With databases, large storage needs, integrity database requirements, no loss before unexpected incidents are very high. So people came up with the concept of "cloning", creating a database version that is identical to the existing one, and storing it somewhere else, in case something goes wrong.
* **Aggregation:** Aggregation operations process data records and return calculated results. Aggregation operations group values ​​from multiple Documents together, and a variety of operations can be performed on the grouped data to return a single result.
* **In SQL**, count(\*) and GROUP BY are equivalent to Aggregation in MongoDB.
* **File storage:** MongoDB is used as a file system that takes advantage of the above functions and acts as a way of distribution over sharding.

1. **Pros and cons of MongoDB**

* **Advantages of MongoDB**
* **Open Source:** MongoDB is free and open source software with a huge developer community.
* **Hiệu năng cao:**
* The query speed (Find, Update, Insert, Delete) of MongoDB is much faster than that of relational database management systems (RDBMS)
* Tests show that the Insert speed of MongoDB can be up to 100 times faster than MySQL.

**So why does Mongo have such high performance?**

* MongoDB stores data as JSON, when you Insert many objects, it will be Insert a JSON array almost like the case of Insert 1 object.
* Data in MongoDB is not bound together like in RDBMS, when Insert, Delete or Update does not need to spend time checking whether related tables are satisfied like in RDBMS.
* Data in MongoDB is indexed, when querying it will be found very quickly.
* When performing Insert, Find....Mongo will lock the operations, for example when it performs Find(), during the Find process that adds Insert and Update operations, it will stop all to wait for Find () accomplished.
* **Flexible data**
* MongoDB is a document database, saves data in the form of Json, is not constrained about the quantity field, data type .... you can insert freely the data you want.
* **Availability**
* MongoDBsupport replication has been set up to ensure data backup and restore.
* **Rich** **query language:**
* MongoDB is a rich Query language i.e. it has built-in methods to perform Create, Read, Update, Delete data.
* **Horizontal scalability of the function:**
* In MongoDB there is a conceptual cluster that is a cluster of nodes containing data communication with each other, when we want to expand the system, we just need to add a new node to the cluster.
* **Disadvantages of MongoDB**
* MongoDB does not have binding properties like in RDBMS -> Easy to falsify data.
* Does not support join like RDBMS, so when writing the join function in the code, we have to do it manually, causing the query speed to be reduced.
* Use a lot of memory: Since the data is stored as Key - Value, the collections only differ in value so the Key will be repeated. Join is not supported, so there will be data redundancy (in RDBMS, we only need to save one record and then the other records refer to it, but not in MongoDB)

**Why use MongoDB**

* Document storage-oriented: Data is stored in document-style JSON
* Replication and high performance
* Query variety
* Update faster
* Professional support by MongoDB

1. **Data Model**

**MongoDB provides two model data types: Embedded Data Model and Normalized Data Model**. Depending on the case, you can use the appropriate model during document preparation

### **3.1 Embedded data model:**

In this model you can have (embed) all related data into a single document, it is also known as normalized data model.

For the example below, suppose we get employee details in 3 different documents, namely: Personal\_details, Contact, Address.

We can completely embed 3 documents as follows:A picture containing table

Description automatically generated

### **3.2 Normalized Data Model**

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text

Description automatically generated with medium confidence

1. **Data types in MongoDB**

**MongoDB supports many data types:**

* **String:** This is the most used data type to store data. Strings in MongoDB must be valid UTF-8.
* **Integer:** This data type is used to store a numeric value. Integer can be 32 bit or 64 bit depending on your server.
* **Boolean:** This data type is used to store a Boolean value (True/False).
* **Double:** This data type is used to store floating point numeric values.
* **Min/Max keys:** This data type is used to compare a value with the lowest and highest BSON elements.
* **Array**: This data type is used to store arrays or lists or multiple values ​​into a key.
* **Timestamp:** Convenient for recording or marking the time a Document was modified or added.
* **Object:** This data type is used for embedded documents.
* **Null:** This data type is used to store a Null value.
* **Symbol**: This data type is used like a string.
* **Date:** This data type is used to store the current date and time in UNIX time format. You can define your own time by creating a Date object and passing the day, month, and year into it.
* **Object ID:** This data type is used to store the Document's ID.
* **Binary data:** This data type is used to store binary data.
* **Code:** This data type is used to store binary data.
* **Regular expression**: This data type is used to store Regular Expresion.

# **II OPERATIONS IN MONGODB**

1. **Create Database in MongoDB**
   1. **Command use**

The command use **DATABASE\_NAME in MongoDB** is used to create the database. This command will create a new database if it does not already exist, otherwise this command will return the existing database.

**Syntax: use DATABASE\_NAME**

*Create a new database named Camvan, use the use DATABASE command as follows: use Camvan* Rectangle

Description automatically generated with medium confidence

*To check the currently selected database, we use the db command.A picture containing shape

Description automatically generated*

*If you want to check the list of databases, use the show dbs command.A picture containing shape

Description automatically generated*

**In MongoDB, the default database is Test. If no database is created then the Collections will be kept in Test.**

* 1. **View the database in use**

**To see the database in use (current database), we use the command: db**A picture containing shape

Description automatically generated

* 1. **View all databases in the system**

**To see all the databases created on MongoDB, use the command : show dbs**

Shape, rectangle

Description automatically generated with medium confidence

This command will only show databases that already have at least one collection, if there is no collection, it won't show.

1. **Delete Database in Mongo**

**To delete a database in MongoDB need to use the following syntax: Db.dropDatabase()**

This command is used to delete the current database, so when using it, you must switch to the database to be deleted to succeed.

1. **Create Collection in MongoDB**

**The db.createCollection() method in MongoDB is as follows: db.createCollection(name, options)**

* **List of available options:**

|  |  |  |
| --- | --- | --- |
|  | Type | Description |
| capped | **Boolean** | Nếu true, kích hoạt một capped collection.  Đây là một collection có kích cỡ cố định mà tự động ghi đề các entry cũ nhất khi nó tiếp cận kích cỡ tối đa.  Nếu bạn xác định là true, thì bạn cũng cần xác định tham số size. |
| autoIndexID | **Boolean** | Nếu true, tự động tạo chỉ mục trên các trường \_id.  Giá trị mặc định là false |
| size | **Số** | Xác định kích cỡ tối đa (giá trị byte) cho một Capped Collection.  Nếu tham số Capped là true, thì bạn cũng cần xác định trường này. |
| max | **Số** | Xác định Document tối đa được cho phép trong một Capped Collection. |

While performing insert data into the Document, MongoDB first checks the size field of the Cappped Collection, then it checks the max field.

1. **Delete Collection in MongoDB**

The db.collection.drop() method in MongoDB is used to delete a Collection from the database.

**Syntax: db.COLLECTION\_NAME.drop()**

*Check the available collections inside the Camvan database.A picture containing graphical user interface

Description automatically generated*

*Delete Collection with the name CamVan as follows:*

A picture containing shape

Description automatically generated

The drop() method will return true, if the selected Collection is successfully deleted, otherwise it will return false

1. **Insert Document in MongoDB**

### **5.1 Method Insert**

To insert data into a Collection in MongoDB, you need to use the Insert() method.

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

Text

Description automatically generated

Camvan is the name of the Collection, which was created in the previous chapter. If this collection does not already exist in the database, then MongoDB will create this Collection and insert the Document into it.

In the inserted Document, if we do not specify the \_id parameter, then MongoDB will assign a unique Objectld to this Document.

\_id is a unique, 12-byte hexadecimal number for each Document in a Collection.

To insert multiple Documents in a single query, you can pass an array of Documents in the Insert() command.





To insert data into a Document, you can also use db.Camvan.save(Document).

If you do not specify \_id in the Document, then the save() method will work the same as the insert() method. If you specify \_id, it will replace all Document data without \_id as defined in save() method.

### **5.2 Method insertOne()**

If you only need to insert one library into the collection, you can use this method: **db.COLLECTION\_NAME.insertOne(Document)**

Text

Description automatically generated

### **5.3 Method insertMany()**

You can insert multiple libraries using the insertMany() method.

To use this library, you need to include the array in the library

Text

Description automatically generated with medium confidence

1. **Query Document**

**6.1 Method Find()**

To query data from Collection in MongoDB, we need to use Find() method.

**Syntax: db.COLLECTION\_NAME.find()**

Find() method will display all Document in unstructured form.

*Suppose we have created a Collection with the name Camvan*Text

Description automatically generated

*Insert 3 documents using Insert() method*

Text

Description automatically generated

Retrieve all documents in Document

### **6.2 Method Pretty()**

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

**Syntax: db.Camvan.find().pretty()**

### **6.3 Method FindOne()**

As part of the Find() method, which is the FindOne() method, it only returns 1 Document.

**Syntax: db.COLLECTION\_NAME.findOne()Text

Description automatically generated with medium confidence**

### **6.4 AND in MongoDB**

In the find() method, if you pass multiple keys by separating them by commas (,), then MongoDB treats it as an AND condition.

**Syntax: db.COLLECTION\_NAME.find({key1: value1, key2: value2}).pretty()**

### **6.5 OR in MongoDB**

To query a Document based on an OR condition, you need to use $or.

**Syntax: db.COLLECTION\_NAME.find({$or:[{key1: value1}, {key2 : value2}]}.pretty()**

### **6.6 NOR in MongoDB**

To query the document based on the NOT condition, you need to use $not.

**Syntax: db.COLLECTION\_NAME.find($not: [ { key1: value1 }, { key2: value2 }])**

1. **Update Document**

### **7.1 Method Update**

* The update() method updates the values in the existing Document.

**Syntax: db.COLLECTION\_NAME.update(SELECTION\_CRITERIA, UPDATE\_DATA)**

### **7.2 Method Save**

* The Save() method replaces the existing Document with the new Document passed in this save() method.

**Syntax: db.COLLECTION\_NAME.save({ \_id: ObjectID(), NEW\_DATA })**

### **7.3 FindOneAndUpdate**

* The FindOneAndUpdate() method updates the values that exist in the Documents.

**Syntax: db.COLLECTION\_NAME.findOneAndUpdate(SELECTION\_CRITERIA, UPDATED\_DATA)**

### **7.4 UpdateOne**

* This method updates a Document that matches the filtered value.

**Syntax: db.COLLECTION\_NAME.updateOne( <filter>, <update> )**

### **7.5 UpdateMany**

* This method updates a Document that matches the filtered value.

**Syntax: db.COLLECTION\_NAME.updateOne( <filter>, <update> )**

1. **Delete Document**

Remove() method in MongoDB is used to remove Document from Collection.

**Syntax: db.COLLECTION\_NAME.remove(DELLETION\_CRITTERIA, JUSTONE)**

- Delete only one Document

If there are multiple records and you only want to delete the first one, set the JustOne parameter in the Remove() method.

**Syntax: db.COLLECTION\_NAME.remove(DELETION\_CRITERIA, 1)**

- Delete all Document

If you do not specify deletion criteria, then MongoDB will delete all documents from the Collection.

**Syntax: db.COLLECTION.remove()**

# **III MONGODB ATLAS**

* **What is Atlas?**

**Is a mongo-managed cloud database built for applications use mongo batabase**

**Atlas has many different versions for different users.**

* To create and deploy a Mongo atlas cluster . First open a web browser go to mongodb.com

**B1: Create and name an Organization**

**B2: Click new project to create a project**

- Name the project to create and click next

- Continue to click create project

**B3: Click build a Database**

Here there are 3 cloud server regions for us to choose, we choose 1 out of 3

Click on the cluster to rename the cluster

Click on create cluster

**B4: Now we can connect the database**

To configure the connection, we click connect. To ensure a secure connection, mongo requires the IP address of the machine to connect to.

Click Add Your Current IP Address to add the current machine IP address

If you want to be able to connect from anywhere, click allow access form anywhere

We create user adim to manage our database.

Mongo supports us with three methods of connecting to the database.

Connect through mongodb shell your applications.

Connect via uqa mongodb compass let's say I choose to connect via mongofb shell choose the mongodb shell version you have copy this connection string to mongodb shell we can connect to this mongodb database

Link drive: https://drive.google.com/drive/folders/1nILU-ByCQ8eVTbfJgeIyx-YHKRd43xx-?usp=sharing