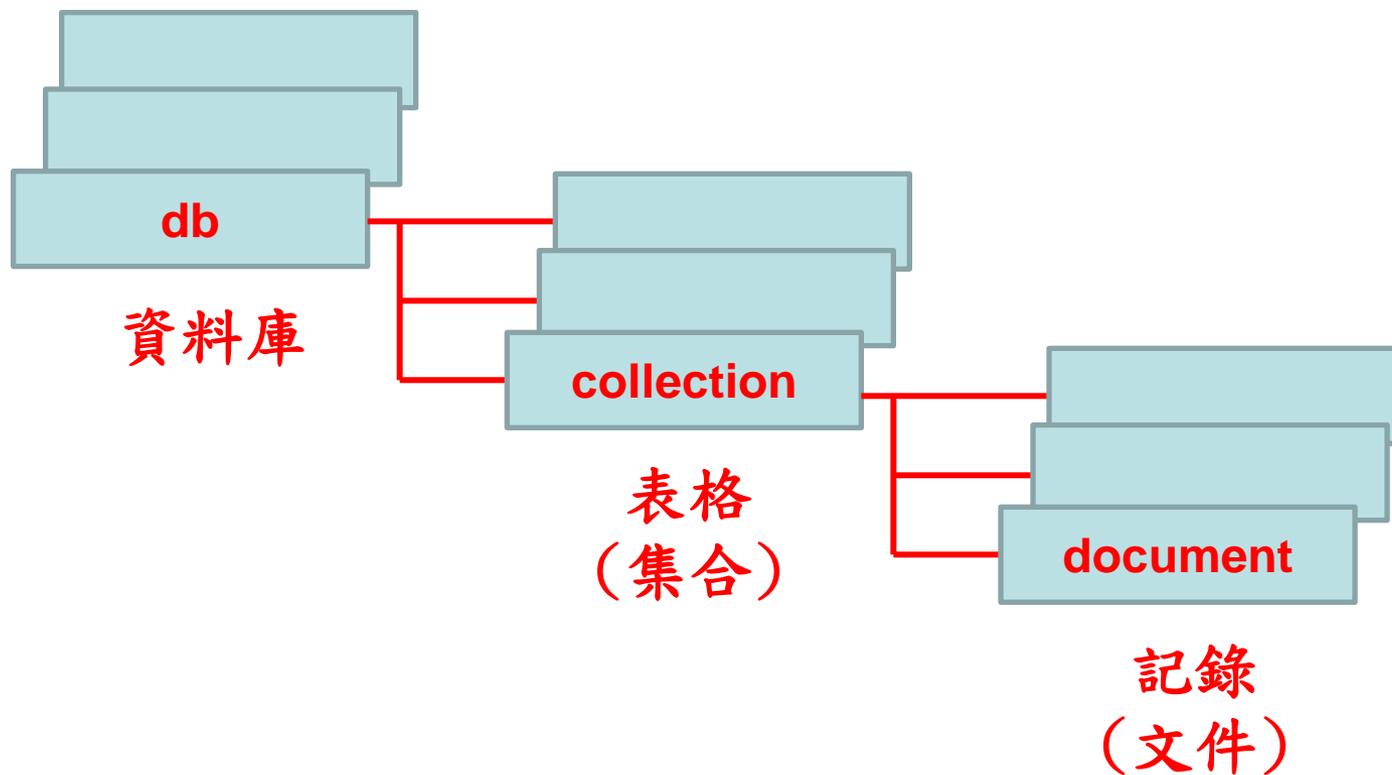

MongoDB 雲端資料庫簡介 (二)



日期 2016 06/08
地點 校務研究辦公室

MongoDB 操作-基本觀念 01



MongoDB 操作-基本觀念 02

MongoDB 與傳統資料庫比較

RDBMS	MongoDB
Table	Collection
Column	Key
Value	Value
Records / Rows	Document / Object

MongoDB 操作-基本觀念 03

Collections

```
{
  "_id" : ObjectId("527b3cc65ceafed9b2254a94"),
  "f_name" : "Zenny",
  "sex" : "Female",
  "class" : "VIII",
  "age" : 13,
  "grd_point" : 28.2514
}
```

← Document1

```
{
  "_id" : ObjectId("527b3cc65ceafed9b2254a95"),
  "f_name" : "Paul",
  "sex" : "Male",
  "class" : "VII",
  "age" : 13,
  "grd_point" : 28.5004
}
```

Document2 →

```
{
  "_id" : ObjectId("527b3cc65ceafed9b2254a97"),
  "f_name" : "Lassy",
  "sex" : "Female",
  "class" : "VIII",
  "age" : 13,
  "grd_point" : 28.2514
}
```

← Document3

MongoDB 操作-基本觀念 04

比較運算

Name	Description
\$gt	Matches values that are greater than the value specified in the query. 大於
\$lt	Matches values that are less than the value specified in the query. 小於
\$gte	Matches values that are greater than or equal to the value specified 大或等於
\$lte	Matches values that are less than or equal to the value specified in the query 小或等於
\$ne	Matches all values that are not equal to the value specified in the query 不等於
\$in	Matches any of the values that exist in an array specified in the query 存在陣列內
\$nin	Matches values that do not exist in an array specified to the query 不存在陣列內

MongoDB 操作-基本觀念 05

邏輯運算

Name	Description
<code>\$and</code>	Joins query clauses with a logical AND returns all documents that match the conditions of both clauses.
<code>\$not</code>	Inverts the effect of a query expression and returns documents that do not match the query expression.
<code>\$or</code>	Joins query clauses with a logical OR returns all documents that match the conditions of either clause.
<code>\$nor</code>	Joins query clauses with a logical NOR returns all documents that fail to match both clauses.

MongoDB 操作-基本指令 01

mongod --dbpath C:\Users\user\mongo01

```
C:\Program Files\MongoDB\Server\3.0\bin>mongod --dbpath C:\Users\user\mongo01
2015-07-25T09:55:06.366+0800 I JOURNAL [initandlisten] journal dir=C:\Users\user\mongo01\journal
2015-07-25T09:55:06.366+0800 I JOURNAL [initandlisten] recover : no journal files present, no recovery needed
2015-07-25T09:55:06.541+0800 I JOURNAL [durability] Durability thread started
2015-07-25T09:55:06.543+0800 I JOURNAL [journal writer] Journal writer thread started
2015-07-25T09:55:06.635+0800 I CONTROL [initandlisten] MongoDB starting : pid=7012 port=27017 dbpath=C:\Users\user\mongo01 64-bit host=nb
2015-07-25T09:55:06.635+0800 I CONTROL [initandlisten] targetMinOS: Windows 7/Windows Server 2008 R2
2015-07-25T09:55:06.636+0800 I CONTROL [initandlisten] db version v3.0.4
2015-07-25T09:55:06.636+0800 I CONTROL [initandlisten] git version: 0481c958dae
```

須先新增目錄C:\Users\user\mongo01(資料庫儲存區)

```
2015-07-25T09:55:06.637+0800 I CONTROL [initandlisten] build info: windows sys.getwindowsversion(major=6, minor=1, build=7601, platform=2, service_pack='Service Pack 1') BOOST_LIB_VERSION=1_49
2015-07-25T09:55:06.637+0800 I CONTROL [initandlisten] allocator: tcmalloc
2015-07-25T09:55:06.638+0800 I CONTROL [initandlisten] options: { storage: { dbPath: "C:\Users\user\mongo01" } }
2015-07-25T09:55:06.642+0800 I INDEX [initandlisten] allocating new ns file C:\Users\user\mongo01\local.ns, filling with zeroes...
2015-07-25T09:55:07.106+0800 I STORAGE [FileAllocator] allocating new datafile C:\Users\user\mongo01\local.0, filling with zeroes...
2015-07-25T09:55:07.108+0800 I STORAGE [FileAllocator] creating directory C:\Users\user\mongo01\_tmp
2015-07-25T09:55:07.118+0800 I STORAGE [FileAllocator] done allocating datafile C:\Users\user\mongo01\local.0, size: 64MB, took 0.005 secs
2015-07-25T09:55:07.138+0800 I NETWORK [initandlisten] waiting for connections on port 27017
```

27017 通訊埠

MongoDB 操作-基本指令 02

```
C:\Program Files\MongoDB\Server\3.0\bin>mongo → 登入資料庫
MongoDB shell version: 3.0.4 → 資料庫版本
connecting to: test
> show dbs; → 顯示所有資料庫
local 0.078GB

>
> db. → 按 Tab 鍵以顯示所有資料庫指令
db.adminCommand<      db.grantRolesToUser<
db.auth<              db.group<
db.changeUserPassword< db.groupcmd<
db.cloneCollection<   db.groupeval<
db.cloneDatabase<     db.hasOwnProperty<
db.commandHelp<      db.help<
db.constructor         db.hostInfo<
db.copyDatabase<      db.isMaster<
db.createCollection<  db.killOP<
db.createRole<        db.killOp<
db.createUser<        db.listCommands<
db.currentOP<         db.loadServerScripts<
db.currentOp<         db.logout<
db.dbEval<            db.printCollectionStats<
db.dropAllRoles<     db.printReplicationInfo<

>
> exit → 離開資料庫
bye

C:\Program Files\MongoDB\Server\3.0\bin>
```

MongoDB 操作-基本指令 03

```
> show dbs;
```

顯示所有資料庫

```
admin 0.078GB
local 0.078GB
test 0.078GB
```

```
> use ctu;
```

切換（並新增）資料庫

```
switched to db ctu
```

```
> db.cce.insert( {math:80, name:"emma"} );
```

新增（集合表格與）文件記錄

```
WriteResult({ "nInserted" : 1 })
```

```
> db.cce.insert( {math:70, name:"john"} );
```

共新增三筆文件記錄

```
WriteResult({ "nInserted" : 1 })
```

```
> db.cce.insert( {math:90, name:"tom"} );
```

```
WriteResult({ "nInserted" : 1 })
```

```
> db.cce.find();
```

顯示集合表格 cce 的所有文件記錄

```
{ "_id" : ObjectId("55b353a62aa5051728393bbe"), "math" : 80, "name" : "emma" }
```

```
{ "_id" : ObjectId("55b353b92aa5051728393bbf"), "math" : 70, "name" : "john" }
```

```
{ "_id" : ObjectId("55b353c62aa5051728393bc0"), "math" : 90, "name" : "tom" }
```

```
> show dbs;
```

顯示所有資料庫

```
admin 0.078GB
ctu 0.078GB
local 0.078GB
test 0.078GB
```



MongoDB 操作-基本指令 04

```
> db.cce.insert( {math:70, name:"john"} );  
WriteResult({ "nInserted" : 1 })
```

{math:70, name:"john"} 一筆文件記錄

欄位名稱：欄位內容
(70整數, "john"字串)

```
> db.cce.find();  
{ "_id" : ObjectId("55b353a62aa5051728393bbe"), "math" : 80, "name" : "emma" }  
{ "_id" : ObjectId("55b353b92aa5051728393bbf"), "math" : 70, "name" : "john" }  
{ "_id" : ObjectId("55b353c62aa5051728393bc0"), "math" : 90, "name" : "tom" }
```

主鍵名稱：主鍵內容(內定自動產生)

MongoDB 操作-基本指令 05

```
> db.cce.insert( {math:85, name:"mary", sex:"女" } );
WriteResult({ "nInserted" : 1 })
> db.cce.find();
;
{ "_id" : ObjectId("55b353b92aa5051728393bbf"), "math" : 70, "name" : "john" }
{ "_id" : ObjectId("55b353c62aa5051728393bc0"), "math" : 90, "name" : "tom" }
{ "_id" : ObjectId("55b399a52aa5051728393bc4"), "math" : 80, "name" : "emma" }
{ "_id" : ObjectId("55b399de2aa5051728393bc5"), "math" : 85, "name" : "mary", "s
ex" : "女" }
> _
```

請注意

新增集合表格無須預先宣告欄位格式 (Schema)

若存在有一筆 { _id: 1, name: "mary" }

- insert({ _id: 1, name: "john" }) //會提示錯誤
- save({ _id: 1, name: "john" }) //會把 mary 改為 john

save 操作在 _id 存在的狀況下，會執行更新動作；
若 _id 找不到的時候 save 才會執行 insert 動作。

MongoDB 操作-基本指令 06

```
> db.cce.find();
{ "_id" : ObjectId("55b353b92aa5051728393bbf"), "math" : 70, "name" : "john" }
{ "_id" : ObjectId("55b353c62aa5051728393bc0"), "math" : 90, "name" : "tom" }
{ "_id" : ObjectId("55b359652aa5051728393bc2"), "math" : 80, "name" : "emma" }
→> db.cce.update( {name:"emma"}, {math:60, name:"emma"} ); 修改文件記錄內容
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.cce.find();
{ "_id" : ObjectId("55b353b92aa5051728393bbf"), "math" : 70, "name" : "john" }
{ "_id" : ObjectId("55b353c62aa5051728393bc0"), "math" : 90, "name" : "tom" }
{ "_id" : ObjectId("55b359652aa5051728393bc2"), "math" : 60, "name" : "emma" }
→> db.cce.remove( {name:"emma"} ); 刪除文件記錄內容
WriteResult({ "nRemoved" : 1 })
> db.cce.find();
{ "_id" : ObjectId("55b353b92aa5051728393bbf"), "math" : 70, "name" : "john" }
{ "_id" : ObjectId("55b353c62aa5051728393bc0"), "math" : 90, "name" : "tom" }
>
```

文件記錄的操作指令

insert新增, **remove**刪除, **update**修改, **find**查詢

MongoDB 操作-基本指令 07

```
> db.cce.find();
{ "_id" : ObjectId("55b353b92aa5051728393bbf"), "math" : 70, "name" : "john" }
{ "_id" : ObjectId("55b353c62aa5051728393bc0"), "math" : 90, "name" : "tom" }
{ "_id" : ObjectId("55b399a52aa5051728393bc4"), "math" : 30, "name" : "emma" }
{ "_id" : ObjectId("55b399de2aa5051728393bc5"), "math" : 85, "name" : "mary", "sex" : "女" }

> db.cce.find( {math: {$gt:60}} ); 數學大於 60
{ "_id" : ObjectId("55b353b92aa5051728393bbf"), "math" : 70, "name" : "john" }
{ "_id" : ObjectId("55b353c62aa5051728393bc0"), "math" : 90, "name" : "tom" }
{ "_id" : ObjectId("55b399de2aa5051728393bc5"), "math" : 85, "name" : "mary", "sex" : "女" }

> db.cce.find( {math: {$gt:60, $lt:90}} ); 數學大於 60 小於 90
{ "_id" : ObjectId("55b353b92aa5051728393bbf"), "math" : 70, "name" : "john" }
{ "_id" : ObjectId("55b399de2aa5051728393bc5"), "math" : 85, "name" : "mary", "sex" : "女" }

>
```

MongoDB 操作-基本指令 08

> 數學由小到大排列

```
> db.cce.find( {math: {$gt:60, $lt:95}} ).sort( {math:1} );
{ "_id" : ObjectId("55b353b92aa5051728393bbf"), "math" : 70, "name" : "john" }
{ "_id" : ObjectId("55b399de2aa5051728393bc5"), "math" : 85, "name" : "mary", "sex" : "女" }
{ "_id" : ObjectId("55b353c62aa5051728393bc0"), "math" : 90, "name" : "tom" }
>
```

> 數學由大到小排列

```
> db.cce.find( {math: {$gt:60, $lt:95}} ).sort( {math:-1} );
{ "_id" : ObjectId("55b353c62aa5051728393bc0"), "math" : 90, "name" : "tom" }
{ "_id" : ObjectId("55b399de2aa5051728393bc5"), "math" : 85, "name" : "mary", "sex" : "女" }
{ "_id" : ObjectId("55b353b92aa5051728393bbf"), "math" : 70, "name" : "john" }
>
```

MongoDB 操作-基本指令 09

> 限制查詢文件筆數

```
> db.cce.find().limit(3);
[ "_id" : ObjectId("55b353b92aa5051728393bbf"), "math" : 70, "name" : "john" ]
[ "_id" : ObjectId("55b353c62aa5051728393bc0"), "math" : 90, "name" : "tom" ]
[ "_id" : ObjectId("55b399a52aa5051728393bc4"), "math" : 30, "name" : "emma" ]
> db.cce.find().limit(4);
[ "_id" : ObjectId("55b353b92aa5051728393bbf"), "math" : 70, "name" : "john" ]
[ "_id" : ObjectId("55b353c62aa5051728393bc0"), "math" : 90, "name" : "tom" ]
[ "_id" : ObjectId("55b399a52aa5051728393bc4"), "math" : 30, "name" : "emma" ]
[ "_id" : ObjectId("55b399de2aa5051728393bc5"), "math" : 85, "name" : "mary", "sex" : "女" ]
>
```

MongoDB 操作-基本指令 10

```
> var aaa = db.cce.find();           定義 aaa 物件
> aaa[2]
{
  "_id" : ObjectId("55b399a52aa5051728393bc4"),
  "math" : 30,
  "name" : "emma"
}
> aaa[1]
{
  "_id" : ObjectId("55b353c62aa5051728393bc0"),
  "math" : 90,
  "name" : "tom"
}
> aaa[0]
{
  "_id" : ObjectId("55b353b92aa5051728393bbf"),
  "math" : 70,
  "name" : "john"
}
```

MongoDB 操作-基本指令 11

直接登入時就執行驗證

```
mongo -u user01 -p 1qaz --authenticationDatabase ctu
```

```
C:\Program Files\MongoDB\Server\3.0\bin>mongo -u user01 -p 1qaz --authenticationDatabase ctu
MongoDB shell version: 3.0.6
connecting to: test
> db                                顯示目前資料庫
test
> use ctu;                          切換資料庫
switched to db ctu
> db                                顯示目前資料庫
ctu
> show collections;                顯示所有集合表格
cce
myctu
system.indexes
>
```

MongoDB 操作-基本指令 12

```
> show collections;
```

顯示所有集合表格

```
cce  
system.indexes
```

```
> db.cce.drop();
```

刪除 cce 集合表格

```
true
```

```
> show collections;  
system.indexes
```

!!!已刪除 cce 集合表格!!!

```
> show dbs;
```

顯示所有資料庫

```
admin 0.078GB  
ctu 0.078GB  
local 0.078GB  
test 0.078GB
```

```
> db;
```

顯示目前資料庫

```
ctu
```

```
> db.dropDatabase();
```

刪除目前資料庫

```
{ "dropped" : "ctu", "ok" : 1 }
```

```
> show dbs;
```

```
admin 0.078GB  
local 0.078GB  
test 0.078GB
```

!!!已刪除 ctu 資料庫!!!

```
> _
```

謝謝指教

