

## **Non-RDBMS (mongoDB)**

ปัจจุบันผู้มีส่วนใหญ่ก็จะนิยมใช้ RDBMS(Relation DataBase Management System) ในการจัดเก็บข้อมูล โดยใช้ภาษา SQLเป็นการบริหารจัดการฐานข้อมูลในเชิงสัมพันธ์กัน สามารถนำข้อมูลมาซึ่งเชื่อมกันและกันได้ แต่หากมีข้อมูลมากๆ อาจจะทำให้การเข้าถึงข้อมูลนั้นช้า มีความยุ่งยากมาก และการขยายระบบนั้นทำได้ยาก ดังนั้นในปัจจุบันนี้จึงมีการเก็บข้อมูลแบบใหม่ที่สามารถแก้ปัญหานี้ได้ คือ NOSQL

### **NoSQL (Non-RDBMS) คืออะไร**

NoSQL (Non-RDBMS) คือ ฐานข้อมูลที่ไม่มีภาษา SQL ให้ใช้ ซึ่งมีการใช้งานต่างจาก MySQL, Microsoft SQL, Oracle ที่เราเคยใช้กัน โดยที่อย่างแรกเลยมันไม่มีความสัมพันธ์ เพราะมันไม่ใช่ RDBMS เพราะฉะนั้นจะไม่มีการใช้งานในส่วนของการคำสั่ง Join, WHERE ในการทำงาน และ สำหรับ NoSQL (Non-RDBMS) นี้จะเน้นการ Denormalize คือ เน้นให้การทำงานมันเร็วที่สุดโดยไม่ต้องไป join อะไรทั้งสิ้น โดยหาระเบียน(Record)เดี่ยว แล้วเอามาใช้งานได้เลย ส่วนเวลาที่เชื่อมต่อก็น่าจะง่ายกว่า SQL โดยที่ไม่ต้องมี Overhead มากมาย เพียงแค่มี IP กับ Port ก็เพียงพอแล้วสำหรับการใช้งานแล้ว

NoSQL หรือจะเรียกว่า **Object Database** หรือ **Document Database** ซึ่ง NoSQL database นั้นเป็นที่สนใจมากขึ้นเนื่องจาก Google นำมาใช้ในองค์กร และเรียกชื่อมันว่า "BigTable" แนวคิดของ Object Database นั้นจะเน้นไปที่ข้อมูลที่ไม่มีความสัมพันธ์กัน และต้องการการเข้าถึงที่รวดเร็ว (**Performance**) รวมทั้งความสามารถในการขยายระบบได้ในอนาคต

ส่วนใหญ่การทำงานของ NoSQL เน้นใช้งานกับปริมาณข้อมูลที่มีจำนวนมากมายมหาศาล เช่น facebook , twitter และ google ถ้าใช้คำสั่งในส่วนของ SQL คงใช้เวลานานมากถึงจะได้ข้อมูลนั้นๆ ออกมา จากที่ NoSQL มันไม่มีโครงสร้างตายตัวและไม่แน่นอน จึงทำให้สามารถทำงานได้หลากหลายกว่า ประสิทธิภาพสูงกว่า แต่ก็ยากกว่า RDBMS เนื่องจากไม่ต้องคอยจัดการเรื่องความสัมพันธ์ จึงอาจจะทำให้เกิดความยุ่งยาก

ตัวอย่างของฐานข้อมูลแบบ NoSQL Database สามารถแบ่งตามประเภท Document store ดังนี้

1. CouchDB
2. Apache Jackrabbit
3. MongoDB
4. XML database

- MarkLogic Server

- eXist

ที่มาโดย : <http://www.wikipedia.org>

## **CouchDB**

CouchDB คือ ส่วนหนึ่งใน NoSQL ที่เป็นการทำงานเกี่ยวกับเอกสาร document-oriented database ที่สามารถ queried และ indexed โดยการทำให้ MapReduce ผ่าน JavaScript โดย CouchDB จะใช้งานผ่าน RESTful JSON API ซึ่งจะทำให้เราสามารถเรียกจากที่ไหนก็ได้โดยผ่าน HTTP protocol ที่สำคัญไม่ต้องมี Driver ในการทำงาน

### **เราสามารถทำอะไรได้บ้างใน CouchDB**

- 1.เป็นงานเอกสาร (Document database) ที่มีการเรียกใช้ผ่าน Function ของ RESTful JSON API ที่เป็นส่วนใช้ในคำสั่งต่างๆ เช่น printjson หมายถึงการแสดงผลข้อมูลออกมา
- 2.ไม่มี Schema และสามารถเพิ่มเติมแก้ไข field ได้อย่างทันที
- 3.สามารถทำ replication โดยใช้วิธี bi-direction และสามารถจัดการปัญหาเรื่องการ conflict ของ data ได้ด้วย
- 4.สามารถ Query และ Index ได้โดยการใช้ Javascript แทนการใช้ SQL Query เพื่อเป็นความสะดวกและรวดเร็วกว่าการใช้งานแบบเดิม

## **mongoDB**

สำหรับ NoSQL ชนิด mongoDB นี้เป็นฐานข้อมูลที่สำคัญเหมาะกับการใช้งานด้านเอกสาร (Document (Document oriented)) ซึ่ง mongoDB นี้สามารถรองรับโปรแกรมได้หลายภาษา และการทำงานของ mongoDB เป็นการทำงานแบบ client-server มีการทำงานทั้ง 32 บิต และ 64 บิต ซึ่งจะทำงานใน Memory (RAM)

ดังตามที่กล่าวไว้แล้วจะมีการทำงาน 2 ส่วนด้วยกัน คือ

- 1.ไฟล์ mongod.exe เป็นส่วนการทำงานของ server
- 2.ไฟล์ mongo.exe เป็นส่วนการทำงานของ client

mongoDB เป็นการสนับสนุนของ Open source ซึ่งเขียนขึ้นด้วยภาษา C++ และมีความสามารถรองรับภาษาอื่นๆได้ดังต่อไปนี้

C	C++
Java	Javascript
Perl	PHP
Python	Ruby

สำหรับการใช้ฐานข้อมูลแบบ NoSQL คือ CouchDB และ mongoDB เป็นออกแบบมาเพื่อแก้ปัญหาช่องว่างของ NoSQL กับ RDBMS ให้ทำงานด้วยกันได้ง่ายขึ้น เนื่องจากการทำงานของทั้ง 2 อย่าง (NoSQL กับ RDBMS) มีส่วนที่แตกต่างกันออกไปตัวอย่างเช่น

MySQL Program ใช้ mysql                      ส่วน mongo Program ใช้ mongod  
 MySQL Program ใช้ mysql                      ส่วน mongo Program ใช้ mongo

### ตารางการเปรียบเทียบคำสั่งระหว่าง Sql Statement กับ mongoDB Statement

SQL Statement	mongo Query Language Statement
<u>คำสั่งในการสร้างตารางใน Mysql</u>	
CREATE TABLE USERS (a Number, b Number)	
<u>คำสั่งในการเพิ่มข้อมูล</u>	
INSERT INTO USERS VALUES(1,1)	db.users.insert({a:1,b:1})
<u>คำสั่งในการแสดงข้อมูล</u>	
SELECT * FROM users	db.users.find()
<u>คำสั่งในการแสดงข้อมูลตามเงื่อนไข age=44</u>	
SELECT * FROM users WHERE age=44	db.users.find({age:44})
<u>คำสั่งในการแสดงข้อมูลตามเงื่อนไข age=44 โดยมีการเรียงตามชื่อ</u>	
SELECT * FROM users WHERE age=44 ORDER BY name	db.users.find({age:44}).sort({name:1})
<u>คำสั่งในการแสดงข้อมูล โดยแสดงเพียงแถวเดียว (การกำหนด limit คือจำนวนแถวที่จะแสดง)</u>	
SELECT * FROM users LIMIT 1	db.users.findOne()
<u>คำสั่งในการแสดงข้อมูล ถ้าข้อมูลซ้ำกันให้แสดงออกมาแถวเดียว</u>	
SELECT DISTINCT last_name FROM users	db.users.distinct('last_name')
<u>คำสั่งในการแสดงอัปเดตข้อมูลแบบมีเงื่อนไข</u>	
UPDATE users SET a=1 WHERE b='q'	db.users.update({'b':'q'}, {'\$set': {'a':1}}, false, true)
<u>คำสั่งในการแสดงนับข้อมูล</u>	
SELECT COUNT(*y) FROM users	db.users.count()

จากที่กล่าวมาข้างต้นเป็นส่วนหนึ่งเท่านั้นของการทำงานของ mongo ในส่วนของคำสั่งอื่นๆ ก็เช่นเดียวกัน จะมีการใช้คำสั่งที่แตกต่างกันออกไป เช่น การเลือกมีหลายลักษณะคือ มีการเลือกแบบมีเงื่อนไขการเขียนคำสั่งจะมีรูปแบบที่คล้ายกันแต่จะมีส่วนที่แตกต่างกันออกไปแล้วแต่การเลือกใช้งาน Database จึงทำให้ใช้คำสั่งที่แตกต่างกันออกไป ซึ่งสามารถศึกษาเพิ่มเติมได้ที่

ที่มาของข้อมูล : <http://www.mongodb.org/display/DOCS/SQL+to+Mongo+Mapping+Chart>

### **คุณสมบัติของ mongoDB**

1. การเข้ารหัสอย่างสม่ำเสมอ มีการเรียกข้อมูลไปมาระหว่างพิเศษ
2. สนับสนุนข้ามแพลตฟอร์ม ไปนารีที่ใช้ได้สำหรับ Windows, Linux และ OS X
3. สนับสนุนการแสดงผลปกติ รหัส ข้อมูลไปนารีและอื่น ๆ

จุดเด่นของ mongoDB อยู่ที่การ insert ได้รับการยอมรับในการทำระบบ Tracking , Logging , History เป็นอย่างมาก ดังนั้น mongoDB จะเป็น NoSQL ที่มีประโยชน์มาก เพราะช่วยให้เราสามารถเก็บและเข้าถึงข้อมูลได้ง่าย และสามารถเข้าถึงข้อมูลได้โดยตรงและมีความยืดหยุ่นสำหรับการทำงาน

### **การติดตั้ง mongoDB ลงบน windows**

1. ดาวน์โหลดไฟล์ mongoDB สำหรับวินโดวส์ ขนาด 32 บิต จาก <http://www.mongodb.org/downloads>
  2. แยกไฟล์ zip (unzip)
  3. ทำการรัน mongoDB (cmd)
    - ไปที่ start -> run
    - พิมพ์ cmd -> ไปที่ root (cd C:\)
    - 1. สร้างไดเรกทอรีที่ root ของเครื่อง (C:)
      - สร้างไดเรกทอรี data
- ```
C:\> mkdir \data
```
- สร้างไดเรกทอรี db
- ```
C:\> mkdir \data\db
```
1. กลับมายังที่ที่เก็บไฟล์เดอริไว้ แล้วเข้าใช้งานในโฟลเดอร์ของ mongoDB และ bin
  2. รันไฟล์ mongod.exe (ไฟล์นี้จะเป็นการรันไฟล์ในฝั่งของ server)

หมายเหตุ ถ้ามีการถามถึงการบล็อกไม่ให้ลงภายใน window ควรจะตอบว่า Unblock

## การใช้งาน mongoDB

1. ทำการรัน mongoDB (cmd)

- ไปที่ start -> run

- พิมพ์ cmd -> ไปที่ root (cd C:\)

2. เข้าไปยังโฟลเดอร์ที่เก็บไฟล์ของ mongoDB -> โฟลเดอร์ bin

3. รันไฟล์ที่ชื่อว่า mongo.exe (ไฟล์นี้จะเป็นการรันไฟล์ในฝั่งของ client)

ในส่วนของการทำงานจะแบ่งออกเป็น client-server ในส่วนของ server คือ ส่วนของการรันโปรแกรม และ ส่วนของ client คือ ส่วนของการทำงาน ถ้าต้องการจะออกจากส่วนของ server ให้กด Ctrl+C และถ้าต้องการจะออกจากส่วนของ client ให้ใช้คำสั่ง exit

## คำสั่งพื้นฐานในการใช้งาน mongoDB

1. คำสั่ง show dbs คือ การแสดงชื่อของ database ทั้งหมดที่มีอยู่
2. คำสั่ง use database\_name เป็นการเปลี่ยน database ที่ต้องการจะใช้งาน ณ ตอนนั้นๆ
3. คำสั่ง db เพื่อดูว่าปัจจุบันเราอยู่ใน database ที่ชื่อว่าอะไร
4. คำสั่ง show users คือการแสดงผลผู้ใช้ database ปัจจุบัน
5. คำสั่ง show collections คือ การแสดงชื่อของ collection ใน database ปัจจุบัน
6. คำสั่ง help สำหรับดูคำสั่งพื้นฐานของ mongo shell ที่เกี่ยวกับ database คำสั่งพื้นฐานต่างๆ
7. คำสั่ง db.coll\_name.find({a:1}) คือการแสดงผล list objects ใน collection ที่ชื่อนั้นๆ โดยมีเงื่อนไขว่า a=1
8. คำสั่ง db.coll\_name.help() คือ ตัวช่วยเหลือในกลุ่มของ collection
9. คำสั่ง db.help() คือ ตัวช่วยเหลือในกลุ่มของ database

## การเข้าถึงข้อมูลจาก Query

คำว่า “Query” หมายถึง การนำข้อมูลจากฐานข้อมูลมาแสดง โดยวิธีการทำงานกับผลลัพธ์ของ Query เรียกว่า “cursor object” ในที่นี้เราจะใช้คำสั่ง Query find() ซึ่งจะแสดงทุกอย่างที่อยู่ใน Collection

## Operators พื้นฐาน

\$gt คือ >

\$lt คือ <

\$gte คือ >=

\$lte คือ <=

\$ne คือ !=

\$in อยู่ในarray

\$nin ไม่อยู่ในarray

## การติดต่อ Database ของ mongoDB

1. run -> พิมพ์ cmd
2. C:\mongoDB-win32-i386-1.6.1\bin\mongo

```
ผลลัพธ์ MongoDB shell version: 1.6.1
connecting to: test
```

โดยในตอนแรก mongoDB จะตั้งค่า database ให้อยู่ที่ database ที่ชื่อว่า “test” และผู้ใช้สามารถเปลี่ยน database ได้โดยใช้คำสั่ง use database\_name และจะสร้างดาต้าเบสได้ด้วยคำสั่งเดียวกัน แต่จะต้องมีการทำงานบนดาต้าเบสนั้นๆ ด้วย เช่น การเพิ่มข้อมูลลงบนดาต้าเบส ซึ่งจะประกอบไปด้วย collections จะต้องมีการเพิ่มข้อมูลใน collection ก่อน ซึ่งก็คือการสร้าง collection และ database ด้วย จะทำให้มีการเซฟดาต้าเบสนั้นไว้

## การเลือกใช้ Database

การเลือกใช้ Database ที่ชื่อ ชื่อของดาต้าเบส

```
> use ชื่อของดาต้าเบส
switched to db ชื่อของดาต้าเบส
```

## ศึกษาเพิ่มเติมได้จาก

<http://www.mongodb.org/>