

# HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU



Đỗ Thanh Nghị  
dtngghi@cit.ctu.edu.vn

10-2016

# Nội dung

2

- Giới thiệu
- Quản trị cơ bản CSDL
- Tạo, đọc, cập nhật, xóa (CRUD)
- MongoDB – PHP
- MongoDB – Java
- MongoDB – Python

# Giới thiệu

3

- MongoDB = Hum**ongous** **DB**: huge, monstrous (data)
- Nguồn mở với giấy phép  
MongoDB server và tools: GNU AGPL v3.0  
Trình điều khiển (drivers): Apache License v2.0  
Tài liệu: Creative Commons
- Hiệu năng cao, tính sẵn dùng cao, dễ dàng co giãn
- Ngôn ngữ truy vấn mềm dẻo
- Nền: Redhat, CentOS, Fedora, Debian, Ubuntu, Linux khác, Unix, OS X, Windows
- Trình điều khiển: C/C++, Java, Javascript, .NET, Perl, PHP, Python, Ruby, Scala

# Giới thiệu

4

- Hướng tài liệu
- Tài liệu được lưu theo dạng BSON (Binary-encoded serialization of JSON-like), gồm các cặp trường-giá trị
- Bộ sưu tập (collection)
  - Tương tự như bảng trong CSDL quan hệ
  - Có tập chỉ mục chung
  - Tập các tài liệu
  - Các tài liệu có thể có cấu trúc không đồng nhất
- Cơ sở dữ liệu
  - Chứa tập các bộ sưu tập

# Giới thiệu

5

- Ví dụ tài liệu:

```
{
  _id: ObjectId('5816bed4a2b7a9f009f2f2bb')
  title: 'MongoDB Overview',
  by: 'John',
  likes: 100,
  comments: [
    {
      user: 'user1',
      message: 'My first comment',
      like: 0
    },
    {
      user: 'user2',
      message: 'My second comments',
      like: 5
    }
  ]
}
```

# Giới thiệu

6

<b>RDBMS</b>	<b>MongoDB</b>
Database	Database
Table	Collection
Tuple/Row	Document
column	Field
Table Join	Embedded Documents
Primary Key	Primary Key (Default key <code>_id</code> provided by mongodb itself)
<b>Database Server and Client</b>	
Mysqld/Oracle	mongod
mysql/sqlplus	mongo

# Nội dung

7

- Giới thiệu
- **Quản trị cơ bản CSDL**
- Tạo, đọc, cập nhật, xóa (CRUD)
- MongoDB – PHP
- MongoDB – Java
- MongoDB – Python

# Môi trường MongoDB

8

- MongoDB server

```
sudo service mongod [start|stop|restart]
```

- MongoDB client (shell)

```
mongo --username <user> --password <pass> --  
host <host> --port <port> --  
authenticationDatabase <admin>
```

```
mongo -u <user> -p <pass> --host <host> --  
port <port>
```



# Môi trường MongoDB

9

- MongoDB client (shell)

```
mongo script-file.js -u <user> -p
```

```
mongo --eval '<javascript>'
```

```
mongo (mặc định là localhost, cổng 27017)
```

dấu nhắc lệnh là:

```
>
```

# Cơ sở dữ liệu (CSDL)

10

- Tạo CSDL  
**use** <DBNAME>
- Ví dụ  
>**use** mydb  
switched to db mydb
- CSDL hiện hành  
>**db**  
mydb

# Cơ sở dữ liệu

11

- Hiển thị danh sách các CSDL

**>show dbs**

local            0.78125GB

test            0.23012GB

# Cơ sở dữ liệu

12

- Xóa CSDL

**db.dropDatabase()**

- Ví dụ

**>show dbs**

local            0.78125GB

mydb            0.23012GB

test            0.23012GB

**>use mydb**

switched to db mydb

**>db.dropDatabase()**

**>{ "dropped" : "mydb", "ok" : 1 }**

# Quản trị người dùng

13

- Tạo người dùng **root** có quyền **root**

```
>use admin
```

```
>db.createUser(
```

```
{
```

```
  user: "root",
```

```
  pwd: "passwd",
```

```
  roles: [ "root" ]
```

```
}
```

```
)
```

- Cần soạn thảo lại tập tin /etc/mongod.conf

```
security:
```

```
  authorization: enabled
```

# Quản trị người dùng

14

- Tạo người dùng **user1** có quyền **đọc/ghi** trên CSDL **mydb**

```
>db.createUser(  
{  
  user: "user1",  
  pwd: "xxx",  
  roles: [{ role: "readWrite", db: "mydb" }]  
}  
)
```

# Nội dung

15

- Giới thiệu
- Quản trị cơ bản CSDL
- **Tạo, đọc, cập nhật, xóa (CRUD)**
- MongoDB – PHP
- MongoDB – Java
- MongoDB – Python

# Tạo bộ sưu tập (collection)

16

- Cú pháp tạo bộ sưu tập  
**db.createCollection(name, options)**
- Ví dụ  
    **>use mydb**  
switched to db mydb  
    **>db.createCollection("mycollection")**  
    { "ok" : 1 }  
    **>show collections**  
mycollection  
system.indexes



# Tạo bộ sưu tập

17

- MongoDB có thể tự động tạo ra bộ sưu tập  
`>db.tut.insert({"name" : "tutorial"})`  
`>show collections`  
mycollection  
system.indexes  
tut

# Xóa bộ sưu tập

18

- Cú pháp xóa bộ sưu tập  
**db.COLLECTION\_NAME.drop()**
- Ví dụ  
    **>use mydb**  
switched to db mydb  
    **>db.mycollection.drop()**  
true  
    **>show collections**  
system.indexes  
tut

# Kiểu dữ liệu

19

- Chuỗi UTF-8
- Số nguyên
- Luận lý (true/ false)
- Số thực
- Mảng
- Timestamp – ctimestamp, Date
- Đối tượng
- Object ID
- Binary data
- Null, Symbol, giá trị Min/ Max, etc

# Thêm tài liệu (document) vào bộ sưu tập

20

- Cú pháp thêm tài liệu vào bộ sưu tập  
**`db.COLLECTION_NAME.insert(document)`**
- Ví dụ  

```
>db.mycol.insert({  
  _id: ObjectId('5816baa0a2b7a9f009f2f2b7'),  
  title: 'MongoDB Overview',  
  description: 'MongoDB is no sql database',  
  by: 'tutorials point',  
  tags: ['mongodb', 'database', 'NoSQL'],  
  likes: 100  
})
```

# Thêm tài liệu vào bộ sưu tập

21

- MongoDB tự tạo ra trường **\_id** có giá trị duy nhất

```
>db.post.insert([
  {
    title: 'MongoDB Overview',
    description: 'MongoDB is no sql database',
    by: 'tutorials point',
    tags: ['mongodb', 'database', 'NoSQL'],
    likes: 100
  },
  {
    title: 'NoSQL Database',
    description: 'NoSQL db doesn't have tables',
```

# Thêm tài liệu vào bộ sưu tập

22

```
by: 'tutorials point',
tags: ['mongodb', 'database', 'NoSQL'],
likes: 20,
comments: [
  {
    user: 'user1',
    message: 'My first comment',
    dateCreated: new Date(2013,11,10,2,35),
    like: 0
  }
]
}
])
```

# Truy vấn tài liệu

23

- Phương thức truy vấn

`db.COLLECTION_NAME.find()`

`db.COLLECTION_NAME.findOne()`

- Ví dụ

```
>db.mycol.find().pretty()
```

```
{
```

```
  "_id": ObjectId("5816baa0a2b7a9f009f2f2b7"),
```

```
  "title": "MongoDB Overview",
```

```
  "description": "MongoDB is no sql database",
```

```
  "by": "tutorials point",
```

```
  "tags": ["mongodb", "database", "NoSQL"],
```

```
  "likes": "100"
```

```
}
```

# Truy vấn tài liệu

24

Operation	Syntax	Example	RDBMS Equivalent
Equality	{<key>: <value>}	db.mycol.find({"by":"tutorials point"}).pretty()	where by = 'tutorials point'
Less Than	{<key>: { \$lt: <value> } }	db.mycol.find({"likes": { \$lt: 50 } }).pretty()	where likes < 50
Less Than Equals	{<key>: { \$lte: <value> } }	db.mycol.find({"likes": { \$lte: 50 } }).pretty()	where likes <= 50
Greater Than	{<key>: { \$gt: <value> } }	db.mycol.find({"likes": { \$gt: 50 } }).pretty()	where likes > 50
Greater Than Equals	{<key>: { \$gte: <value> } }	db.mycol.find({"likes": { \$gte: 50 } }).pretty()	where likes >= 50
Not Equals	{<key>: { \$ne: <value> } }	db.mycol.find({"likes": { \$ne: 50 } }).pretty()	where likes != 50



# Truy vấn tài liệu

25

- Lọc các trường (1: quan tâm, 0: bỏ qua)  
`db.COLLECTION_NAME.find({}, {KEY:1})`
- Ví dụ  

```
>db.mycol.find({}, {title: 1, _id:0}).pretty()  
{ "title" : "MongoDB Overview" }  
{ "title" : "NoSQL Overview" }  
{ "title" : "Neo4j Overview" }
```

# Truy vấn tài liệu

26

- Toán tử AND

```
db.mycol.find(  
  {  
    $and: [  
      {key1: value1}, {key2:value2}  
    ]  
  }  
).pretty()
```

- Ví dụ

```
>db.mycol.find({$and:[{"by":"tutorials point"},"title":  
"MongoDB Overview"]}).pretty()
```

# Truy vấn tài liệu

27

- Toán tử OR

```
db.mycol.find(  
  {  
    $or: [  
      {key1: value1}, {key2:value2}  
    ]  
  }  
).pretty()
```

- Ví dụ

```
>db.mycol.find({$or:[{"by":"tutorials point"}, {"title":  
"MongoDB Overview"}]}).pretty()
```

# Truy vấn tài liệu

28

- Giới hạn

```
db.COLLECTION_NAME.find().limit(NUMBER)
```

```
db.COLLECTION_NAME.find().limit(NUMBER).skip(NUMBER)
```

- Bộ sưu tập **mycol** bao gồm

```
{ "_id" : ObjectId("5816bed4a2b7a9f009f2f2bb"), "title" : "MongoDB Overview" }
```

```
{ "_id" : ObjectId("5816bed4a2b7a9f009f2f2bc"), "title" : "NoSQL Overview" }
```

```
{ "_id" : ObjectId("5816bed4a2b7a9f009f2f2bd"), "title" : "Tutorials Point Overview" }
```

- Ví dụ

```
>db.mycol.find({},{"title":1,_id:0}).limit(2)
```

```
{"title":"MongoDB Overview"}
```

```
{"title":"NoSQL Overview"}
```

```
>db.mycol.find({},{"title":1,_id:0}).limit(2).skip(1)
```

```
{"title":"NoSQL Overview"}
```

# Truy vấn tài liệu

29

- Sắp xếp (1: thứ tự tăng, -1: thứ tự giảm)  
`db.COLLECTION_NAME.find().sort({KEY:1})`
- Ví dụ  

```
>db.mycol.find({},{"title":1,_id:0}).sort({"title":-1})
{"title":"Tutorials Point Overview"}
{"title":"NoSQL Overview"}
{"title":"MongoDB Overview"}
```

# Cập nhật tài liệu

30

- Cập nhật

**`db.COLLECTION_NAME.update(SELECTION_CRITERIA, UPDATED_DATA)`**

- Ví dụ

```
>db.mycol.update({'title':'MongoDB Overview'},{$set:
{'title':'New MongoDB Tutorial'}})
```

```
>db.mycol.find()
```

```
{ "_id" : ObjectId("5816bed4a2b7a9f009f2f2bb"), "title" : "New MongoDB
Tutorial" }
```

```
{ "_id" : ObjectId("5816bed4a2b7a9f009f2f2bc"), "title" : "NoSQL Overview" }
```

```
{ "_id" : ObjectId("5816bed4a2b7a9f009f2f2bd"), "title" : "Tutorials Point
Overview" }
```

# Cập nhật tài liệu

31

- Mặc định MongoDB chỉ cập nhật một tài liệu, nếu muốn cập nhật nhiều tài liệu cần sử dụng **multi:true**

- Ví dụ

```
>db.mycol.update({'title':'MongoDB Overview'},
  {$set: {'title':'New MongoDB Tutorial'}},{multi:true})
>db.mycol.find()
{ "_id" : ObjectId("5816bed4a2b7a9f009f2f2bb"), "title" : "New MongoDB Tutorial" }
{ "_id" : ObjectId("5816bed4a2b7a9f009f2f2bc"), "title" : "NoSQL Overview" }
{ "_id" : ObjectId("5816bed4a2b7a9f009f2f2bd"), "title" : "Tutorials Point Overview" }
```

# Cập nhật tài liệu

32

- Thay thế tài liệu

```
db.COLLECTION_NAME.save({_id:ObjectId(),NEW_DATA})
```

- Ví dụ

```
>db.mycol.save(  
  {  
    "_id" : ObjectId("5816bed4a2b7a9f009f2f2bb"), "title":"New  
Topic", "by":"Tutorials Point"  
  }  
)  
>db.mycol.find()  
{ "_id" : ObjectId("5816bed4a2b7a9f009f2f2bb"), "title" : "New Topic", "by" :  
"Tutorials Point" }  
{ "_id" : ObjectId("5816bed4a2b7a9f009f2f2bc"), "title" : "NoSQL Overview" }  
{ "_id" : ObjectId("5816bed4a2b7a9f009f2f2bd"), "title" : "Tutorials Point  
Overview" }
```



# Xóa tài liệu

33

- Xóa tài liệu

**`db.COLLECTION_NAME.remove(DELETION_CRITERIA)`**

- Ví dụ

```
>db.mycol.remove({'title':'New Topic'})
```

```
>db.mycol.find()
```

```
{ "_id" : ObjectId("5816bed4a2b7a9f009f2f2bc"), "title" : "NoSQL Overview" }
```

```
{ "_id" : ObjectId("5816bed4a2b7a9f009f2f2bd"), "title" : "Tutorials Point  
Overview" }
```

# Xóa tài liệu

34

- Xóa 1 tài liệu

```
db.COLLECTION_NAME.remove(DELETION_CRITERIA,1)
```

- Xóa tất cả tài liệu

```
db.COLLECTION_NAME.remove()
```

- Ví dụ

```
>db.mycol.remove()
```

```
>db.mycol.find()
```

# Tổng hợp tài liệu

35

- Bộ sưu tập **mycol** có các tài liệu như sau

```
{
  _id: ObjectId('...'), title: 'MongoDB Overview', description: 'MongoDB is no sql
database', by_user: 'tutorials point', url: 'http://www.tutorialspoint.com',
  tags: ['mongodb', 'database', 'NoSQL'], likes: 100
},
{
  _id: ObjectId('...'), title: 'NoSQL Overview', description: 'No sql database is
very fast', by_user: 'tutorials point', url: 'http://www.tutorialspoint.com',
  tags: ['mongodb', 'database', 'NoSQL'], likes: 10
},
{
  _id: ObjectId('...'), title: 'Neo4j Overview', description: 'Neo4j is no sql
database', by_user: 'Neo4j', url: 'http://www.neo4j.com', tags: ['neo4j',
'database', 'NoSQL'], likes: 750
}
```

# Tổng hợp tài liệu

36

- Phương thức tổng hợp tài liệu  
`db.COLLECTION_NAME.aggregate(AGGREGATE_OPERATION)`
- Ví dụ nhóm các tài liệu theo **by\_user** (sử dụng toán tử **\$group**) và đếm số tài liệu (sử dụng toán tử **\$sum**)  

```
>db.mycol.aggregate([{$group : {_id : "$by_user",  
num_tutorial : {$sum : 1}}}]])  
{ "_id" : "Neo4j", "num_tutorial" : 1 }  
{ "_id" : "tutorials point", "num_tutorial" : 2 }
```

# Tổng hợp tài liệu

37

Expression	Description	Example
\$sum	Sums up the defined value from all documents in the collection.	<code>db.mycol.aggregate([{\$group : {_id : "\$by_user", num_tutorial : {\$sum : "\$likes"}}}])</code>
\$avg	Calculates the average of all given values from all documents in the collection.	<code>db.mycol.aggregate([{\$group : {_id : "\$by_user", num_tutorial : {\$avg : "\$likes"}}}])</code>
\$min	Gets the minimum of the corresponding values from all documents in the collection.	<code>db.mycol.aggregate([{\$group : {_id : "\$by_user", num_tutorial : {\$min : "\$likes"}}}])</code>
\$max	Gets the maximum of the corresponding values from all documents in the collection.	<code>db.mycol.aggregate([{\$group : {_id : "\$by_user", num_tutorial : {\$max : "\$likes"}}}])</code>

# Tổng hợp tài liệu

38

Expression	Description	Example
\$push	Inserts the value to an array in the resulting document.	<code>db.mycol.aggregate([{\$group : {_id : "\$by_user", url : {\$push: "\$url"}}}])</code>
\$addToSet	Inserts the value to an array in the resulting document but does not create duplicates.	<code>db.mycol.aggregate([{\$group : {_id : "\$by_user", url : {\$addToSet : "\$url"}}}])</code>
\$first	Gets the first document from the source documents according to the grouping. Typically this makes only sense together with some previously applied "\$sort"-stage.	<code>db.mycol.aggregate([{\$group : {_id : "\$by_user", first_url : {\$first : "\$url"}}}])</code>
\$last	Gets the last document from the source documents according to the grouping. Typically this makes only sense together with some previously applied "\$sort"-stage.	<code>db.mycol.aggregate([{\$group : {_id : "\$by_user", last_url : {\$last : "\$url"}}}])</code>

# Tổng hợp tài liệu

39

- Kết hợp các bước xử lý dạng ống dẫn (pipeline)

`$project`: chọn vài trường

`$match`: lọc các tài liệu

`$group`: nhóm các tài liệu

`$sort`: sắp xếp tài liệu

`$skip`: bỏ qua các tài liệu

`$limit`: lấy các tài liệu top đầu

- Ví dụ

```
>db.mycol.aggregate(  
  [  
    {$group : {_id : "$by_user", num_tutorial : {$sum : 1}}},  
    {$sort: {num_tutorial : -1}}  
  ]  
)
```

# Biểu thức chính quy (regular expression)

40

- Toán tử **\$regex**

```
{ $regex: "pattern-matching" }
```

```
{ $regex: "pattern-matching", $options: 'i' }
```

- Ví dụ

```
>db.mycol.find(
```

```
{ title : { $regex : "nosql", $options : 'i' } }, { title:1, _id:0 }  
) .pretty()
```

```
{ "title" : "NoSQL Overview" }
```

```
>db.mycol.find(
```

```
{ title : { $regex : /over/, $options : 'i' } }, { title:1, _id:0 }  
) .pretty()
```

```
{ "title" : "MongoDB Overview" }
```

```
{ "title" : "NoSQL Overview" }
```

```
{ "title" : "Neo4j Overview" }
```



# Các chủ đề khác

41

- Nhân bản (replication)
- Lưu dự phòng (backup)
- MongoDB trên PC cluster
- Xử lý giao dịch (transaction)
- Map reduce
- Lưu trữ đối tượng nhị phân như ảnh, âm thanh, văn bản, video (GridFS)
- Tìm kiếm chuỗi
- Tạo chỉ mục

# Nội dung

42

- Giới thiệu
- Quản trị cơ bản CSDL
- Tạo, đọc, cập nhật, xóa (CRUD)
- **MongoDB – PHP**
- MongoDB – Java
- MongoDB – Python

# MongoDB - PHP

43

- Thực hiện kết nối đến server
- Chọn CSDL
- Thực hiện thao tác CRUD
- Xử lý kết quả
- Đóng kết nối

# MongoDB - PHP

44

- Thực hiện kết nối đến server

```
$connection = new MongoDB\Driver\Manager  
("mongodb://username:password@hostname:port")
```

- Chọn CSDL, bộ sưu tập, và thực hiện truy vấn

```
$query = new MongoDB\Driver\Query();  
$rows = $connection->executeQuery("mydb.mycol",  
$query);
```

- Thực hiện thao tác CRUD, chú ý các tham số ở dạng mảng `array('key'=>'val')`

# Ví dụ: p1.php

45

```
<?php
try {
    // connect to mongodb
    $m = new MongoDB\Driver\Manager("mongodb://user:pass@
                                   ip-server:27017");

    // query
    $query = new MongoDB\Driver\Query([]);
    $rows = $m->executeQuery("mydb.mycol", $query);
    // show results in table
    echo "<table border='1'>";
    echo "<tr><th>Title</th> <th>Description</th> <th>By</th>
        <th>Likes</th> </tr>";
```

# Ví dụ: p1.php

46

```
// iterate cursor to display title of documents
foreach ($rows as $row) {
    echo "<tr>";
    echo "<td>" . $row->title . "</td>";
    echo "<td>" . $row->description . "</td>";
    echo "<td>" . $row->by . "</td>";
    echo "<td>" . $row->likes . "</td>";
    echo "</tr>";
}
echo "</table>";
} catch (MongoDB\Driver\Exception\Exception $e) {
    echo "Exception:", $e->getMessage(), "\n";
}
?>
```

# Nội dung

47

- Giới thiệu
- Quản trị cơ bản CSDL
- Tạo, đọc, cập nhật, xóa (CRUD)
- MongoDB – PHP
- **MongoDB – Java**
- MongoDB – Python

# MongoDB - JAVA

48

- Thực hiện kết nối đến server

```
MongoClientURI uri = new MongoClientURI("mongodb://  
username:password@hostname:port/?authSource=admin");  
MongoClient mongoClient = new MongoClient(uri);
```

- Chọn CSDL

```
DB db = mongoClient.getDB("dbname");
```

- Chọn bộ sưu tập

```
DBCollection col = db.getCollection("colname");
```



# MongoDB - JAVA

49

- Thực hiện kết nối đến server

```
MongoClientURI uri = new MongoClientURI("mongodb://  
username:password@hostname:port/?authSource=admin");  
MongoClient mongoClient = new MongoClient(uri);
```

- Chọn CSDL

```
DB db = mongoClient.getDB("dbname");
```

- Tạo, chọn bộ sưu tập

```
DBCollection col = db.createCollection("colname");  
DBCollection col = db.getCollection("colname");
```

# MongoDB - JAVA

50

- Thực hiện thao tác CRUD

```
BasicDBObject doc = new BasicDBObject();  
doc.put("key1", "val1");  
doc.put("key2", "val2"); ...  
col.insert(doc);
```

```
DBCursor cursor = col.find();
```

```
BasicDBObject q = new BasicDBObject();  
q.put("key", "val");  
DBCursor cursor = col.find(q);
```

# MongoDB - JAVA

51

- Thực hiện thao tác CRUD

```
doc.put("key", "val");
```

```
doc.get("key");
```

```
col.update(doc);
```

```
col.remove();
```

```
col.remove(doc);
```

# Ví dụ: MongoDBJDBC.java

52

```
import com.mongodb.MongoClient;
import com.mongodb.MongoClientURI;
import com.mongodb.MongoException;
import com.mongodb.DB;
import com.mongodb.DBCollection;
import com.mongodb.DBObject;
import com.mongodb.DBCursor;

public class MongoDBJDBC {
    public static void main(String args[]) {
        try{
            // To connect to mongodb server
            MongoClientURI uri = new MongoClientURI("mongodb://
                user:pass@ip-server:27017/?authSource=admin");
            MongoClient mongoClient = new MongoClient(uri);
```

# Ví dụ: MongoDBJDBC.java

53

```
// Now connect to your databases
DB db = mongoClient.getDB("mydb");
DBCollection coll = db.getCollection("mycol");
DBCursor cursor = coll.find();
while (cursor.hasNext()) {
    DBObject doc = cursor.next();
    System.out.println(doc);
}
} catch(Exception e) {
    System.err.println(e.getClass().getName() + ": " +
        e.getMessage() );
}

} // main
} // class
```

# Nội dung

54

- Giới thiệu
- Quản trị cơ bản CSDL
- Tạo, đọc, cập nhật, xóa (CRUD)
- MongoDB – PHP
- MongoDB – Java
- **MongoDB – Python**

# MongoDB - Python

55

- Thực hiện kết nối đến server

```
connection = MongoClient  
("mongodb://username:password@hostname:port")
```

- Chọn CSDL

```
db = connection["dbname"]
```

hoặc

```
db = connection.dbname
```

- Chọn bộ sưu tập

```
collection = db.colname
```

# MongoDB - Python

56

- Thực hiện thao tác CRUD

```
collection.insert_one({"key1": "val1", "key2":  
"val2", ...})
```

```
collection.find()
```

```
collection.update_one({"key": "criteria"}, {"$set":  
{"key1": "val1", "key2": "val2", ...}})
```

```
collection.delete_one({"key": "criteria"})
```

```
collection.insert_many()
```

```
collection.update_many()
```

```
collection.delete_many()
```



# Ví dụ: Mongo.py

57

```
from pymongo import MongoClient

try:
    # To connect to mongodb server
    connection = MongoClient('mongodb://user:pass@
                             ip-server:27017')

    # Now connect to your databases
    db = connection.mydb
    # Exec query
    collection = db.mycol
    result = collection.find()
    for doc in result:
        print doc

except Exception, e:
    print str(e)
```

# Tài liệu tham khảo

58

- E. Plugge, D. Hows and P. Membrey, "*MongoDB Basics*", Apress, 2014
- K. Banker, P. Bakkum, S. Verch, D. Garrett and T. Hawkins, "*MongoDB in Action*", Manning Publications, 2016
- Tutorialspoint, "*MongoDB Tutorial*", 2016
- MongoDB, <https://www.mongodb.com>