Find Query

You can query documents in a MongoDB collection using the .find() method. This method returns a **cursor** (an object you can iterate over to get results).

Sample Collection Setup

- Before running queries, here's a sample dataset you can insert into your database:

```
await db.products.insertMany([
 {
  name: Wireless Mouse,
  category: Electronics,
  price: 25.50,
  stock: 200,
  tags: [accessories, wireless],
  details: { brand: ErgoGear, color: black },
  is_available: true
 },
 {
  name: Mechanical Keyboard,
  category: Electronics,
  price: 80.00,
  stock: 75,
  tags: [accessories, gaming, mechanical],
  details: { brand: ClickyKeys, layout: US },
  is_available: true
 },
 {
  name: Organic Coffee Beans,
  category: Groceries,
  price: 15.00,
  stock: 150,

  tags: [coffee, organic, beverages],
  details: { origin: Colombia, weight_kg: 0.5 },
  is_available: true
 },
 {
  name: Smart Speaker Echo,
  category: Smart Home,
  price: 99.99,
  stock: 120,
  tags: [smart-home, voice-assistant],
  details: { brand: HomeAI, assistant: Echo },
  is_available: false // Currently out of stock/unavailable
 },
 {
  name: Yoga Mat Pro,
  category: Fitness,
  price: 40.00,
  stock: 80,
  tags: [fitness, yoga, accessories],
```

```
    details: { material: TPE, thickness_mm: 6 },
    reviews: [
      { user: Charlie, rating: 5, comment: Very comfortable! }
    ],
    is_available: true
  }
]);
```

Query Examples

1. Get all products: db.products.find({})

Returns every product in the collection without any filters.

Result: [


```
        {"name":"Wireless Mouse", "category":"Electronics", "price":25.5, "stock":200, ...},
        {"name":"Mechanical Keyboard", "category":"Electronics", "price":80, "stock":75, ...},
        {"name":"Organic Coffee Beans", "category":"Groceries", "price":15, "stock":150, ...},
        {"name":"Smart Speaker Echo", "category":"Smart Home", "price":99.99, "stock":120, ...},
        {"name":"Yoga Mat Pro", "category":"Fitness", "price":40, "stock":80, ...}
]
```


2. Find all available products

```
db.products.find({ is_available: true })
```

Returns only products where is_available is true.

Result: [


```
        {"name":"Wireless Mouse", "category":"Electronics", "price":25.5, "stock":200, ...},
        {"name":"Mechanical Keyboard", "category":"Electronics", "price":80, "stock":75, ...},
        {"name":"Organic Coffee Beans", "category":"Groceries", "price":15, "stock":150, ...},
        {"name":"Yoga Mat Pro", "category":"Fitness", "price":40, "stock":80, ...}
]
```


3. Find all products in the "Electronics" category

```
db.products.find({ category: "Electronics" })
```

Returns all products whose category field is exactly "Electronics".

Result: [


```
        {"name":"Wireless Mouse", "category":"Electronics", "price":25.5, "stock":200, ...},
        {"name":"Mechanical Keyboard", "category":"Electronics", "price":80, "stock":75, ...}
]
```

4. Find products with price greater than 50

```
db.products.find({ price: { $gt: 50 } })
```

Returns products where the price field is greater than 50.

Result: [

{"name":"Mechanical Keyboard", "category":"Electronics", "price":80, "stock":75, ...},
{"name":"Smart Speaker Echo", "category":"Smart Home", "price":99.99, "stock":120, ...}

]

5. Find products with the tag "accessories"

```
db.products.find({ tags: "accessories" })
```

Returns products whose tags array contains the value "accessories".

Result: [

{"name":"Wireless Mouse", "category":"Electronics", "price":25.5, "stock":200, ...},
{"name":"Mechanical Keyboard", "category":"Electronics", "price":80, "stock":75, ...},
{"name":"Yoga Mat Pro", "category":"Fitness", "price":40, "stock":80, ...}

]

6. Get only the name and price of all products

```
db.products.find({}, { name: 1, price: 1, _id: 0 })
```

Returns all products but only includes the name and price fields (excludes _id).

Result: [


```
{ "name": "Wireless Mouse", "price": 25.5 },
{ "name": "Mechanical Keyboard", "price": 80 },
{ "name": "Organic Coffee Beans", "price": 15 },
{ "name": "Smart Speaker Echo", "price": 99.99 },
{ "name": "Yoga Mat Pro", "price": 40 }
```

]

7. Find a product by brand inside **details**

```
db.products.find({ "details.brand": "ErgoGear" })
```

Returns products where the nested field details.brand equals "ErgoGear".

Result: [{

"name":"Wireless Mouse",
"category":"Electronics",
"price":25.5,
"stock":200,
"tags":["accessories","wireless"],
"details":{"brand":"ErgoGear","color":"black"},
"is_available":true
}]

8. Find products that have reviews

```
db.products.find({ reviews: { $exists: true } })
```

Returns products that contain the reviews field (regardless of its content).

Result: [{

"name":"Yoga Mat Pro",
"category":"Fitness",
"price":40,
"stock":80,
"tags":["fitness","yoga","accessories"],
"details":{"material":"TPE","thickness_mm":6},
"reviews":[{"user":"Charlie","rating":5,"comment":"Very comfortable!"}],
"is_available":true
}]