Data Engineering

MongoDB Introduction

Taught by Pichaya Tandayya

PRINCE OF SONGKLA UNIVERSITY

NoSQL Database

- SQL เดิม เป็นภาษาที่ใช้สำหรับเข้าถึงฐานข้อมูล ที่จัดเก็บข้อมูลอยู่ในรูปแบบความสัมพันธ์ของตารางข้อมูล ที่เรียกว่า Relational Database Management System (RDBMS)
- NoSQL (Non SQL หรือ Not only SQL) เป็นการจัดเก็บข้อมูลที่มีมากกว่าความสัมพันธ์แบบตาราง เพื่อจัดเก็บข้อมูลได้หลากหลายรูปแบบยิ่งขึ้น แบ่งเป็น
  - **Key-value Stores**
  - **Document databases**
  - **Wide Column Stores**
  - **Graph Databases**

Source: ดร.อารีย์ ธีรภาพเสรี, ปฏิบัติการหัวข้อ **แนะนำ MongoDB**

NoSQL Database

Key-value Stores ข้อมูลจะเก็บในรูปของ key และค่า value ของ key นั้น ๆ โดยจะไม่มี key ที่ซ้ำกันเลย ส่วนค่า value สามารถเป็นชนิดข้อมูลแบบใดก็ได้ ฐานข้อมูลประเภทนี้เช่น Redis, Riak

Document databases เก็บข้อมูลในรูปแบบโครงสร้าง JSON หรือเป็น document ที่ซับซ้อนกว่าแบบ key-value stores โดยแต่ละ document จะมี id เพื่อเข้าถึง document นั้น ๆ ฐานข้อมูลประเภทนี้เช่น MongoDB และ CouchDB

Source: ดร.อารีย์ ธีรภาพเสรี, ปฏิบัติการหัวข้อ **แนะนำ MongoDB**

NoSQL Database

Wide Column Stores ข้อมูลจะถูกจัดเก็บใน Column แทน ซึ่งต่างจาก SQL ที่เก็บข้อมูลในแถว โดยข้อมูลแต่ละแถว อาจมีจำนวนและรูปแบบคอลัมน์ที่แตกต่างกันได้ แต่ละแถวจะเรียกว่าเป็น column family แต่ละคอลัมน์ก็จะเก็บข้อมูลแบบ key-value ฐานข้อมูลประเภทนี้เช่น Google Bigtable, Cassandra

Graph Databases เก็บข้อมูลเป็นเครือข่ายของความสัมพันธ์ของโหนดต่าง ๆ เช่น ความสัมพันธ์ใน social network อย่าง facebook ซึ่งแต่ละโหนดก็มีความสัมพันธ์กับโหนดอื่น ๆ ได้หลากหลายรูปแบบอีกด้วย โดยแต่ละโหนดสามารถเก็บข้อมูลได้อย่างอิสระ ฐานข้อมูลประเภทนี้เช่น Neo4J, Infinite Graph

Source: ดร.อารีย์ ธีรภาพเสรี, ปฏิบัติการหัวข้อ **แนะนำ MongoDB**

Need of NoSQL

- การมาของ Web 2.0 และ Social Media ทำให้มีข้อมูลที่หลากหลายเผยแพร่อยู่บน Internet
- ข้อมูลกลุ่ม Social Media และบริการต่างๆที่ใกล้เคียงมักมีการเปลี่ยนแปลงรูปแบบ หรือเพิ่มเติม Feature ตลอดเวลา ทำให้ระบบ หรือนักวิเคราะห์ข้อมูลจำเป็นต้องปรับระบบเพื่อรับ Field ของข้อมูลใหม่ ๆ ตลอดเวลา
- การปรับโครงสร้างของ Relational Database ที่มีหลายตารางนั้นทำได้ยากและค่อนข้างใช้เวลา เนื่องจากการเพิ่ม/ลด Field ใดๆนั้นมักกระทบความสัมพันธ์ระหว่างตารางด้วยเสมอ

Source: Dr. Chinnapong Angsuchotmetee, NoSQL: Non-Relational Database, DEPA Basic Data Engineering, 4 ก.ค. 2564

NoSQL

- NoSQL เปลี่ยนจากการเก็บข้อมูลในรูปแบบตารางและความสัมพันธ์ระหว่างตารางเป็นแบบอื่นๆ ตามแต่ที่ระบบเลือกใช้
- ตัวอย่าง
  - เก็บในรูปแบบชุดของ “key” : “value” (BerkelyDB, Apache Ignite)
  - เก็บในรูปแบบ Graph และการเชื่อมโยงกันระหว่างข้อมูล (AllegroGraph, Jena, Neo4J)
  - เก็บในรูปแบบชุดของเอกสาร (MongoDB)

MongoDB

MongoDB เป็น document database ที่จัดเก็บข้อมูลในรูปแบบ collection ของ document

- แต่ละ document จะมีโครงสร้างแบบ JSON ประกอบไปด้วยฟิลด์ข้อมูลต่าง ๆ
- แต่ละ document สามารถมีจำนวนฟิลด์ข้อมูลที่ไม่เท่ากันได้

ข้อดี

- เป็นฐานข้อมูลที่รองรับจำนวนข้อมูลที่เพิ่มขนาดขึ้นอย่างรวดเร็วได้
- มีการค้นหาข้อมูลที่รวดเร็ว
- สำรองข้อมูลได้ง่าย
- สามารถเขียนชุดคำสั่งเป็นสคริปต์แล้วสั่งรันทีเดียวได้

Source: ดร.อารีย์ ธีรภาพเสรี, ปฏิบัติการหัวข้อ **แนะนำ MongoDB**

MongoDB

- รูปแบบในการแลกเปลี่ยนข้อมูล
  - JavaScript Object Notation (JSON)
  - Binary JavaScirpt Object Notation (BSON)
- เนื่องด้วยภาษา JavaScript เป็นภาษาหลักในการเขียนโปรแกรมระบบเว็บ และ JSON เป็นรูปแบบหลักในการแลกเปลี่ยนข้อมูล ทำให้โปรแกรมที่เขียนด้วยภาษา JavaScript หรือโปรแกรมอื่นๆที่รองรับ JSON สามารถส่งข้อมูลเข้าสู่ MongoDB ได้ทันที

Source: ดร.อารีย์ ธีรภาพเสรี, ปฏิบัติการหัวข้อ **แนะนำ MongoDB**

ชุดเอกสารที่อยู่ในรูปแบบ JavaScript Object Notation (JSON)

```
{
  "_id": 1,
  "name" : { "first" : "John", "last" : "Backus" },
  "contribs" : [ "Fortran", "ALGOL", "Backus-Naur Form", "FP" ],
  "awards" : [
    {
      "award" : "W.W. McDowell Award",
      "year" : 1967,
      "by" : "IEEE Computer Society"
    }, {
      "award" : "Draper Prize",
      "year" : 1993,
      "by" : "National Academy of Engineering"
    }
  ]
}
```

- แต่ละ document มีโครงสร้างแบบ JSON ประกอบไปด้วยฟิลด์ข้อมูลต่าง ๆ
- แต่ละ document มีจำนวนฟิลด์ข้อมูลที่ไม่เท่ากันได้

ชุดเอกสารที่อยู่ในรูปแบบ Binary JavaScirpt Object Notation (BSON)

```
{"hello": "world"} →    \x16\x00\x00\x00              // total document size
                        \x02                         // 0x02 = type String
                        hello\x00                    // field name
                        \x06\x00\x00\x00world\x00    // field value
                        \x00                         // 0x00 = type EOO ('end of object')
```

```
{"BSON": ["awesome", 5.05, 1986]} →    \x31\x00\x00\x00
                                       \x04BSON\x00
                                       \x26\x00\x00\x00
                                       \x02\x30\x00\x08\x00\x00\x00awesome\x00
                                       \x01\x31\x00\x33\x33\x33\x33\x33\x33\x14\x40
                                       \x10\x32\x00\xc2\x07\x00\x00
                                       \x00
                                       \x00
```

JavaScript Object Notation

1 เอกสาร JSON แทนด้วย { ..... } ดังโครงสร้างต่อไปนี้

{
“ชื่อข้อมูล” : ข้อมูล,
“ชื่อข้อมูล” : ข้อมูล,
..
..
..
}

```
{
   "firstName": "Duke",
   "lastName": "Java",
   "age": 18,
   "streetAddress": "100 Internet Dr",
   "city": "JavaTown",
   "state": "JA",
   "postalCode": "12345",
   "phoneNumbers": [
      { "Mobile": "111-111-1111" },
      { "Home": "222-222-2222" }
   ]
}
```

Source: Dr. Chinnapong Angsuchotmetee, NoSQL: Non-Relational Database, DEPA Basic Data Engineering, 4 ก.ค. 2564

สรุปความแตกต่าง: MongoDB เทียบกับ MySQL

	MongoDB	MySQL
โมเดลข้อมูล	เก็บข้อมูลในเอกสาร JSON จากนั้นจัดระเบียบเป็น collection	เก็บข้อมูลเป็นคอลัมน์และแถว พื้นที่เก็บข้อมูลเป็นแบบตารางและเชิงสัมพันธ์
ความสามารถในการปรับขนาด	ใช้การจำลองแบบและการแบ่งเป็นส่วนข้อมูลเพื่อปรับขนาดในแนวนอน	ใช้การปรับขนาดแนวตั้งและแบบจำลองการอ่านเพื่อปรับปรุงประสิทธิภาพตามขนาด
ภาษาการสืบค้น	ใช้ภาษาการสืบค้น MongoDB	ใช้ SQL
ประสิทธิภาพ	เก่งในการแทรกหรืออัปเดตบันทึกจำนวนมาก	เร็วกว่าเมื่อเลือกบันทึกจำนวนมาก
ความยืดหยุ่น	ไม่มี Schema ทำให้มีความยืดหยุ่นมากขึ้นและอนุญาตให้ทำงานกับข้อมูลที่ไม่มีโครงสร้าง กึ่งโครงสร้าง และข้อมูลที่มีโครงสร้าง	มี Schema ที่เข้มงวดซึ่งทำงานได้ดีกับข้อมูลที่มีโครงสร้าง
การรักษาความปลอดภัย	ใช้ใบรับรอง Kerberos X.509 และ LDAP เพื่อยืนยันตัวตนผู้ใช้	ใช้วิธีการยืนยันตัวตนในตัว

ควรใช้เมื่อใด: MongoDB vs MySQL

- ดูที่ ACID (Atomicity, Consistency, Isolation, Durability)

Source: https://aws.amazon.com/th/compare/the-difference-between-mongodb-vs-mysql/

ACID (Atomicity, Consistency, Isolation, Durability)

- Atomicity requires that each transaction be "all or nothing“
- Consistency ensures that any transaction will bring the database from one valid state to another
- Isolation ensures that the concurrent execution of transactions
- Durability ensures that once a transaction has been committed

Source: https://saixiii.com/acid-database/

ACID (Atomicity, Consistency, Isolation, Durability)

- Atomicity คือ “all or nothing” หมายถึง ถ้ามีกระบวนการใดหรือส่วนหนึ่งส่วนใดใน transaction นั้น ล้มเหลวแล้ว ทั้ง transaction นั้นถือว่าล้มเหลวทั้งหมด และ database จะยกเลิกการเปลี่ยนแปลงที่เกิดจาก transaction นั้น รวมถึงรับประกันว่าต้องรองรับทุกสถานการณ์ที่อาจจะเกิดขึ้น ไม่ว่าจะไฟดับ หรือ ระบบล่ม
- Consistency คือคุณสมบัติที่จะต้องแน่ใจได้ว่า ไม่ว่าจะทำถึงกระบวนการไหนใน transaction ข้อมูลจะต้องถูกเขียนลงบน database อย่างถูกต้องตามกฎที่ตั้งไว้ แต่ไม่ได้หมายความว่าผลลัพธ์จะได้ถูกต้องตามที่ application หวัง เพราะเป็นความรับผิดชอบระดับ code ของ application เองที่จะต้องรองรับตาม logic

Source: https://saixiii.com/acid-database/

ACID (Atomicity, Consistency, Isolation, Durability)

- Isolation คือคุณสมบัติที่จะมั่นใจว่า ทุก ๆ ผลลัพธ์จากการทำงานของ transactions จะถูกต้อง แม้กระบวนการของ transactions ก่อนจะไม่สมบูรณ์ ก็จะไม่มีผลกระทบต่อ transactions อื่นๆ
- Durability คือ คุณสมบัติที่เมื่อใดก็ตามที่ transaction มีการ “committed” ข้อมูลของ transaction นั้นจะต้องยังคงอยู่ครบถ้วน ถึงแม้จะเกิดไฟดับ หรือระบบล่มหลังจากนั้น ซึ่งใน relational database เมื่อใดก็ตามที่ รัน SQL ผลลัพธ์จะต้องถูกเก็บลงถาวรเพื่อป้องกันระบบล่ม โดยจะต้องถูกเก็บลงใน non-volatile memory

Source: https://saixiii.com/acid-database/

MySQL

- รูปแบบพื้นที่เก็บข้อมูลภายใน MySQL เหมาะสำหรับคลังข้อมูลและการประมวลผลเชิงวิเคราะห์ออนไลน์ เป็นไปตามมาตรฐาน ACID ซึ่งหมายความว่าการทำธุรกรรมเป็นแบบอะตอม สม่ำเสมอ แยกจากกัน และคงทน สิ่งนี้ทำให้ MySQL มีประโยชน์เมื่อทำงานกับธุรกรรมที่ซับซ้อน เช่น กรณีการใช้งานทางอีคอมเมิร์ซ ธุรกรรม และการเงิน
- ข้อมูลที่มีโครงสร้างสูงและการจัดทำดัชนีของ MySQL นั้นดีต่อการสืบค้นแบบเฉพาะกิจ ซึ่งมักดำเนินการโดยผู้ใช้ปลายทางหรือนักวิเคราะห์ข้อมูลที่ต้องการการเข้าถึงข้อมูลอย่างรวดเร็ว ซึ่งไม่ผ่านรายงานหรือการสืบค้นที่กำหนดไว้ล่วงหน้า

Source: https://aws.amazon.com/th/compare/the-difference-between-mongodb-vs-mysql/

MongoDB

- MongoDB เป็นฐานข้อมูลแบบ NoSQL ซึ่งเหมาะสมกับการทำงานกับข้อมูลที่ไม่มีโครงสร้าง ในกรณีการใช้งาน เช่น เครือข่ายสังคม สื่อ หรือ Internet of Things (IoT) เนื่องจาก MongoDB ไม่มี Schema จึงเป็นทางเลือกที่ดีในการจัดการกับข้อมูลที่เปลี่ยนแปลงและขยายตัวตลอดเวลา
- MongoDB is ACID-compliant at the document level.

คุณสมบัติ MongoDB

- โมเดลข้อมูลเชิงเอกสาร
- ความสามารถในการปรับขนาด
- ความพร้อมใช้งานสูง
- การทำดัชนี
- การรวมข้อมูล
- รองรับภาษาที่ หลากหลาย

Source: https://appmaster.io/th/blog/mongodb-khuue-aair

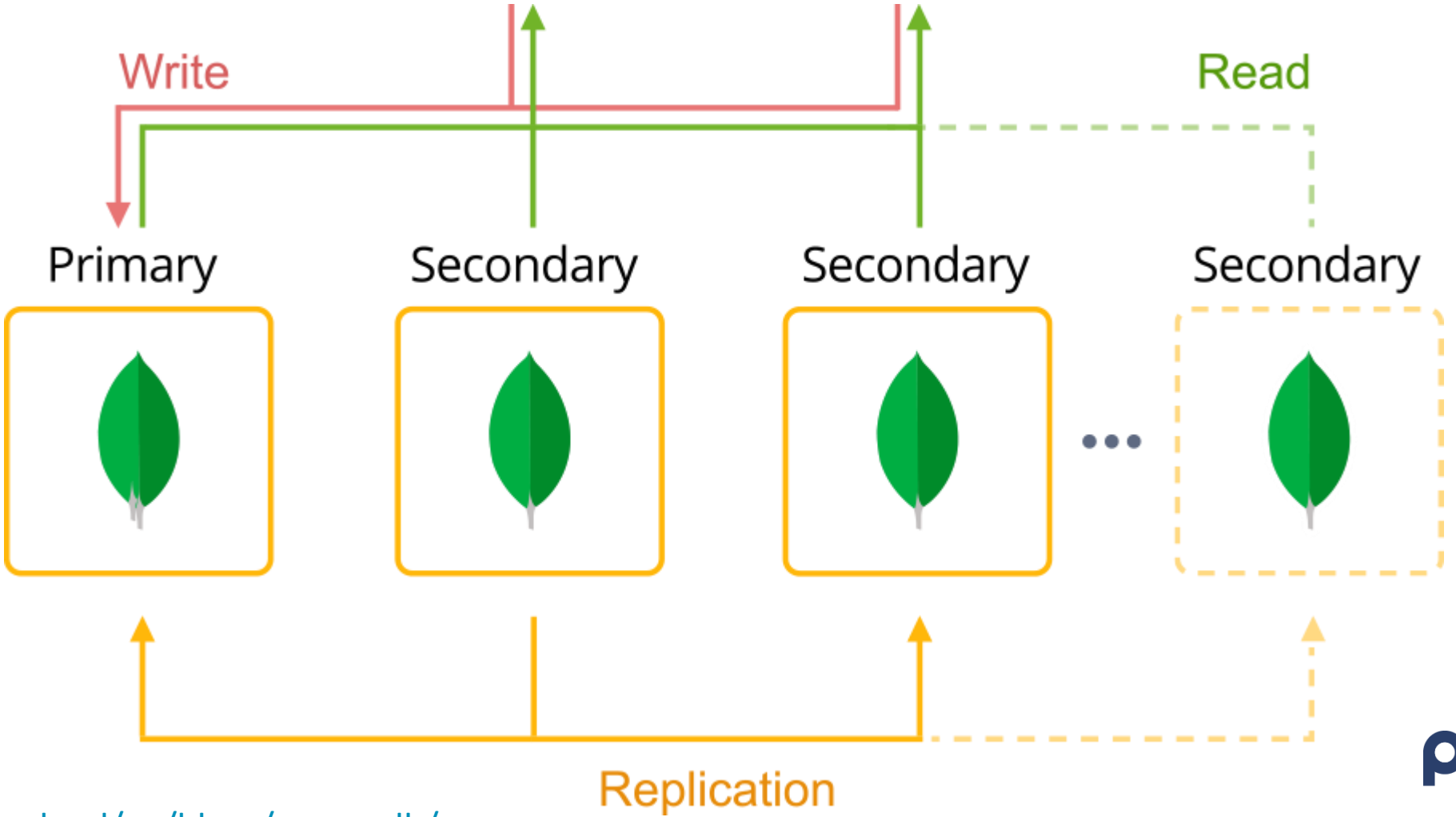
คุณสมบัติ MongoDB

- โมเดลข้อมูลเชิงเอกสาร : MongoDB จัดเก็บข้อมูลในรูปแบบของเอกสารคล้าย JSON ทำให้มีความยืดหยุ่นมากขึ้นและพัฒนาได้เร็วขึ้น
- ความสามารถในการปรับขนาด : MongoDB ใช้สถาปัตยกรรมแบบกระจายและรองรับการปรับขนาดในแนวนอน ซึ่งหมายความว่าสามารถจัดการข้อมูลจำนวนมากขึ้น ตลอดจนอ่านและเขียนเวิร์กโหลดโดยไม่จำเป็นต้องอัปเกรดฮาร์ดแวร์ราคาแพง
- ความพร้อมใช้งานสูง : MongoDB มีคุณสมบัติในตัวสำหรับการ failover อัตโนมัติและชุด replica เพื่อให้มั่นใจว่าฐานข้อมูลยังคงพร้อมใช้งานและสามารถเข้าถึงได้ระหว่างความล้มเหลวของฮาร์ดแวร์หรือการหยุดชะงักอื่นๆ

Source: https://appmaster.io/th/blog/mongodb-khuue-aair

MongoDB สามารถสร้างเป็น Cluster เพื่อตอบสนอง High Availability (HA)

เหมาะกับการทำ Big Data



Source: https://www.proen.cloud/en/blogs/mongodb/

คุณสมบัติ MongoDB

- การทำดัชนี : MongoDB รองรับการทำดัชนีเพื่อปรับปรุงประสิทธิภาพของข้อความค้นหาและการค้นหา ทำให้ค้นหาเอกสารเฉพาะภายใน collection ได้เร็วและง่ายขึ้น
- การรวมข้อมูล : MongoDB มีเครื่องมือในตัวสำหรับการรวมข้อมูล ช่วยให้นักพัฒนาสามารถวิเคราะห์และจัดการข้อมูลจำนวนมากได้อย่างง่ายดาย
- รองรับภาษาที่ หลากหลาย : MongoDB สามารถใช้ได้กับภาษาโปรแกรมต่างๆ มากมาย รวมถึง JavaScript , Python , Java , C++ และอื่นๆ

คุณลักษณะเหล่านี้ทำให้ MongoDB เป็นเครื่องมือที่ทรงพลังและยืดหยุ่นสำหรับการจัดเก็บและจัดการข้อมูลจำนวนมากในแอปพลิเคชันสมัยใหม่ที่ขับเคลื่อนด้วยข้อมูล

Source: https://appmaster.io/th/blog/mongodb-khuue-aair

Go to Document Structure

เริ่มต้นใช้งาน MongoDB บน Windows

ดาวน์โหลดซอฟต์แวร์ **3** ตัว

(1) ดาวน์โหลด MongoDB Community Server ที่
https://www.mongodb.com/try/download/community

(2) ดาวน์โหลด MongoDB Shell ที่
https://www.mongodb.com/try/download/shell

(3) ดาวน์โหลด MongoDB Database Tools ที่
https://www.mongodb.com/try/download/database-tools

- เมื่อ unzip ทั้ง 3 ไฟล์แล้ว ให้ copy คำสั่ง mongoimport.exe ใน folder bin ของ database-tools และ mongosh.exe ใน folder bin ของ Mongo shell ไปเก็บรวมไว้ ใน folder bin ของ MongoDB Community

Source: ดร.อารีย์ ธีรภาพเสรี, ปฏิบัติการหัวข้อ **แนะนำ MongoDB**

การเริ่มใช้งาน MongoDB

แบ่งเป็น 2 ขั้นตอนใหญ่ ๆ คือ

1. รัน MongoDB Service (mongod) ค้างไว้
2. พิมพ์คำสั่งต่าง ๆ ของ MongoDB ซึ่งทำได้ 2 แบบคือ

2.1. พิมพ์คำสั่งบน MongoDB Shell

2.2. ใช้งาน MongoDB GUI

Source: ดร.อารีย์ ธีรภาพเสรี, ปฏิบัติการหัวข้อ **แนะนำ MongoDB**

ขั้นตอนการรัน MongoDB Service

3.1. สร้าง directory สำหรับเก็บฐานข้อมูลเช่น สร้างที่ C:\Users\student\Desktop\ชื่อฐานข้อมูล\db\

3.2. รัน cmd บน windows

3.3. บน cmd ให้ cd ไปยัง directory ที่ลงโปรแกรม MongoDB ไว้ ซึ่งโดยทั่วไปจะเป็น

c:\>cd "Program Files\MongoDB\Server\5.0\bin"

3.4. รัน mongod ด้วยคำสั่ง mongod –-dbpath ตามด้วยชื่อ path ที่สร้างไว้ในข้อ 3.1 เช่น

mongod --dbpath c:\Users\student\Desktop\6410110999\db

ถ้าการทำงานถูกต้อง mongod จะรันข้อความต่าง ๆ เกี่ยวกับสถานะของ MongoDB แล้วหยุดค้างไว้ ให้ค้างหน้าต่าง mongod นี้ไว้ โดยอาจย่อให้เล็กลง แต่ไม่ต้องปิดหน้าต่าง

Source: ดร.อารีย์ ธีรภาพเสรี, ปฏิบัติการหัวข้อ **แนะนำ MongoDB**

ขั้นตอนการรัน MongoDB Shell

3.5. รัน cmd บน หน้าต่างใหม่

3.6. บน cmd ให้ cd ไปยัง directory ที่ลง Mongosh โดยทั่วไปจะเป็น

c:\>cd "Program Files\mongosh"

3.7. บน cmd พิมพ์คำสั่งว่า mongosh เพื่อรัน MongoDB Shell

หมายเหตุ การออกจาก mongod ให้กด Ctrl+c ส่วนการออกจาก MongoDB Shell ให้พิมพ์ exit

Source: ดร.อารีย์ ธีรภาพเสรี, ปฏิบัติการหัวข้อ **แนะนำ MongoDB**

MongoDB CRUD Operations

CRUD (**C**reate, **R**ead, **U**pdate, **D**elete) เป็นกลุ่มคำสั่งพื้นฐานสำหรับการทำงานกับฐานข้อมูล

- **Create** การสร้างมี 3 ระดับคือ สร้างฐานข้อมูล, สร้าง collection และสร้าง document
- **Read** การอ่านและค้นหาข้อมูล MongoDB
- **Update** การอัพเดตข้อมูล document ของ MongoDB
- **Delete** การลบข้อมูลมี 3 ระดับคือ การลบ document, การลบ collection และการลบ database

Source: ดร.อารีย์ ธีรภาพเสรี, ปฏิบัติการหัวข้อ **แนะนำ MongoDB**

Query และ Aggregation

- **Query** เป็นการเรียกดูข้อมูลแบบใส่เงื่อนไขที่ผู้ใช้ต้องการ ผลลัพธ์ที่ได้จะเป็น document ที่ตรงกับเงื่อนไขนั้น
- **Aggregation** เป็นกระบวนการประมวลผลข้อมูล โดยการจัดกลุ่มข้อมูลที่มีบางฟิลด์ที่เหมือนกัน แล้วคำนวณค่าผลลัพธ์บางอย่างจากกลุ่มข้อมูลนี้ เช่น การหาผลรวม การหาค่าเฉลี่ย เพื่อให้ได้ผลการค้นหาเพียงชุดเดียว

Source: ดร.อารีย์ ธีรภาพเสรี, ปฏิบัติการหัวข้อ **แนะนำ MongoDB**

Create

- การสร้างมี 3 ระดับ
  - สร้างฐานข้อมูล
  - สร้าง collection
  - สร้าง document
- ใช้ คำสั่ง use สำหรับสร้างและเรียกใช้งานฐานข้อมูล
- คำสั่ง insert document ลงใน collection
  - `db.xxx.insertOne()` เพิ่มข้อมูลครั้งละ 1 document
  - `db.xxx.insertMany()` เพิ่มข้อมูลหลาย documents ในคำสั่งเดียว

Source: ดร.อารีย์ ธีรภาพเสรี, ปฏิบัติการหัวข้อ **แนะนำ MongoDB**

Document

- เครื่องหมาย { } แทนข้อมูล 1 document หรือ 1 object
- [] หมายถึง ข้อมูลแบบอาร์เรย์
- เครื่องหมาย : เป็นตัวคั่นระหว่าง `<field>:<value>` โดย value สามารถเป็น ตัวเลข, ข้อความ,
- อาร์เรย์, object หรือ โครงสร้างอื่น ๆ โดยแต่ละ `<field>` คั่นด้วย ,

Read and Update

- **Read** อ่านและค้นหาข้อมูล MongoDB โดยใช้คำสั่ง `db.xxx.find()`
  - ระบุเงื่อนไขที่ต้องการค้นหาได้ในรูปแบบ {<field>:<value>,…}
- **Update** การอัพเดตข้อมูล document ของ MongoDB ใช้คำสั่ง
  - `db.xxx.updateOne(<filter>,<update>,<option>)`
  - `db.xxx.updateMany (<filter>,<update>,<option>)`
  - `$set:{<field>:<value>}` เป็นการกำหนดค่า <value> ใหม่ให้กับ <field>

Source: ดร.อารีย์ ธีรภาพเสรี, ปฏิบัติการหัวข้อ **แนะนำ MongoDB**

Delete

- **Delete** การลบข้อมูลมี 3 ระดับ
  - การลบ document
    - `db.xxx.deleteMany({<field>:<value>,…})`
      - ถ้าไม่ระบุเงื่อนไขในวงเล็บจะเป็นการลบ document ทั้งหมดใน collection
    - `db.xxx.deleteOne({<field>:<value>,…})`
      - ลบเฉพาะ document แรกที่ตรงกับเงื่อนไข
  - การลบ collection
    - `db.xxx.drop()`
  - การลบ database
    - `db.dropdatabase()`

Source: ดร.อารีย์ ธีรภาพเสรี, ปฏิบัติการหัวข้อ **แนะนำ MongoDB**

Import JSON Data

When importing data from a **JSON** file, you can format your data as:

- Newline-delimited documents

```
{ "type": "home", "number": "212-555-1234" }
{ "type": "cell", "number": "646-555-4567" }
{ "type": "office", "number": "202-555-0182"}
```

- Comma-separated documents in an array

```
[{ "type": "home", "number": "212-555-1234" },
 { "type": "cell", "number": "646-555-4567" },
 { "type": "office", "number": "202-555-0182"}]
```

Source: https://www.mongodb.com/docs/compass/current/import-export/

Database Tools: Import data

Run mongoimport from the system command line, not the mongo shell.

- Syntax

mongoimport <options> <connection-string> <file>

mongoimport –jsonArray –db database_name –collection collection_name –file file_location

- Example

mongoimport primer-dataset.json

mongoimport --jsonArray --db test --collection timeline-all-casses --file C:\Users\Zaw\timeline-cases-all.json

Source: https://www.mongodb.com/docs/database-tools/mongoimport/#mongodb-binary-bin.mongoimport

Database Tools: Import data

- Syntax ที่ระบุ host

mongoimport --host (host name) -u (name of user) -p (password of user) --authenticationDatabase (admin database used for authentication) --db (Name of database) --collection (name of collection) --drop --file /name_of_file (Name of file which was we have used to import into collection)

Sourcehttps://www.mongodb.com/docs/database-tools/mongoimport/#mongodb-binary-bin.mongoimport

Importing JSON Files

Where,

- **DB_Name** represents the name of the database that contains the collection **Collection_Name**.
- **type** specifies the file type JSON (Optional field).
- **Name-of-file-to-import** represents the name and path of the JSON file to be imported/restored.

```
mongoimport --db DB_Name --collection Collection_Name --type=json --file Name-of-file-to-import
```

```
mongoimport -d DB_NAME -c COLLECTION_name --file Name-of-file-to-import
```

```
mongoimport -d DB_NAME -c COLLECTION_name --drop --file Name-of-file-to-import
```

Source: https://hevodata.com/learn/mongoimport/#part1

Import CSV Data

- When importing data from a CSV file, the first line of the file must be a comma-separated list of the document field names.
- Subsequent lines in the file must be comma-separated field values in the order corresponding with the field order in the first line.
- The following .csv file imports three documents:

```
name,age,fav_color,pet
Jeff,25,green,Bongo
Alice,20,purple,Hazel
Tim,32,red,Lassie
```

Source: https://www.mongodb.com/docs/compass/current/import-export/

Importing CSV Files

Where,

Source: https://hevodata.com/learn/mongoimport/#part1

- **DB_Name** represents the name of the database that contains the collection **Collection_Name**.
- **type** specifies the file type CSV (Optional field).
- **headerline** details the mongoimport command to take the 1st record of CSV file(s) as field names.
- **Name-of-file-to-import** represents the name and path of the CSV file to be imported/restored.

```
mongoimport --db DB_Name --collection Collection_Name --type=csv --headerline --file=Name-of-file-to-import
```

```
mongoimport --db test --collection confirmed-casses --type csv --file C:\Users\Zaw\confirmed-cases-since-120465.csv --headerline
```

Import CSV Data

1. With header row: We can import data with header row with the help of –header that shows the name of the fields will use the first line of the CSV file.

Syntax:

mongoimport –db database_name –collection collection_name –type csv –file file_location –header

2. Without header row: We can import data without header row by excluding –header. But in the place of the header, we have to put –fields that show the name of the field that we want to give. Field names are separated by a comma.

Syntax:

mongoimport –db database_name –collection collection_name –type csv –fields field_names –file file_location

Source: https://www.geeksforgeeks.org/import-data-to-mongodb/

Database Tools: Export data

Run mongoexport from the system command line, not the mongo shell.

- Syntax

```
mongoexport --collection=<coll> <options> <connection-string>
```

- Example

```
mongoexport --db=test --collection=inventory --out=inventory.json
```

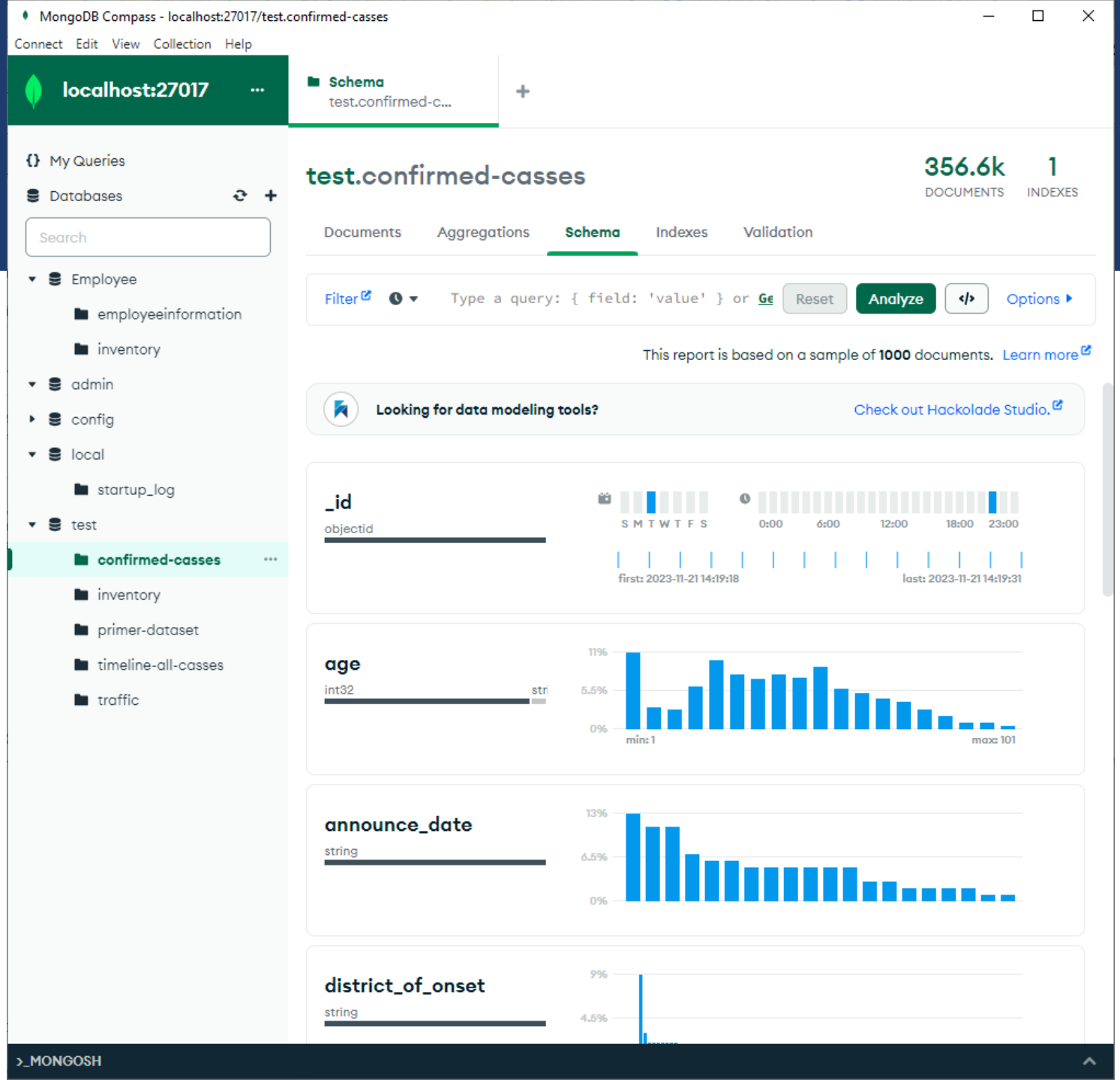
Source: https://www.mongodb.com/docs/database-tools/logs/

คำสั่งเพิ่มเติม

อ้างอิงคำสั่งต่าง ๆ เพิ่มเติมได้ที่

- http://docs.mongodb.com/manual/
- http://www.tutorialspoint.com/mongodb/
- https://www.mongodb.com/docs/compass/current/import-export/
- https://www.mongodb.com/docs/compass/master/
- https://www.mongodb.com/docs/compass/master/import-export/#import-and-export-data-from-the-command-line
- https://www.mongodb.com/docs/database-tools/

ดู Schema ใน MongoDB Compass



แนะนำ Studio 3T

- Studio 3T เป็น GUI ที่ช่วยให้การทำงานกับ MongoDB ง่ายขึ้น ดาวน์โหลดได้ที่ https://robomongo.org
- ก่อนเริ่มใช้งาน Studio 3T จะต้องรัน mongod เพื่อเปิด mongo service

Source: ดร.อารีย์ ธีรภาพเสรี, ปฏิบัติการหัวข้อ **แนะนำ MongoDB**

Assignment

- ทดลองดาวน์โหลด Dataset จาก https://data.go.th/dataset/covid-19-daily
- นำเข้าข้อมูลเข้าสู่ MariaDB หรือ MongoDB ก็ได้ พร้อม ประมวลผลด้วย Pandas และแสดงผลตามต้องการ
- บันทึกภาพถ่ายหน้าจอของ Source Code ผลการรันโปรแกรม มาเป็นการส่งงาน


PRINCE OF
SONGKLA
UNIVERSITY