

MongoDB Data Type & basic commands

Data:

```
//Data Types
{
  string: "String of text",
  int: 405,
  double: 3.565,
  array:[1,2,3],
  object:{attr: "attr", attr:"attr"},
  date: new Date("<YYYY-mm-dd>"),
  object_id: <ObjectId>,
  no_value: null
}
```

Data type storing in mongoDB is Bson

Insert information

```
//insert a record
db.student.insertOne({name: "Jack", major:"Biology", gpa: 3.5})
db.student.insertOne({_id:2, name: "Claire", major:"Marketing", gpa: 3},
awards:["Valedictorian", "Summa cum laude"]})

//insert many code
db.student.insertMany([
{name: "Mike", major: "Computer Science", gpa: 2.7},
{
_id:2, name: "Claire", major:"Marketing", gpa: 3},
awards:["Valedictorian", "Summa cum laude"]
}
])
```

ObjectId datatype is auto generated for new record

Data structure of mongoDB is so flexible unlike SQL need prepare every columns to the table

Finding information

```
//find all record inside the collection
db.student.find({})
```

```

//limit the result top 3
db.student.find({},{-id:0}).limit(3)

//sort 1 ascending, -1 descending order
db.student.find({}).sort({gpa:-1, name:1})

//filter and logic
db.student.find({major: "Biology", name: "Jack"},{-id:0})

//find all students who are chemistry majors or named jack
db.student.find({$or:[{major: "Chemistry"}, {name: "Jack"}]} , {_id:0})

//find all student whose gpa is greater than 3.5
db.student.find({gpa:{ $gt: 3.5}}, {_id:0})

//find all student whose gpa is less than 3.2
db.student.find({gpa:{ $lte: 3.2}}, {_id:0}).sort({gpa:-1})

//other command $eq, $ne, $lt, $lte, $gt, $gte

//find whose name is in the array
db.student.find({name: { $in: ["Kate", "Claire"]} }, {_id:0}) // $nin -> not in

//find student who has award field
db.student.find({awards: { $exists: true}}, {_id:0})

//find student whose name field is string
db.student.find({ name: { $type: 2}}, {_id: 0})

//find student whose first item in grade field is 90. (find )
db.student.find({"grades.0: 90"}, {_id: 0})

//find student whose gpa can be greater than 80.
db.student.find({ grades: { $elemMatch : { $gte: 80}}}, {_id: 0})

//find student whose array size is 4
db.student.find({ grades: { $size: 4} }, {_id: 0})

```

Data type Table:

| Type | | | |
|----------------------------|----|-----------------------|----------------------------|
| Number | | | |
| Alias | | | |
| Notes | | | |
| Double | 1 | "double" | |
| String | 2 | "string" | |
| Object | 3 | "object" | |
| Array | 4 | "array" | |
| Binary data | 5 | "binData" | |
| Undefined | 6 | "undefined" | Deprecated. |
| ObjectId | 7 | "objectId" | |
| Boolean | 8 | "bool" | |
| Date | 9 | "date" | |
| Null | 10 | "null" | |
| Regular Expression | 11 | "regex" | |
| DBPointer | 12 | "dbPointer" | Deprecated. |
| JavaScript | 13 | "javascript" | |
| Symbol | 14 | "symbol" | Deprecated. |
| JavaScript code with scope | 15 | "javascriptWithScope" | Deprecated in MongoDB 4.4. |
| 32-bit integer | 16 | "int" | |
| Timestamp | 17 | "timestamp" | |
| 64-bit integer | 18 | "long" | |

| | | | |
|------------|-----|-----------|---------------------|
| Decimal128 | 19 | "decimal" | New in version 3.4. |
| Min key | -1 | "minKey" | |
| Max key | 127 | "maxKey" | |

Update information

```
//first filter, then update the field, but only update one field
db.student.updateOne(
  {major: "Biology"},
  {
    $set:
      {major: "chemistry"}
  }
)

db.student.updateMany(
  {major: "Biology"},
  {
    $set:
      {major: "chemistry"}
  }
)

//replace with new entry
db.student.replaceOne(
  {major: "Biology"},
  {name: "new name", major: "new major"}
)
```

Delete information

```
//delete one
db.student.deleteOne({gpa:{$gte: 3.5}})
//delete all
db.student.deleteMany({gpa:{$gte: 3.5}})
```