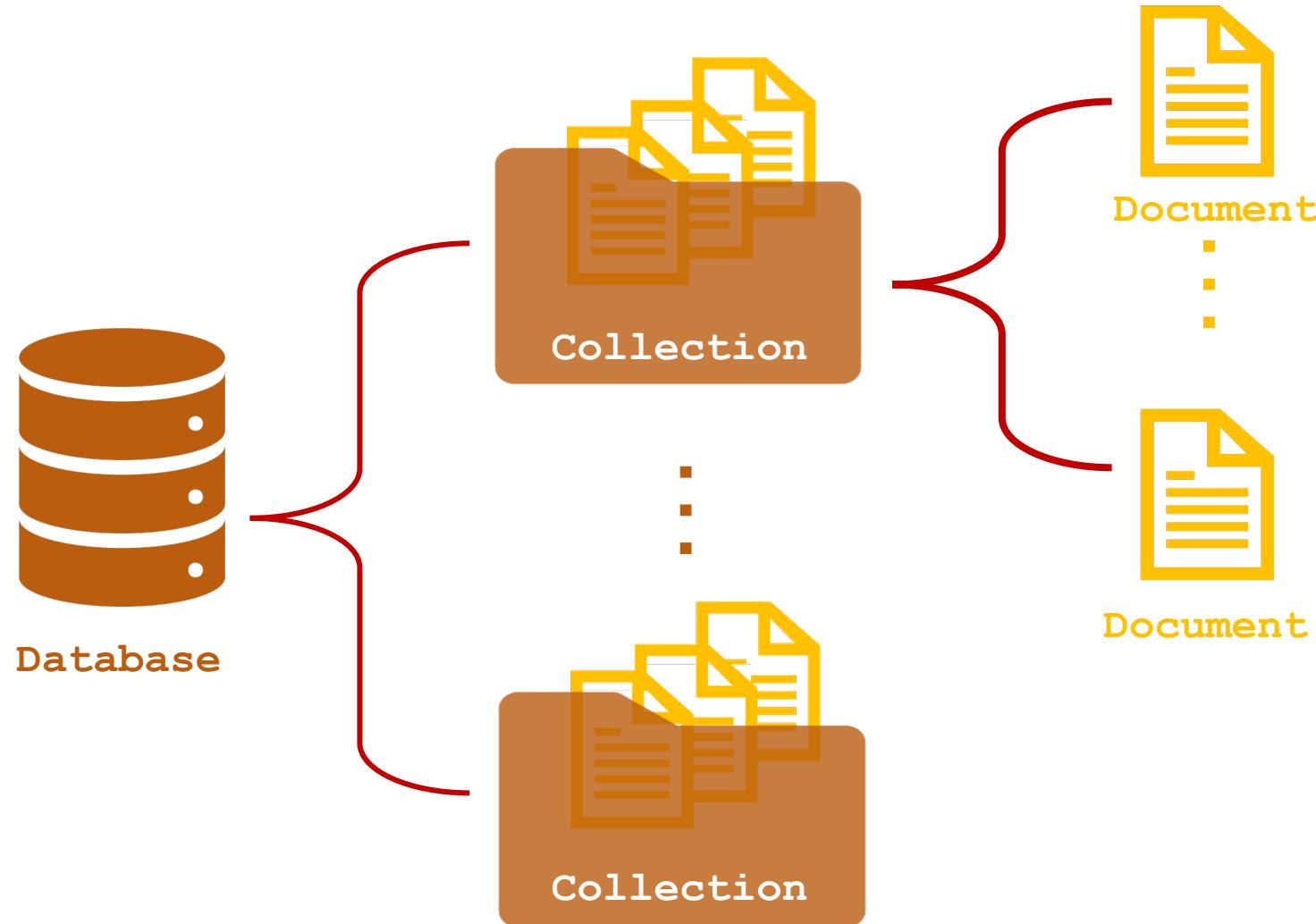




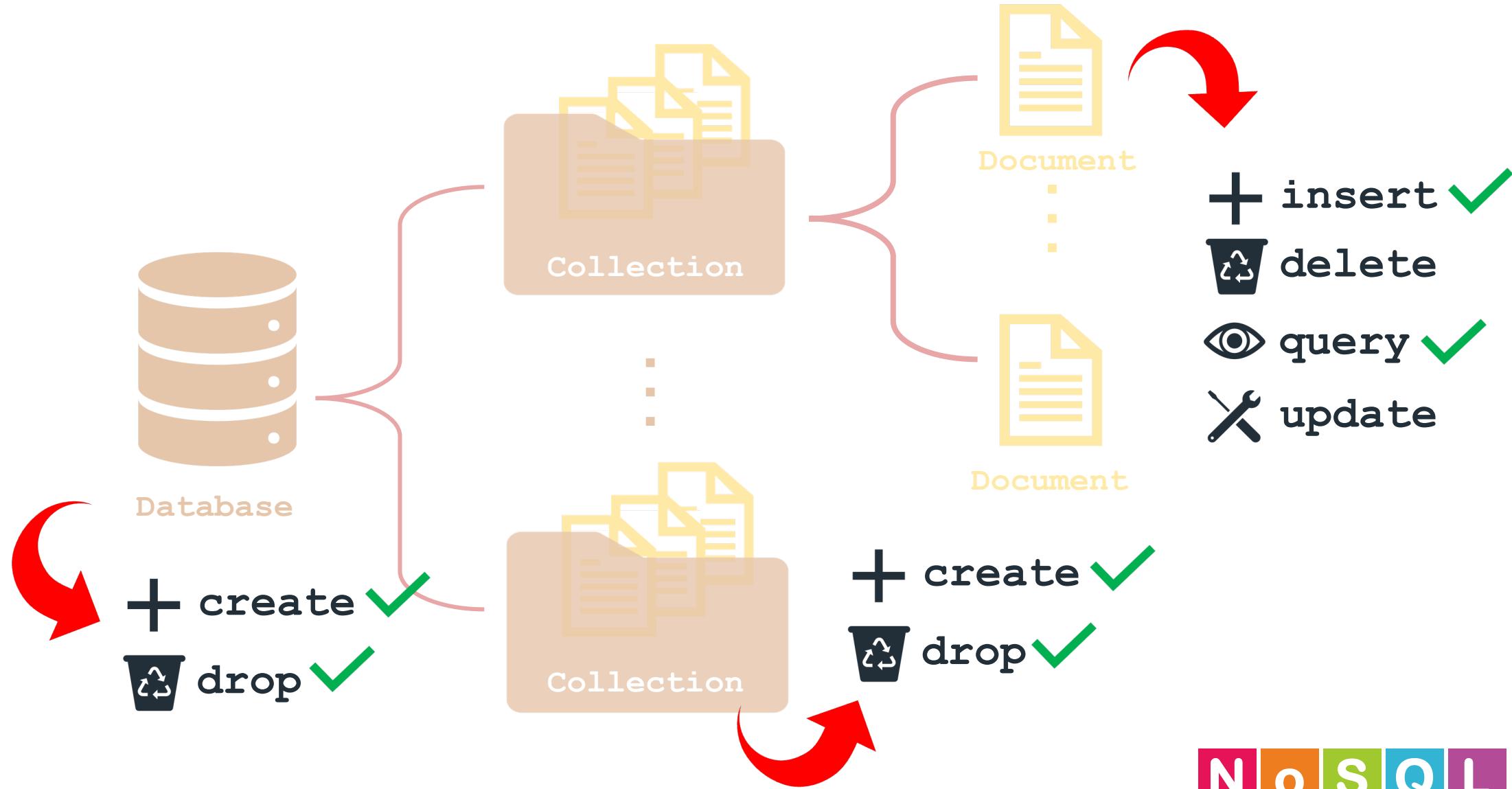
# Chapter 14: MongoDB Implementation (Part 2)

By Asst.Prof. Dr. Taravichet Titijaroonroj

# โครงสร้าง MongoDB



# คำสั่งและตัวดำเนินการของ MongoDB





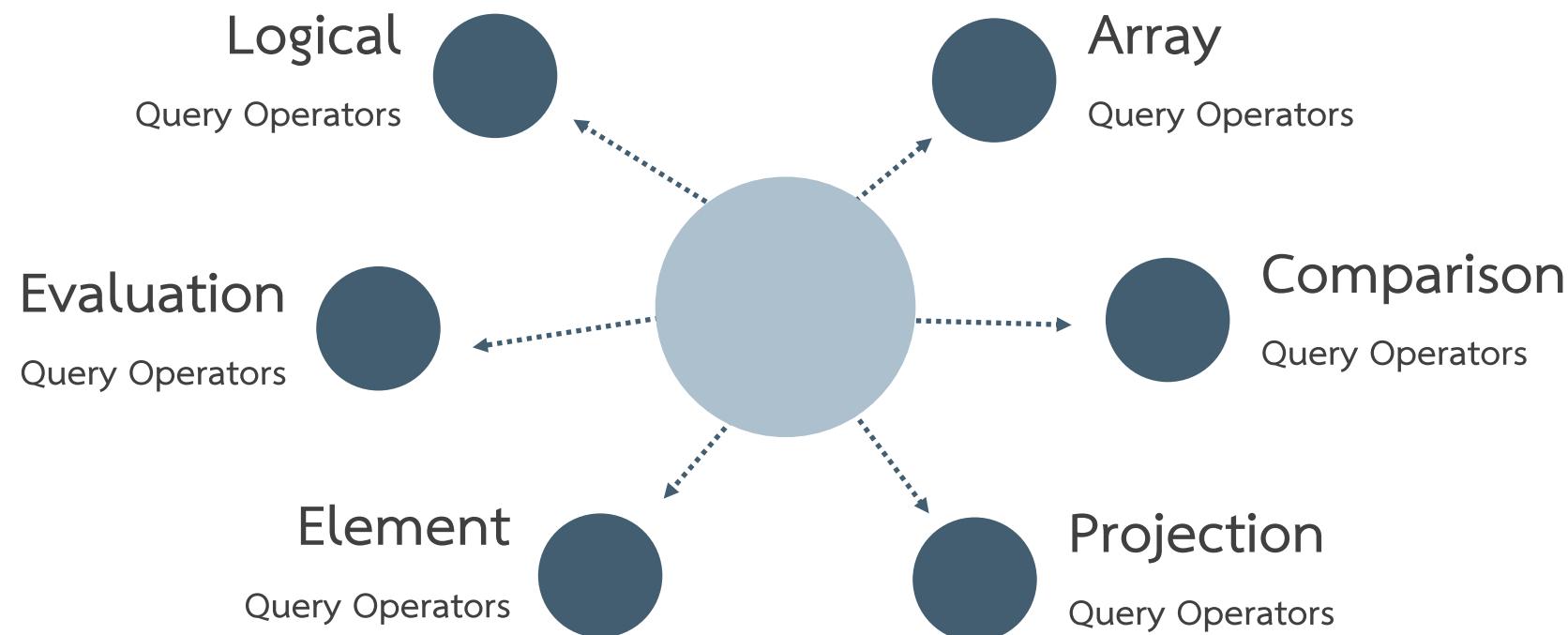
# การเรียกดู Document แบบมีเงื่อนไข

# การเรียกดู Document แบบมีเงื่อนไข



```
db.COLLECTION_NAME.find(condition, projection)
```

การเรียกดูข้อมูลแบบมีเงื่อนไข คือ การเลือก document มาแสดงแค่บางรายการโดยอาศัยตัวดำเนินการ เพื่อสร้างเงื่อนไขดังต่อไปนี้



# ข้อมูล Wizard Collection



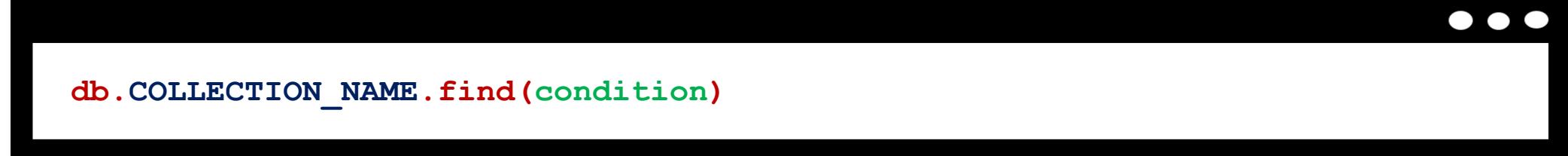
```
{sex:"m", name: "Severus Snape", school: "Hogwarts", house: "Slytherin", pets: [], money: 35000, position: "teacher"}  
{sex:"f", name: "Minerva McGonagall", school: "Hogwarts", house: "Gryffindor", pets: [], money: 50000, position: "teacher"}  
{sex:"f", name: "Sybill Trelawney", school: "Hogwarts", house: "Ravenclaw", pets: [], money: 15000, position: "teacher"}  
{sex:"m", name: "Quirinus Quirrell", school: "Hogwarts", house: "Ravenclaw", pets: [], money: 5000, position: "teacher"}  
{sex:"m", name: "Albus Dumbledore", school: "Hogwarts", house: "Gryffindor", pets: ["Phoenix"], money: 95000, position: "teacher"}  
{sex:"m", name: "Remus Lupin", school: "Hogwarts", house: "Gryffindor", pets: [], money: 52000, position: "teacher"}  
{sex:"m", name: "Igor Karkaroff", school: "Durmstrang", house: "", pets: [], money: 35000, position: "teacher"}  
{sex:"f", name: "Pomona Sprout", school: "Hogwarts", house: "Hufflepuff", pets: [], money: 25000, position: "teacher"}  
{sex:"f", name: "Madame Maxime", school: "Beauxbatons", house: "", pets: [], money: 45000, position: "teacher"}  
  
{sex:"m", name: "Harry Potter", school: "Hogwarts", house: "Gryffindor", pets: ["cat", "bird"], money: 9000, position: "student"}  
{sex:"f", name: "Cho Chang", school: "Hogwarts", house: "Ravenclaw", pets: ["cat", "bird"], money: 7500, position: "student"}  
{sex:"f", name: "Hermione Granger", school: "Hogwarts", house: "Gryffindor", pets: ["cat"], money: 9000, position: "student"}  
{sex:"m", name: "Neville Longbottom", school: "Hogwarts", house: "Gryffindor", pets: ["toad"], money: 9000, position: "student"}  
{sex:"m", name: "Newt Scamander", school: "Hogwarts", house: "Hufflepuff", pets: [], money: 7900, position: "student"}  
{sex:"m", name: "Cedric Diggory", school: "Hogwarts", house: "Hufflepuff", pets: [], money: 8900, position: "student"}  
{sex:"m", name: "Viktor Krum", school: "Durmstrang", house: "", pets: [], money: 8600, position: "student"}  
{sex:"f", name: "Fleur Delacour", school: "Beauxbatons", house: "", pets: [], money: 9000, position: "student"}
```

# ข้อมูล Account Collection

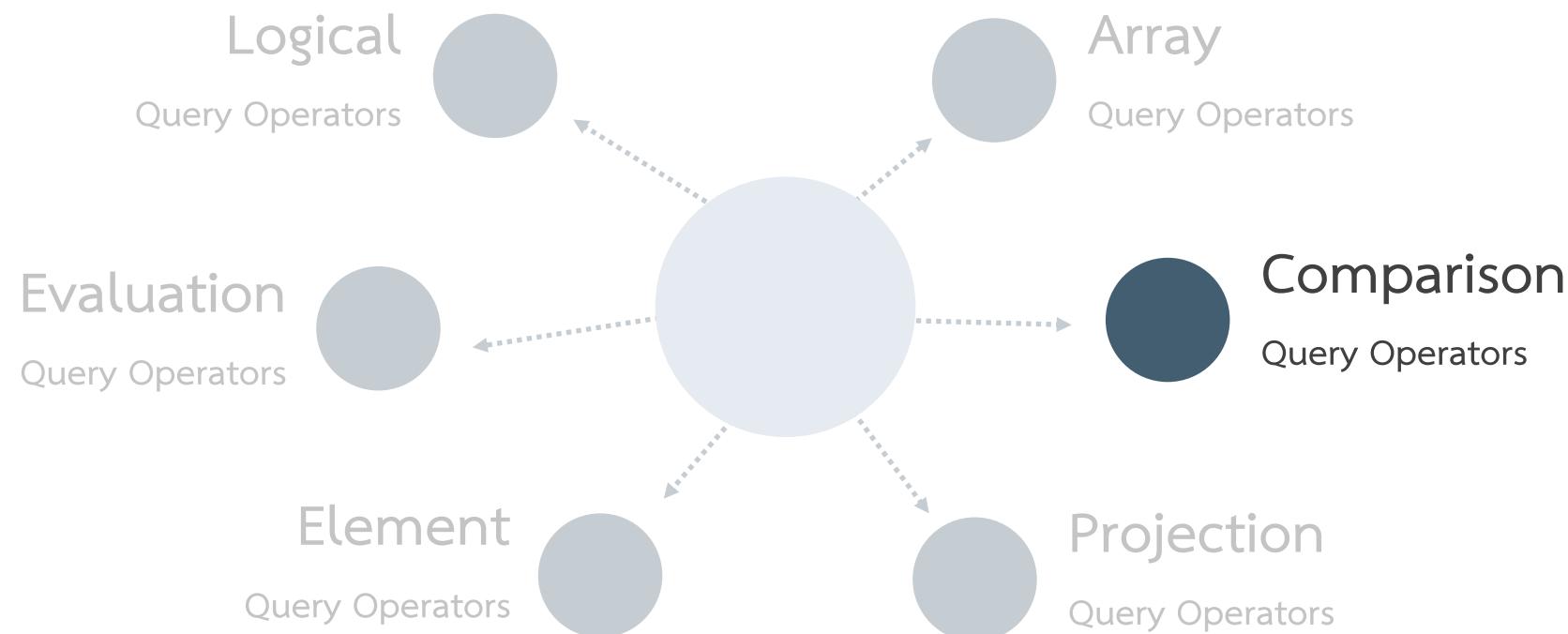


```
{account:"Bank", balance: 35000, creditcard: "1111-2222-3333-4444"}  
{account:"Adam", balance: 5000, creditcard: "1111-2222-3333-4444"}  
{account:"Alex", balance: 3000}  
{account:"Antony", balance: 15000, point:50}  
{account:"Apart", balance: 2000}
```

# การเรียกดู Document แบบมีเงื่อนไข



การเรียกดูข้อมูลแบบมีเงื่อนไข คือ การเลือก document มาแสดงแค่บางรายการโดยอาศัยตัวดำเนินการ เพื่อสร้างเงื่อนไขดังต่อไปนี้



# Comparison Query Operators



ตัวดำเนินการในกลุ่มนี้ทำหน้าที่เป็นตัวเปรียบเทียบปริมาณมากกว่าหรือน้อยกว่า และเปรียบเทียบความแตกต่างหรือความเหมือนกัน

ตัวดำเนินการ	ไวยากรณ์	ตัวอย่าง	ความหมาย
เท่ากับ	{<key>:<value>} {<key>:{ <b>\$eq</b> :<value>}} }	db.myCol.find({ "name": "Peter" }).pretty() db.myCol.find({ "name": { <b>\$eq</b> : "Peter" } }).pretty()	where name = 'Peter'
น้อยกว่า	{<key>:{ <b>\$lt</b> :<value>}} }	db.myCol.find({ "money": { <b>\$lt</b> :50 } }).pretty()	where money < 50
น้อยกว่าเท่ากับ	{<key>:{ <b>\$lte</b> :<value>}} }	db.myCol.find({ "money": { <b>\$lte</b> :50 } }).pretty()	where money <= 50
มากกว่า	{<key>:{ <b>\$gt</b> :<value>}} }	db.myCol.find({ "money": { <b>\$gt</b> :50 } }).pretty()	where money > 50
มากกว่าเท่ากับ	{<key>:{ <b>\$gte</b> :<value>}} }	db.myCol.find({ "money": { <b>\$gte</b> :50 } }).pretty()	where money >= 50
ไม่เท่ากับ	{<key>:{ <b>\$ne</b> :<value>}} }	db.myCol.find({ "money": { <b>\$ne</b> :50 } }).pretty()	where money != 50

## ตัวอย่างที่ 1

```
>>> db.Wizard.find({"school": "Durmstrang"}).pretty()
{
    "_id" : ObjectId("5d199a29c745b527b7abdb08"),
    "sex" : "m",
    "name" : "Igor Karkaroff",
    "school" : "Durmstrang",
    "house" : "",
    "pets" : [ ],
    "money" : 35000,
    "position" : "teacher"
}
{
    "_id" : ObjectId("5d199a29c745b527b7abdb11"),
    "sex" : "m",
    "name" : "Viktor Krum",
    "school" : "Durmstrang",
    "house" : "",
    "pets" : [ ],
    "money" : 8600,
    "position" : "student"
}
```

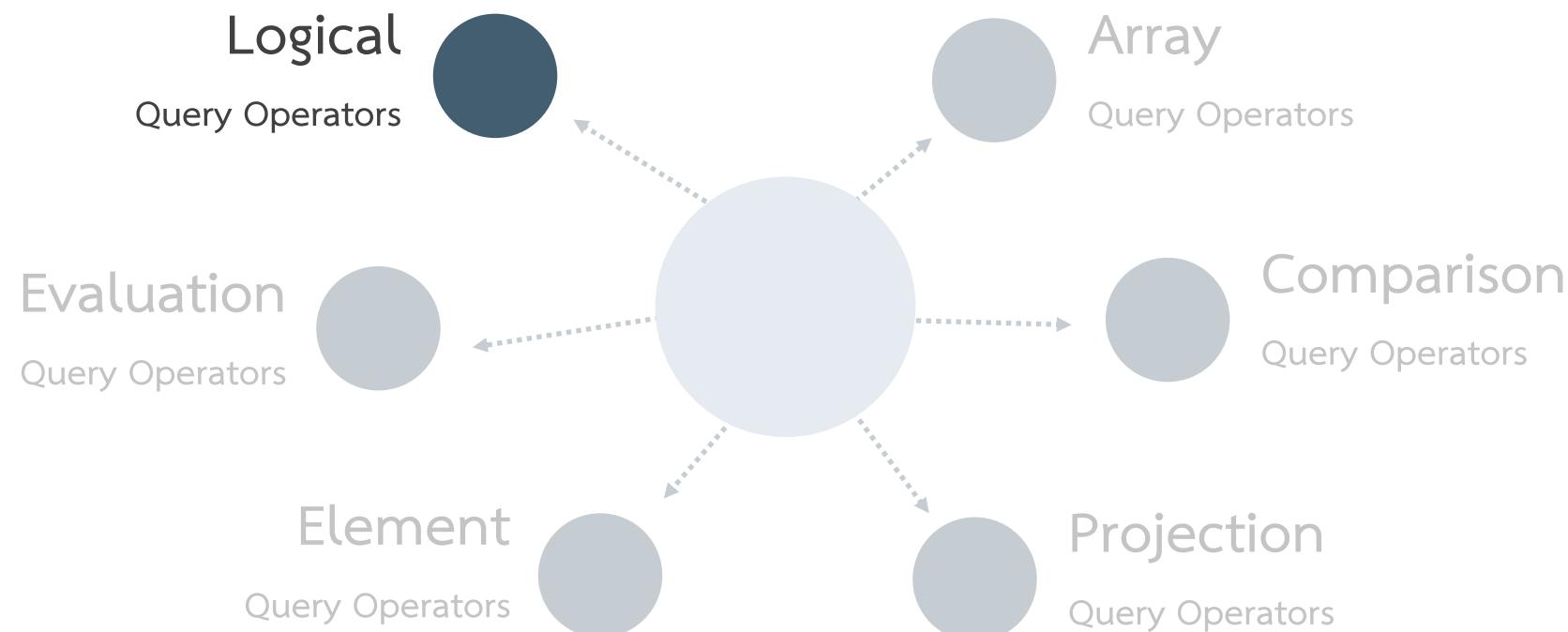
## ตัวอย่างที่ 2

```
>>> db.Wizard.find({"money": {"$gte: 90000}}).pretty()
{
    "_id" : ObjectId("5d199a29c745b527b7abdb06"),
    "sex" : "m",
    "name" : "Albus Dumbledore",
    "school" : "Hogwarts",
    "house" : "Gryffindor",
    "pets" : [
        "Phoenix"
    ],
    "money" : 95000,
    "position" : "teacher"
}
```

# การเรียกดู Document แบบมีเงื่อนไข



การเรียกดูข้อมูลแบบมีเงื่อนไข คือ การเลือก document มาแสดงแค่บางรายการโดยอาศัยตัวดำเนินการ เพื่อสร้างเงื่อนไขดังต่อไปนี้



# Logical Query Operators



ตัวดำเนินการในกลุ่มนี้ทำหน้าที่เป็นตัวเชื่อมระหว่างเงื่อนไขได้ ๆ ได้แก่ and, or, not และ nor

**AND** หมายความว่า **ทุก** นิพจน์ (expression) ในวงเล็บก้ามปู “[.....]” ต้องมีผลลัพธ์เป็น **จริง** ทั้งหมด

```
$and: [{expression 1}, {expression 2}, ... , {expression N}, ]
```

**OR** หมายความว่า **มี** นิพจน์ **อย่างน้อย** หนึ่งอันในวงเล็บก้ามปู “[.....]” ที่มีผลลัพธ์เป็น **จริง**

```
$or: [{expression 1}, {expression 2}, ... , {expression N}, ]
```

## ตัวอย่างที่ 1

```
> db.Wizard.find({$or: [{"school": "Durmstrang"}, {"school" : "Beauxbatons"}]}).pretty()

{
  "_id" : ObjectId("5d199a29c745b527b7abdb08"),
  "sex" : "m",
  "name" : "Igor Karkaroff",
  "school" : "Durmstrang",
  "house" : "",
  "pets" : [ ],
  "money" : 35000,
  "position" : "teacher"
}
{
  "_id" : ObjectId("5d199a29c745b527b7abdb0a"),
  "sex" : "f",
  "name" : "Madame Maxime",
  "school" : "Beauxbatons",
  "house" : "",
  "pets" : [ ],
  "money" : 45000,
  "position" : "teacher"
}
{
  "_id" : ObjectId("5d199a29c745b527b7abdb11"),
  "sex" : "m",
  "name" : "Viktor Krum",
  "school" : "Durmstrang",
  "house" : "",
  "pets" : [ ],
  "money" : 8600,
  "position" : "student"
}
{
  "_id" : ObjectId("5d199a29c745b527b7abdb12"),
  "sex" : "f",
  "name" : "Fleur Delacour",
  "school" : "Beauxbatons",
  "house" : "",
  "pets" : [ ],
  "money" : 9000,
  "position" : "student"
}
```

## ตัวอย่างที่ 2

```
>>> db.Wizard.find( { $and: [ {"house": "Hufflepuff"}, {"money" : {$gt:8000}} ] } ).pretty()  
{  
    "_id" : ObjectId("5d199a29c745b527b7abdb09") ,  
    "sex" : "f" ,  
    "name" : "Pomona Sprout" ,  
    "school" : "Hogwarts" ,  
    "house" : "Hufflepuff" ,  
    "pets" : [ ] ,  
    "money" : 25000 ,  
    "position" : "teacher"  
}  
{  
    "_id" : ObjectId("5d199a29c745b527b7abdb10") ,  
    "sex" : "m" ,  
    "name" : "Cedric Diggory" ,  
    "school" : "Hogwarts" ,  
    "house" : "Hufflepuff" ,  
    "pets" : [ ] ,  
    "money" : 8900 ,  
    "position" : "student"  
}
```

```

>>> db.Wizard.find({
    $and: [
        {"school": "Hogwarts"} ,
        {"money: {$gt:50000}} ,
        {$or: [
            {"house" : "Hufflepuff"} ,
            {"house" : "Gryffindor"}
        ]}
    ]
}).pretty()
    
```

ผลลัพธ์

```

{
    "_id" : ObjectId("5d199a29c745b527b7abdb06") , "sex" : "m" , "name" : "Albus Dumbledore" , "school" : "Hogwarts" , "house" : "Gryffindor" , "pets" : [ "Phoenix" ] , "money" : 95000 , "position" : "teacher" }

{
    "_id" : ObjectId("5d199a29c745b527b7abdb07") , "sex" : "m" , "name" : "Remus Lupin" , "school" : "Hogwarts" , "house" : "Gryffindor" , "pets" : [ ] , "money" : 52000 , "position" : "teacher" }
    
```

# Logical Query Operators



**NOT** หมายความว่าผลลัพธ์ของนิพจน์ในวงเล็บปีกกา “`{...}`” จะตรงกันข้ามกับ **operator-expression** เสมอ

```
{$not: {operator-expression}}
```

ตัวอย่าง

```
db.inventory.find( { price: { $not: { $gt: 1.99 } } } )
```

จากคำสั่งข้างต้นจะคืนค่าเป็นทุก document ที่

- แออททริบิวท์ `price` มีค่าน้อยกว่าหรือเท่ากับ 1.99 **หรือ**
- **ไม่มี**แออททริบิวท์ `price`

# Logical Query Operators



## NOR หมายความว่าทุกนิพจน์ในวงเล็บก้ามปุ “[.....]” ต้องเป็นเท็จทั้งหมด



```
$nor: [{expression 1}, {expression 2}, ... , {expression N} ]
```

### ตัวอย่าง

```
db.inventory.find( { $nor: [ { price: 1.99 }, { sale: true } ] } )
```

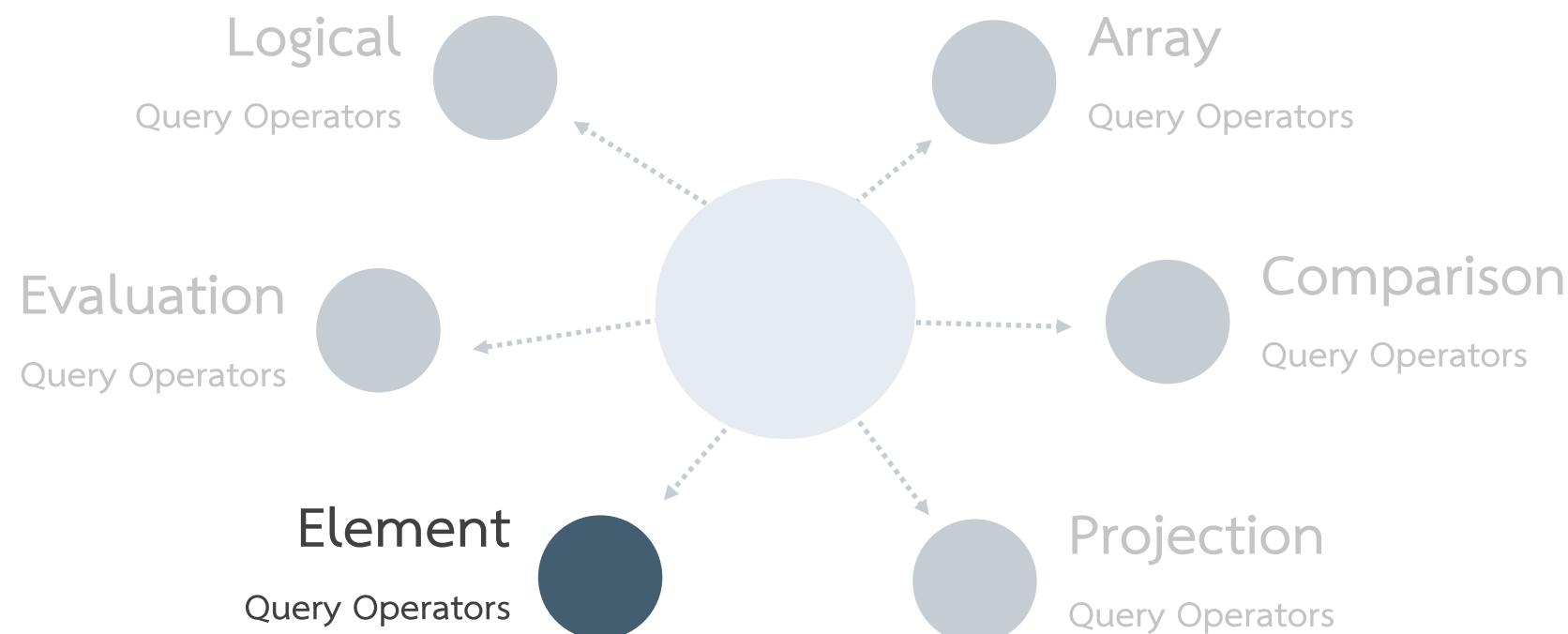
จากคำสั่งข้างต้นจะคืนค่าเป็นทุก document ที่

- แออททริบิวท์ price มีค่าไม่เท่ากับ 1.99 และแออททริบิวท์ sale มีค่าไม่เท่ากับ true **หรือ**
- แออททริบิวท์ price มีค่าไม่เท่ากับ 1.99 และ**ไม่มี**แออททริบิวท์ sale **หรือ**
- **ไม่มี**แออททริบิวท์ price และแออททริบิวท์ sale มีค่าไม่เท่ากับ true **หรือ**
- **ไม่มี**แออททริบิวท์ price และ**ไม่มี**แออททริบิวท์ sale

# การเรียกดู Document แบบมีเงื่อนไข



การเรียกดูข้อมูลแบบมีเงื่อนไข คือ การเลือก document มาแสดงแค่บางรายการโดยอาศัยตัวดำเนินการ เพื่อสร้างเงื่อนไขดังต่อไปนี้



# Element Query Operators



ตัวดำเนินการในกลุ่มนี้ทำหน้าที่สำหรับตรวจสอบคีย์ (key, field, attribute)

## EXISTS ใช้เพื่อตรวจสอบว่ามีคีย์ดังกล่าวหรือไม่

```
{"key" : { $exists : true } }
```

## TYPE ใช้เพื่อตรวจสอบว่าคีย์ดังกล่าวมีชนิดข้อมูลสอดคล้องกับที่กำหนดไว้หรือไม่

```
{"key" : { $type : n } }
```

# Element Query Operators



Type Description	Type value
Double	1
String	2
Object	3
Array	4
Binary data	5
Object id	7
Boolean	8
Date	9
Null	10

Type Description	Type value
Regular expression	11
JavaScript code	13
Symbol	14
JavaScript code with scope	15
32-bit integer	16
Timestamp	17
64-bit integer	18
Min / Max key	255 / 127

```
> db.Account.find({point:{ $exists: true}})  
{ "_id" : ObjectId("5d940a046644ad7f2532185e") , "account" : "Antony", "balance" :  
15000,  
"point" : 50 }  
> db.Account.find({point:{ $exists: false}})  
{ "_id" : ObjectId("5d940a046644ad7f2532185b") , "account" : "Bank", "balance" :  
35000,  
"creditcard" : "1111-2222-3333-4444"}  
{ "_id" : ObjectId("5d940a046644ad7f2532185c") , "account" : "Adam", "balance" :  
5000,  
"creditcard" : "1111-2222-3333-4444" }  
{ "_id" : ObjectId("5d940a046644ad7f2532185d") , "account" : "Alex", "balance" :  
3000 }  
{ "_id" : ObjectId("5d940a046644ad7f2532185f") , "account" : "Apart", "balance" :  
2000 }
```

# ข้อมูล Wizard Collection



```
{sex:"m", name: "Severus Snape", school: "Hogwarts", house: "Slytherin",pets: [], money: 35000, position: "teacher"}  
{sex:"f", name: "Minerva McGonagall", school: "Hogwarts", house: "Gryffindor",pets: [], money: 50000, position: "teacher"}  
{sex:"f", name: "Sybill Trelawney", school: "Hogwarts", house: "Ravenclaw",pets: [], money: 15000, position: "teacher"}  
{sex:"m", name: "Quirinus Quirrell", school: "Hogwarts", house: "Ravenclaw",pets: [], money: 5000, position: "teacher"}  
{sex:"m", name: "Albus Dumbledore", school: "Hogwarts", house: "Gryffindor", pets: ["Phoenix"], money: 95000, position: "teacher"}  
{sex:"m", name: "Remus Lupin", school: "Hogwarts", house: "Gryffindor",pets: [], money: 52000, position: "teacher"}  
{sex:"m", name: "Igor Karkaroff", school: "Durmstrang", house: "",pets: [], money: 35000, position: "teacher"}  
{sex:"f", name: "Pomona Sprout", school: "Hogwarts", house: "Hufflepuff", pets: [], money: 25000, position: "teacher"}  
{sex:"f", name: "Madame Maxime", school: "Beauxbatons", house: "",pets: [], money: 45000, position: "teacher"}  
  
{sex:"m", name: "Harry Potter", school: "Hogwarts", house: "Gryffindor", pets: ["cat", "bird"], money: 9000, position: "student"}  
{sex:"f", name: "Cho Chang", school: "Hogwarts", house: "Ravenclaw", pets: ["cat", "bird"], money: 7500, position: "student"}  
{sex:"f", name: "Hermione Granger", school: "Hogwarts", house: "Gryffindor", pets: ["cat"], money: 9000, position: "student"}  
{sex:"m", name: "Neville Longbottom", school: "Hogwarts", house: "Gryffindor", pets: ["toad"], money: 9000, position: "student"}  
{sex:"m", name: "Newt Scamander", school: "Hogwarts", house: "Hufflepuff",pets: [], money: 7900, position: "student"}  
{sex:"m", name: "Cedric Diggory", school: "Hogwarts", house: "Hufflepuff",pets: [], money: 8900, position: "student"}  
{sex:"m", name: "Viktor Krum", school: "Durmstrang", house: "",pets: [], money: 8600, position: "student"}  
{sex:"f", name: "Fleur Delacour", school: "Beauxbatons", house: "",pets: [], money: 9000, position: "student"}
```

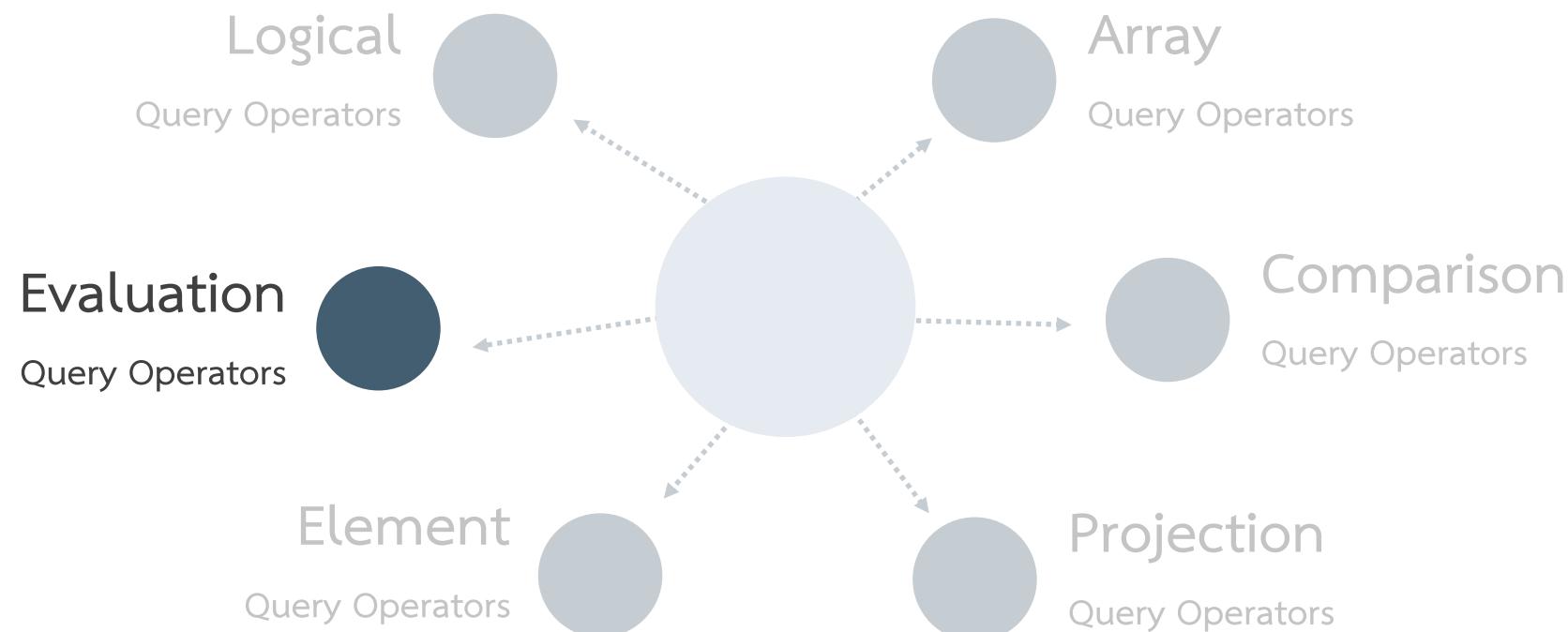
```
> db.Account.find({point:{ $type: 2}})

> db.Account.find({point:{ $type: 1}})
{ "_id" : ObjectId("5d940a046644ad7f2532185e") , "account" : "Antony" , "balance" :
15000 , "point" : 50 }
```

# การเรียกดู Document แบบมีเงื่อนไข



การเรียกดูข้อมูลแบบมีเงื่อนไข คือ การเลือก document มาแสดงแค่บางรายการโดยอาศัยตัวดำเนินการ เพื่อสร้างเงื่อนไขดังต่อไปนี้



# Evaluation Query Operators



ตัวดำเนินการในกลุ่มนี้ทำหน้าที่ช่วยตรวจสอบหรือค้นหาบางอย่าง

**MOD** ใช้เพื่อตรวจสอบว่ามีคีย์ดังกล่าวหารด้วย **M** และเหลือเศษ **N** หรือไม่

```
{key : { $mod : [M, N] } }
```

**EXPR** ใช้เพื่อเปรียบเทียบค่าระหว่างคีย์ใน document อันเดียวกัน

```
{ $expr: {expression} }
```

```
// Inventory Data
{ "_id" : 1, "item" : "pencil", "amount" : 0 }
{ "_id" : 2, "item" : "pen", "amount" : 5 }
{ "_id" : 3, "item" : "ruler", "amount" : 12 }
```

```
// Code and its result
> db.inventory.find( {amount : { $mod: [ 4, 0 ] } } )

{ "_id" : 1, "item" : "pencil", "amount" : 0 }
{ "_id" : 3, "item" : "ruler", "amount" : 12 }
```

```
// Monthly Budget Data
{ "_id" : 1, "category" : "food", "budget": 400, "spent": 450 }
{ "_id" : 2, "category" : "drinks", "budget": 100, "spent": 150 }
{ "_id" : 3, "category" : "clothes", "budget": 100, "spent": 50 }
{ "_id" : 4, "category" : "misc", "budget": 500, "spent": 300 }
{ "_id" : 5, "category" : "travel", "budget": 200, "spent": 650 }
```

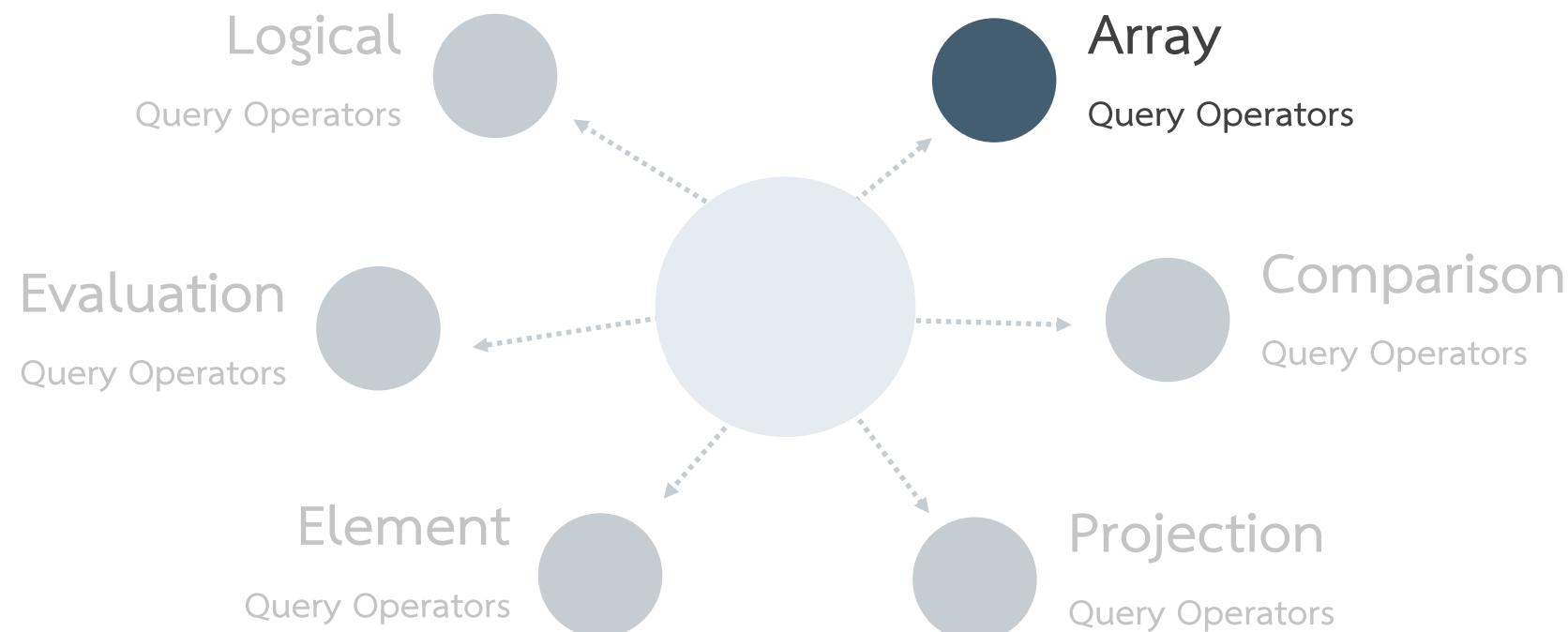
```
// Code and its result
> db.monthlyBudget.find( { $expr: { $gt: [ "$spent" , "$budget" ] } } )

{ "_id" : 1, "category" : "food", "budget" : 400, "spent" : 450 }
{ "_id" : 2, "category" : "drinks", "budget" : 100, "spent" : 150 }
{ "_id" : 5, "category" : "travel", "budget" : 200, "spent" : 650 }
```

# การเรียกดู Document แบบมีเงื่อนไข



การเรียกดูข้อมูลแบบมีเงื่อนไข คือ การเลือก document มาแสดงแค่บางรายการโดยอาศัยตัวดำเนินการ เพื่อสร้างเงื่อนไขดังต่อไปนี้



# Array Query Operators



ตัวดำเนินการในกลุ่มนี้ใช้ดำเนินการเกี่ยวกับการตรวจสอบจำนวนสมาชิกหรือค่ากับข้อมูลชนิดอาร์เรย์

**ALL** ใช้เพื่อตรวจสอบว่าสมาชิกในอาร์เรย์ดังกล่าวมีค่าเท่ากับ `<value1>` , `<value2>` ... ทั้งหมดหรือไม่

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

**SIZE** ใช้เพื่อตรวจสอบว่าจำนวนสมาชิกในอาร์เรย์ดังกล่าวมีค่าเท่ากับ `N` หรือไม่

```
{ key: { $size: N } }
```

จะได้รับแค่ Document ที่มีทั้ง “cat” และ “bird” เท่านั้น ไม่ครบไม่ได้ แต่เกินได้

## ตัวอย่างที่ 1



```
> db.Wizard.find({pets:{$all:[["cat","bird"]]}})  
{ "_id" : ObjectId("5d940245c21d5d01e7f00494") , "sex" : "m", "name" : "Harry  
Potter", "school" : "Hogwarts", "house" : "Gryffindor", "pets" : [ "cat", "bird"  
] , "money" : 9000, "position" : "student"  
{ "_id" : ObjectId("5d9414d47e80ab466b8f6f94") , "sex" : "f", "name" : "Cho Chang",  
"school" : "Hogwarts", "house" : "Ravenclaw", "pets" : [ "cat", "bird" ] , "money"  
: 7500, "position" : "student" }
```

```
> db.Wizard.find({$and:[{pets:["cat","bird"]}]})  
{ "_id" : ObjectId("5d940245c21d5d01e7f00494") , "sex" : "m", "name" : "Harry  
Potter", "school" : "Hogwarts", "house" : "Gryffindor", "pets" : [ "cat", "bird"  
] , "money" : 9000, "position" : "student"  
{ "_id" : ObjectId("5d9414d47e80ab466b8f6f94") , "sex" : "f", "name" : "Cho Chang",  
"school" : "Hogwarts", "house" : "Ravenclaw", "pets" : [ "cat", "bird" ] , "money"  
: 7500, "position" : "student" }
```

## ตัวอย่างที่ 2

```
> db.Wizard.find({pets:{$size:1}})  
{ "_id" : ObjectId("5d940245c21d5d01e7f0048f") , "sex" : "m" , "name" : "Albus  
Dumbledore" , "school" : "Hogwarts" , "house" : "Gryffindor" , "pets" : [ "Phoenix"  
] , "money" : 95000 , "position" : "teacher" }  
{ "_id" : ObjectId("5d940245c21d5d01e7f00495") , "sex" : "f" , "name" : "Cho Chang" ,  
"school" : "Hogwarts" , "house" : "Ravenclaw" , "pets" : [ "bird" ] , "money" : 7500 ,  
"position" : "student" }  
{ "_id" : ObjectId("5d940245c21d5d01e7f00496") , "sex" : "f" , "name" : "Hermione  
Granger" , "school" : "Hogwarts" , "house" : "Gryffindor" , "pets" : [ "cat" ] ,  
"money" : 9000 , "position" : "student" }  
{ "_id" : ObjectId("5d940245c21d5d01e7f00497") , "sex" : "m" , "name" : "Neville  
Longbottom" , "school" : "Hogwarts" , "house" : "Gryffindor" , "pets" : [ "toad" ] ,  
"money" : 9000 , "position" : "student" }
```

# Array Query Operators



## elemMatch

ใช้เพื่อตรวจสอบว่า document ได้ที่สมาชิกในอาร์เรย์มีสมาชิกสอดคล้องกับ *<expression>*

```
{ key: { $elemMatch: { <expression 1>, <expression 2>, ... } } }
```

```
// Scores Data
{ _id: 1, results: [ 82, 85, 88 ] }
{ _id: 2, results: [ 75, 88, 89 ] }
```

```
// Code and its result
db.scores.find({ results:{$elemMatch:{ $gte: 80, $lt: 85 }}})
{ "_id" : 1, "results" : [ 82, 85, 88 ] }
```

*// The following query matches only those documents where the results array contains at least one element that is both greater than or equal to 80 and is less than 85.*

## ตัวอย่างที่ 4 :Array of Embedded Documents

```
// Survey Data
{ _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 } ] }
```

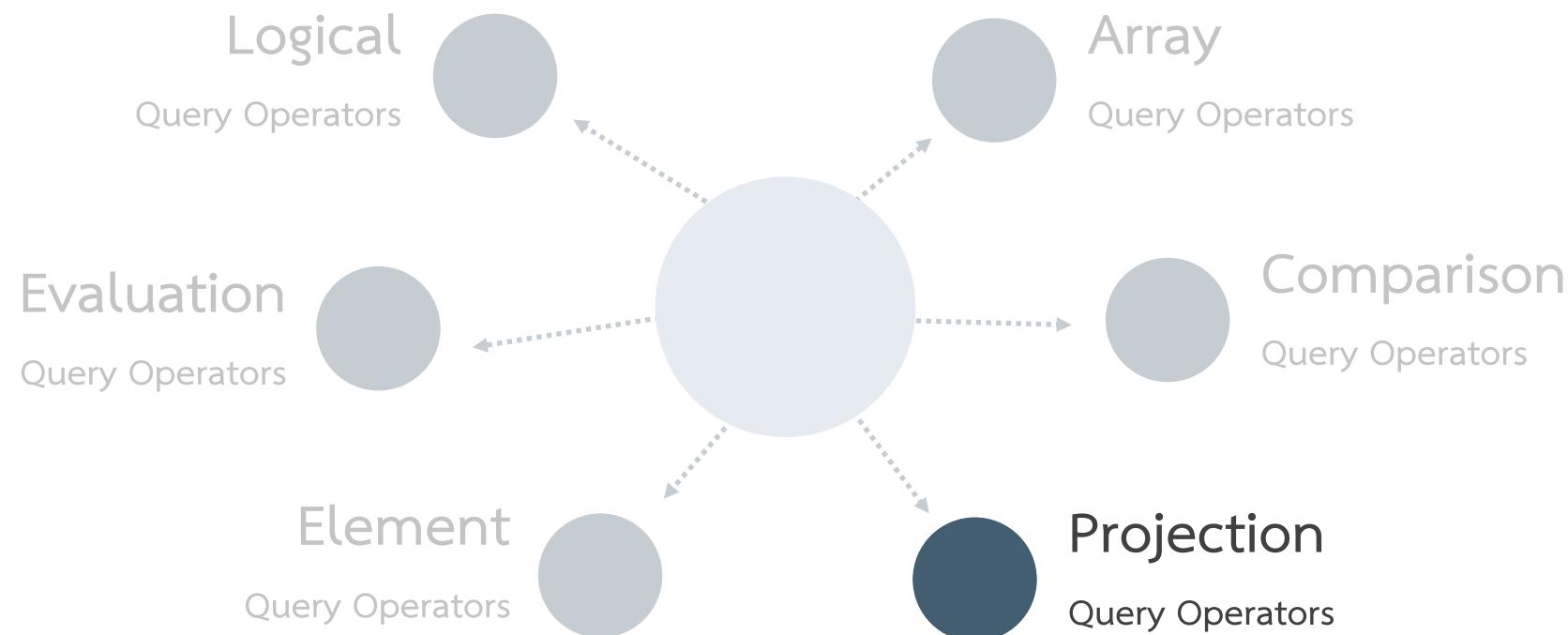
```
// Code and its result
db.survey.find({results:{ $elemMatch:{ product: "xyz", score: { $gte: 8 } }}})

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

# การเรียกดู Document แบบมีเงื่อนไข



การเรียกดูข้อมูลแบบมีเงื่อนไข คือ การเลือก document มาแสดงแค่บางรายการโดยอาศัยตัวดำเนินการ เพื่อสร้างเงื่อนไขดังต่อไปนี้



# การเรียกดู Document แบบมีเงื่อนไข



นอกจากนี้ ยังมีคำสั่ง **findOne()** สำหรับค้นหา Document แรกที่พบว่าสอดคล้องกับเงื่อนไขที่กำหนด กรณีมี Document ที่สอดคล้องกับมีมากกว่า 1 จะแสดงเฉพาะ Document แรกของผลลัพธ์ที่ถูกจัดเรียงเรียบร้อยแล้วเท่านั้น

```
db.COLLECTION_NAME.findOne(condition, projection)
```

อย่างไรก็ตาม การค้นหาด้วยคำสั่ง **findOne()** ทำให้การค้นหามีความรวดเร็วมากกว่าการใช้งานคำสั่ง **find()** เนื่องจากคำสั่ง **findOne()** จะจัดเรียงข้อมูลก่อนจากนั้นจะเปรียบเทียบจนเจอ Document ที่สอดคล้องกับเงื่อนไขแรก และจะหยุด ซึ่งแตกต่างจากคำสั่ง **find()** ที่จะค้นหาทุก Document ใน Collection จนครบ