

mongoDB

สำหรับผู้เริ่มต้น

[ฉบับปรับปรุง 2021]

แนะนำเนื้อหา



- รู้จักกับฐานข้อมูล & MongoDB
- การใช้งาน MongoDB
- โครงสร้างฐานข้อมูล ชนิดข้อมูลและการดำเนินการกับข้อมูล
- การสร้าง Index และการสืบค้นข้อมูล
- การสำรองและกู้ข้อมูล



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



เล่นทั้งหมด

สอน MongoDB (NOSQL) พื้นฐาน

วิดีโอ 10 รายการ • การดู 29,209 ครั้ง • อัปเดตล่าสุดเมื่อ 21 เม.ย. 2016

≡+ ↺ ↻ ...

Kong Ruksiamติดตามแล้ว

- 1  สอน MongoDB (Nosql) ตอนที่ 1 - สร้าง Database และ Collection
Kong Ruksiam 10:24
- 2  สอน MongoDB (Nosql) ตอนที่ 2 - Insert Document Part 1
Kong Ruksiam 6:37
- 3  สอน MongoDB (Nosql) ตอนที่ 3 - Insert Document Part 2
Kong Ruksiam 3:50
- 4  สอน MongoDB (Nosql) ตอนที่ 4 Find() และ FindOne()
Kong Ruksiam 3:53
- 5  สอน MongoDB (Nosql) ตอนที่ 5 - Query Document แบบมีเงื่อนไข
Kong Ruksiam 6:23
- 6  สอน MongoDB (Nosql) ตอนที่ 6 - การใช้ตัวดำเนินการ AND
Kong Ruksiam



ฐานข้อมูล (Database)



กลุ่มข้อมูลที่ถูกรวบรวมและจัดเก็บไว้อย่างมีโครงสร้าง
โดยมีรูปแบบการจัดเก็บที่ชัดเจน เป็นระเบียบและง่ายต่อการ
ทำไปใช้ตามวัตถุประสงค์ที่ต้องการ



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

ฐานข้อมูล (Database)



ฐานข้อมูลที่นิยมใช้ในปัจจุบันแบ่งออกเป็น 2 แบบ ได้แก่

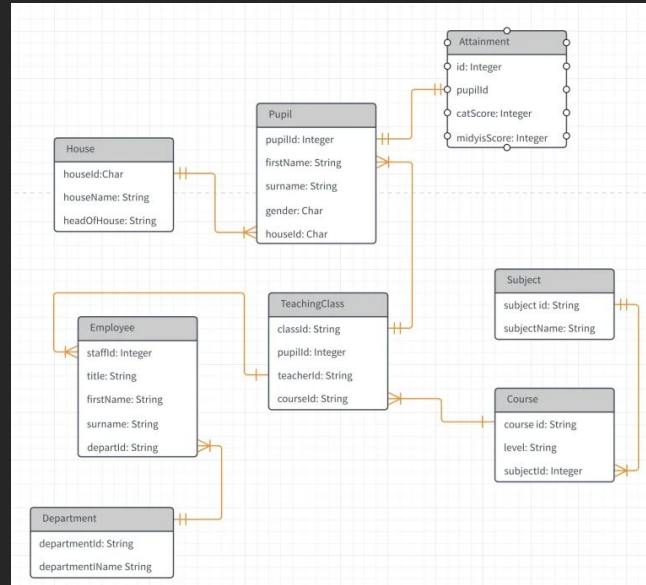
- ฐานข้อมูลเชิงสัมพันธ์ (Relational Database)
- ฐานข้อมูลไม่ใช่เชิงสัมพันธ์ (Non-Relational Database)

ฐานข้อมูลเชิงสัมพันธ์ (Relational Databases)

- เป็นฐานข้อมูลที่ใช้ภาษา SQL (Structure Query Language) มาดำเนินการกับข้อมูล (เพิ่ม , ลบ , แก้ไข , ค้นหา)
- จัดเก็บข้อมูลแบบตาราง (Table)
- *กำหนดโครงสร้างการจัดเก็บข้อมูลที่ชัดเจน (Structure Data)
- มีการเชื่อมโยงความสัมพันธ์กับข้อมูลตารางอื่นๆด้วย



ฐานข้อมูลเชิงสัมพันธ์ (Relational Databases)



<https://stackoverflow.com/questions/51542341/relational-database-design-query>



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

ฐานข้อมูลเชิงสัมพันธ์ (Relational Databases)

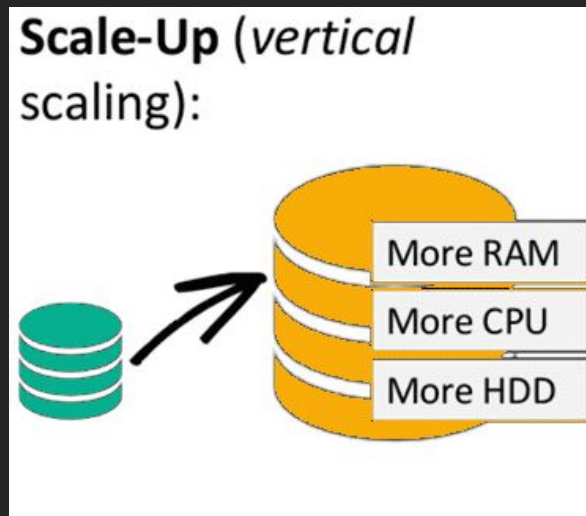
การปรับขนาด

เมื่อฐานข้อมูลมีการจัดเก็บข้อมูลในปริมาณมาก และอนาคตมีการปรับขนาดฐานข้อมูลให้ใหญ่ขึ้น ฐานข้อมูลเชิงสัมพันธ์จะเป็นการปรับขนาดในแนวตั้ง (**Vertical-Scaling**) โดยติดตั้งอุปกรณ์เก็บข้อมูลเพื่อเพิ่มความจุ เพิ่มแรมและหน่วยประมวลผลให้มีประสิทธิภาพสูงขึ้น ส่งผลให้มีค่าใช้จ่ายที่สูงขึ้นและการอัปเดตอุปกรณ์ต้องทำการปิดระบบชั่วคราว



ฐานข้อมูลเชิงสัมพันธ์ (Relational Databases)

การปรับขนาดในแนวตั้ง (Vertical-Scaling)

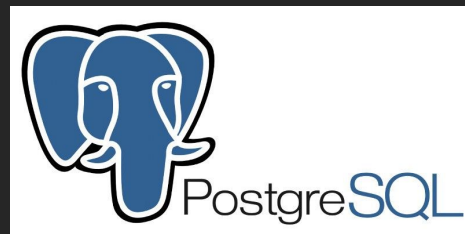


<https://sangeethraaj21.medium.com/sql-vs-nosql-faef10e3852d>

Database Management System (DBMS)

ระบบจัดการฐานข้อมูล SQL ที่ได้รับความนิยมในปัจจุบัน

- MySQL
- Oracle
- PostgreSQL
- SQL Server



เนื้อหา SQL จำนวน 6 ชั่วโมงเต็ม!





ปูพื้นฐาน SQL สำหรับจัดการฐานข้อมูล [2020]

วิดีโอ 44 รายการ • การดู 56,563 ครั้ง • อัปเดตล่าสุดเมื่อ 24 ต.ค. 2020

≡+ ↺ ↻ ...

 **KongRuksiam Official** 

1 **ปูพื้นฐาน SQL** สำหรับจัดการฐานข้อมูล [Full Course] 6 ชั่วโมงเต็ม !!!
KongRuksiam Official
6:24:38

2 **ปูพื้นฐาน SQL** เบื้องต้น [2020] ตอนที่ 1 - Introduction
KongRuksiam Official
ตอนที่ 1 - Intro 12:50

3 **ปูพื้นฐาน SQL** เบื้องต้น [2020] ตอนที่ 2 - ดาวน์โหลด SQLite
KongRuksiam Official
ตอนที่ 2 - ดาวน์โหลด 5:28

4 **ปูพื้นฐาน SQL** เบื้องต้น [2020] ตอนที่ 3 - สร้างฐานข้อมูลและตาราง
KongRuksiam Official
ตอนที่ 3 - สร้างฐานข้อมูล 21:41

5 **ปูพื้นฐาน SQL** เบื้องต้น [2020] ตอนที่ 4 - บันทึกข้อมูล (Insert)
KongRuksiam Official
ตอนที่ 4 - บันทึกข้อมูล 15:17



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

ฐานข้อมูลไม่ใช่เชิงสัมพันธ์ (Non-Relational Databases)

ฐานข้อมูลไม่ใช่เชิงสัมพันธ์ หรือเรียกอีกชื่อ คือ NoSQL (Not-Only SQL) ถูกคิดค้นขึ้นเพื่อจัดการข้อจำกัดของการใช้งาน Relational Database ในอดีต เพื่อปรับเปลี่ยนโครงสร้างการจัดเก็บข้อมูลให้มีความยืดหยุ่นที่มากขึ้นด้วย



ฐานข้อมูลไม่ใช่เชิงสัมพันธ์ (Non-Relational Databases)

NoSQL ถูกนำมาใช้งานการสำหรับประมวลผลข้อมูลจากการใช้งาน Internet อย่างแพร่หลายในยุคปัจจุบันไม่ว่าจะมาจาก Social Media ต่างๆ ที่มีข้อมูลขนาดใหญ่ (Big Data) เพื่อแปลข้อมูลที่มีความหลากหลายมากขึ้น เช่น ข้อมูลรูปภาพ วิดีโอ เสียง ตัวอักษร เป็นต้น ซึ่งจะตอบโจทย์ในเรื่อง ความเร็วและความยืดหยุ่นในการจัดเก็บข้อมูล



ฐานข้อมูลไม่ใช่เชิงสัมพันธ์ (Non-Relational Databases)

- ไม่มีการจัดเก็บข้อมูลในแบบตารางที่มีความสัมพันธ์กัน
- ไม่มีโครงสร้างการจัดเก็บข้อมูลที่แน่นอน (UnStructure Data)
- ง่ายต่อการปรับขนาดและโครงสร้าง

ฐานข้อมูลไม่ใช่เชิงสัมพันธ์ (Non-Relational Databases)

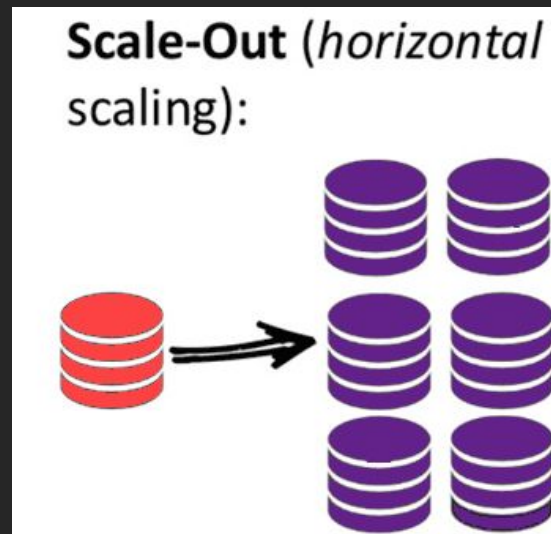
การปรับขนาด

เมื่อฐานข้อมูลมีการจัดเก็บข้อมูลในปริมาณมาก และอนาคตมีการปรับขนาดฐานข้อมูลให้ใหญ่ขึ้น ฐานข้อมูลไม่ใช่เชิงสัมพันธ์จะเป็นการปรับขนาดในแนวนอน (**Horizontal-Scaling**) โดยเพิ่ม Server เข้าไปแบ่งภาระของข้อมูล สามารถรองรับข้อมูลขนาดใหญ่ขึ้น มีประสิทธิภาพสูงขึ้น ติดตั้งง่ายและเปลี่ยนแปลงได้ตลอดเวลาโดยไม่ต้องปิดระบบ



ฐานข้อมูลไม่ใช่เชิงสัมพันธ์ (Non-Relational Databases)

การปรับขนาดในแนวนอน (Horizontal-Scaling)



<https://sangeethraaj21.medium.com/sql-vs-nosql-faef10e3852d>

รูปแบบการจัดเก็บข้อมูล (NoSQL Store)



NoSQL ไม่มีโครงสร้างการจัดเก็บข้อมูลที่แน่นอน เพื่อที่จะสามารถปรับเปลี่ยนโครงสร้างได้ในอนาคต โดยจะเก็บข้อมูลอยู่ 4 รูปแบบ (ตามลักษณะการใช้งาน)

- Key - Value Store
- Document Store
- Graph Store
- Wide Column Store

รูปแบบการจัดเก็บข้อมูล (NoSQL Store)



1.Key-Value Store เป็นระบบจัดเก็บข้อมูลในรูปแบบที่เรียกว่า "Associative Arrays " หรืออาร์เรย์ของชุดข้อมูลที่มีความสัมพันธ์กันในรูปแบบคู่คีย์ (key : value) หรือเรียกอีกชื่อว่า Dictionary

โดยคีย์ (key) จะทำหน้าที่เป็นตัวเชื่อมโยงไปยังค่า (value) ในโครงสร้างของฐานข้อมูลซึ่งจะมีได้แค่คีย์เดียว และการดึงข้อมูลต้องระบุคีย์ที่เก็บค่าข้อมูลนั้นๆไว้

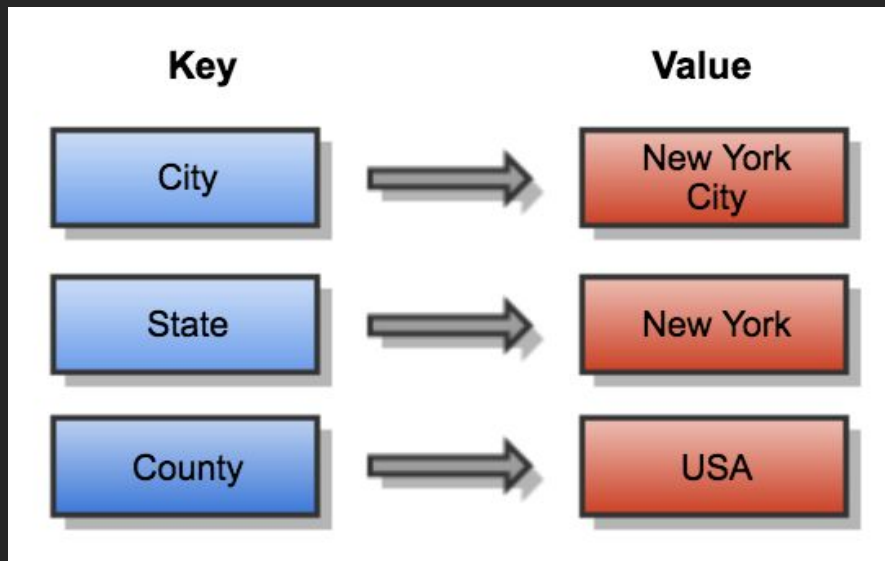


<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

ตัวอย่าง Key - Value Store



<https://dv-website.s3.amazonaws.com/uploads/2018/09/kvd-pic1.png>

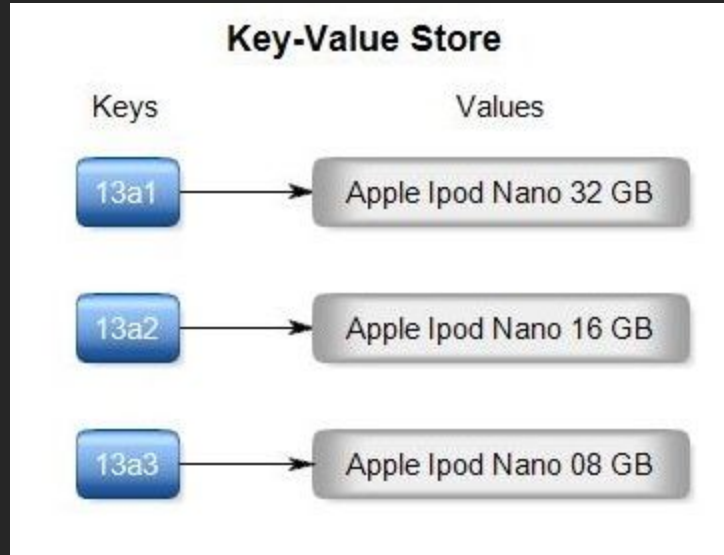


<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

ตัวอย่าง Key - Value Store



<https://www.analyticsvidhya.com/blog/wp-content/uploads/2014/11/keyvalue.jpg>



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

ฐานข้อมูลแบบ Key - Value Store



- Redis
- Oracle NoSQL Database
- Voldemorte
- Aerospike

The Oracle NoSQL Database logo, consisting of the word "ORACLE" in red with a registered trademark symbol, followed by a horizontal line and the words "NOSQL DATABASE" in black.

รูปแบบการจัดเก็บข้อมูล (NoSQL Store)



2.Document Store เป็นฐานข้อมูลเชิงเอกสารที่ออกแบบมาเพื่อจัดเก็บข้อมูล ดึงข้อมูล และจัดการข้อมูล ที่เรียกว่า **“Document-Oriented”** คือ ใช้โครงสร้างภายในของเอกสารเพื่อระบุการจัดเก็บข้อมูล ซึ่งมีลักษณะแบบกึ่งโครงสร้าง (Semi Structure Data) โดยเอกสาร (Document) จะถูกเข้ารหัสข้อมูลและจัดให้อยู่ในรูปแบบของ XML , JSON



ตัวอย่าง Document Store



C1	C2	C3	C4
—	—	—	—
—	—	—	—
—	—	—	—
—	—	—	—

Relational data model

Highly-structured table organization with rigidly-defined data formats and record structure.



Document data model

Collection of complex documents with arbitrary, nested data formats and varying "record" format.

Document 1

```
{
  "id": "1",
  "name": "John Smith",
  "isActive": true,
  "dob": "1964-30-08"
}
```

Document 2

```
{
  "id": "2",
  "fullName": "Sarah Jones",
  "isActive": false,
  "dob": "2002-02-18"
}
```

Document 3

```
{
  "id": "3",
  "fullName": {
    "first": "Adam",
    "last": "Stark"
  },
  "isActive": true,
  "dob": "2015-04-19"
}
```

<https://lennilobel.wordpress.com/2015/06/01/relational-databases-vs-nosql-document-databases/>

<https://dataconomy.com/wp-content/uploads/2014/07/SQL-vs.-NoSQL.png>



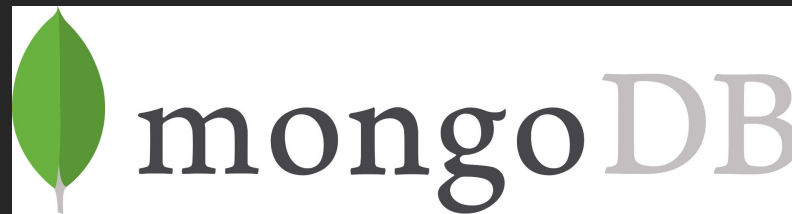
<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

ฐานข้อมูลแบบ Document Store

- MongoDB
- CouchDB



รูปแบบการจัดเก็บข้อมูล (NoSQL Store)



3. Graph Store เป็นการจัดเก็บข้อมูลและความสัมพันธ์ของข้อมูลในรูปแบบกราฟ ข้อมูลด้านในจะเรียกว่า โหนด (Node) ซึ่งจะเก็บ Entity ของข้อมูลและกำหนดความสัมพันธ์ที่เชื่อมโยงความสัมพันธ์ดังกล่าวผ่าน Edge การใช้งานกราฟจะถูกนำเสนอผ่านความสัมพันธ์ของข้อมูล เช่น Social Network , AI และ Machine Learning เป็นต้น

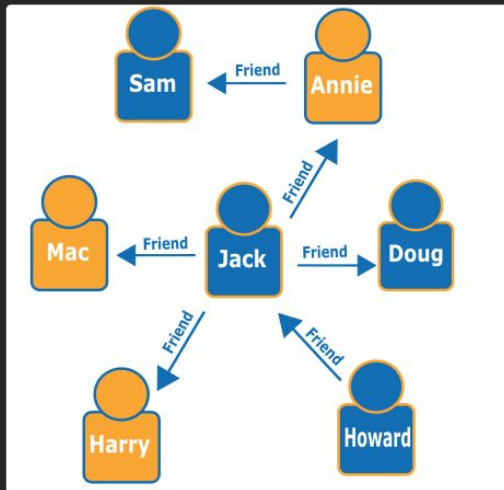


<https://www.youtube.com/c/KongRuksiamOfficial/>

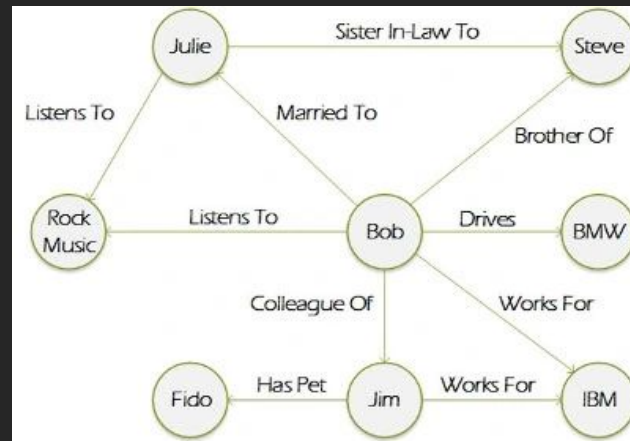


<https://www.facebook.com/KongRuksiamTutorial/>

ตัวอย่าง Graph Store



<https://d1.awsstatic.com/diagrams/foaf-graph.e5e42865e0ee97a2972f9014d28f525ef68a981b.png>



<https://mkhernandez.files.wordpress.com/2019/01/graph-database-sketch1.jpg?w=400>

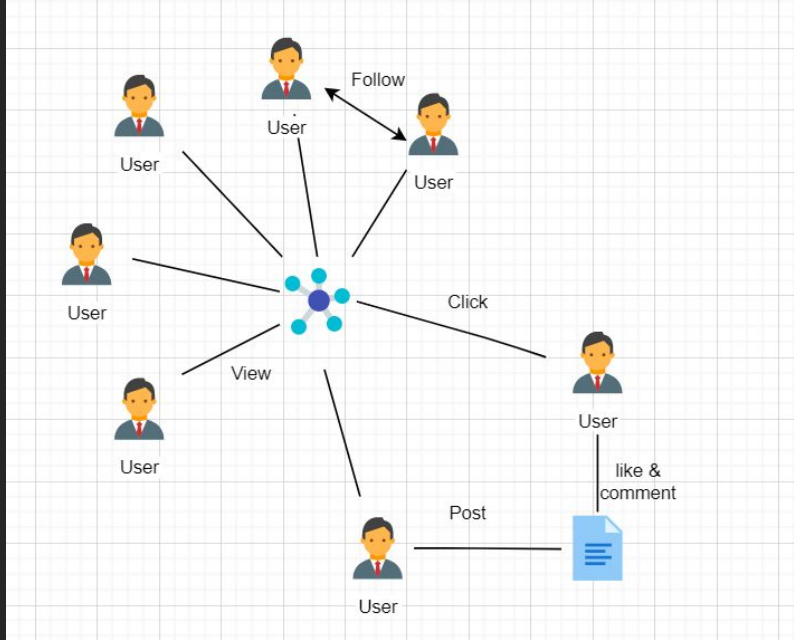


<https://www.youtube.com/c/KongRuksiamOfficial/>

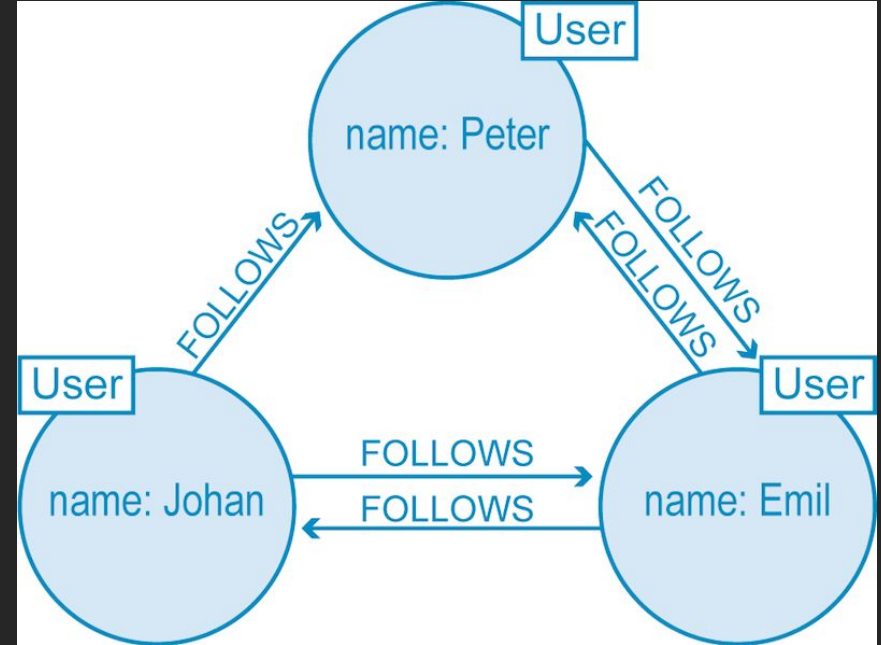


<https://www.facebook.com/KongRuksiamTutorial/>

ตัวอย่าง Graph Store



<https://dz2cdn1.dzone.com/storage/temp/13696862-1593695775580.png>



<https://dist.neo4j.com/wp-content/uploads/20180711200201/twitter-users-graph-database-model-peter-emil-johan.png>

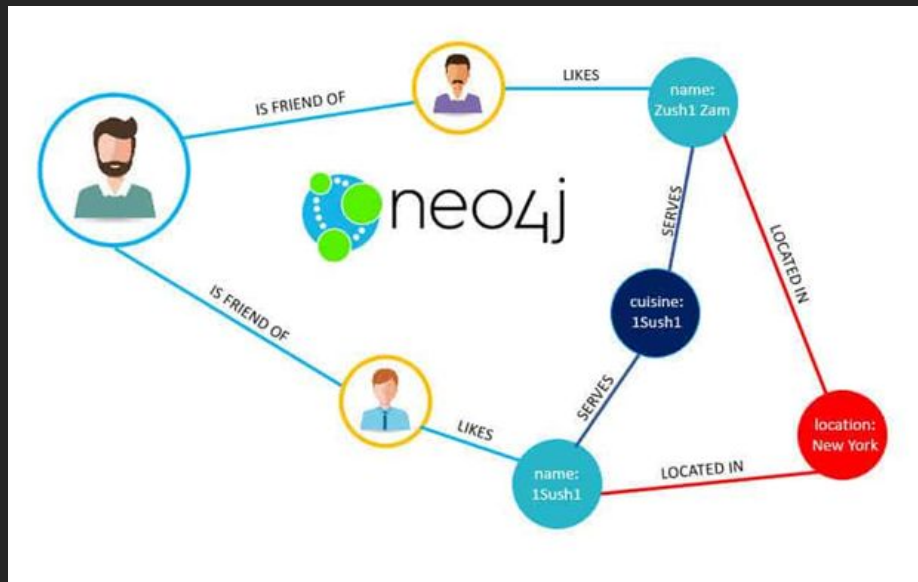


<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

ฐานข้อมูลแบบ Graph Store



<https://www.ctidata.com/graph-databases-hype/>



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

รูปแบบการจัดเก็บข้อมูล (NoSQL Store)



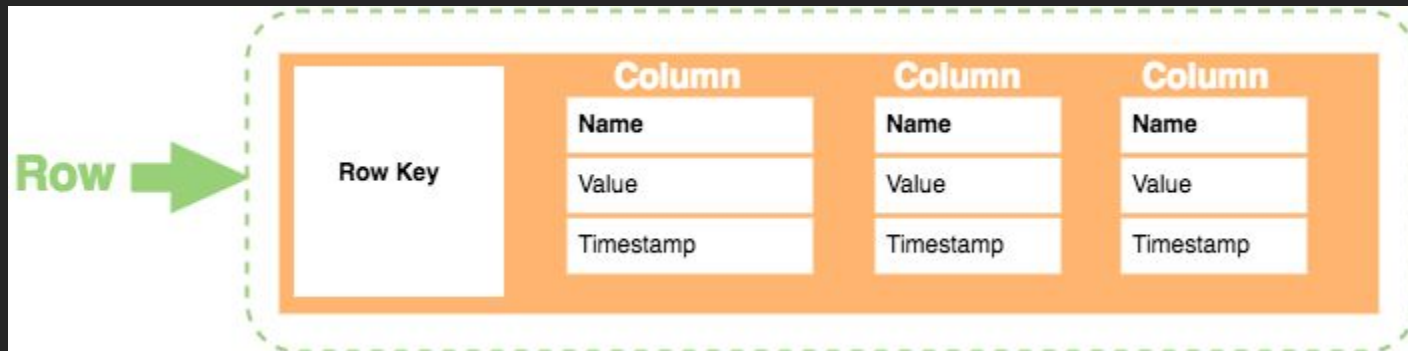
4 Wide Column Storage เป็นการจัดเก็บข้อมูลเป็นเรคอร์ดหรือในรูปแบบตาราง แถว และคอลัมน์ ซึ่งจะแตกต่างจาก Relational Database คือจะมีความยืดหยุ่นมากกว่า ชื่อและรูปแบบของคอลัมน์สามารถเปลี่ยนจากแถวหนึ่งไปอีกแถวภายในตารางเดียวกันได้และขยายคอลัมน์ได้จำนวนมาก

(Extensible Record Store : ขยายการจัดเก็บข้อมูลในแต่ละเรคอร์ด)

อาจจะมองเป็นรูปแบบ key:value ในรูปแบบ 2 มิติก็ได้



โครงสร้างของ Wide Column Storage mongoDB



<https://i.imgur.com/8laydo6.png>

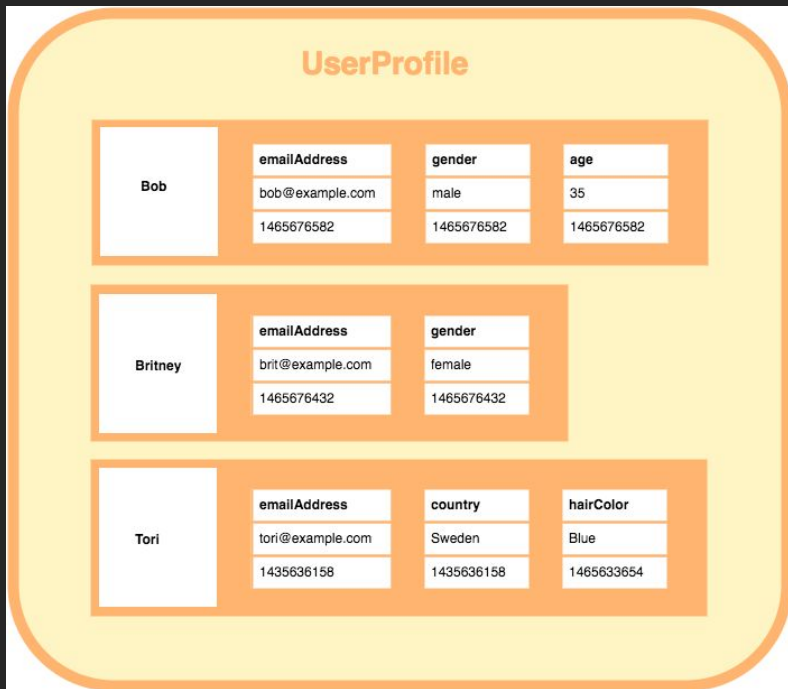


<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

โครงสร้างของ Wide Column Storage



https://database.guide/wp-content/uploads/2016/06/wide_column_store_database_example_column_family-1.png



<https://i1.wp.com/technicalsand.com/wp-content/uploads/2020/12/Wide-column-database-example.png?resize=452%2C256&ssl=1>



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

รูปแบบการจัดเก็บข้อมูล

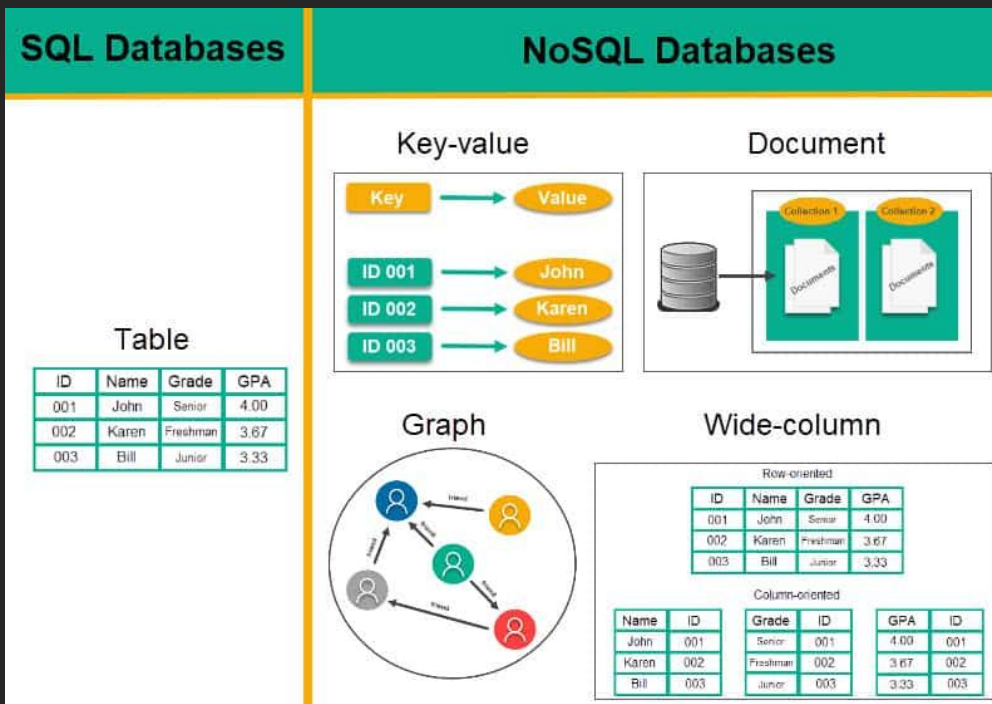


ตัวอย่างฐานข้อมูลแบบ Wide Column Store

- Cassandra
- HBase
- DynamoDB
- Microsoft Azure Cosmos DB



สรุป SQL และ NoSQL



สรุป SQL และ NoSQL



	SQL	NoSQL
โมเดล	Relational	Non Relational
ภาษาที่ใช้	SQL	ตามลักษณะการใช้งาน
รูปแบบการจัดเก็บ	ตาราง	Key-Value , Document , Graph , Wide Column
โครงสร้าง (Schema)	โครงสร้างแน่นอน	โครงสร้างแบบไดนามิก
การปรับขนาด	แนวตั้ง (Vertical)	แนวนอน (Horizontal)
การค้นหา	สำหรับการค้นหาที่ซับซ้อน	ไม่เหมาะกับการค้นหาที่ซับซ้อน



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

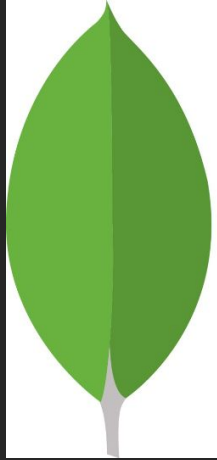
Break



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



mongoDB

รู้จักกับ MongoDB



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

MongoDB คืออะไร



เป็นฐานข้อมูลเชิงเอกสาร (Document Store) สำหรับเก็บข้อมูลขนาดใหญ่ที่มีความยืดหยุ่นสูง ง่ายต่อการปรับขนาดและทำงานข้าม Platform ได้ การจัดเก็บข้อมูลไม่ได้อยู่ในรูปแบบตาราง แต่จะอยู่ในรูปแบบเอกสาร JSON แล้วเซฟเก็บไว้ในเอกสารรูปแบบไบนารี BSON



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

โครงสร้างการจัดเก็บข้อมูลของ MongoDB

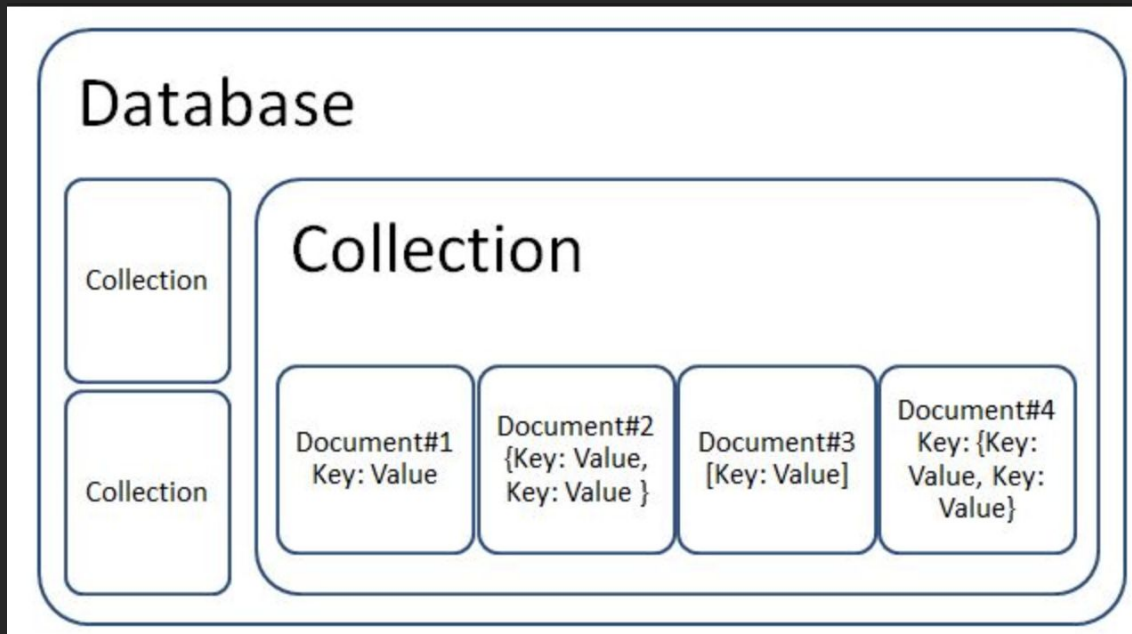
มีองค์ประกอบอยู่ 3 ส่วน ได้แก่

- Database
- Collection
- Documents



<https://phoenixnap.com/kb/wp-content/uploads/2021/05/document-database-illustration.png>

โครงสร้างการจัดเก็บข้อมูลของ MongoDB



https://miro.medium.com/max/1313/1*vhKORnX6ZQ5HNUaeqxbQGg.png



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

โครงสร้างการจัดเก็บข้อมูลของ MongoDB

- Database (ฐานข้อมูล) เป็นส่วนที่ใช้เก็บ Collection หรือชุดข้อมูล



โครงสร้างการจัดเก็บข้อมูลของ MongoDB

- Collection หรือชุดข้อมูลเทียบได้กับตารางในฐานข้อมูลเชิงสัมพันธ์
- Document เอกสารที่จัดเก็บข้อมูลของคู่คีย์ (Key) และค่า (Value)



ความแตกต่างระหว่าง SQL กับ MongoDB

SQL	MONGODB
Database	Database
Table	Collection
Row	document or BSON document
Column	Field
Index	Index
table joins	embedded documents and linking
primary key (specify any unique column or column combinations as primary key)	primary key (the primary key is automatically set to the _id field in MongoDB)
aggregation (e.g. by group)	aggregation pipeline

ความแตกต่างระหว่าง SQL กับ MongoDB

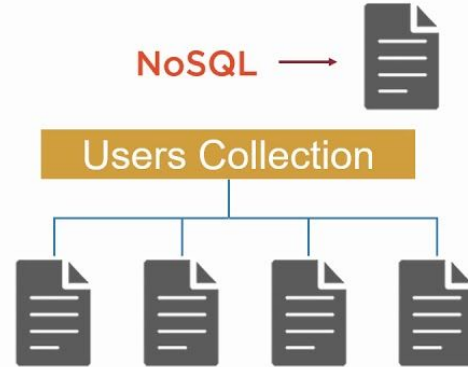
SQL

Table: Users

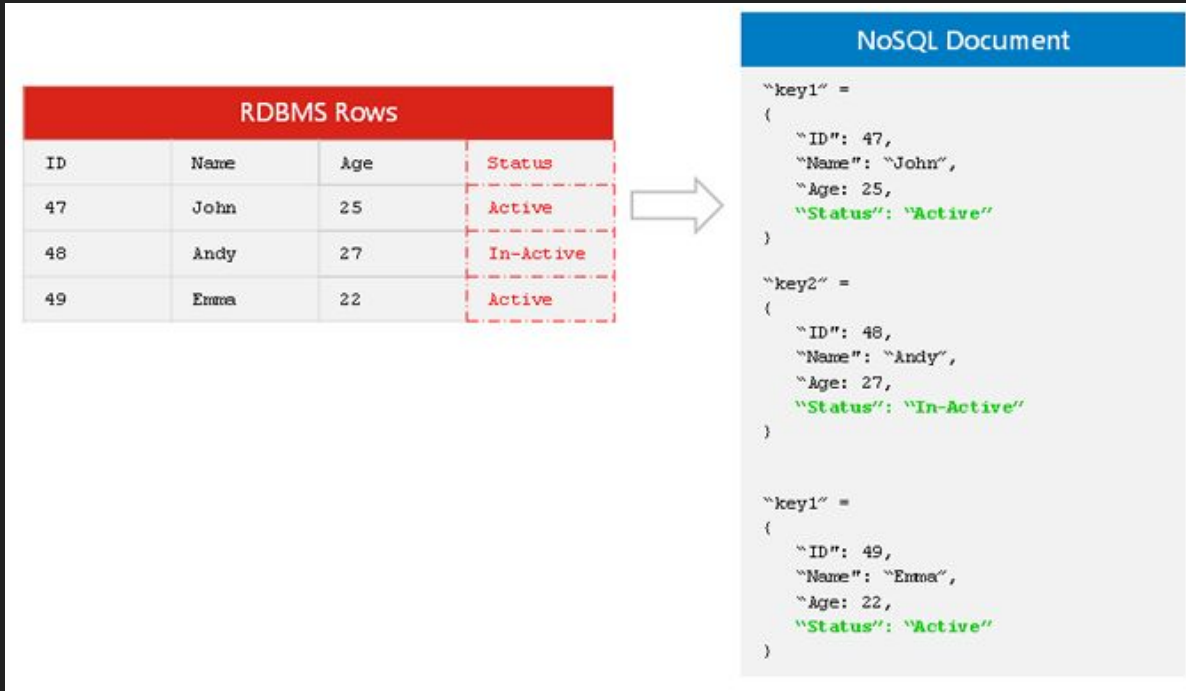
	id	name	createdAt	updatedAt
	Filter	Filter	Filter	Filter
1	1	Ashton	2018-08-12 1...	2018-08-12 1...
2	2	Jakeem	2018-08-12 1...	2018-08-12 1...
3	3	Uma	2018-08-12 1...	2018-08-12 1...

Relationships with JOINS

NoSQL →



ความแตกต่างระหว่าง SQL กับ MongoDB



ความแตกต่างระหว่าง SQL กับ MongoDB



<https://www.alachisoft.com/nosdb/why-nosql.html>

Break



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

รู้จักกับ JSON และ BSON

JSON (JavaScript Object Notation) เป็นรูปแบบการแลกเปลี่ยนหรือรับส่งข้อมูลในระบบคอมพิวเตอร์หรือแอปพลิเคชัน ในอดีตมีการแลกเปลี่ยนหรือรับส่งข้อมูลโดยใช้รูปแบบ xml แต่เนื่องจาก xml มีโครงสร้างข้อมูลที่มีความซับซ้อนและมีขนาดใหญ่จึงมีการใช้เปลี่ยนมาใช้รูปแบบ JSON แทน เนื่องจากมีขนาดเบาและมีโครงสร้างข้อมูลที่มนุษย์สามารถอ่านและเขียนได้ง่าย โดยรูปแบบของ JSON จะอยู่ในรูปแบบข้อความธรรมดา



รู้จักกับ JSON และ BSON

XML

```
<empinfo>
  <employees>
    <employee>
      <name>James Kirk</name>
      <age>40</age>
    </employee>
    <employee>
      <name>Jean-Luc Picard</name>
      <age>45</age>
    </employee>
    <employee>
      <name>Wesley Crusher</name>
      <age>27</age>
    </employee>
  </employees>
</empinfo>
```

JSON

```
{ "empinfo" :
  {
    "employees" : [
      {
        "name" : "James Kirk",
        "age" : 40,
      },
      {
        "name" : "Jean-Luc Picard",
        "age" : 45,
      },
      {
        "name" : "Wesley Crusher",
        "age" : 27,
      }
    ]
  }
}
```

<https://fahadhussaincs.blogspot.com/2018/09/xml-vs-json.html>

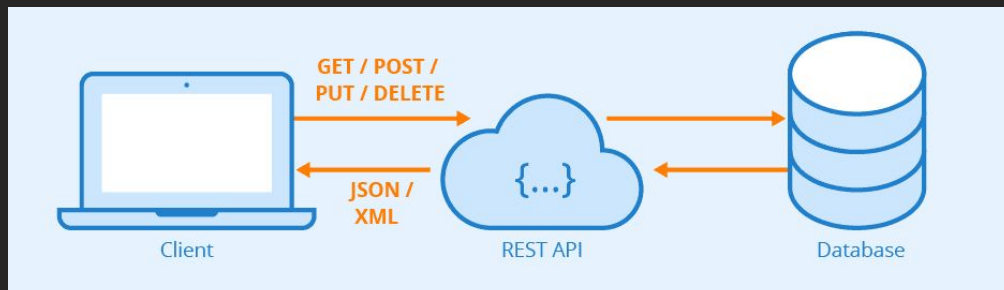


<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

รู้จักกับ JSON และ BSON



<https://www.datamounts.com/difference-rest-api-restful-api/>

JSON (JavaScript Object Notation) ในตอนเริ่มต้นถูกใช้ในการเขียนโปรแกรมด้วย JavaScript ในการแลกเปลี่ยนข้อมูลกับ Server ปัจจุบันนิยมนำ JSON มาใช้เป็นรูปแบบในการแลกเปลี่ยนข้อมูลในเว็บหรือแอปพลิเคชัน มีหลายภาษาที่รองรับการแลกเปลี่ยนข้อมูล เช่น Java , Python , C# เป็นต้น

รู้จักกับ JSON และ BSON

JSON รองรับการทำงานชนิดข้อมูลพื้นฐานทั้งหมด ได้แก่

- Number: ตัวเลข
- String: สตริงหรือชุดข้อความโดยใช้เครื่องหมาย double-quote (“)
- Boolean: ค่าบูลีน (True or False)
- Null: ค่าว่าง

รู้จักกับ JSON และ BSON

- Array: อาร์เรย์หรือชุดข้อมูลซึ่งจะเป็นชนิดข้อมูลใดก็ได้ ใช้สัญลักษณ์ [] เป็นตัวแสดงและคั่นสมาชิกในอาร์เรย์แต่ละค่าด้วยคอมม่า (,) เช่น [var1,var2]
- Object: ชุดข้อมูลที่เป็นคู่ Key-Value แบบสตริงใช้สัญลักษณ์ปีกกา {key1:value1,key2:value2} และคอมม่า (,) เป็นตัวแบ่งแต่ละคู่ และใช้โคลอน (:) เป็นตัวแบ่งฝั่งระหว่าง key และ value



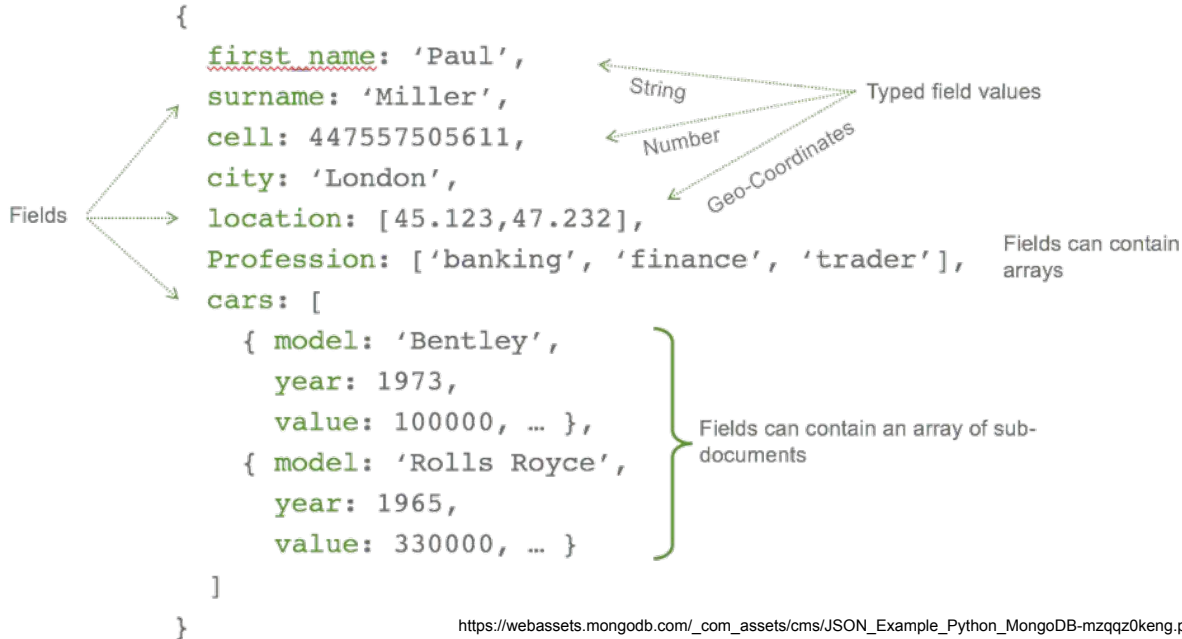
รู้จักกับ JSON และ BSON

ตัวอย่าง Object หรือ Associate Array

```
{  
  name: "sue",  
  age: 26,  
  status: "A",  
  groups: [ "news", "sports" ]  
}
```

← field: value
← field: value
← field: value
← field: value

จัดเก็บข้อมูลแบบ JSON ใน MongoDB



https://webassets.mongodb.com/_com_assets/cms/JSON_Example_Python_MongoDB-mzqqz0keng.png



<https://www.youtube.com/c/KongRuksiamOfficial/>



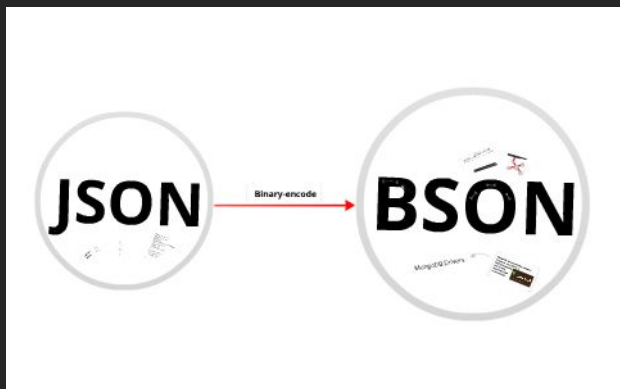
<https://www.facebook.com/KongRuksiamTutorial/>

จัดเก็บข้อมูลแบบ JSON ใน MongoDB

```
{ 1
  _id: "5373aadadcac133aad5b6660", 2
  name: "kaushal patel",
  email: "kp@example.com",
  courses: {
    course_name: "java",
    fees: "5000",
    duration: "3", 3
    professor: "g.r."
  }
}
```

1. เริ่มต้นเอกสารด้วยเครื่องหมายปีกกาเปิด-ปิด {}
2. ข้อมูลแต่ละฟิลด์คั่นด้วยเครื่องหมายคอมม่า (,)
3. สามารถสร้างเอกสารย่อยในเอกสารหลักได้โดยการใส่ข้อมูลในเครื่องหมายปีกกาเปิด-ปิด

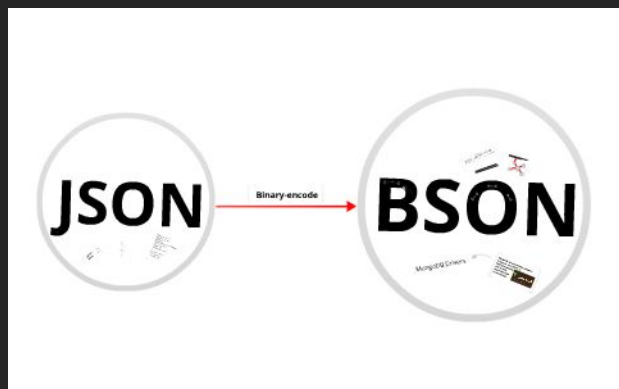
รู้จักกับ JSON และ BSON



ในฐานข้อมูล MongoDB จะทำการแปลงเอกสาร JSON โดยเข้ารหัสให้อยู่ในรูปแบบไบนารีหรือเรียกว่า **BSON** (Binary JSON) เพื่อขยายความสามารถให้ JSON มีชนิดข้อมูลเพิ่มมากขึ้น มีการเข้าและถอดรหัสด้วยภาษาที่แตกต่างกันได้



รู้จักกับ JSON และ BSON



BSON จะมีขนาดเบาและรับส่งข้อมูลได้เร็วเหมือนกับ JSON โดย MongoDB สามารถเข้าถึงข้อมูลหรือ Object ใน BSON เพื่อสร้าง Index ในการทำระบบสอบถามข้อมูล (Query) โดย BSON จะทำงานซ่อนอยู่เบื้องหลัง ส่วนการแสดงผลข้อมูลกับผู้ใช้จะเป็น

JSON



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

Break



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

ติดตั้งเครื่องมือพื้นฐาน

- **MongoDB Commutity** เป็น MongoDB Server รุ่น Open Source ที่ให้ดาวน์โหลดใช้งานได้ฟรี
- **MongoDB Compass** เป็นเครื่องมือในการจัดการข้อมูลใน รูปแบบ GUI ในการอำนวยความสะดวกในการวิเคราะห์ข้อมูล การดำเนินการกับข้อมูล (CRUD) รวมทั้งระบบสืบค้นและจัดการ Index เป็นต้น



การสร้างและลบฐานข้อมูล

- แสดงฐานข้อมูล

show dbs

- สร้างหรือเลือกฐานข้อมูล

use ชื่อฐานข้อมูล (ในกรณีที่ยังไม่มีฐานข้อมูลจะสร้างฐานข้อมูลขึ้นมา แต่ถ้ามีฐานข้อมูลจะเลือกฐานข้อมูลมาใช้)

- ลบฐานข้อมูล

db.dropDatabase() (ต้องเข้าไปทำงานในฐานข้อมูลก่อนค่อยลบ)



จัดการ Collection

- สร้าง Collection

use ชื่อฐานข้อมูล

db.createCollection(“ชื่อ Collection”)

- เปลี่ยนชื่อ Collection

db.ชื่อCollection.renameCollection(“ชื่อ Collection ใหม่”)

- ลบ Collection

db.ชื่อCollection.drop()



ชนิดข้อมูลใน MongoDB

ชนิดข้อมูล	รายละเอียด
String	จัดเก็บ String หรือชุดข้อความและรองรับ UTF-8 เขียนในสัญลักษณ์ “” (double quote)
Integer	จัดเก็บตัวเลขจำนวนเต็ม
Double	จัดเก็บตัวเลขทศนิยม
Boolean	จัดเก็บค่าบูลีน (True/False)
Null	จัดเก็บค่า Null หรือไม่มีค่า



ชนิดข้อมูลใน MongoDB

ชนิดข้อมูล	รายละเอียด
Object	จัดเก็บ document โดยเขียนในสัญลักษณ์ปีกกา {}
Array	จัดเก็บชุดข้อมูลโดยเขียนในวงเล็บ []
ObjectId	จัดเก็บ id ของ document (มีค่าไม่ซ้ำกัน)
Date	จัดเก็บวันที่และเวลา
Binary Data	จัดเก็บข้อมูลแบบไบนารี
Max / Min Key	เปรียบเทียบค่ามากกว่า / น้อยกว่า กับค่าข้อมูลอื่นๆ



การเพิ่ม Document

- เพิ่ม Document

db.ชื่อCollection.insertOne(<Document>)

- การเพิ่มหลาย Document

db.ชื่อCollection.insertMany([<Document>])

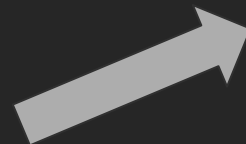


การเพิ่ม Document

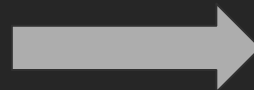
ฐานข้อมูล



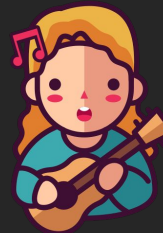
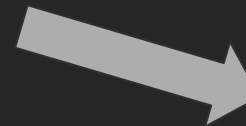
Employees
(collection)



document



document



document

การเพิ่ม Document (เตรียม Document)



document

```
{  
  name:"ก้อง",  
  age:30,  
  email:"kong@gmail.com",  
}
```



การเพิ่ม Document (เตรียม Document)



document

```
{  
  name:"ก้อง", string  
  age:30, integer  
  email:"kong@gmail.com", string  
}
```



การเพิ่ม Document

```
db.ชื่อCollection.insertOne(<Document>)
```

```
db.employees.insertOne(
```

```
{name:"ก้อง",age:30,email:"kong@gmail.com"}
```

```
)
```



การเพิ่ม Document

- เพิ่ม Document

db.ชื่อCollection.insertOne(<Document>)

- การเพิ่มหลาย Document

db.ชื่อCollection.insertMany([<Document>])



การเพิ่ม Document หลายรายการ (เตรียม Document)



document

```
{  
  name:"ก้อง",  
  age:30,  
  email:"kong@gmail.com",  
}
```



การเพิ่ม Document หลายรายการ (เตรียม Document)



document

```
{  
  name:"โจโจ้",  
  age:25,  
  email:"jojo@gmail.com",  
}
```



การเพิ่ม Document หลายรายการ



document

ใช้ [] ครอบ Document เช่น

[

<Document>,

<Document>

]



การเพิ่ม Document หลายรายการ

```
db.ชื่อCollection.insertMany([<Document>])
```

```
db.employees.insertMany(
```

```
[
```

```
{name:"ก้อง",age:30,email:"kong@gmail.com"},
```

```
{name:"โจโจ้",age:25,email:"jojo@gmail.com"}
```

```
]
```

```
)
```



บันทึกข้อมูลแบบ Array



document

```
{  
  name:"ก้อง", string  
  age:30, integer  
  email:"kong@gmail.com", string  
  social:["facebook","line","twitter"] Array  
}
```



บันทึกข้อมูลแบบ Object



document

```
{  
  name:"ก้อง",  
  age:30,  
  email:"kong@gmail.com",  
  general:{  
    weight:60.5,height:170,gender:"ชาย"  
  }  
}
```



การส่งออกและนำเข้า Collection



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

การสอบถามข้อมูล Document ใน Collection

- **findOne()**

สอบถามข้อมูล Document จาก Collection ตามเงื่อนไขที่กำหนดผลจากการสอบถามจะได้ผลลัพธ์เพียง Document เดียว หากเจอ Document หลายรายการจะเลือกเอารายการแรกมาใช้งาน ถ้าไม่เจอจะได้ค่า Null



การสอบถามข้อมูล Document ใน Collection

- **find()**

สอบถามข้อมูล Document จาก Collection ถ้าไม่กำหนดเงื่อนไขจะเลือก Document ทั้งหมดมาใช้งาน แต่สามารถกรอง (Filter) หรือเลือกข้อมูลภายใน Document ได้โดยระบุเงื่อนไข หรือ Parameter เข้าไป ถ้าเงื่อนไขไม่ตรง จะได้ Document เป็นค่า Null



การสอบถามข้อมูล Document ใน Collection

ข้อมูลพนักงาน (Employee.json) ประกอบด้วย

- ชื่อ (Name)
- เงินเดือน (Salary)
- แผนก / ตำแหน่งงาน (Department)
- ที่อยู่ (Address)
- ข้อมูลทั่วไป (General : Object)
- Social Media (Social : Array)



สอบถามข้อมูล Document แบบเงื่อนไขเดียว

- พนักงานที่มีเงินเดือน 15000 บาท
- พนักงานที่ทำงานแผนกโปรแกรมเมอร์
- พนักงานที่อาศัยอยู่กรุงเทพมหานคร



สอบถามข้อมูล Document แบบ 2 เงื่อนไข

- พนักงานที่อาศัยอยู่ กรุงเทพมหานคร และ มีเงินเดือน 15000
- พนักงานที่อาศัยอยู่ ระยอง และ เป็นโปรแกรมเมอร์
- พนักงานที่อาศัยอยู่ กรุงเทพมหานคร และ เป็นฝ่ายขาย



สอบถามข้อมูล Document แบบ Object

- พนักงานที่เป็นเพศชาย
- พนักงานที่เป็นเพศหญิง
- พนักงานที่เป็นเพศชายและทำงานเป็นโปรแกรมเมอร์
- พนักงานที่มีส่วนสูง 169 cm และทำงานฝ่ายขาย



สอบถามข้อมูล Document แบบ Array

- พนักงานที่ใช้ Facebook อย่างเดียว
- พนักงานที่ใช้ Facebook , Line
- พนักงานที่ใช้ Facebook , Line , Instagram
- พนักงานที่ใช้ Facebook , Line , Twitter



ตัวดำเนินการเปรียบเทียบ

ชื่อ	คำอธิบาย	การใช้งาน
\$eq	เท่ากับ (equal)	{field:{\$eq:value}} หรือ {field:value}
\$gt	มากกว่า (greater)	{field:{\$gt:value}}
\$gte	มากกว่าหรือเท่ากับ (greater or equal)	{field:{\$gte:value}}
\$lt	น้อยกว่า (less)	{field:{\$lt:value}}
\$lte	น้อยกว่าหรือเท่ากับ (less or equal)	{field:{\$lte:value}}
\$ne	ไม่เท่ากับ (not equal)	{field:{\$ne:value}}

ตัวดำเนินการเปรียบเทียบ

- พนักงานที่มีเงินเดือน 15,000 บาท
- พนักงานที่มีเงินเดือนมากกว่า 40,000 บาท
- พนักงานที่มีเงินเดือน 40,000 บาทขึ้นไป
- พนักงานที่มีเงินเดือนน้อยกว่า 30,000 บาท
- พนักงานที่มีเงินเดือน 40,000 บาทลงไป
- พนักงานที่ไม่ได้ทำงานเป็นโปรแกรมเมอร์



สอบถามข้อมูลด้วย In และ Not In

ชื่อ	คำอธิบาย	การใช้งาน
\$in	มีค่าใน Array	{field:{\$in:[value1,value2,...]}}
\$nin	ไม่มีค่าใน Array	{field:{\$nin:[value1,value2,...]}}

\$in คือการกำหนดเงื่อนไขในรูปแบบ Array สำหรับสอบถามข้อมูล ใช้งานร่วมกับ Field ที่เราสนใจและให้เลือกค่าใดค่าหนึ่งตามที่กำหนดใน Array ดังกล่าว ซึ่งสามารถกำหนดค่าในการเลือกสอบถามข้อมูลได้มากกว่า 1 ค่า

สอบถามข้อมูลด้วย In และ Not In

ชื่อ	คำอธิบาย	การใช้งาน
\$in	มีค่าใน Array	{field:{\$in:[value1,value2,...]}}
\$nin	ไม่มีค่าใน Array	{field:{\$nin:[value1,value2,...]}}

\$nin ตรงกันข้ามกับการใช้ \$in

สรุป

\$in สนใจข้อมูลใน Field ที่กำหนด เทียบกับรายการหรือสมาชิกใน Array

\$nin สนใจข้อมูลใน Field ที่กำหนด เทียบกับรายการหรือสมาชิกที่ไม่ได้อยู่

ใน Array

สอบถามข้อมูลด้วย In และ Not In

- พนักงานที่อยู่ในจังหวัด กรุงเทพมหานคร , ระยอง
- พนักงานที่ ไม่ได้อยู่ใน กรุงเทพมหานคร , ระยอง
- พนักงานที่อยู่แผนก ฝ่ายขาย , ฝ่ายการตลาด
- พนักงานที่ ไม่ได้อยู่แผนก ฝ่ายขาย , ฝ่ายการตลาด
- พนักงานที่มี ส่วนสูง 168 , 169 , 175
- พนักงานที่มีเงินเดือน 15,000 บาท , 40,000 บาท



ตัวดำเนินการตรรกศาสตร์

ชื่อ	คำอธิบาย
\$and	ตรงทั้งสองเงื่อนไข
\$or	ตรงกับเงื่อนไขใดเงื่อนไขหนึ่ง
\$nor	ไม่ตรงกับเงื่อนไขใดเลย
\$not	ตรงข้ามกับเงื่อนไขที่กำหนด

ตัวดำเนินการตรรกศาสตร์

\$and (และ)

เงื่อนไขที่ 1 และ เงื่อนไขที่ 2 เป็นจริง (ทั้งสองอย่าง)

\$or (หรือ)

เงื่อนไขที่ 1 หรือ เงื่อนไขที่ 2 เป็นจริง (อย่างใดอย่างหนึ่ง)

ตัวดำเนินการตรรกศาสตร์

\$not (ไม่)

ไม่ตรงกับเงื่อนไขที่กำหนด (ตรงข้ามเงื่อนไข)

\$nor (not or)

ไม่ตรงกับเงื่อนไขทั้ง 2 ที่กำหนดขึ้นมา



โจทย์สอบถามข้อมูล

- พนักงานที่มีเงินเดือนมากกว่า 15,000 และ น้อยกว่า 45,000 บาท

เงื่อนไขที่ 1

เงินเดือนมากกว่า 15,000 บาท

เงื่อนไขที่ 2

เงินเดือนน้อยกว่า 45,000 บาท



โจทย์สอบถามข้อมูล

- พนักงานที่อยู่แผนกวิศวกร หรือ พนักงานที่เป็นเพศชาย

เงื่อนไขที่ 1

พนักงานที่อยู่แผนกวิศวกร

เงื่อนไขที่ 2

พนักงานที่เป็นเพศชาย



โจทย์สอบถามข้อมูล

- พนักงานที่ไม่ได้อยู่ในกรุงเทพมหานคร และ ไม่ได้ทำงานแผนกฝ่ายขาย

เงื่อนไขที่ 1

พนักงานที่ไม่ได้อยู่ในกรุงเทพมหานคร

เงื่อนไขที่ 2

พนักงานที่ไม่ได้ทำงานแผนกฝ่ายขาย



สอบถามข้อมูล

- พนักงานที่มีน้ำหนักตัวไม่มากกว่าหรือเท่ากับ 52 (น้ำหนักต่ำกว่า 52)



อัปเดต Document

มี 2 รูปแบบ

- อัปเดต Document รายการเดียว
- อัปเดต Document หลายรายการ

อัปเดต Document

โครงสร้างคำสั่ง

db.ชื่อCollection.updateOne({1,2,3})

- 1.Document ที่ต้องการแก้ไข
- 2.ระบุข้อมูลที่ต้องการแก้ไข
- 3.ตัวเลือกเพิ่มเติม



อัปเดต Document

การระบุข้อมูลที่ต้องการแก้ไข [2]

`{ $set: {ชื่อฟิลด์ : ค่าที่ต้องการแก้ไข} }`

กำหนดเงินเดือนพนักงานที่ชื่อสมชายเป็น 50,000 บาท

`updateOne({name:"สมชาย"}, { $set: {salary : 50000} })`



อัปเดต Document หลายรายการ

โครงสร้างคำสั่ง

db.ชื่อCollection.updateMany({1,2,3})

- 1.Document ที่ต้องการแก้ไข
- 2.ระบุข้อมูลที่ต้องการแก้ไข
- 3.ตัวเลือกเพิ่มเติม



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

อัปเดต Document หลายรายการ

การระบุข้อมูลที่ต้องการแก้ไข [2]

{\$set:{ชื่อฟิลด์ : ค่าที่ต้องการแก้ไข}}

พนักงานที่อยู่กรุงเทพมหานครทุกคนมีเงินเดือน 70,000 บาท

updateMany({address:"กรุงเทพมหานคร"},{\$set:{salary :70000}})



ลบ Document

มี 2 รูปแบบ

- ลบ Document รายการเดียว
- ลบ Document หลายรายการ

การลบ Document

โครงสร้างคำสั่ง

```
db.ชื่อCollection.deleteOne(filter)
```

ลบพนักงานที่ชื่อสมชาย

```
deleteOne({name:"สมชาย"})
```



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

การลบ Document หลายรายการ

โครงสร้างคำสั่ง

db.ชื่อCollection.deleteMany(filter)

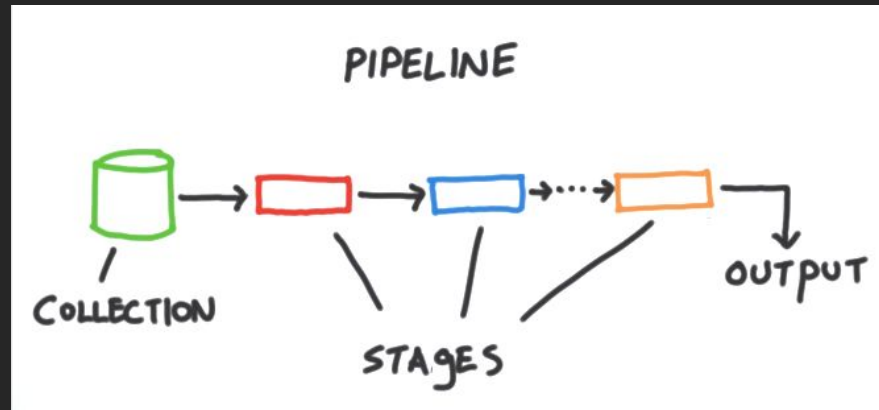
ลบพนักงานที่อยู่จังหวัดกรุงเทพมหานคร

deleteMany({address:"กรุงเทพมหานคร"})



รู้จักกับ Aggregation & Pipeline

ไปป์ไลน์ คือ กระบวนการนำข้อมูล
จากฐานข้อมูลมารวมเข้าด้วยกันและ
วิเคราะห์หาผลลัพธ์ที่ต้องการ โดย
จะนำเอา Document ต่างๆ จาก
Collection ไปเข้าสู่กระบวนการ
กลั่นกรอง (Stage) เพื่อวิเคราะห์หา
ผลลัพธ์ (Output)



<https://i.stack.imgur.com/S555v.png>



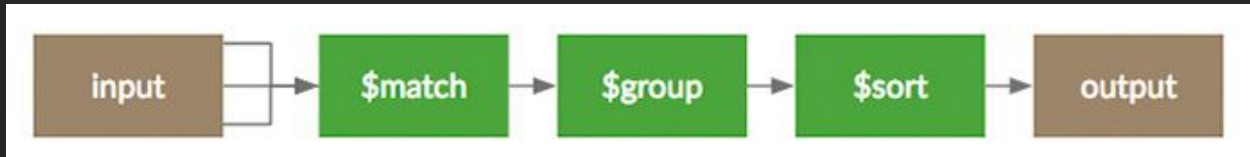
<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

รู้จักกับ Aggregation & Pipeline

ไพล์ไลน์จะใช้เมธอดหรือคำสั่ง aggregate() สำหรับรวบรวม Document ภายใน Collection ซึ่งด้านในจะประกอบด้วย Array ของ Stage โดย Document ทั้งหมดก็จะถูกดำเนินการกลั่นกรองข้อมูลด้านในกลุ่มของ Stage ดังกล่าวนั่นเอง



<https://intellipaath.com/mediaFiles/2015/08/Aggregation.jpg>



<https://www.youtube.com/c/KongRuksiamOfficial/>

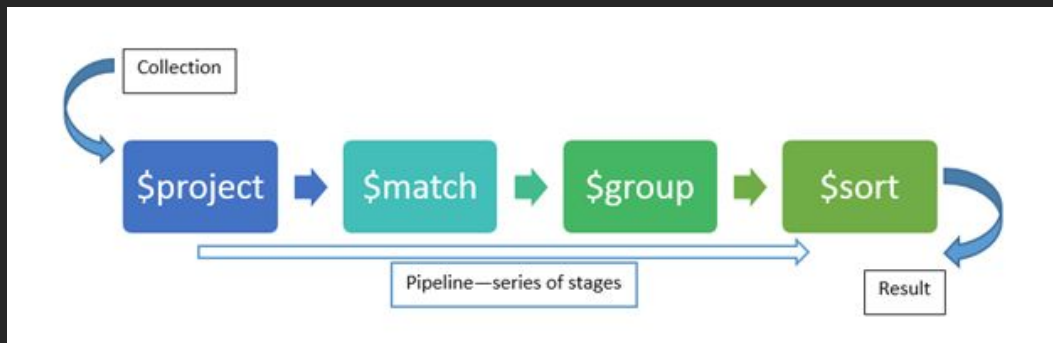


<https://www.facebook.com/KongRuksiamTutorial/>

รู้จักกับ Aggregation & Pipeline

โครงสร้างคำสั่ง

db.ชื่อCollection.aggregate([stage])



<https://www.codeproject.com/KB/database/1149682/image001.png>



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

กระบวนการกลั่นกรอง (Stage)

ชื่อ Stage	คำอธิบาย
\$match	กรองเอาเฉพาะ Document ตรงตามเงื่อนไขที่กำหนด
\$group	จัดกลุ่ม Document และคำนวณค่าเก็บใน Output
\$project	แสดงข้อมูล Document เอาเฉพาะ Field ที่กำหนด
\$sort	จัดเรียง Document
\$skip	ข้าม Document ตามจำนวนที่ระบุ
\$limit	จำกัดการแสดงผลจำนวน Document
\$unwind	แยกสมาชิก Field Array ออกเป็น Document
\$count	นับจำนวน Document
\$lookup	ดูข้อมูล Document ที่มี Collection ต่างกัน

เปรียบเทียบ SQL กับ Stage

SQL	Stage
WHERE	\$match
GROUP BY	\$group
SELECT	\$project
ORDER BY	\$sort
LIMIT	\$limit
JOIN	\$lookup

สอบถามข้อมูลด้วย Projection (SELECT)

เป็นรูปแบบการสอบถามข้อมูลและให้แสดงผลเฉพาะฟิลด์ที่ต้องการ
และส่งออกไปโดยการแสดงฟิลด์จะกำหนดเป็นหมายเลข 1 ส่วนการซ่อน
หรือปิดฟิลด์จะใช้หมายเลข 0

โครงสร้าง

{ \$project: { ชื่อฟิลด์ : หมายเลข, ชื่อฟิลด์ : หมายเลข } }



สอบถามข้อมูลด้วย Projection (SELECT)

- แสดงรหัสพนักงานและชื่อพนักงาน
- แสดงชื่อพนักงานอย่างเดียว
- แสดงชื่อพนักงานและแผนก
- แสดงชื่อพนักงานและเงินเดือน
- แสดงชื่อพนักงาน แผนก เงินเดือน ที่อยู่



สอบถามข้อมูลด้วย \$match (WHERE)

เป็นรูปแบบการสอบถามข้อมูลโดยเลือกเอาเฉพาะ Document
ที่ผ่านเงื่อนไขที่ระบุแล้วส่งออกไป

โครงสร้าง

{ \$match: { ชื่อฟิลด์ : เงื่อนไข } }



สอบถามข้อมูลด้วย \$match (WHERE)

- แสดงข้อมูลพนักงานที่มีเงินเดือน 15,000 บาท

ใช้งาน (\$match & \$project)

- แสดงชื่อ , แผนก ของพนักงานที่มีเงินเดือน 15,000 บาท
- แสดงชื่อ , แผนก , เงินเดือน ของพนักงานที่มีเงินเดือน 15,000 บาท
- แสดงชื่อ , แผนก , เงินเดือน ของพนักงานที่มีเงินเดือน
มากกว่า 15,000 บาท



นับจำนวนด้วย \$count

เป็นการนับจำนวน Document ใน Collection และส่งออกเป็น
จำนวน Document ที่นับได้

โครงสร้าง

{ \$count: ชื่อฟิลด์ที่แสดงการนับจำนวน <string> }

นับจำนวนด้วย \$count

- จำนวนพนักงานที่มีเงินเดือน 15,000 บาท
- จำนวนพนักงานที่มีเงินเดือน 40,000 บาท
- จำนวนพนักงานที่มีเงินเดือนมากกว่า 50,000 บาท



เรียงลำดับด้วย \$sort (ORDER BY)

การดึงข้อมูลใน Collection จะมีลำดับในการจัดเก็บข้อมูลถ้าเราต้องการอยากจะเรียงลำดับข้อมูลใหม่จะใช้ \$sort โดยกำหนดค่าเป็นหมายเลข ดังนี้

1 คือ เรียงจากน้อยไปมาก

-1 คือ เรียงจากมากไปน้อย

เรียงลำดับด้วย \$sort (ORDER BY)

โครงสร้าง

{ \$sort: { ชื่อฟิลด์ : หมายเลข, ชื่อฟิลด์ : หมายเลข } }

หมายเลข 1 คือ เรียงจากน้อยไปมาก / (พหุคูณชนะ (ก-ฮ) - สระ)

หมายเลข-1 คือ เรียงจากมากไปน้อย / (สระ - พหุคูณชนะ (ก-ฮ))

เรียงลำดับด้วย \$sort (ORDER BY)

- เรียงลำดับเงินเดือนพนักงานจากน้อยไปมาก
- เรียงลำดับเงินเดือนพนักงานจากมากไปน้อย
- เรียงลำดับชื่อของพนักงานจาก ก-ฮ (พยัญชนะ- สระ)
- เรียงลำดับชื่อของพนักงานจาก ฮ- ก (สระ - พยัญชนะ)
- เรียงลำดับที่อยู่ของพนักงาน



จำกัดข้อมูลด้วย \$limit (LIMIT)

เนื่องจากฐานข้อมูลมีจำนวนข้อมูลจำนวนมาก หากดึงข้อมูล Document มาทั้งหมดอาจจะไม่สะดวกในการอ่านรายละเอียดข้อมูล ต้องมีการจำกัดการแสดงผลจำนวน Document โดยใช้ \$limit

โครงสร้าง

{ \$limit:จำนวน Document <integer> }



ข้าม Document ที่ไม่ต้องการด้วย \$skip

เนื่องจากฐานข้อมูลมีจำนวน Document จำนวนมาก หากดึงข้อมูล Document มาทั้งหมดแล้วต้องการอยากจะข้ามไปทำงานที่ Document ที่ต้องการจะใช้ \$skip มาจัดการส่วนนี้

โครงสร้าง

{ \$skip:จำนวน Document <integer> }



จัดกลุ่มข้อมูลด้วย \$group (GROUP BY)

เป็นการจัดกลุ่มข้อมูลของ Document ไว้เป็นกลุ่มเดียวกันโดยมี _id
เก็บรายการข้อมูลแต่ละกลุ่มที่ไม่ซ้ำกันและยังสามารถเพิ่มฟิลด์ในการ
คำนวณหาผลลัพธ์ภายในกลุ่มข้อมูลที่สามารถสร้างขึ้นได้ โดยใช้งานกลุ่มร่วมกับ
ฟังก์ชันทางสถิติ

จัดกลุ่มข้อมูลด้วย \$group (GROUP BY)

โครงสร้าง

`{ $group: { ชื่อฟิลด์สำหรับจัดกลุ่ม : ตัวดำเนินการทางสถิติ } }`



ตัวดำเนินการทางสถิติทำงานร่วมกับ \$group

ชื่อ	ความหมาย
\$sum	หาผลรวม Document ภายในกลุ่ม
\$avg	หาค่าเฉลี่ย Document ภายในกลุ่ม
\$min	หาค่าต่ำสุด Document ภายในกลุ่ม
\$max	หาค่าสูงสุด Document ภายในกลุ่ม
\$first	หาค่า Document ลำดับแรกในกลุ่ม
\$last	หาค่า Document ลำดับท้ายในกลุ่ม
\$push	แทรกค่า Array ไปยังผลลัพธ์ในกลุ่ม
\$addToSet	แทรกค่า Array ไปยังผลลัพธ์ในกลุ่มโดยไม่ซ้ำกัน

จัดกลุ่มข้อมูลด้วย \$group (GROUP BY)

ตัวอย่างการนับจำนวนทั้งหมด

```
{ $group:{ _id:null , myField:{$count:{}} } }
```

- **_id** คือ ระบุชื่อ Field ว่าต้องการแบ่งกลุ่มโดยใช้ Field ไหน เป็น Field อ้างอิง ถ้าระบุ null คือ ไม่มี Field อ้างอิง
- **myField** คือ ชื่อฟิลด์ที่เก็บผลลัพธ์เอาไว้และใช้ \$count มานับจำนวน



จัดกลุ่มข้อมูลด้วย \$group (GROUP BY)

- นับจำนวนพนักงานแต่ละแผนก
- นับจำนวนพนักงานแต่ละจังหวัด
- หาผลรวมเงินเดือนพนักงานทุกคน
- หาผลรวมเงินเดือนพนักงานโดยจำแนกเป็นแต่ละแผนก
- หาผลรวมเงินเดือนพนักงานที่อยู่แผนกฝ่ายขาย



จัดกลุ่มข้อมูลด้วย \$group (GROUP BY)

- หาเงินเดือนเฉลี่ยพนักงานแต่ละแผนก
- หาเงินเดือนสูงสุดของพนักงานแต่ละจังหวัด (ที่อยู่)
- หาเงินเดือนต่ำสุดของพนักงานแต่ละจังหวัด (ที่อยู่)



เพิ่มข้อมูลกลุ่มด้วย \$push , \$addToSet

ชื่อ	ความหมาย
\$sum	หาผลรวม Document ภายในกลุ่ม
\$avg	หาค่าเฉลี่ย Document ภายในกลุ่ม
\$min	หาค่าต่ำสุด Document ภายในกลุ่ม
\$max	หาค่าสูงสุด Document ภายในกลุ่ม
\$first	หาค่า Document ลำดับแรกในกลุ่ม
\$last	หาค่า Document ลำดับท้ายในกลุ่ม
\$push	แทรกค่า Array ไปยังผลลัพธ์ในกลุ่ม
\$addToSet	แทรกค่า Array ไปยังผลลัพธ์ในกลุ่มโดยไม่ซ้ำกัน

เพิ่มข้อมูลกลุ่มด้วย \$push , \$addToSet

- หาผลรวมเงินเดือนพนักงานโดยจำแนกเป็นแต่ละแผนกและให้ระบุชื่อของพนักงานพร้อมเงินเดือนแต่ละคน
- หาผลรวมเงินเดือนพนักงานโดยจำแนกเป็นแต่ละแผนกและให้ระบุกลุ่มชื่อพนักงาน



เชื่อม Collection ด้วย \$lookup

เป็นการสอบถามข้อมูล Document ที่เก็บข้อมูลใน Collection ที่แตกต่างกันโดยกำหนดให้ Collection หลักเชื่อมกับ Collection ด้านนอก ซึ่ง Collection ดังกล่าวต้องอยู่ในฐานข้อมูลเดียวกัน ผลลัพธ์ที่ได้ คือ กลุ่ม Document ที่เชื่อมโยงข้อมูลระหว่าง Collection นั้นเอง



เชื่อม Collection ด้วย \$lookup

ชื่อพนักงาน (name)	รหัสแผนก (department_id)
สมชาย	DP001
ก้อง	DP002
โจโจ้	DP003
แจ่ม	DP001

Employees

รหัสแผนก (id)	ชื่อแผนก (name)
DP001	โปรแกรมเมอร์
DP002	ฝ่ายการตลาด
DP003	ฝ่ายขาย

Departments

เชื่อม Collection ด้วย \$lookup



ชื่อพนักงาน (name)	รหัสแผนก (department_id)
สมชาย	DP001
ก้อง	DP002
โจโจ้	DP003
แจ่ม	DP001

Employees

รหัสแผนก (id)	ชื่อแผนก (name)
DP001	โปรแกรมเมอร์
DP002	ฝ่ายการตลาด
DP003	ฝ่ายขาย

Departments

เชื่อม Collection ด้วย \$lookup

localField

foreignField

ชื่อพนักงาน (name)	รหัสแผนก (department_id)
สมชาย	DP001
ก้อง	DP002
โจโจ้	DP003
แจ่ม	DP001

Employees

รหัสแผนก (id)	ชื่อแผนก (name)
DP001	โปรแกรมเมอร์
DP002	ฝ่ายการตลาด
DP003	ฝ่ายขาย

Departments

เชื่อม Collection ด้วย \$lookup

โครงสร้าง

```
{ $lookup: {  
  from : "ชื่อ Collection ด้านนอกที่เชื่อมโยง",  
  localField : "Field จาก Collection หลัก",  
  foreignField : "Field Collection ที่เชื่อมโยง",  
  as : "ชื่อ Field ที่เก็บผลลัพธ์"  
}
```

```
}
```

