

# เทคโนโลยีการจัดการข้อมูลแบบใหม่ด้วย

# NoSQL

อาจารย์อดิศักดิ์ เลือสมับ

adisak@tni.ac.th

คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีไทย-ญี่ปุ่น



**ปัจจุบัน** ปริมาณข้อมูลบนอินเทอร์เน็ตมีมากมาย และมีแนวโน้มจะมากขึ้นโดยตลอด ตัวอย่างที่ชัดเจน เช่น เว็บไซต์ในกลุ่ม Social Network และ E-Commerce ทั้งหลาย ท่านผู้อ่านเคยคิดกันเล่นๆ ไหมว่าในแต่ละวันมีคนโพสต์ข้อความ รูปภาพ หรือวิดีโอ บน Facebook กันมากแค่ไหนหรือมีการทำธุรกรรมซื้อขายบน Amazon นาทีละกี่รายการ แล้วท่านผู้อ่านเคยสงสัยต่อไปอีกไหมว่า ข้อมูลที่น่าจะมีมากมายขนาดนั้น หรือที่เราเรียกกันในศัพท์เฉพาะว่า Big Data เค้าจัดเก็บกันอย่างไร? วันนี้เราจะมาทำความรู้จักกับเทคโนโลยีการจัดการข้อมูลในรูปแบบใหม่ที่เรียกว่า NoSQL (อ่านว่า โน-เอส-คิว-แอล) ซึ่งเป็นเทคโนโลยีที่แม้แต่ไพร่พลของเฮาส์คูเปอร์ (PwC) บริษัทผู้ตรวจสอบบัญชีระดับโลกยังมองว่าเป็นเทรนด์ดีจิดอลในปี 2015 [1]

โน-เอส-คิว-แอล มาจากคำว่า Not Only SQL แปลตรงๆ ก็คือ ไม่ใช่แค่ เอส-คิว-แอล ปัญหา คือ เรายังหรือไม่ว่า SQL คืออะไร? สำหรับคนที่เรียน หรือทำงานในสาย IT คงตอบได้ไม่ยากว่า SQL (Structured Query Language) นั้นเป็นภาษาที่ใช้ในการเขียนโปรแกรม เพื่อจัดการกับฐานข้อมูลเชิงสัมพันธ์ (Relational Database) ซึ่งเป็นมาตรฐานที่ใช้กันมานานหลายสิบปี และแอปพลิเคชันที่เราใช้กันอยู่ในโลกปัจจุบันส่วนใหญ่ออกแบบโดยใช้ Relational Database กันเกือบทั้งนั้น

ปัญหาคือ ทำไมเว็บไซต์ขนาดใหญ่ เช่น Google, Facebook (ส่วน Inbox Messenger), Twitter, Amazon และอื่นๆ ถึงได้หันหลังให้กับสิ่งที่เป็นมาตรฐานที่ใช้กันมาอย่างยาวนานแบบ Relational

Database หันมาพัฒนา NoSQL Database แทน สาเหตุก็เพราะ NoSQL มีคุณสมบัติที่ถูกพัฒนาขึ้นเพื่อแก้ปัญหของการใช้ SQL แบบเดิมกับข้อมูลปริมาณมากๆ (เน้นย้ำนะครับว่าใช้กับข้อมูลปริมาณมากๆ) ที่สำคัญ 3 ข้อดังนี้

## 1. Dynamic Schema

ข้อนี้สำหรับผู้เขียนถือว่า Amazing ที่สุด เพราะปรกติการจัดเก็บข้อมูลด้วย Relational Database จะต้องมีกำหนดว่าฐานข้อมูลจะเก็บอะไรบ้าง มีความสัมพันธ์ระหว่างกันอย่างไร ที่เราเรียกกันว่า Schema โดยเมื่อต้องการจัดเก็บข้อมูลบางรายการเพิ่มเติมในภายหลัง ก็จะต้องแก้ไข Schema ซึ่งในทางปฏิบัติจะต้องหยุดการทำงานของระบบก่อน (ลองจินตนาการว่าคุณเป็นเจ้าของเว็บ E-Commerce ที่มียอดขายชั่วโมงละหลายสิบล้าน คุณอยากหยุดระบบใหม่สักวินาทีไหม?) แต่สำหรับ NoSQL Database นั้นไม่จำเป็นต้องมี Schema ที่ตายตัว หรือจะพูดให้ถูกจริง ๆ คือ ไม่ต้องมี Schema ก่อนที่จะจัดเก็บข้อมูลก็ได้ โดยข้อมูลสามารถจัดเก็บได้ตามต้องการ จะเพิ่มหรือลดรายการใดๆ ก็ไม่มีผลกระทบกับระบบทั้งสิ้น

## 2. Auto-Sharding

Sharding คือ การแบ่งข้อมูลเก็บเป็นส่วนๆ โดยเมื่อข้อมูลมีขนาดใหญ่ เราอาจจะแยกเก็บข้อมูลเป็นส่วนๆ เช่น สินค้าชนิดที่ 1-10 เก็บข้อมูลที่ Server A และสินค้าชนิดที่ 11-20 เก็บข้อมูลที่ Server



B (จากตัวอย่างนี้ก็ประกอบไปด้วย 2 shard) ซึ่งการกระจายข้อมูลออกไปเก็บในหลายๆ Server นี้มีข้อดีคือ ประหยัดต้นทุนในการขยายระบบ เพราะสามารถนำ Server ปกติทั่วไปมาใช้งานได้ ไม่จำเป็นต้องเป็นเกรด Enterprise Server และยังช่วยเพิ่มประสิทธิภาพการอ่าน-เขียนข้อมูลปริมาณมากๆ

โดยการทำ Sharding เช่นนี้ ถ้าเป็น Relational Database ผู้พัฒนาจะต้องการเขียนโปรแกรมในการกระจายข้อมูลเอง แต่ใน NoSQL Database จะทำการกระจายข้อมูลไปยัง Server ต่างๆ ให้โดยอัตโนมัติ

### 3. Integrated Caching

SQL VS NoSQL	
<p><b>Pros</b></p> <p>Relational databases are good at structured data and transactional, high-performance workloads.</p> <p>Offerings are proven and mature with a wide variety of tools available.</p> <p><b>Cons</b></p> <p>Can be difficult to scale.</p> <p>Fixed schema for organizing data.</p> <p><b>EXAMPLES</b></p> <p>MySQL, PostgreSQL, SQL Server, Oracle</p>	<p><b>Pros</b></p> <p>Good for non-relational data. Schema-less architecture allows for frequent changes to the database and easy addition of varied data to the system.</p> <p>Easily scalable, runs well on distributed systems (the cloud).</p> <p><b>Cons</b></p> <p>Installation, management and toolsets still maturing.</p> <p>Can have slower response time.</p> <p><b>EXAMPLES</b></p> <p>Amazon DynamoDB, MongoDB, Couchbase, Riak</p>

#### เปรียบเทียบ SQL กับ NoSQL

NoSQL Database จะจัดเก็บข้อมูลที่ใช้งานบ่อยๆ เข้าไว้ใน Memory (RAM) ซึ่งเป็นคุณสมบัติเด่นของ NoSQL ที่รวม Caching ไว้ในตัวเอง ซึ่งคุณสมบัตินี้จะทำให้การทำงานกับข้อมูลมีความรวดเร็วขึ้น

### NoSQL Database ในปัจจุบันมี 4 รูปแบบหลัก คือ

- 1. Key-Value Store** คือ เก็บข้อมูลโดยการระบุ Key และ Value ดึงข้อมูลโดยการระบุ Key แล้ว Database จะคืน Value กลับมาให้ รูปแบบนี้เหมาะกับการเก็บข้อมูลที่ไม่ซับซ้อน และต้องการความเร็วสูง เช่น การเก็บข้อมูล Session ของ Web app
- 2. Wide Column Store** นอกจากจะเก็บข้อมูลโดยการเพิ่ม Record แล้วยังเพิ่ม Column ได้อีกด้วย เหมาะกับระบบที่ต้องการ distribute และ scale ใหญ่มากๆ
- 3. Document Store** พัฒนาเพิ่มจาก key value store คือ สามารถเพิ่ม index เพื่อค้นหาจากตัวข้อมูล แตกต่างจาก key value ที่ต้องค้นหาจาก key เท่านั้น เหมาะกับการเก็บข้อมูลทั่วไป
- 4. Graph Store** รูปแบบนี้แตกต่างจาก 3 ตัวด้านบน เป็น

ชนิดเดียวที่รู้จักความสัมพันธ์ และอาจจะซับซ้อนเกินกว่าที่ relational database จะรองรับได้ด้วยซ้ำ เช่น เก็บข้อมูล social network ว่าใครเป็น friend ใคร การค้นหาที่สามารถระบุได้ว่าต้องการเพื่อนของเพื่อนของเพื่อน ในระดับชั้นไหน เป็นต้น

Type	Examples
Key-Value Store	redis, riak
Wide Column Store	HBASE, Cassandra
Document Store	mongoDB, CouchDB
Graph Store	Neo4j, the graph database

Increasing Data Complexity

ตัวอย่าง Product NoSQL Database ในรูปแบบต่างๆ ที่เป็นที่ยอมรับในปัจจุบัน

ที่นี้ประเด็นสำคัญ เราควรให้ความสำคัญกับ NoSQL มากแค่ไหน ? เริ่มต้นต้องเข้าใจก่อนว่า NoSQL ไม่ใช่มาแทน SQL นะครับ ไม่ได้หมายความว่าเราควรจะเปลี่ยนแอปพลิเคชันของเราไปใช้ NoSQL Database ให้หมดเพราะมันใหม่กว่า เพราะจริงๆ แล้วสองอย่างนี้มันเสริมกัน ในระบบใหญ่ๆ อาจจะมีทั้ง Relational และ NoSQL ก็ได้ แต่ขอไฮไลท์ไว้อย่างนึงเลยว่า NoSQL นั้นไม่เหมาะกับงานที่มีความซับซ้อนสูง มีความสัมพันธ์เยอะ เช่น ระบบการเงิน แต่เหมาะกับงานที่ไม่ได้ซับซ้อนมาก เช่น Search Engine, การทำ Log File ของแอปพลิเคชัน เป็นต้น

ดังนั้นหลักจริงๆ ถ้าจะใช้ SQL NoSQL ก็ให้ดูว่าคุณต้องการประโยชน์จากคุณสมบัติ 3 ข้อข้างบนของ NoSQL หรือไม่? ถ้าใช่ ก็สมควรอย่างยิ่งที่จะเริ่มศึกษา NoSQL ได้แล้ว ทั้งนี้เพื่อเพิ่มประสิทธิภาพของแอปพลิเคชัน และลดรายจ่ายในการซื้อ Server ขององค์กร แต่ในทางกลับกัน ถ้าไม่ต้องการประโยชน์ดังกล่าว หรือระบบเดิมยังรองรับการใช้งานได้อย่างมีประสิทธิภาพ ก็ไม่ต้องรีบเปลี่ยนหรอกครับ เพราะเทคโนโลยีใหม่ที่ยังไม่เสถียรมากนักอย่าง NoSQL เอาแค่ปัญหาเรื่อง ระบบ Admin ของ Database ที่แย่มากๆ กับ เรื่อง Support ที่เวลามีปัญหาแล้วจะหาแหล่งข้อมูลหรือผู้เชี่ยวชาญจริงๆ ได้ลำบาก เอาแค่สองข้อนี้ก็เหนื่อยแล้วครับ