KB금융그룹 | 국민의 평생 금융파트너

2025년 상반기 K-디지털 트레이닝

FIND 명령 위주로!!1

쿼리 작성하기

[KB] IT's Your Life

KB 국민은행

1 전자상거래 쿼리

- **상품, 카테고리, 리뷰**

슬러그라는 필드명을 자주 쓴다.
시스템을 위한 키는 몽고디비에서는 _머시기

하나 찾기

```
product = db.products.findOne({'slug': 'wheel-barrow-9092'})
db.categories.findOne({'_id': product['main_cat_id']})
db.reviews.find({'product_id': product['_id']})
```

사용자가 인식할수있는 키
문자열로 표ㅕ현하는 필드명 슬러그

보통 카테고리-제품명-제조번호를
조합해서
사용자용식별키를 만들어 사용한다

product[슬러그명]를 통해서 사용자가 UUID를
직접 안써도 되게끔 활용한다

UUID는
지역정보(아이피맥주소)
+시간정보+
프로세스ID+순차번호
=>
전세계적으로
유일한 식별자
샤딩때문에 클라이언트
드라이버에서 발급함

sql에서는 auto_increment
와 대응됨
디비서버에서 발급

전자상거래 쿼리

skip, limit 그리고 쿼리 옵션

- skip(n): 건너뛸 데이터 개수
- limit(m): 추출할 데이터 개수
- sort({키1:1, 키2: -1, ... }): 정렬. 1(오름차순), -1(내림차순)

→ 연결 순서는 상관 없음

```
db.reviews.find({'product_id': product['_id']}).skip(0).limit(12)

product  = db.products.findOne({'slug': 'wheel-barrow-9092'})
db.reviews.find({'product_id': product['_id']})
          .sort({'helpful_votes': -1})
          .limit(12)
```

처음부터 12개 얻겠다는뜻
순서는 상관 없음 리미트 먼저와도돼
find뒤에는 순서 상관 없음.
내부에서 올바른 쿼리가 생성됨

동률시 콤마하고 또 키-값 적어주면 돼

전자상거래 쿼리

skip, limit 그리고 쿼리 옵션

- 페이지네이션

어떤 제품에 대한 리뷰를 얻고 싶어

```
page_number = 1
product  = db.products.findOne({'slug': 'wheel-barrow-9092'})
category = db.categories.findOne({'_id': product['main_cat_id']})
reviews_count = db.reviews.count({'product_id': product['_id']})
reviews = db.reviews.find({'product_id': product['_id']})
                    .skip((page_number - 1) * 12)
                    .limit(12)
                    .sort({'helpful_votes': -1})
```

특정 제품 행을 찾았고
해당 제품행에서
카네고리를 얻었고
해당 제품의 리뷰수를 찾았고

스킵과 리미트를 서서 페이지 내이션!

순서는 상관 없다!!!
정해진 순서가 내부에서 동작한다

정렬 먼저 되겠지 내부에선

전자상거래 쿼리

카테고리는 보통 트리 형태죠
트리형태 정보를 저장하는 기법 3가지
1. 노드가 있고 자식이 배열
2. 데이터가 있고 부모에 대한 참조를 가짐
3. 1번2번 둘다

- **상품 리스트 페이지**

```
page_number = 1
category = db.categories.findOne({'slug': 'outdoors'})
siblings = db.categories.find({'parent_id': category['_id']}) //  'outdoors' 상품과 같은 카테고리에 있는 상품 얻기
products = db.products.find({'category_id': category['_id']})
                      .skip((page_number - 1) * 12)
                      .limit(12)
                      .sort({'helpful_votes': -1})
```

이번 예시는 2번째 방법으로
되어있음

  - **최상위 상품 카테고리 얻기**

```
categories = db.categories.find({'parent_id': null})
```

MongoDB의 쿼리 언어

질의 조건과 셀렉터

AND 조건

db.users.find({'last_name': "Banker"})

db.users.find({'first_name': "Smith", birth_year: 1975})

범위

연산자	설명
$lt	~보다 작은
$gt	~보다 큰
$lte	~보다 작거나 같은
$gte	~보다 크거나 같은

db.users.find({'birth_year': {'$gte': 1985}, 'birth_year': {'$lte': 2015}})

db.users.find({'birth_year': {'$gte': 1985, '$lte': 2015}})

MongoDB의 쿼리 언어

질의 조건과 셀렉터

키 이름 형식은 같은데
밸류의 형식이 다를 수 있음

범위

- 테스트 데이터

```
db.items.insert({ "_id" : ObjectId("4caf82011b0978483ea29ada"), "value" : 97 })
db.items.insert({ "_id" : ObjectId("4caf82031b0978483ea29adb"), "value" : 98 })
db.items.insert({ "_id" : ObjectId("4caf82051b0978483ea29adc"), "value" : 99 })
db.items.insert({ "_id" : ObjectId("4caf820d1b0978483ea29ade"), "value" : "a" })
db.items.insert({ "_id" : ObjectId("4caf820f1b0978483ea29adf"), "value" : "b" })
db.items.insert({ "_id" : ObjectId("4caf82101b0978483ea29ae0"), "value" : "c" })
```

- 범위 질의

```
db.items.find({'value': {'$gte': 97}}) // 정수에 대한 검사이므로 정수 값 데이터만 출력

db.items.find({'value': {'$gte': "a"}}) // 문자열ㄹ에 대한 검사이므로 문자열 데이터만 출력
```

검사 대상의 값
타입에 따라
분리해서
범위 검색함

MongoDB의 쿼리 언어

집합 연산자

연산자	설명
$in	어떤 인수든 하나라도 참고 집합에 있는 경우 일치
$all	모든 인수가 참고 집합에 있고 배열이 포함된 도큐먼트에서 사용되는 경우 일치
$nin	그 어떤 인수도 참고 집합에 있지 않을 경우 일치

not in

```
db.products.find({
    'main_cat_id': {
      '$in': [
        ObjectId("6a5b1476238d3b4dd5000048"),
        ObjectId("6a5b1476238d3b4dd5000051"),
        ObjectId("6a5b1476238d3b4dd5000057")
      ]
    }
  })
```

MongoDB의 쿼리 언어

부울 연산자

연산자	설명
$ne	인수가 요소와 같지 않은 경우 일치
$not	일치 결과를 반전시킴(반대로 만듦)
$or	제공된 검색어 집합 중 하나라도 TRUE인 경우 일치
$nor	제공된 검색어 집합 중 그 어떤 것도 TRUE가 아닌 경우 일치
$and	제공된 검색어 집합이 모두 TRUE인 경우 일치
$exists	요소가 도큐먼트 안에 존재할 경우 일치

not equal

not or

특이
존재하느냐

MongoDB의 쿼리 언어

부울 연산자

```
db.products.find({
   '$or': [
    {'details.color': 'blue'},
    {'details.manufacturer': 'ACME'}
   ]
 })
```

파인드 응용
db.collection.find(
조건문서,
projection문서(출력 속성 고르기)(비어있음 다 출력)
//{username:1}
);

```
db.products.find({
   $and: [
    {
     tags: {$in: ['gift', 'holiday']}
    },
    {
     tags: {$in: ['gardening', 'landscaping']}
    }
   ]
 })
```

_id처럼 기본키는 항상 나온다.

projection문서에서 {username:0}처럼 하면
해당 속성 빼고 출력이라는 뜻

출력 지정은 추가만 할거면 추가하고 제거는 제거만된다.

출력 추가하면서 제거하는 지정은 안된다
_id만 빼고

보통 서비스에서는 주력DB로는 sql 보조DB로는 NOSQL을 쓴다. rdb로 성능 안나오는 구간
캐시(좋아요, 싫어요, 해쉬태그,최근 상품보기)이런거는 nosql(몽고,레디스)로

MongoDB의 쿼리 언어

부울 연산자

해당 필드가 있냐 없냐에 따라

```
db.products.find({'details.color': {$exists: false}})
```

디테일 속성에 color 하부속성이 있냐 없냐

```
db.products.find({'details.color': {$exists: true}})
```

이때 null을 쓰면 안돼

null이라는 값을 가지는 필드가 있잖아

특정 필드의 데이터가 null이잖아

MongoDB의 쿼리 언어

배열

배열에 대한 연산자

연산자	설명
$elemMatch	제공된 모든 조건이 동일한 하위 도큐먼트에 있는 경우 일치
$size	배열 하위 도큐먼트의 크기가 제공된 리터럴 값과 같으면 일치

```
{
    _id: ObjectId("4c4b1476238d3b4dd5003981"),
    slug: "wheel-barrow-9092",
    sku: "9092",
    tags: ["tools", "equipment", "soil"]
}
```

db.products.find({tags: "soil"}) soil이라는 태그가 있냐 일종의 in연산

db.products.find({'tags.0': "soil"})

.인덱스 표시

반드시 따옴표 쓰시고

MongoDB의 쿼리 언어

배열

```
{
    _id: ObjectId("4c4b1476238d3b4dd5000001")
    username: "kbanker",
    addresses: [
        {
            name: "home",
            street: "588 5th Street",
            city: "Brooklyn",
            state: "NY",
            zip: 11215
        },
        {
            name: "work",
            street: "1 E. 23rd Street",
            city: "New York",
            state: "NY",
            zip: 10010
        },
    ]
}
```

배열안에 객체들이 들어가있는 형태

```
db.users.find({'addresses.0.state': "NY"})
```

첫번째 state필드가

```
db.users.find({'addresses.state': "NY"})
```

배열 요소중에 state필드가

```
db.users.find({'addresses.name': 'home', 'addresses.state': 'NY'})
```

이거 약간 or처럼 등장함.두개를 찾는걸로

```
db.users.find({
    'addresses': {
      '$elemMatch': {
        'name': 'home',
        'state': 'NY'
      }
    }
  })
```

내부의 필드에 관해서 and 연산하고 싶으면 $elemMatch를 써야해

배열 내부의 필드를 통해서 검색하려함. 내부 문서에 접근