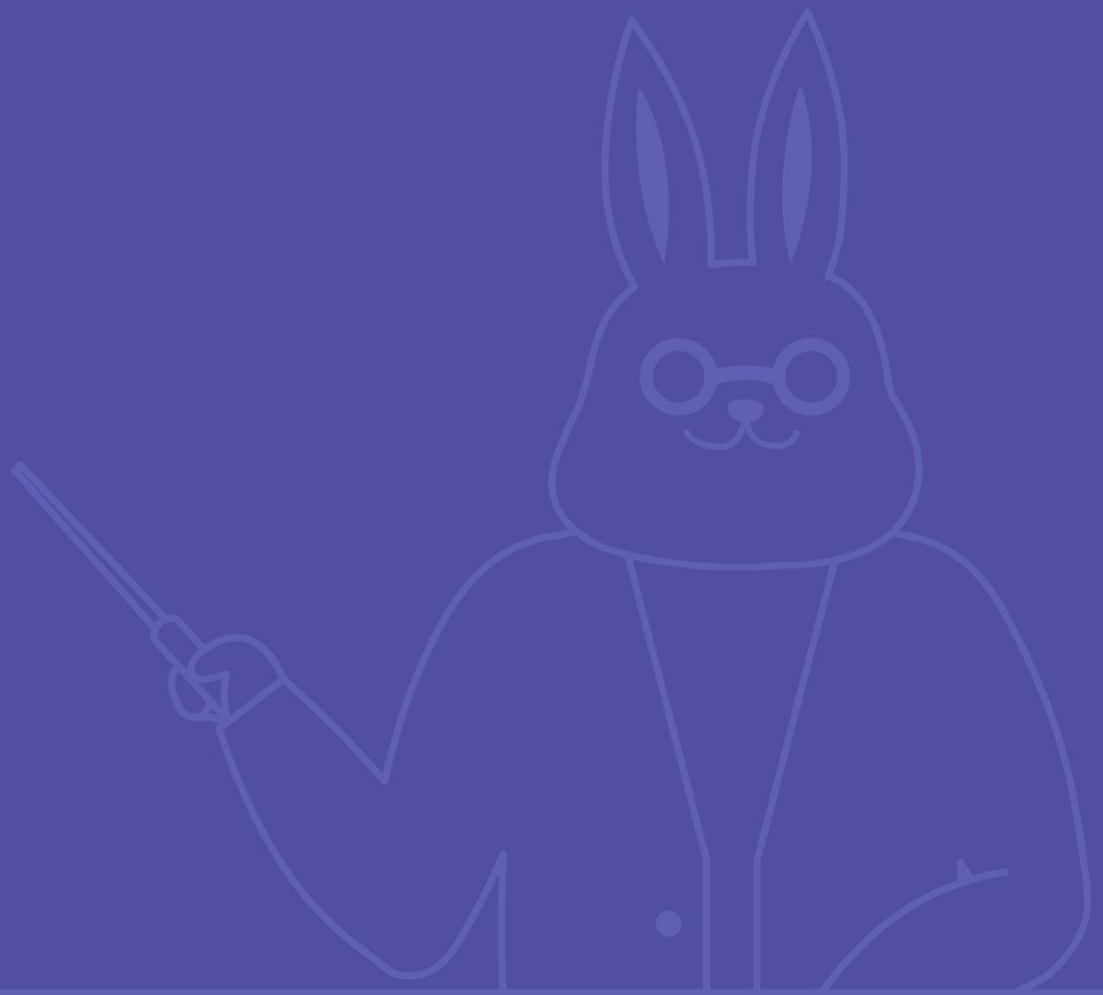
MongoDB 기초

03 쿼리 연산자



목차

01

쿼리의 구조

앞으로 다양하게 배우게 될 쿼리와 연산자

SQL의 모습

```
SELECT * FROM table WHERE column="value"
```

몽고디비 쿼리의 모습

```
collection.find({"field": "value"}, {})
```

BSON 형식으로 쿼리를 표현하다 보니 많이 헷갈린다!
이번 시간에 쿼리에 연산자가 사용되는 형식에 대해 알아보자

쿼리의 형식

몽고디비 쿼리

```
{ <field>: {<operator1>: <value>, <operator2>: <value>}, <field>: ... }
```

기본적으로 쿼리는 필드가 가장 바깥에 있고, 안쪽에 연산자가 들어간다

출처: 맛있는 몽고디비

쿼리의 형식

몽고디비 쿼리

```
{ "height": {"$gte": 175, "$lte": 180}, "width": {"$gte": 60} }
```

기본적으로 쿼리는 필드가 가장 바깥에 있고, 안쪽에 연산자가 들어간다

출처: 맛있는 몽고디비

$or, $and, $nor이 들어간 쿼리의 형식

몽고디비 쿼리

```
{
        <$or,$and,$nor>: [<query>, <query>, ...],
        <field>: {<operator1>: <value>, <operator2>: <value>}, <field>: ...
}
```

하지만 예외적으로 **$or, $and, $nor** 세 개의 연산자는 가장 바깥에 쓰인다

출처: 맛있는 몽고디비

$or, $and, $nor이 들어간 쿼리의 형식

몽고디비 쿼리

```
{ "$or": [ { "status": "A" }, { "qty": { "$lt": 30 } } ] }
```

하지만 예외적으로 **$or, $and, $nor** 세 개의 연산자는 가장 바깥에 쓰인다

출처: 맛있는 몽고디비

형식을 지키지 않은 예시

잘못된 예

```
{"name": {"first": 'Karoid', "last": 'Jeong'}}
```

올바른 예

```
{"name.first": 'Karoid', "name.last": 'Jeong'}}
```

정해진 형식을 지키지 않은 쿼리는 사용될 수 없다

출처: 맛있는 몽고디비

02

점 표기법

점 표기법이란?

BSON 내부의 Object에 접근하기 위한 방법

객체 내부로 접근하기

BSON 구조 예시

```
{
    "_id": ObjectId("542c2b97 bac059 5474 108b48"),
    "name": {"first": "sue", "last": "Turing" },
    "age": 26,
    "is_alive": true,
    "groups": ["news", "sports"],
    "viewTime": ISODate("2017-10-24T05:02:46.395Z")
}
```

만약 name 안의 first 값을 조회하려면 어떻게 할까?

객체 내부로 접근하기

BSON 구조 예시

```
{
 "_id": ObjectId("542c2b97 bac059 5474 108b48"),
 "name": {"first": "sue", "last": "Turing" },
 "age": 26,
 "is_alive": true,
 "groups": ["news", "sports"],
 "viewTime": ISODate("2017-10-24T05:02:46.395Z")
}
```

잘못된 쿼리

```
{"name": {"first": 'sue'}}
```

올바른 쿼리

```
{"name.first": 'sue'}
```

점 연산자로 내부에 접근할 수 있다

배열의 요소에 접근하기

BSON 구조 예시

```
{
 "_id": ObjectId("542c2b97 bac059 5474 108b48"),
 "name": {"first": "sue", "last": "Turing" },
 "age": 26,
 "is_alive": true,
 "groups": ["news", "sports"],
 "viewTime": ISODate("2017-10-24T05:02:46.395Z")
}
```

잘못된 쿼리

```
{"groups":["news"]}
```

올바른 쿼리

```
{"groups.0": "news"}
```

마찬가지로 **배열의 첫 번째 요소**로 news 값을 갖는 도큐먼트를 찾고 싶다면?

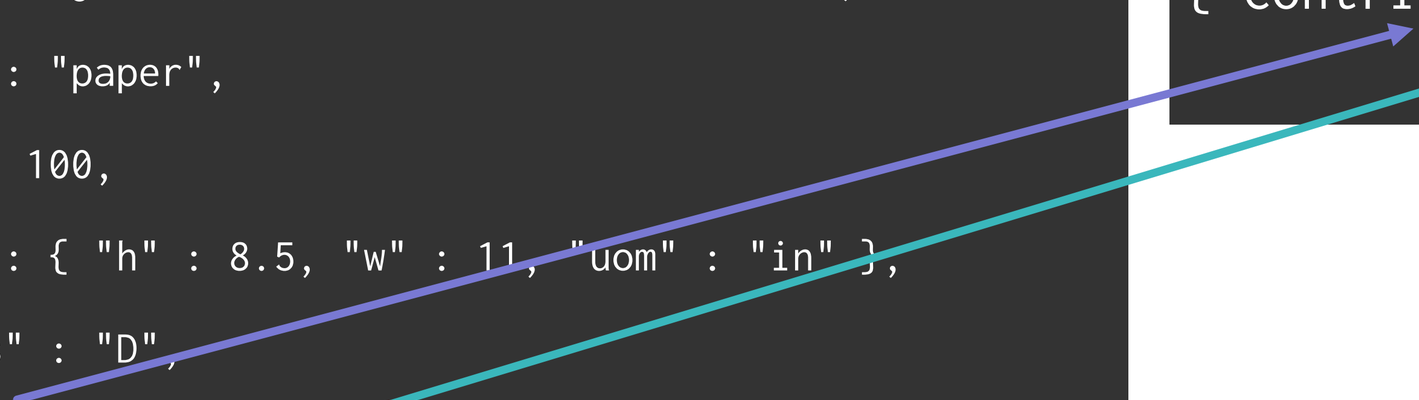
더 복잡한 구조로 검색하기

BSON 구조 예시

```
{
  "_id" : ObjectId("5a105606e8762a54e6c1ee78"),
  "item" : "paper",
  "qty" : 100,
  "size" : { "h" : 8.5, "w" : 11, "uom" : "in" },
  "status" : "D",
  "contribs" : [
    { "name" : "karoid", "password" : 123 },
    { "name" : "db", "password" : 4433 },
    { "size" : 3 }
  ]
 }
```

올바른 쿼리

```
{"contribs.0.name": "karoid"}
```



03

비교 연산자

operator	설명
$eq	(equals) 주어진 값과 일치하는 값
$gt	(greater than) 주어진 값보다 큰 값
$gte	(greather than or equals) 주어진 값보다 크거나 같은 값
$lt	(less than) 주어진 값보다 작은 값
$lte	(less than or equals) 주어진 값보다 작거나 같은 값
$ne	(not equal) 주어진 값과 일치하지 않는 값
$in	주어진 배열 안에 속하는 값
$nin	주어진 배열 안에 속하지 않는 값

대소 비교 쿼리 예시

쿼리

```
articles.find( { "likes": { "$gt": 10, "$lt": 30 } } )
```

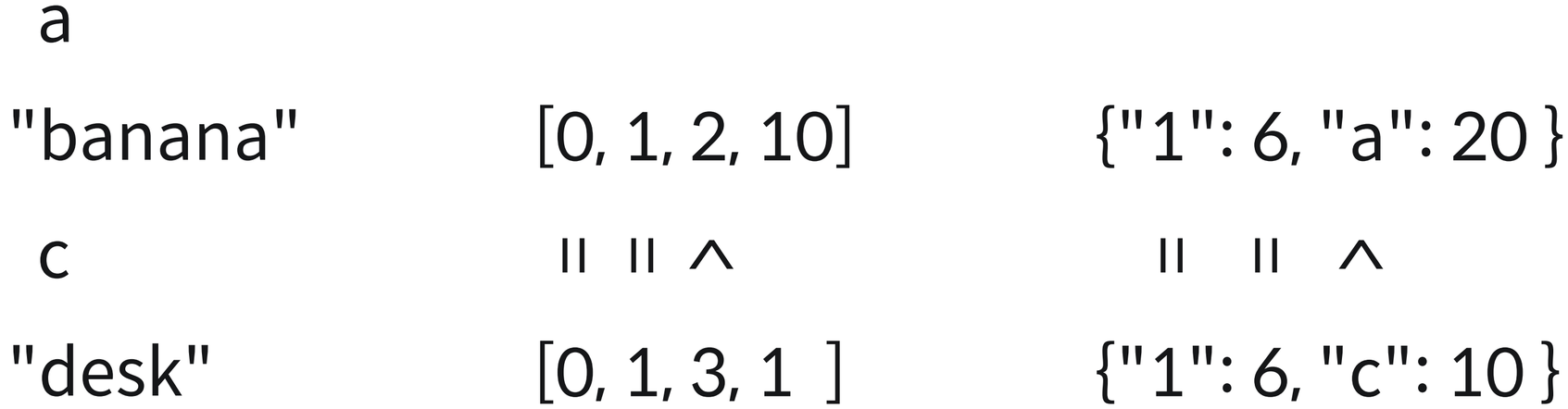
좋아요 수가 10 초과 30 미만인 도큐먼트 검색

쿼리

```
articles.find( { "likes": { "$gte": 10, "$lte": 30 } } )
```

좋아요 수가 10 이상 30 이하인 도큐먼트 검색

숫자 외의 다른 타입 비교



포함 쿼리 예시

쿼리

```
inventory.find( { "qty": { "$in": [ 5, 15 ] } } )
```

수량이 5 또는 15인 아이템 도큐먼트

쿼리

```
inventory.find( { "tags": { "$nin": [
    re.compile("^be"),
    re.compile("^st")
]} } )
```

태그가 정규표현식 ^be 또는 ^st에 일치하지 않는 도큐먼트

04

논리 연산자

operator	설명
$or	주어진 조건 중 하나라도 true 일 때 true
$and	주어진 모든 조건이 true 일 때 true
$nor	주어진 조건 중 하나라도 false 일 때 true
$not	주어진 조건이 false 일 때 true

연산자의 위치 주의

쿼리 예시

```
articles.find({ "$or": [ { "title": "article01" }, { "writer": "Alpha" } ] })
```

게시글 중 제목이 article01 이거나 작가가 Alpha인 도큐먼트

쿼리 예시

```
articles.find({ "likes": { "$not": { "$lte": 11 } } })
```

좋아요 수가 11 이하가 아닌 도큐먼트

and 연산자는 쓸 일이 많지 않다

쿼리

```
inventory.find({"$and": [
        {"qty": {"$gt": 10}},
        {"qty": {"$lt": 100 }}
]})
```

쿼리

```
inventory.find({"qty": {"$gt": 10, "$lt": 100} })
```

두 쿼리는 서로 같은 의미를 갖는다

복합적인 논리 연산자의 사용

쿼리

```
inventory.find( {
    "$and" : [
        { "$or" : [ { "price" : 0.99 }, { "price" : 1.99 } ] },
        { "$or" : [ { "sale" : True }, { "qty" : { "$lt" : 20 } } ] }
    ]
} )
```

논리 연산자를 복합적으로 사용할 때 and 연산자가 쓰일 수 있다

05

문자열 연산자

operator	설명
$mod	그 필드에 modulo operation을 통해 특정 결과가 나온 Document를 선택한다.
$regex	특정 정규 표현식과 맞는 Document를 선택한다
$text	문자열 검색의 기능을 수행한다
$where	자바스크립트로 알맞은 Document를 선택한다

정규표현식 연산자

쿼리 예시

```
{ <field>: { "$regex": 'pattern', "$options": '<options>' } }
```

options	설명
i	대소문자 무시
m	정규식에서 anchor(^) 를 사용할 때 값에 \n 이 있다면 무력화
x	정규식 안에 있는 whitespace를 모두 무시
s	dot (.) 사용 할 때 \n 을 포함해서 매치

정규표현식을 이용한 검색

정규표현식 예시

쿼리

```
articles.find( { "title" : { "$regex":'article0[1-2]' } } )
```

결과

```
{
  "_id" : ObjectId("56c0ab6c639be5292edab0c4"),
  "title" : "article01", "content" : "content01", "writer" : "Taker", "likes" : 0, "comments" : [ ]
}
{
  "_id" : ObjectId("56c0ab6c639be5292edab0c5"), "title" : "article02", "content" : "content02",
  "writer" : "Alpha", "likes" : 23, "comments" : [ { "name" : "Bravo", "message" : "Hey Man!" } ]
}
```

Text 연산자

쿼리 예시

```
{
  "$text":    { "$search": <string>, "$language": <string>,
      "$caseSensitive": <boolean>, "$diacriticSensitive": <boolean> }
}
```

options	설명
$search	검색할 내용
$language	선택적.검색하는 언어
$caseSensitive	선택적.False일 경우 대소문자 무시. False가 기본값
$diacriticSensitive	선택적. ğ와 g 같이 diacritical mark를 구분할지 선택. False가 기본값

컬렉션당 하나만 만들 수 있는 **문자열 인덱스**에서만 작동함

문자열 인덱스 설정

명령어 예시

```
collection.create_index([('field', pymongo.TEXT)], default_language='english')
```

컬렉션당 하나만 만들 수 있는 **문자열 인덱스**에서만 작동함

안타깝게도 한국어는 문자열 인덱스로 지원하지 않음

Text 연산자 예시: 한 단어

쿼리

```
articles.find( { "$text": { "$search": "coffee" } } )
```

결과

```
{ "_id" : 2, "subject" : "Coffee Shopping", "author" : "efg", "views" : 5 }
{ "_id" : 7, "subject" : "coffee and cream", "author" : "efg", "views" : 10 }
{ "_id" : 1, "subject" : "coffee", "author" : "xyz", "views" : 50 }
```

Text 연산자 예시: 여러 단어

쿼리

```
articles.find( { "$text": { "$search": "bake coffee cake" } } )
```

결과

```
{ "_id" : 2, "subject" : "Coffee Shopping", "author" : "efg", "views" : 5 }
{ "_id" : 7, "subject" : "coffee and cream", "author" : "efg", "views" : 10 }
{ "_id" : 1, "subject" : "coffee", "author" : "xyz", "views" : 50 }
{ "_id" : 3, "subject" : "Baking a cake", "author" : "abc", "views" : 90 }
{ "_id" : 4, "subject" : "baking", "author" : "xyz", "views" : 100 }
```

Text 연산자 예시: 구절을 검색

쿼리

```
articles.find( { "$text": { "$search": "\"coffee shop\"" } } )
```

결과

```
{ "_id" : 2, "subject" : "Coffee Shopping", "author" : "efg", "views" : 5 }
```

06

배열 연산자

도큐먼트에서 배열의 의미

쿼리

```
inventory.find({"tags":"school"})
```

결과

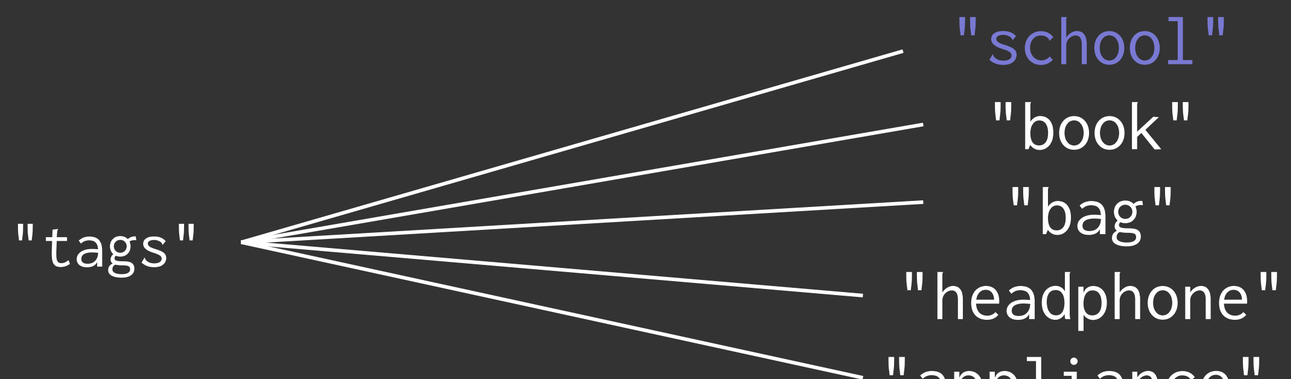
```
{ "_id" : ObjectId("5a11937eabccf6b418a70421"),
 "tags" : [ "school", "book", "bag", "headphone", "appliance" ] }
{ "_id" : ObjectId("5a119396abccf6b418a70422"),
 "tags" : [ "appliance", "school", "book" ] }
```

도큐먼트에서 배열의 의미

쿼리

```
inventory.find({"tags":"school"})
```

결과



배열은 **값이 여러 개**(순서 상관 있음)인 것으로 인식

연산자 소개

operator	설명
$all	순서와 상관없이 배열 안의 요소가 모두 포함되면 선택한다
$elemMatch	$elemMatch 조건과 맞는 배열 속 요소를 가진 Document를 선택한다.
$size	해당 배열의 크기가 같은 Document를 선택한다.

$all 연산자: 배열 속 모든 값을 포함하는 Document를 찾는다

쿼리

```
{ <field>: { "$all": [ <value1> , <value2> ... ] } }
```

```
items.find( { "tags": { "$all": [ "book", "appliance" ] } } )
```

결과

```
{
 "_id" : ObjectId("5a11937eabccf6b418a70421"),
 "tags" : [ "school", "book", "bag", "headphone", "appliance" ]
}
{
 "_id" : ObjectId("5a119396abccf6b418a70422"),
 "tags" : [ "appliance", "school", "book" ]
}
```

$elemMatch 연산자: 해당 field가 query들을 모두 만족하는 값을 갖는 도큐먼트를 검색

쿼리

```
{ <field>: { "$elemMatch": { <query1>, <query2>, ... } } }
```

```
score.find({ "results": { "$elemMatch": { "$gte": 80, "$lt": 85 } } })
```

결과

```
{ "_id" : 1, "results" : [ 82, 85, 88 ] }
```

$elemMatch를 이용하면 Query 안에 Query 넣어서 사용할 수 있다

쿼리

```
survey.find(
   {"results":{"$elemMatch":{"product":"xyz","score": { "$gte": 8 } } }}
)
```

결과

```
{ _id: 1, results: [ { product: "abc", score: 10 }, { product: "xyz", score: 5 } ] }
{ _id: 2, results: [ { product: "abc", score: 8 }, { product: "xyz", score: 7 } ] }
{ _id: 3, results: [ { product: "abc", score: 7 }, { product: "xyz", score: 8 } ] }
```

$size 연산자: 해당 field가 모든 query를 만족하는 값을 갖는 Document를 선택

쿼리

```
{ <field>: { "$size": <array size> } }
```

```
scores.find(    {"results": {"$size": 3} }    )
```

결과

```
{ "_id" : 1, "results" : [ 82, 85, 88 ] }
```

크레딧

/* elice */

코스 매니저
이재성

콘텐츠 제작자
정승호

강사
정승호

감수자
정승호

디자이너
강혜정

연락처

TEL
070-4633-2015

WEB
https://elice.io

E-MAIL
contact@elice.io