

이 MongoDB 쿼리들은 다양한 검색 및 데이터 조작 작업을 수행하는 방법을 보여줍니다. 아래는 각 쿼리의 설명입니다:

1. 문서 삽입

```
db.inventory.insertMany([
  { item: "journal", qty: 25, size: { h: 14, w: 21, uom: "cm" }, status: "A" },
  { item: "notebook", qty: 50, size: { h: 8.5, w: 11, uom: "in" }, status: "A" },
  { item: "paper", qty: 100, size: { h: 8.5, w: 11, uom: "in" }, status: "D" },
  { item: "planner", qty: 75, size: { h: 22.85, w: 30, uom: "cm" }, status: "D" },
  { item: "postcard", qty: 45, size: { h: 10, w: 15.25, uom: "cm" }, status: "A" }
]);
```

- 목적: inventory 컬렉션에 여러 문서를 한 번에 삽입합니다. 각 문서는 item, qty, size, status 필드를 포함하고 있습니다.

2. 컬렉션에서 모든 문서 선택

```
db.inventory.find( {} )
```

- 목적: inventory 컬렉션의 모든 문서를 반환합니다.

3. 동등성 조건 지정

```
db.inventory.find( { status: "D" } )
```

- 목적: status 필드가 "D" 인 문서만 반환합니다.

4. 쿼리 연산자를 사용하여 조건 지정

```
db.inventory.find( { status: { $in: [ "A", "D" ] } } )
```

- 목적: `status` 필드가 "A" 또는 "D" 인 문서만 반환합니다. `$in` 연산자는 배열에 포함된 값 중 하나와 일치하는 문서를 찾습니다.

5. AND 조건 지정

```
db.inventory.find( { status: "A", qty: { $lt: 30 } } )
```

- 목적: `status` 가 "A" 이고 `qty` 가 30 미만인 문서만 반환합니다.

6. OR 조건 지정

```
db.inventory.find( { $or: [ { status: "A" }, { qty: { $lt: 30 } } ] } )
```

- 목적: `status` 가 "A" 이거나 `qty` 가 30 미만인 문서만 반환합니다. `$or` 연산자는 배열 내의 조건 중 하나라도 일치하는 문서를 찾습니다.

7. AND와 OR 조건 지정

```
db.inventory.find( {  
  status: "A",  
  $or: [ { qty: { $lt: 30 } }, { item: /^p/ } ]  
} )
```

- 목적: `status` 가 "A" 이고, `qty` 가 30 미만이거나 `item` 이 "p" 로 시작하는 문서만 반환합니다. `$or` 내의 두 조건 중 하나가 맞으면 문서가 반환됩니다.

8. 점 표기법을 사용하는 중첩 필드 쿼리

```
db.inventory.find( { "size.uom": "in" } )
```

- 목적: `size.uom` 필드가 "in" 인 문서만 반환합니다.

```
db.inventory.find( { "size.h": { $lt: 15 } } )
```

- 목적: `size.h` 필드가 15 미만인 문서만 반환합니다.

9. 배열 쿼리

```
db.inventory.find( { tags: ["red", "blank"] } )
```

- 목적: tags 배열이 정확히 ["red", "blank"] 인 문서만 반환합니다.

```
db.inventory.find( { tags: { $all: ["red", "blank"] } } )
```

- 목적: tags 배열에 "red" 와 "blank" 가 모두 포함된 문서만 반환합니다. 배열에 특정 값이 포함되어야 함을 검사합니다.

10. 배열 요소에 대한 쿼리

```
db.inventory.find( { tags: "red" } )
```

- 목적: tags 배열에 "red" 가 포함된 문서만 반환합니다.

```
db.inventory.find( { dim_cm: { $gt: 25 } } )
```

- 목적: dim_cm 배열에 25보다 큰 값이 포함된 문서만 반환합니다.

11. 임베디드 문서 배열 쿼리

```
db.inventory.find( { "instock": { qty: 5, warehouse: "A" } } )
```

- 목적: instock 배열에 qty 가 5이고 warehouse 가 "A" 인 문서만 반환합니다.

```
db.inventory.find( { 'instock.qty': { $lte: 20 } } )
```

- 목적: instock 배열의 qty 필드가 20 이하인 문서만 반환합니다.

12. 문서 배열에 여러 조건 지정

```
db.inventory.find( { "instock": { $elemMatch: { qty: 5, warehouse: "A" } } } )
```

- 목적: instock 배열의 요소 중 qty 가 5이고 warehouse 가 "A" 인 문서만 반환합니다.

13. 프로젝트

```
db.inventory.find( { status: "A" }, { item: 1, status: 1 } )
```

- 목적: status 가 "A" 인 문서에서 item 과 status 필드만 반환합니다. _id 필드는 기본적으로 포함됩니다.

```
db.inventory.find( { status: "A" }, { item: 1, status: 1, _id: 0 } )
```

- 목적: status 가 "A" 인 문서에서 item 과 status 필드만 반환하며 _id 필드는 제외합니다.

14. 스위치와 문자열 결합

```
db.inventory.find(
  { },
  {
    _id: 0,
    item: 1,
    status: {
      $switch: {
        branches: [
          {
            case: { $eq: [ "$status", "A" ] },
            then: "Available"
          },
          {
            case: { $eq: [ "$status", "D" ] },
            then: "Discontinued"
          },
        ],
        default: "No status found"
      }
    },
    area: {
      $concat: [
        { $toString: { $multiply: [ "$size.h", "$size.w" ] } },
        " ",
        "$size.uom"
      ]
    },
    reportNumber: { $literal: 1 }
  }
)
```

- 목적: 문서에서 status 에 따라 status 를 변환하거나, size.h 와 size.w 를 곱해 면적을 계산하고, 문자열을 결합하여 area 를 생성하며, reportNumber 를 1로 설정합니다.

15. null 필드

```
db.inventory.insertMany([
  { _id: 1, item: null },
  { _id: 2 }
])
```

- 목적: item 이 null 인 문서와 item 필드가 없는 문서를 삽입합니다.

```
db.inventory.find( { item: null } )
```

- 목적: item 필드가 null 인 문서를 찾습니다.

```
db.inventory.find( { item: { $ne : null } } )
```

- 목적: item 필드가 null 이 아닌 문서를 찾습니다.

```
db.inventory.find( { item : { $exists: false } } )
```

- 목적: item 필드가 존재하지 않는 문서를 찾습니다.

이 쿼리들은 MongoDB에서 다양한 방식으로 데이터를 쿼리하고 조작하는 방법을 보여줍니다. 데이터의 필터링, 정렬, 조건부 검색, 배열 및 중첩 문서 쿼리, 프로젝션 등 여러 기능을 활용할 수 있습니다.