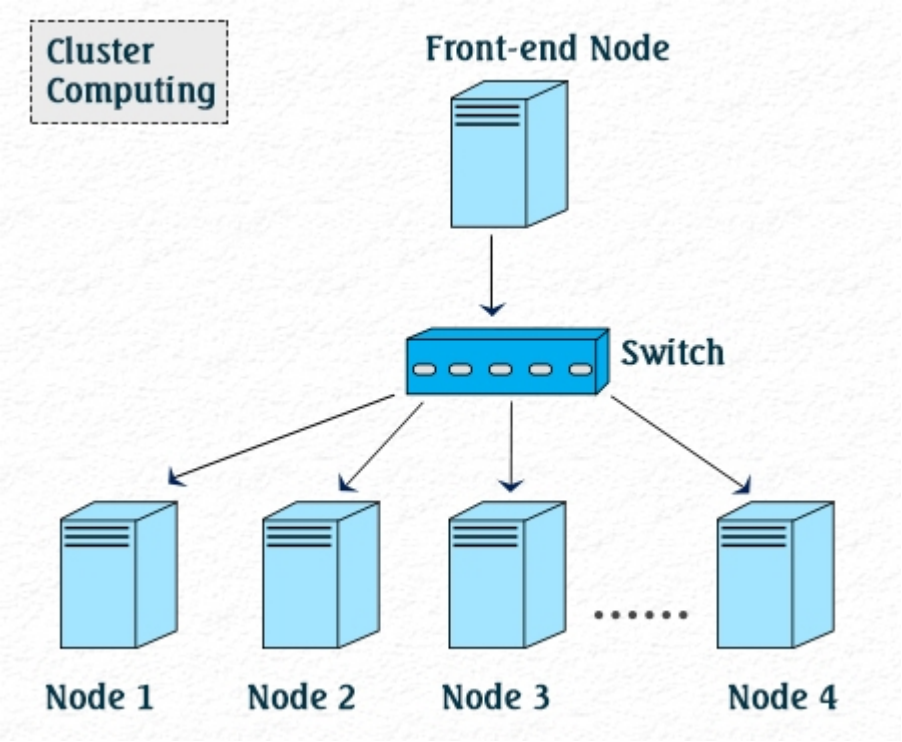
Exercise 8 Hadoop

มีกี่ข้อ?

5. ข้อเอง

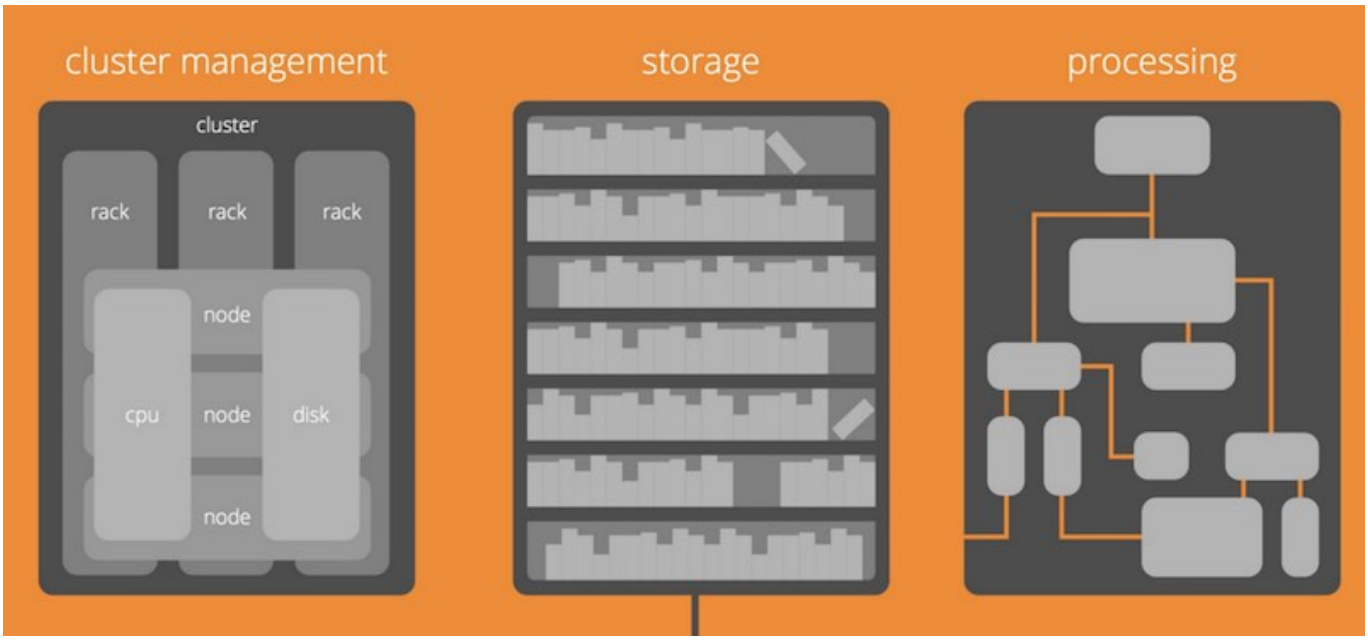
Hadoop คืออะไร?

Hadoop คือ ซอฟท์แวร์ประเภท Open Source ที่จัดทำขึ้นเพื่อเป็นแพลตฟอร์มในการจัดเก็บข้อมูล ซึ่งมีกรอบการทำงานเพื่อใช้ในการจัดเก็บข้อมูลและประมวลผลข้อมูลที่มีขนาดใหญ่มากๆ ที่เราเรียกกันว่า Big Data ซึ่งเจ้าตัว Hadoop เนี่ยก็สามารถปรับขยาย ยืดหยุ่น เพื่อรองรับข้อมูลที่มีจำนวนมากมายมหาศาลได้ ทั้งนี้ก็เพราะมันมีกระบวนการประมวลผลที่แข็งแกร่งมากซึ่งเป็นผลมาจากการประมวลผลข้อมูลแบบกระจายผ่านเครื่องคอมพิวเตอร์ที่ถูกจัดอยู่ในรูปแบบ Cluster อันนำไปสู่ความสามารถในการรองรับข้อมูลที่ไม่จำกัดแถมยังมีความน่าเชื่อถือสูงอีกด้วย



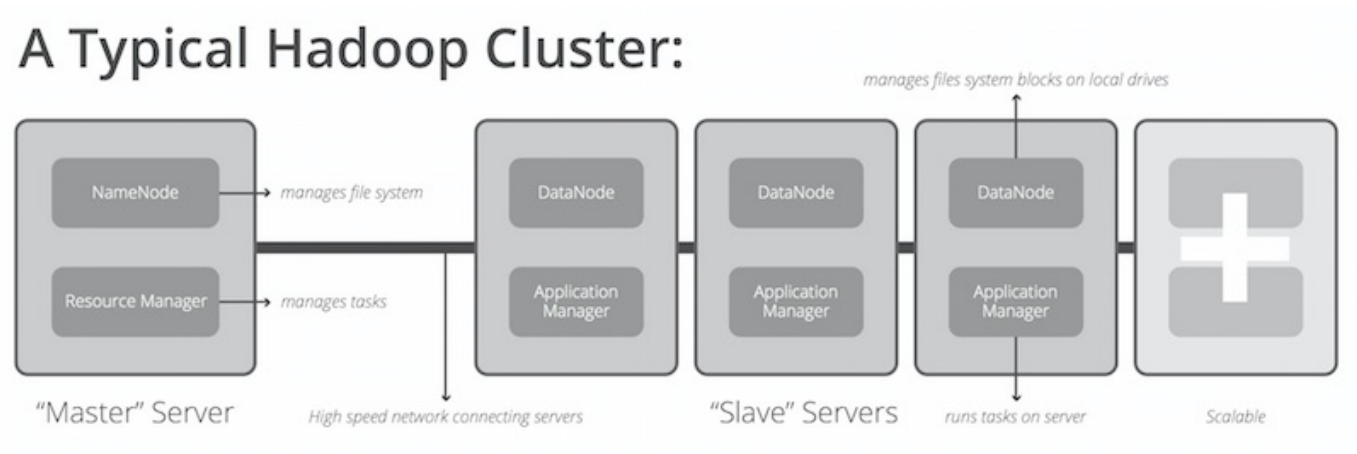
ส่วนประกอบของ Hadoop

```
1. Cluster management
2. Storage
3. Processing
```

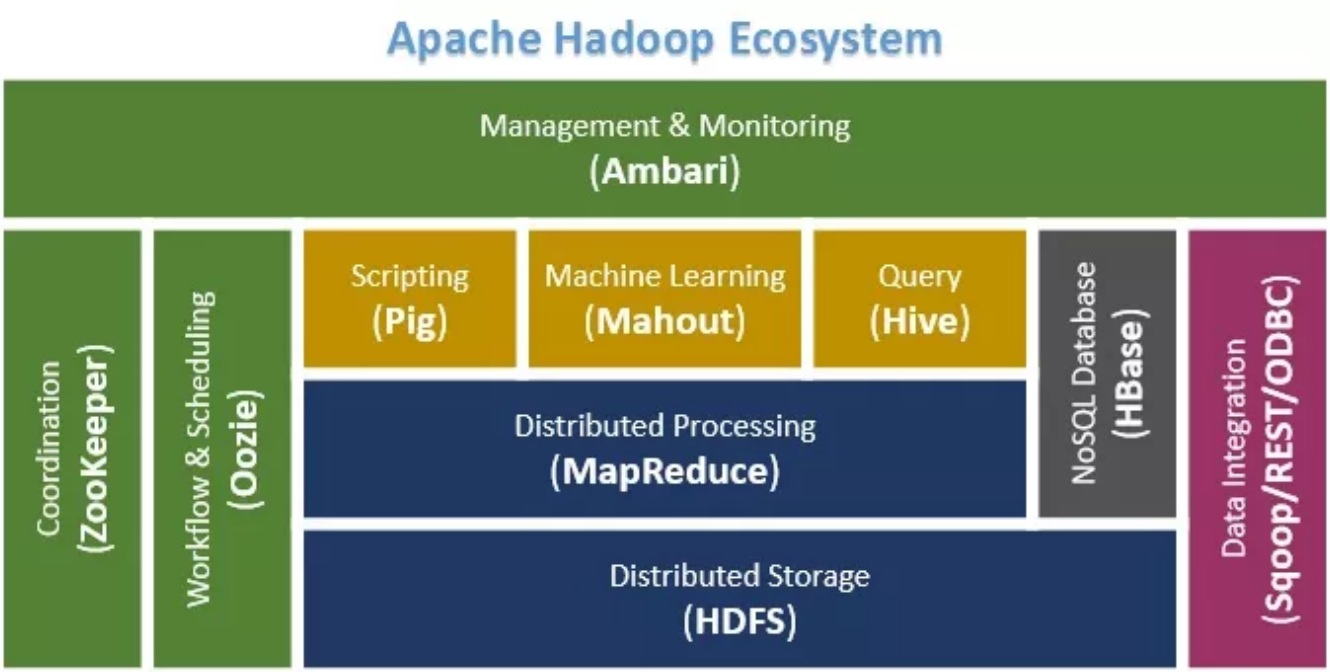


โครงสร้างของ Hadoop

```
1. Master Server
2. Slave Server
```

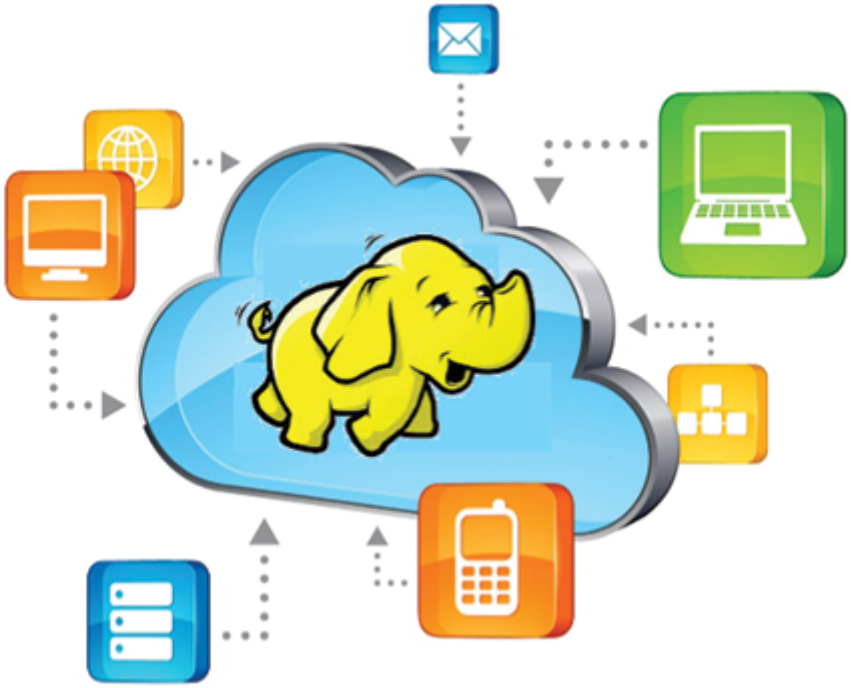


Ecosystem **ของ** Hadoop



ข้อดีของ Hadoop

1. **ความสามารถในการรองรับการจัดเก็บข้อมูลขนาดใหญ่มากๆทุกประเภทได้อย่างรวดเร็ว – ด้วยปริมาณข้อมูลในปัจจุบันที่เพิ่มขึ้นอย่างต่อเนื่อง โดยเฉพาะจากแหล่งอย่างโซเชี่ยลมีเดีย และ** Internet of Things (IoT) **คุณสมบัติข้อนี้ของ** Hadoop **จึงสำคัญมาก**
2. **พลังแห่งการประมวลผล – ด้วยรูปแบบการประมวลผลที่รวดเร็วจากการทำงานแบบ** Cluster **จึงทำให้** Hadoop **กลายเป็นแพลตฟอร์มที่เป็นที่นิยมอย่างกว้างขวางในปัจจุบัน**
3. **มีระบบรองรับความผิดพลาด – ด้วยการทำงานแบบ** Cluster **เมื่อ** node **ใด** node **หนึ่งพังลง งานที่มีการทำอยู่ในระบบจะถูกส่งไปยัง** node **อื่นทันทีเพื่อให้เกิดความต่อเนื่อง รวมถึงระบบเองยังมีการทำข้อมูลสำรองเก็บไว้แบบอัตโนมัติหลายชุดอีกด้วย**
4. **ความยืดหยุ่นในการใช้งาน –** Hadoop **แตกต่างจากระบบฐานข้อมูลดั้งเดิม ที่ต้องมีการแยกประเภทของข้อมูลคร่าวๆก่อนการจัดเก็บ สำหรับ** Hadoop **เราจะเก็บข้อมูลประเภทไหนก็ได้ มากเท่าไหร่ก็ได้ทันที โดยไม่ต้องมีการแยกประเภทล่วงหน้าแถมยังสามารถเลือกได้อีกว่าจะเอาไปใช้งานด้านใด**
5. **ต้นทุนต่ำ –** เพราะเป็นแพลตฟอร์มแบบ Open Source **จึงสามารถนำมาใช้งานได้ฟรี!**
6. **การจัดเก็บข้อมูลไปได้เรื่อยๆตามแต่เราต้องการ**



เครื่องมือที่สามารถทำงานร่วมกันกับ Hadoop ได้

```
1. Spark
2. NOSQL
3. Pig
4. Samza
5. Hive
6. Cascading
7. etc.
```

ว่ากันด้วย NoSQL (MongDB นะ ไม่ใช่ MangoDB)

MongoDB คือ โปรแกรมช่วยจัดการฐานข้อมูลในรูปแบบ NoSQL (Not only structured query language) ข้อมูลของเราไม่ได้ถูกจัดเก็บในรูปแบบเพียงแค่ SQL หรือรูปแบบที่เราเข้าใจได้ง่ายๆ เป็นตาราง มี row มี column ชัดเจน แต่ยังมีข้อมูลที่เป็นในรูปแบบ ของ “Video”, “Image”, “Voice”, และอื่นๆอีกมากมาย และด้วยปริมาณที่ ข้อมูลต่างๆมีมากขึ้นแบบ Exponential (แบบ f(x) = x^n) ทำให้การใช้ฐานข้อมูลแบบเดิมๆเป็นไปได้ยากขึ้น

วิธีติดตั้งก็แสนจะง่าย โหลดตรงนี้

```
1.  เราจะเลือกเวอร์ชั่น 4.X for windows 64-Bit (.msi)
2.  ดับเบิ้ลคลิกเพื่อติดตั้งตามขั้นตอนไปเลย
3.  หลังจากลงเสร็จตัวโปรแกรมจะไปอยู่ที่ C:\Program Files\MongDB
4.  สร้างโฟลเดอร์ที่ไหนก็ได้เอาไว้เก็บตัวอย่าง DB (Database) เช่น C:\u6088XXX
```

วิธีการเรียก MongoDB มาใช้

```
1.  เรียก PowerShell/Command Prompt อะไรก็ได้
2.  แล้วใช้คำสั่งนี้เพื่อใช้เป็นที่อยู่ของ Database
```

```
"C:\Program Files\MongoDB\Server\4.0\bin\mongod.exe" --dbpath "C:\u6088xxx"
```

3. ตัวโปรแกรมจะสร้างเซิร์ฟเวอร์ที่ใช้พอร์ต 27017 ขึ้นมา (อย่าไปปิดแท็บนี้)

ตัวอย่าง Database

```
{   "address": {
    "building": "1007",
    "coord": [ -73.856077, 40.848447 ],
    "street": "Morris Park Ave",
    "zipcode": "10462"
},
    "borough": "Bronx",
    "cuisine": "Bakery",
    "grades": [
    { "date": { "$date": 1393804800000 }, "grade": "A", "score": 2 },
    { "date": { "$date": 1378857600000 }, "grade": "A", "score": 6 },
    { "date": { "$date": 1358985600000 }, "grade": "A", "score": 10 },
    { "date": { "$date": 1322006400000 }, "grade": "A", "score": 9 },      {
"date": { "$date": 1299715200000 }, "grade": "B", "score": 14 }
],
"name": "Morris Park Bake Shop",   "restaurant_id": "30075445"
}
```

4. เรียก PowerShell/Command Prompt อะไรก็ได้ขึ้นมาอีกอันนึง
5. แล้วใช้คำสั่งนี้ เพื่อนำเข้าไฟล์ Dataset หรือข้อมูลเข้ามา

```
"c:\Program Files\MongoDB\Server\4.0\bin\mongoimport.exe" --db test --collection
restaurants --drop --file "%HOMEPATH%\Downloads\primer-dataset.json"
```

6.เมื่อเสร็จแล้วให้ใช้คำสั่งนี้เพื่อรัน MongoDB ขึ้นมา(MongoDB CLI หรือ Command Line)

```
C:\Program Files\MongoDB\Server\4.0\bin\mongo.exe
```

7. ใช้คำสั่ง use test เพื่อสลับไปใช้ Database ที่ชื่อ test (database ตัวนี้จะมีชื่อ restuarant)

8. **ใช้คำสั่งนี้** db.restaurants.count() **เพื่อนับ** query **ทั้งหมด**

db.restaurants.count() **คืออะไร**

```
   db.restaurants.count()
            ^           ^
            |           |
            |           |
        ชื่อคอลเลคชั่น      ชื่อเมธอด
     COLLECTION_NAME  METHOD
```

Syntax ที่น่าจะต้องจำ(อ่ะนะ)

```
db.mycol.find().pretty() -- pretty() ใช้สำหรับการแสดงผลลัพธ์ให้ดูสวยงาม
```

พวกเครื่องหมาย Operator **จำไว้หน่อยก็ดีนะ**

Operation	Syntax	Example	RDBMS Eqivalent
Equality	{key:value}	db.mycol.find({"by":"tutorials point"}).pretty()	where by = 'tutorials point'
Less Than	{key:{$lt:value}}	db.mycol.find({"likes":{$lt:50}}).pretty()	where likes < 50
Less Than Equals	{key: {$lte:value}}	db.mycol.find({"likes":{$lte:50}}).pretty()	where likes <= 50
Greater Than	{key: {$gt:value}}	db.mycol.find({"likes":{$gt:50}}).pretty()	where likes > 50
Greater Than Equals	{key: {$gte:value}}	db.mycol.find({"likes":{$gte:50}}).pretty()	where likes >= 50
Not Equals	{key: {$ne:value}}	db.mycol.find({"likes":{$ne:50}}).pretty()	where likes != 50

AND syntax in MongoDB ใช้ในเมธอด find() และก็ต้องใช้ คอมม่า(,) เพื่อใช้สำหรับเงื่อนไขแบบนี้

```
>db.mycol.find(
   {
      $and: [
         {key1: value1}, {key2:value2}
      ]
   }
).pretty()
----------------------------------------
>db.mycol.find(
    {
```

```
         $and:[
             {"by":"tutorials point"},{"title": "MongoDB Overview"}]}).pretty() {
   "_id": ObjectId(7df78ad8902c),
   "title": "MongoDB Overview",
   "description": "MongoDB is no sql database",
   "by": "tutorials point",
   "url": "http://www.tutorialspoint.com",
   "tags": ["mongodb", "database", "NoSQL"],
   "likes": "100"
}
```

OR syntax in MongoDB ใช้ในเมธอด find()

```
>db.mycol.find(
   {
      $or: [
         {key1: value1}, {key2:value2}
      ]
   }
).pretty()
-----------------------------------
>db.mycol.find({$or:[{"by":"tutorials point"},{"title": "MongoDB
Overview"}]}).pretty()
{
   "_id": ObjectId(7df78ad8902c),
   "title": "MongoDB Overview",
   "description": "MongoDB is no sql database",
   "by": "tutorials point",
   "url": "http://www.tutorialspoint.com",
   "tags": ["mongodb", "database", "NoSQL"],
   "likes": "100"
}
```

Using AND and OR Together

```
>db.mycol.find({"likes": {$gt:10}, $or: [{"by": "tutorials point"},
   {"title": "MongoDB Overview"}]}).pretty()
{
   "_id": ObjectId(7df78ad8902c),
   "title": "MongoDB Overview",
   "description": "MongoDB is no sql database",
   "by": "tutorials point",
   "url": "http://www.tutorialspoint.com",
   "tags": ["mongodb", "database", "NoSQL"],
   "likes": "100"
}
```

ข้อ 1.

Write map and reduce functions (pseudo code) to find the minimum temperature of each month from the sample input file below. The first column is the datetime in YYYYMMDDhhmm. The second column is Degree Celsius

```
201001010000,25.0
201001010015,24.5
201001010030,24.0
201001010045,24.0
201001010100,23.5
201001010115,23.0
…
201512312345,27.0
```

โค้ดจะเป็นแบบนี้

```
map(key,value):
    (datetime,temperature) = split(value,",");
    currentdate = substring(datetime,0,6);
    Emit(currentdate,temperature);

reduce(key,value):
    min = 0 , isfirst = true;
    if isfirst == true then:
        min = value[0]
    end if
    for each temperature in value :
        if temperature < min then :
            min = temperature ;
        end if
    end for
    Emit(key,min);
```

ข้อ 2.

Repeat Question 1 but find the minimum temperature for each year.

```
map(key,value):
    (datetime,temperature) = split(value,",");
    currentyear = substring(datetime,0,4);
    Emit(currentdate,temperature);

reduce(key,value):
    min = 0 , isfirst = true;
    if isfirst == true then:
        min = value[0]
    end if
    for each temperature in value :
        if temperature < min then :
```

```
            min = temperature ;
        end if
    end for
    Emit(key,min);
```

ข้อ 3.

Given a text file below, write map and reduce functions to output the line number that each word appears in the file. For simplicity, the duplication of some output doesn't matter (e.g. "the 2").

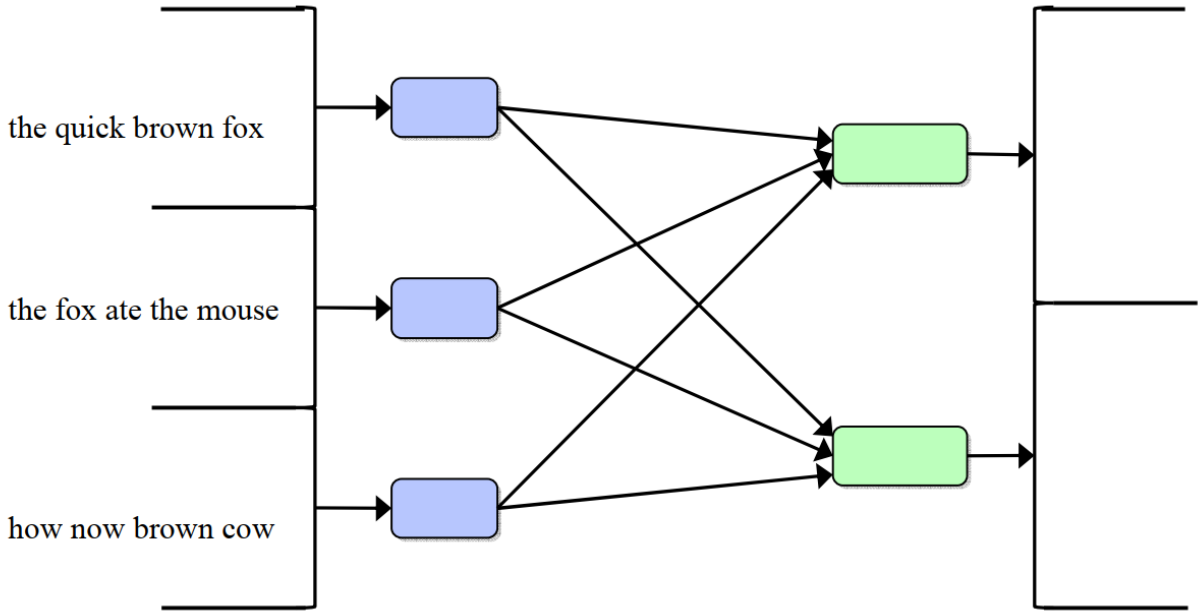
```
INPUT TO MAP (3 lines)
-----------------------
the quick brown fox
the fox ate the mouse
how now brown cow
```

```
OUTPUT FROM REDUCE
-----------------------
ate 2
brown 1
brown 3
…
the 1
the 2
the 2
```

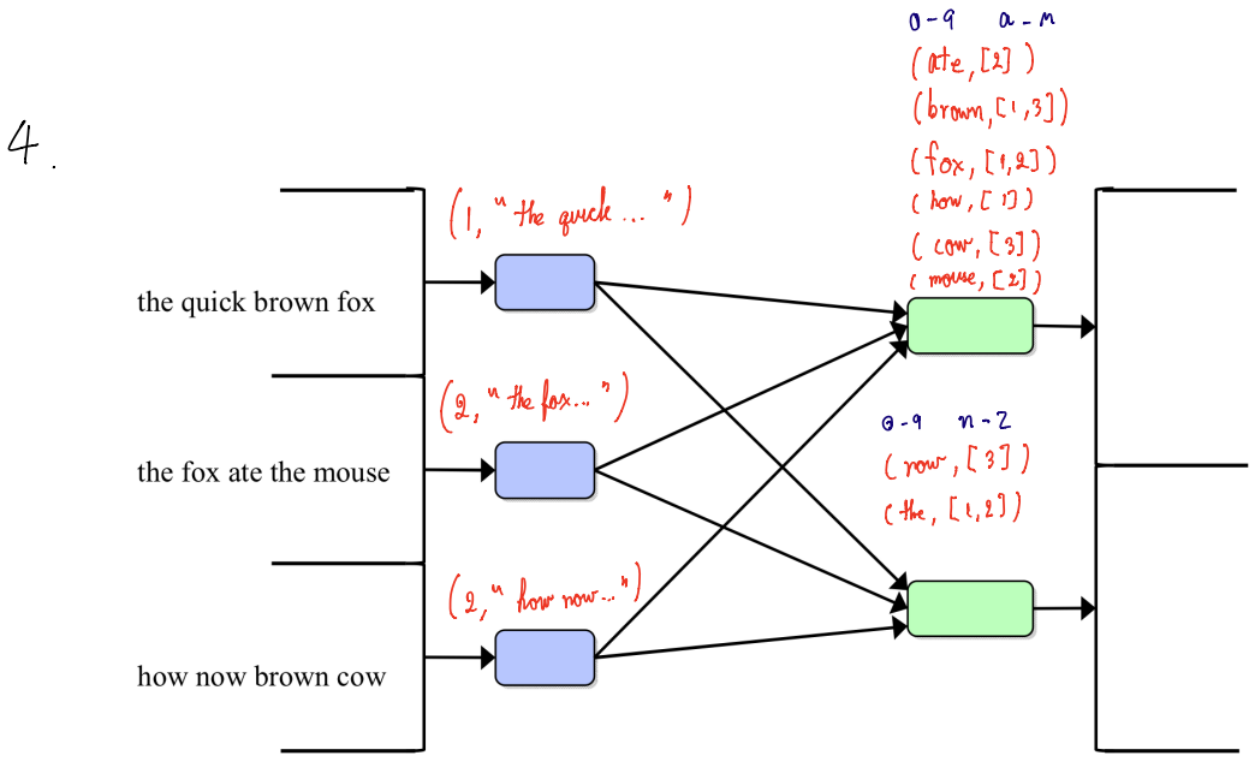
```
map(key,value):
    for each word in split(value," "):
        Emit(word,key);
    end for
reduce(key,value):
    for each word in value:
        Emit(word,key);
    end for
```

ข้อ 4.

Draw the flow of data from Question 3, i.e. the intermediate data from mappers to reducers, and the reducers' output. Given that there are 3 mappers and 2 reducers, and each mapper works on each input line



Result



ข้อ 5.

In a network analysis job, an administrator wants to examine unusual events from a large log file of a web server. Each event is written line by line in the log file. Each line starts with the IP address of browser machines. Write map and reduce functions to filter in all events that contains ERROR. And produce the number of ERROR events of each browser machine. For example, there are 2 ERROR events from 123.125.71.20. Note that the order of events in the output files doesn't matter

```
INPUT TO MAP
----------------------------
1.202.218.8 - - [20/Jun/2012:19:05:12 +0200] "GET /robots.txt OK"
208.115.113.91 - - [20/Jun/2012:19:20:16 +0200] "GET /index.html OK"
123.125.71.20 - - [20/Jun/2012:19:30:40 +0200] "GET /login.html ERROR password
mismatch"
123.125.71.20 - - [20/Jun/2012:19:30:50 +0200] "GET /login.html ERROR password
mismatch"
220.181.108.101 - - [20/Jun/2012:19:31:01 +0200] "GET / OK"
123.125.68.79 - - [20/Jun/2012:19:53:24 +0200] "GET / OK"
178.154.210.252 - - [20/Jun/2012:19:54:10 +0200] "GET /data/test ERROR /data/test/
is forbidden"
74.125.126.103 - - [20/Jun/2012:20:15:29 +0200] "GET /icons/blank.gif OK"
74.125.126.93 - - [20/Jun/2012:20:15:32 +0200] "GET /icons/folder OK"
```

```
OUTPUT FROM REDUCE
---------------------------
178.154.210.252 1
123.125.71.20 2
…
```

```
map(key,value):
    isError = checkword("Error",value);
    if isError == true then
        ip = searchbypattern("(?:[0-9]{1,3}\.){3}[0-9]{1,3}",value);
        Emit(ip,1);
    end if

reduce(key,value):
    sum = 0;
    for each v in value:
        sum = sum + v;
    end for
    Emit(key,sum);
```

Lab from Section 1

Assignment มี 10 ข้อ

```
1.  Display all distinct cuisines
2.  Display only restaurant_id, name, and cuisine fields of all restaurants
without the _id field
3.  Find restaurants (display only name) in Manhattan (borough field equal
“Manhattan”)
4.  Find restaurants (display only name) in Manhattan which has cuisine “Bakery”
5.  Find restaurants (display name and borough) in Manhattan or in Bronx
```

```
6.  Find restaurants (display name, borough, and cuisine) in Manhattan or in Bronx
which has cuisine “Bakery”
7.  Find restaurants (display only name) in Bronx sorted by name in descending
order
8.  Find all restaurants in Bronx with zipcode 10462
9.  Describe what this query returns (in English sentences)
```

```
db.restaurants.find( { "grades.score": { $gt: 30 } } )
```

```
10. Describe what this query returns
```

```
db.restaurants.aggregate([
     { $match:  {"cuisine": "Pizza"} },
     { $group: {
         _id: "$borough", max_score: {$max: { $max: "$grades.score"}}
    }}
])
```

ข้อ 1. Display all distinct cuisines

```
db.restaurants.find({}, {cuisine: 1, _id:0})
```

ผลลัพธ์ที่ได้

```
> db.restaurants.find({}, {cuisine: 1, _id:0})
{ "cuisine" : "Hamburgers" }
{ "cuisine" : "Bakery" }
{ "cuisine" : "American" }
{ "cuisine" : "Jewish/Kosher" }
{ "cuisine" : "American" }
{ "cuisine" : "Chinese" }
{ "cuisine" : "Jewish/Kosher" }
{ "cuisine" : "Ice Cream, Gelato, Yogurt, Ices" }
{ "cuisine" : "Delicatessen" }
{ "cuisine" : "Irish" }
{ "cuisine" : "Ice Cream, Gelato, Yogurt, Ices" }
{ "cuisine" : "American" }
{ "cuisine" : "American" }
{ "cuisine" : "Delicatessen" }
{ "cuisine" : "Delicatessen" }
{ "cuisine" : "Chicken" }
{ "cuisine" : "American" }
{ "cuisine" : "American" }
{ "cuisine" : "Delicatessen" }
{ "cuisine" : "Hamburgers" }
Type "it" for more
```

ข้อ 2. Display only restaurant_id, name, and cuisine fields of all restaurants without the _id field

```
db.restaurants.find({},{"restaurant_id" : 1,"name":1,"cuisine" :1,"_id":0});
```

ผลลัพธ์ที่ได้

```
> db.restaurants.find({},{"restaurant_id" : 1,"name":1,"cuisine" :1,"_id":0});
{ "cuisine" : "Hamburgers", "name" : "Wendy'S", "restaurant_id" : "30112340" }
{ "cuisine" : "Bakery", "name" : "Morris Park Bake Shop", "restaurant_id" : "30075445" }
{ "cuisine" : "American", "name" : "C & C Catering Service", "restaurant_id" : "40357437" }
{ "cuisine" : "Jewish/Kosher", "name" : "Kosher Island", "restaurant_id" : "40356442" }
{ "cuisine" : "American", "name" : "1 East 66Th Street Kitchen", "restaurant_id" : "40359480" }
{ "cuisine" : "Chinese", "name" : "May May Kitchen", "restaurant_id" : "40358429" }
{ "cuisine" : "Jewish/Kosher", "name" : "Seuda Foods", "restaurant_id" : "40360045" }
{ "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Carvel Ice Cream", "restaurant_id" : "40361322" }
{ "cuisine" : "Delicatessen", "name" : "Nordic Delicacies", "restaurant_id" : "40361390" }
{ "cuisine" : "Irish", "name" : "Dj Reynolds Pub And Restaurant", "restaurant_id" : "30191841" }
{ "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Carvel Ice Cream", "restaurant_id" : "40360076" }
{ "cuisine" : "American", "name" : "Glorious Food", "restaurant_id" : "40361521" }
{ "cuisine" : "American", "name" : "The Movable Feast", "restaurant_id" : "40361606" }
{ "cuisine" : "Delicatessen", "name" : "Sal'S Deli", "restaurant_id" : "40361618" }
{ "cuisine" : "Delicatessen", "name" : "Bully'S Deli", "restaurant_id" : "40361708" }
{ "cuisine" : "Chicken", "name" : "Harriet'S Kitchen", "restaurant_id" : "40362098" }
{ "cuisine" : "American", "name" : "P & S Deli Grocery", "restaurant_id" : "40362264" }
{ "cuisine" : "American", "name" : "Regina Caterers", "restaurant_id" : "40356649" }
{ "cuisine" : "Delicatessen", "name" : "Steve Chu'S Deli & Grocery", "restaurant_id" : "40361998" }
{ "cuisine" : "Hamburgers", "name" : "White Castle", "restaurant_id" : "40362344" }
Type "it" for more
>
```

ข้อ 3. Find restaurants (display only name) in Manhattan (borough field equal “Manhattan”)

```
db.restaurants.find( {"borough" :{$in :["Manhattan"]} }, { "name":1,"_id":0 } );
```

ผลลัพธ์ที่ได้

```
> db.restaurants.find( {"borough" :{$in :["Manhattan"]} }, { "name":1,"_id":0 } );
{ "name" : "1 East 66Th Street Kitchen" }
{ "name" : "Dj Reynolds Pub And Restaurant" }
{ "name" : "Glorious Food" }
{ "name" : "Bully'S Deli" }
{ "name" : "Harriet'S Kitchen" }
{ "name" : "P & S Deli Grocery" }
{ "name" : "The Country Cafe" }
{ "name" : "Angelika Film Center" }
{ "name" : "Downtown Deli" }
{ "name" : "Olive'S" }
{ "name" : "Cafe Metro" }
{ "name" : "Lexler Deli" }
{ "name" : "Lorenzo & Maria'S" }
{ "name" : "Berkely" }
{ "name" : "Domino'S Pizza" }
{ "name" : "Texas Rotisserie" }
{ "name" : "21 Club" }
{ "name" : "Metropolitan Club" }
{ "name" : "Palm Restaurant" }
{ "name" : "Spoon Bread Catering" }
Type "it" for more
>
```

ข้อ 4. Find restaurants (display only name) in Manhattan which has cuisine “Bakery”

```
db.restaurants.find( {"borough" :{$in :["Manhattan"]},"cuisine" :{$in :["Bakery"]}
}, {"name":1,"_id":0 } );
```

```
> db.restaurants.find( {"borough" :{$in :["Manhattan"]},"cuisine" :{$in :["Bakery"]} }, { "name":1,"_id":0 } );
{ "name" : "Olive'S" }
{ "name" : "De Robertis Pastry Shop" }
{ "name" : "Little Pie Company" }
{ "name" : "H & H Midtown Bagels East" }
{ "name" : "Fay Da Bakery" }
{ "name" : "Caffe Roma" }
{ "name" : "Tai Pan Bakery" }
{ "name" : "Glaser'S Bakery" }
{ "name" : "La Delice Pastry Shop" }
{ "name" : "Lung Moon Bakery" }
{ "name" : "Capri Bakery" }
{ "name" : "Manna House" }
{ "name" : "Amy'S Bread" }
{ "name" : "Zaro'S Bread Basket" }
{ "name" : "Zaro'S Bread Basket" }
{ "name" : "Amy'S Bread" }
{ "name" : "Levain Bakery" }
{ "name" : "Chiu Hong Bakery" }
{ "name" : "Sweet Chef Southern Styles Bakery" }
{ "name" : "Balthazar Bakery" }
Type "it" for more
>
```

ผลลัพธ์ที่ได้

ข้อ 5. Find restaurants (display name and borough) in Manhattan, or in Bronx

```
db.restaurants.find({"borough":{$in:["Manhattan","Bronx"]}},
{"name":1,borough:1,"_id":0 } );
```

ผลลัพธ์ที่ได้

```
> db.restaurants.find( {"borough" :{$in :["Manhattan","Bronx"]}}, { "name":1,borough:1,"_id":0 } );
{ "borough" : "Bronx", "name" : "Morris Park Bake Shop" }
{ "borough" : "Manhattan", "name" : "1 East 66Th Street Kitchen" }
{ "borough" : "Manhattan", "name" : "Dj Reynolds Pub And Restaurant" }
{ "borough" : "Manhattan", "name" : "Glorious Food" }
{ "borough" : "Manhattan", "name" : "Bully'S Deli" }
{ "borough" : "Manhattan", "name" : "Harriet'S Kitchen" }
{ "borough" : "Manhattan", "name" : "P & S Deli Grocery" }
{ "borough" : "Manhattan", "name" : "The Country Cafe" }
{ "borough" : "Manhattan", "name" : "Angelika Film Center" }
{ "borough" : "Bronx", "name" : "Wild Asia" }
{ "borough" : "Bronx", "name" : "Carvel Ice Cream" }
{ "borough" : "Manhattan", "name" : "Downtown Deli" }
{ "borough" : "Bronx", "name" : "Happy Garden" }
{ "borough" : "Manhattan", "name" : "Olive'S" }
{ "borough" : "Manhattan", "name" : "Cafe Metro" }
{ "borough" : "Manhattan", "name" : "Lexler Deli" }
{ "borough" : "Manhattan", "name" : "Lorenzo & Maria'S" }
{ "borough" : "Manhattan", "name" : "Berkely" }
{ "borough" : "Manhattan", "name" : "Domino'S Pizza" }
{ "borough" : "Bronx", "name" : "Happy Garden" }
Type "it" for more
>
```

ข้อ 6. Find restaurants (display name, borough, and cuisine) in Manhattan or in Bronx which has cuisine “Bakery”

```
db.restaurants.find( {"borough" :{$in :["Manhattan","Bronx"]},"cuisine" :{$in :
["Bakery"]} }, {
"name":1,"borough":1,"cuisine":1,"_id":0 } );
```

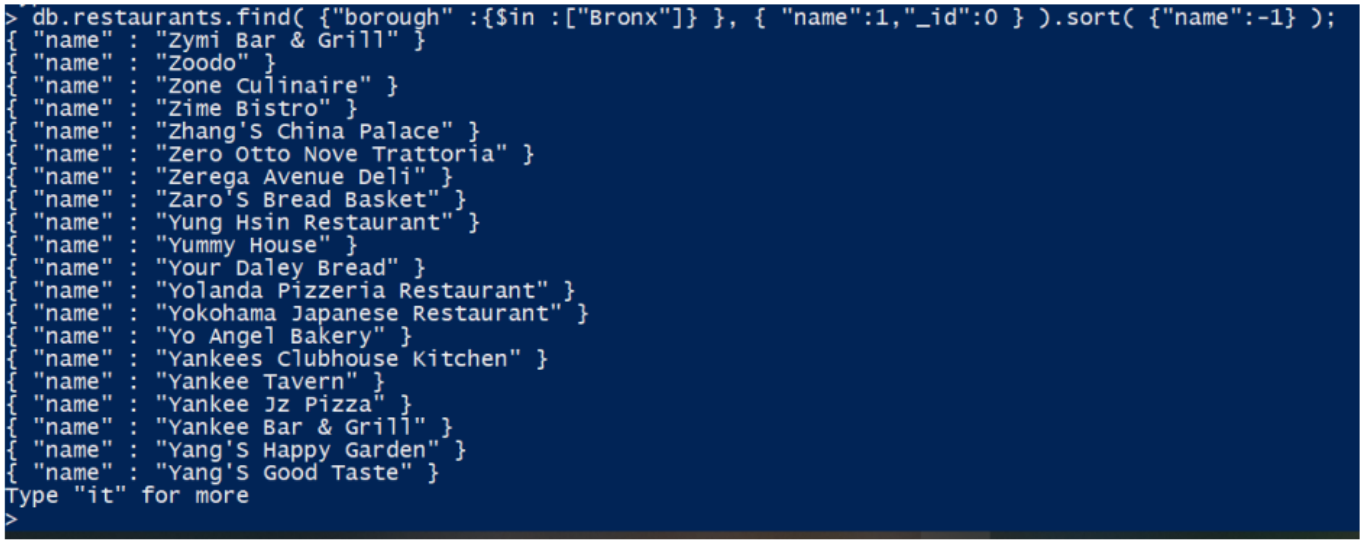
ผลลัพธ์ที่ได้

```
> db.restaurants.find( {"borough" :{$in :["Manhattan","Bronx"]},"cuisine" :{$in :["Bakery"]} }, { "name":1,,"_id":0 } );
2019-11-12T15:50:36.756+0700 E  QUERY    [js] SyntaxError: expected property name, got ',' :
@(shell):1:105
> db.restaurants.find( {"borough" :{$in :["Manhattan","Bronx"]},"cuisine" :{$in :["Bakery"]} }, { "name":1,"borough":1,"cuisine":1,"_id":0 } );
{ "borough" : "Bronx", "cuisine" : "Bakery", "name" : "Morris Park Bake Shop" }
{ "borough" : "Manhattan", "cuisine" : "Bakery", "name" : "Olive'S" }
{ "borough" : "Manhattan", "cuisine" : "Bakery", "name" : "De Robertis Pastry Shop" }
{ "borough" : "Manhattan", "cuisine" : "Bakery", "name" : "Little Pie Company" }
{ "borough" : "Manhattan", "cuisine" : "Bakery", "name" : "H & H Midtown Bagels East" }
{ "borough" : "Manhattan", "cuisine" : "Bakery", "name" : "Fay Da Bakery" }
{ "borough" : "Bronx", "cuisine" : "Bakery", "name" : "E & L Bakery & Coffee Shop" }
{ "borough" : "Manhattan", "cuisine" : "Bakery", "name" : "Caffe Roma" }
{ "borough" : "Bronx", "cuisine" : "Bakery", "name" : "Zaro'S Bread Basket" }
{ "borough" : "Manhattan", "cuisine" : "Bakery", "name" : "Tai Pan Bakery" }
{ "borough" : "Manhattan", "cuisine" : "Bakery", "name" : "Glaser'S Bakery" }
{ "borough" : "Manhattan", "cuisine" : "Bakery", "name" : "La Delice Pastry Shop" }
{ "borough" : "Bronx", "cuisine" : "Bakery", "name" : "Champion Bakery" }
{ "borough" : "Manhattan", "cuisine" : "Bakery", "name" : "Lung Moon Bakery" }
{ "borough" : "Manhattan", "cuisine" : "Bakery", "name" : "Capri Bakery" }
{ "borough" : "Manhattan", "cuisine" : "Bakery", "name" : "Manna House" }
{ "borough" : "Manhattan", "cuisine" : "Bakery", "name" : "Amy'S Bread" }
{ "borough" : "Bronx", "cuisine" : "Bakery", "name" : "Franco Bakery" }
{ "borough" : "Bronx", "cuisine" : "Bakery", "name" : "Biarritz Bakery" }
{ "borough" : "Bronx", "cuisine" : "Bakery", "name" : "Pitusa Bakery" }
Type "it" for more
>
```

ข้อ 7. Find restaurants (display only name) in Bronx sorted by name in descending order

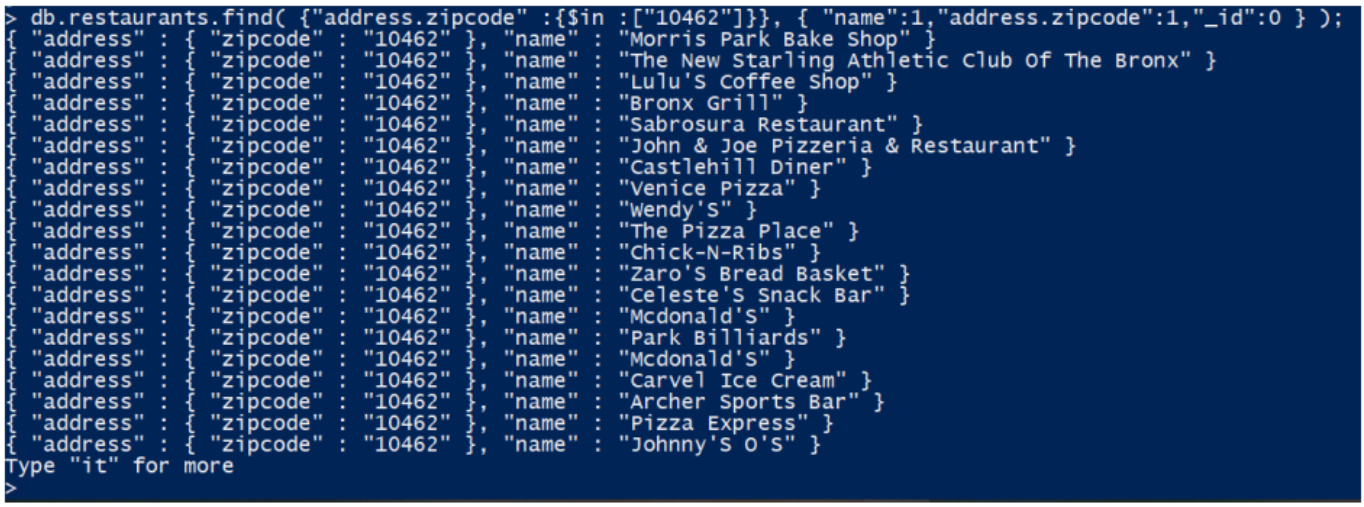
```
db.restaurants.find( {"borough" :{$in :["Bronx"]} }, { "name":1,"_id":0 } ).sort(
{"name":-1} );
```

ผลลัพธ์ที่ได้



ข้อ 8. Find all restaurants in Bronx with zip code 10462

```
db.restaurants.find( {"address.zipcode" :{$in :["10462"]}}, { "restaurant_id" : 1,
"name":1,"address.zipcode":1,"_id":0 } );
```



ข้อ 9. Describe what this query returns (in English sentences)

```
db.restaurants.find({ "grades.score": { $gt: 30 } })
```

คำตอบ $gt (aggregation) used to showing the result the score that's greater than 30 from selects the documents

ข้อ 10. Describe what this query returns

```
db.restaurants.aggregate([
{ $match: {"cuisine": "Pizza"} },
{ $group: {
_id: "$borough", max_score: {$max: { $max: "$grades.score"}}
```

```
}}
])
```

คำตอบ Filters the documents that’s only match the cuisine (pizza) by group from _id and max_score by using $max as max value