



Introduction to NoSQL

By Asst. Prof. Dr. Praphan Pavarangkoon, Dr. Sirasit Lochanachit,
and Asst. Prof. Dr. Taravichet Titijaroonroj

Course Introduction

Course: 06016414 and 06026207

Course Title: NoSQL Database Systems

Duration: 17 weeks (including Midterm and Final examination weeks)

Credit Hours: 3 (2-2-5) credit hours

Lecture: 30 hours

Practice/Computer Lab: 30 hours

Self-Study: 75 hours

Lecturers

Asst. Prof. Dr. Praphan Pavarangkoon, Dr. Sirasit Lochanachit, and Asst. Prof. Dr. Taravichet Titijaroonroj

Course Date/Time/Room

This class meets every **Tuesday at 13:00 – 17:00 (4 hours/week)**,

Wednesday at 9:00 – 13:00 (4 hours/week), and **Wednesday at 14:00 – 18:00 (4 hours/week)**.

Microsoft Teams

https://teams.microsoft.com/l/team/19%3A2uielEOaMZB_WwDV-ao2A5nlgrG9ExKxpiWyDfN-E5w1%40thread.tacv2/conversations?groupId=3368111f-d859-4b98-8ac1-87e16aad31a7&tenantId=fd206715-7509-4ae5-9b96-76bb97886a84

Team code: 9drby5q



Course Description

แนะนำระบบฐานข้อมูลแบบไม่ใช่เชิงสัมพันธ์ ความแตกต่างจากระบบฐานข้อมูลเชิงสัมพันธ์ ข้อดีและข้อเสียของฐานข้อมูลแบบไม่ใช่เชิงสัมพันธ์ ชนิดและโครงสร้างฐานข้อมูลแบบไม่ใช่เชิงสัมพันธ์ แบบคอลัมน์ แบบคีย์แวล्यू แบบเอกสาร และแบบกราฟ ตัวอย่างการออกแบบฐานข้อมูลแบบไม่ใช่เชิงสัมพันธ์ การใช้ภาษาเจสั่นกับฐานข้อมูลแบบไม่ใช่เชิงสัมพันธ์ แนะนำภาษาเจสั่น ไวยากรณ์ของเจสั่น ชนิดข้อมูลของเจสั่น

Topics and Details of 17 Weeks

Week	Date	Topics and Details	Lecturer
1	26,27/11	Introduction to NoSQL	Praphan
2	3,4/12	Graph - 1: Introduction to Graph and Graph Database	Sirasit
3	10,11/12	Graph - 2: Introduction to Cypher	Sirasit
4	17,18/12	Graph - 3: Graph Queries	Sirasit
5	24,25/12	Graph - 4: Creating Nodes and Relationships	Sirasit
6	31/12,1/1	Graph - 5: Reading CSV Data	Sirasit
7	7,8/1	Graph - 6: Graph Algorithms	Sirasit
8	14,15/1	Column Family - 1: Basic Components of Column Family	Praphan
9	<p>Midterm Examination</p> <p>Saturday 25 January at 13:30 – 16:30</p>		

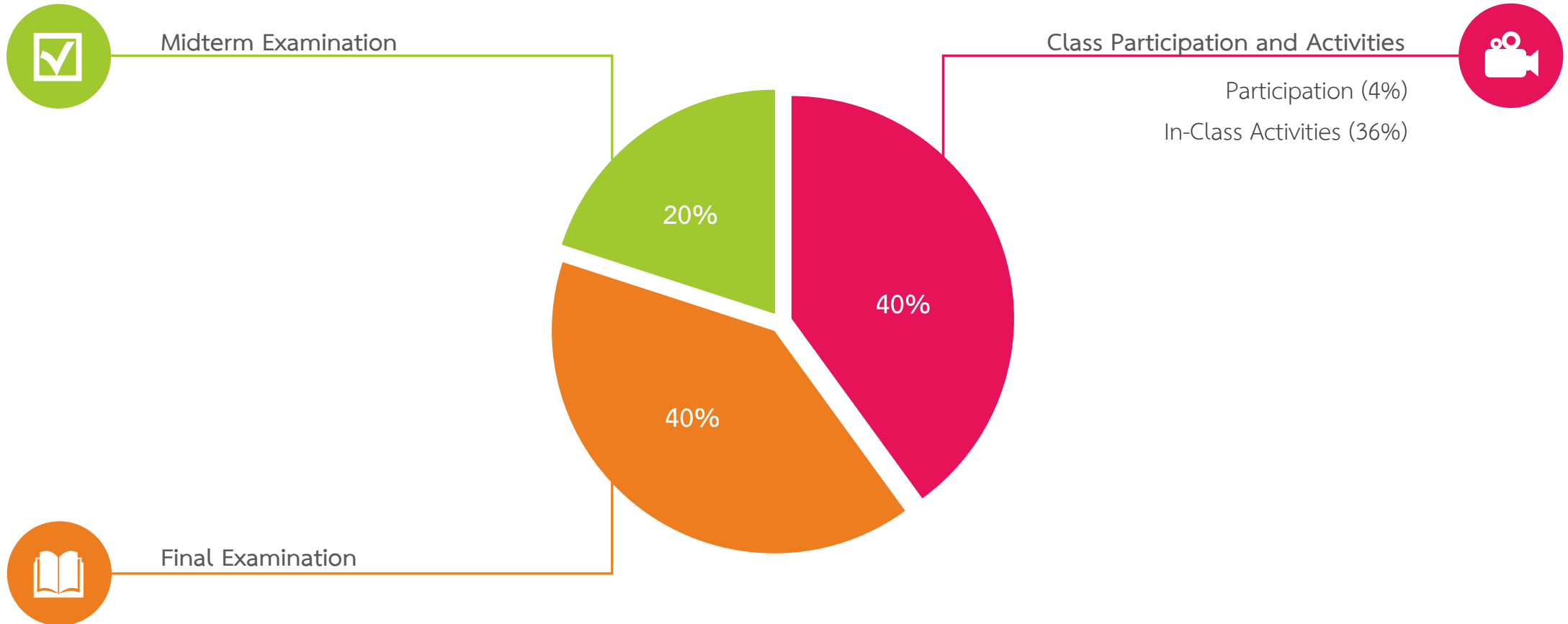
หมายเหตุ วันที่ 10 ธันวาคม 2567 (วันหยุดราชการ ประจำปี 2567) จะเป็นการเรียนการสอนออนไลน์
 วันที่ 31 ธันวาคม 2567 (วันหยุดราชการ ประจำปี 2567) จะเป็นการเรียนการสอนออนไลน์
 วันที่ 1 มกราคม 2568 (วันหยุดราชการ ประจำปี 2568) จะเป็นการเรียนการสอนออนไลน์

Topics and Details of 17 Weeks (cont.)

Week	Date	Topics and Details	Lecturer
10	28,29/1	Column Family - 2: Structures, Processes, and Protocols	Praphan
11	4,5/2	Column Family - 3: Designing for Column Family Databases	Praphan
12	11,12/2	Column Family - 4: Tools for Working with Big Data	Praphan
13	18,19/2	Document - 1: Introduction to JSON Document and JSON Schema Document - 2: NoSQL and MongoDB Data Model Design	Taravichet
14	25,26/2	Document - 3: MongoDB Implementation	Taravichet
15	4,5/3	Document - 4: Advance MongoDB Implementation	Taravichet
16	11,12/3	Document - 5: Indexes and Sharding in MongoDB	Taravichet
17	Final Examination Monday 24 March at 9:30 – 12:30		

หมายเหตุ วันที่ 12 กุมภาพันธ์ 2568 (วันหยุดราชการ ประจำปี 2568) จะเป็นการเรียนการสอนออนไลน์

Course Evaluation and Evaluation Criteria



Course Assessment

Description	Grade		Score (points)
Excellent	A	4.0	80-100
Very Good	B+	3.5	75-79
Good	B	3.0	70-74
Fairly Good	C+	2.5	65-69
Fair	C	2.0	60-64
Poor	D+	1.5	55-59
Very Poor	D	1.0	50-54
Fail	F	0.0	0-49

หมายเหตุ ผู้สอนอาจจะพิจารณาแบบอิงกลุ่มร่วมด้วย



อินเทอร์เน็ต



เครือข่ายสังคมออนไลน์



ข้อมูลมีปริมาณสูง

คุณลักษณะของ NoSQL



NoSQL ได้รับการออกแบบที่อยู่บนพื้นฐาน 3 อย่าง ได้แก่ Non-relational database, Open-source และ Distributed database

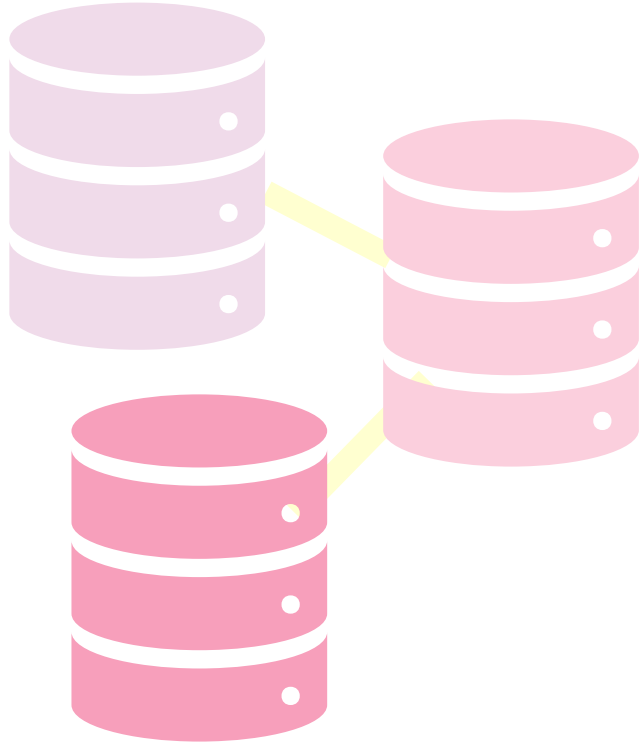
ที่มาและความสำคัญของ NoSQL

ตลอด 10 กว่าปีที่ผ่านมา ระบบฐานข้อมูลเชิงสัมพันธ์ (Relational Database) ได้รับความนิยมอย่างแพร่หลายสำหรับการจัดเก็บข้อมูลประเภทโครงสร้างในงานประเภทต่างๆ ซึ่งข้อมูลดังกล่าวจะแบ่งออกเป็นส่วย่อยๆ และจัดเก็บในรูปแบบของตาราง (Table) ซึ่งประกอบด้วยองค์ประกอบย่อยในรูปแบบของ “คอลัมน์ (Column)” และรวมกันเป็นกลุ่มซึ่งเรียกว่า “แถว (Row)” สิ่งนี้ส่งผลให้ข้อมูลที่จัดเก็บในรูปแบบของตารางไม่สามารถปรับขนาดตามแนวนอนได้ (Scaling Horizontally)



ดังนั้น ในช่วงหลายปีที่ผ่านมา จึงมีงานวิจัยและบริษัทจำนวนมากได้พัฒนาวิธีการสำหรับการปรับขนาดของข้อมูล หนึ่งในวิธีการที่ได้รับความนิยม คือ ระบบฐานข้อมูลที่ไม่ใช่เชิงสัมพันธ์ (Non-Relational Database) หรือสามารถกล่าวได้อีกชื่อว่า NoSQL database ซึ่งสามารถจัดการข้อจำกัดข้างต้นได้อย่างมีประสิทธิภาพ

ความหมายของ NoSQL



- NoSQL ย่อมาจากคำว่า “not only SQL” (ในปัจจุบัน) ถึงแม้ว่าในช่วงแรก NoSQL จะย่อมาจากเกิดจาก “No” ร่วมกับ “SQL”
- NoSQL เป็นที่ได้รับความนิยมอย่างกว้างขวางในปัจจุบัน ซึ่งไม่ยึดหลักการของระบบฐานข้อมูลเชิงสัมพันธ์ นั่นคือ NoSQL ไม่ได้จัดเก็บข้อมูลในรูปแบบของตารางและไม่ได้ใช้คำสั่ง SQL สำหรับการจัดการข้อมูล

ข้อดีของ NoSQL

ฐานข้อมูลประเภท NoSQL มีข้อได้เปรียบฐานข้อมูลประเภท RDBMS ที่สำคัญอยู่มากมาย อาทิเช่น



1

ความสามารถในการรองรับการปรับขนาดมาก (High Scalability)

ข้อมูลที่จัดเก็บ NoSQL สามารถปรับขนาดตามแนวนอน (ปรับเปลี่ยนจำนวนคอลัมน์ หรือ เพิ่มลดคุณลักษณะที่จะจัดเก็บ) ได้ตามความเหมาะสมของภาคธุรกิจ เพื่อหลีกเลี่ยงต้นทุนจำนวนมาก

2

ต้นทุนน้อย (Lower Cost)

มีซอฟต์แวร์ที่เปิดเผยแพร่ซอร์สโค้ดต่อสาธารณชน (Open Source) ของ NoSQL อยู่มากมาย ทำให้เป็นที่สนใจของภาคเอกชน หรือบริษัทขนาดเล็กที่มีข้อจำกัดเกี่ยวกับต้นทุน

3

ความพร้อมใช้งานของระบบ (High Availability)

หลากหลายฐานข้อมูล NoSQL มีการประมวลผลแบบกระจายตัว (distribution) สิ่งที่ทำให้มั่นใจว่าระบบยังคงพร้อมใช้งานตลอดเวลาถึงแม้ว่าจะมีโหนดใดเสียก็ตาม

ข้อดีของ NoSQL (ต่อ)

ฐานข้อมูลประเภท NoSQL มีข้อได้เปรียบฐานข้อมูลประเภท RDBMS ที่สำคัญอยู่มากมาย อาทิเช่น



4

แบบจำลองข้อมูลมีความยืดหยุ่น (Flexible Data Modeling)

NoSQL รองรับการสร้างแบบจำลองที่ยืดหยุ่นและข้อมูลมีการปรับเปลี่ยนได้ สิ่งนี้ทำให้
ง่ายต่อนักพัฒนาระบบดูแลหรือพัฒนาโปรแกรมที่มีความสัมพันธ์กับฐานข้อมูล อีกทั้งยัง
เหมาะสมกับการทำงานแบบ Agile

5

ความจำเป็นสำหรับการโหลด สกัด และแปลงข้อมูลน้อย (Less Need for Extract-Transform-Load)

NoSQL รองรับการเก็บข้อมูลได้หลายแบบ เช่น ตัวเลข ข้อความ ค่าความจริง
นอกจากนี้ ยังสามารถจัดเก็บข้อมูลที่มีโครงสร้างซับซ้อนได้ อาทิเช่น JSON หรือ XML

6

ประสิทธิภาพ (Performance)

NoSQL มีการประมวลผลแบบกระจายตัว (distribution) ทำให้สามารถรองรับ
คำร้องขอรับบริการได้เพิ่มขึ้น ตามจำนวนโหนด สิ่งนี้ส่งผลให้ประสิทธิภาพสูงขึ้น

ความแตกต่างระหว่าง RDBMS และ NoSQL

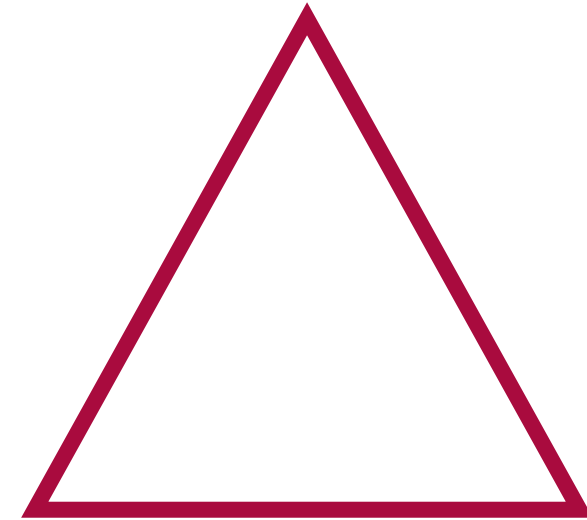
ACID



BASE



Consistency



Availability

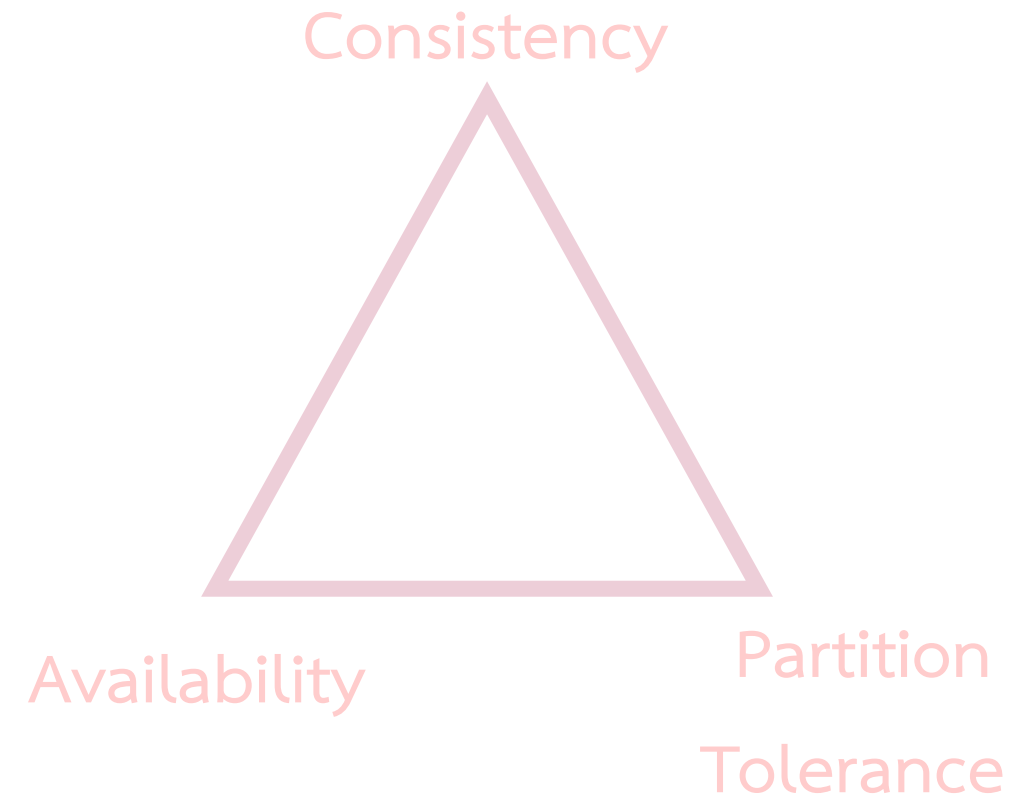
Partition
Tolerance

ความแตกต่างระหว่าง RDBMS และ NoSQL

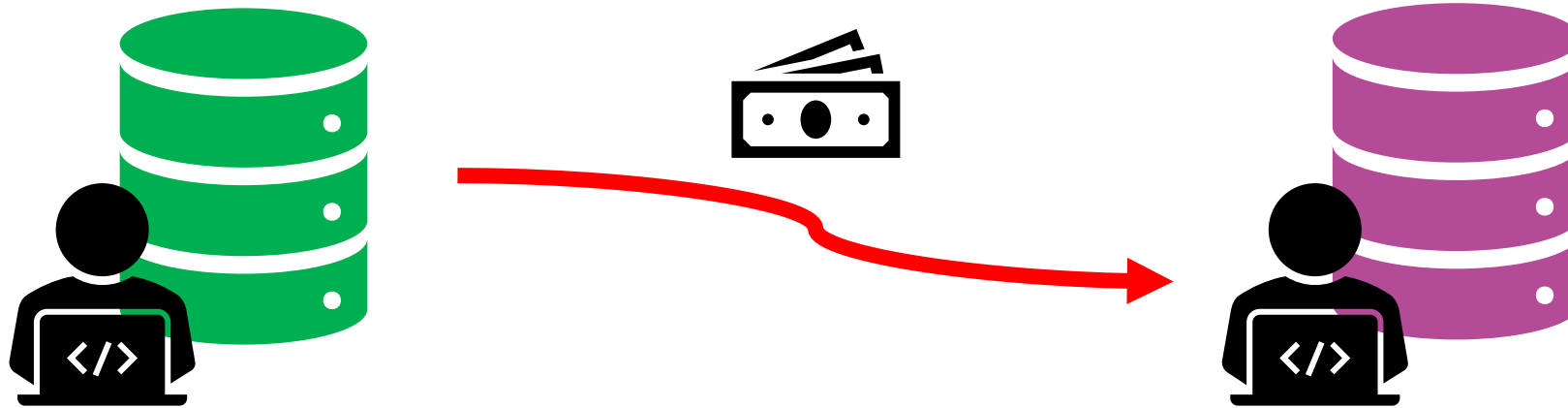
ACID



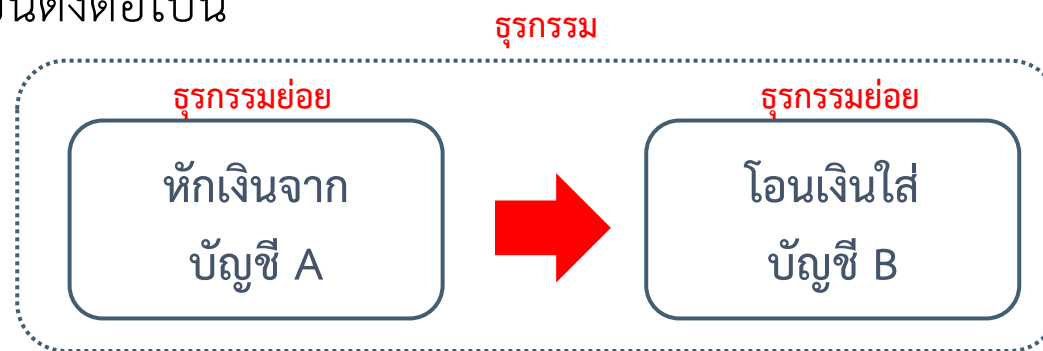
BASE



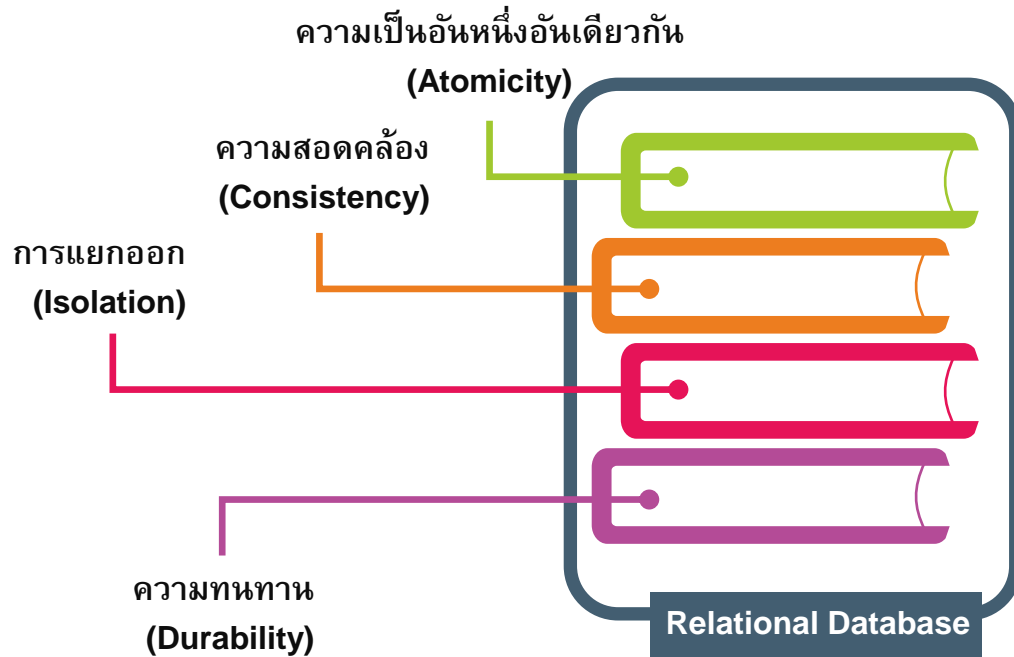
ความหมาย Database Transaction



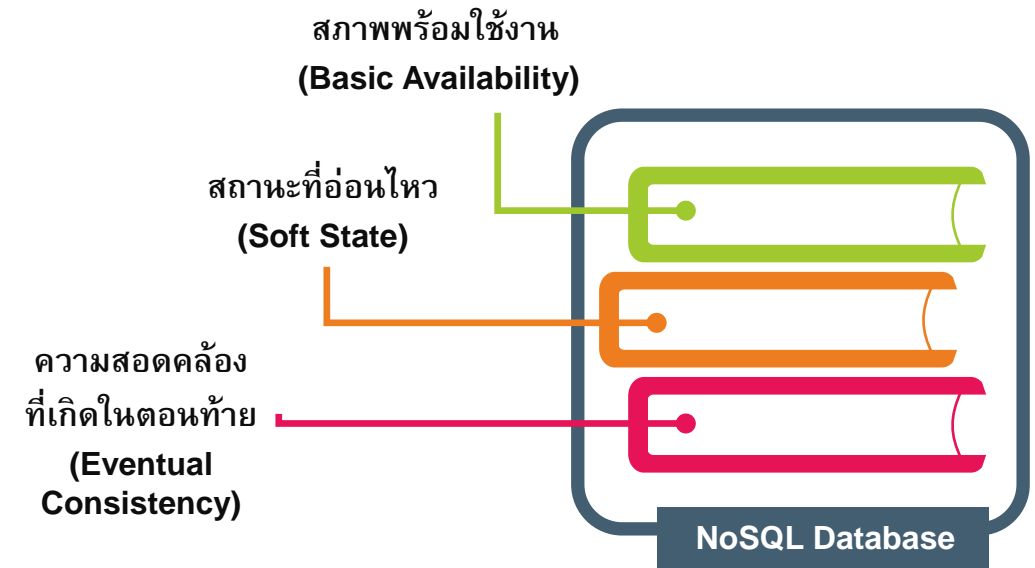
Database transactions จะหมายถึง ชุดคำสั่งที่จัดการกับข้อมูลในฐานข้อมูล อาทิเช่น การโอนเงินระหว่างธนาคาร จะประกอบด้วยขั้นตอนดังต่อไปนี้



ความแตกต่างระหว่าง RDBMS และ NoSQL



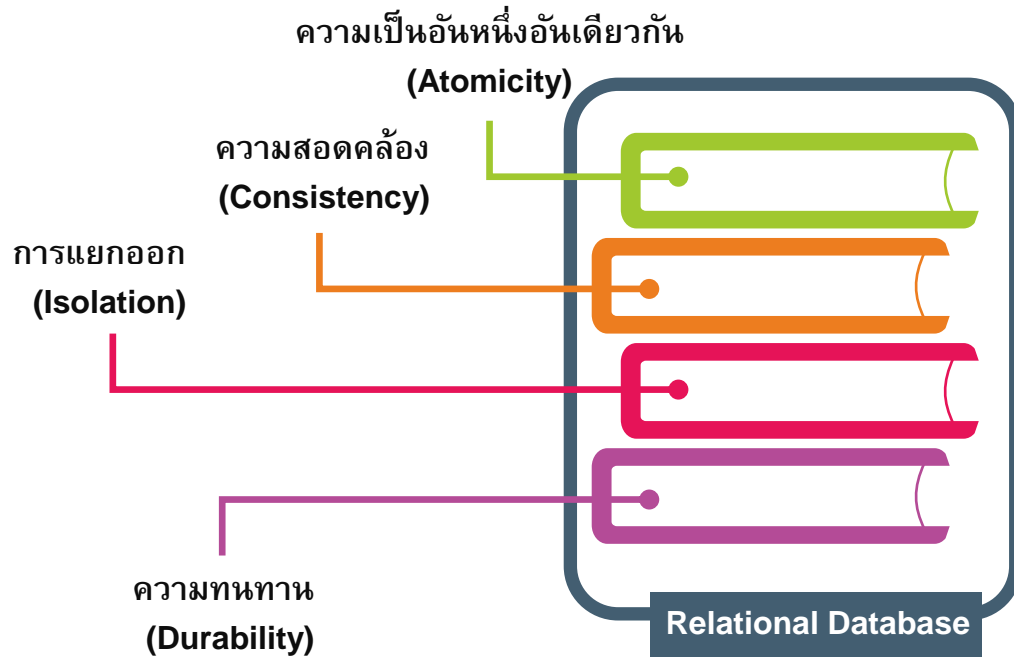
ACID Model



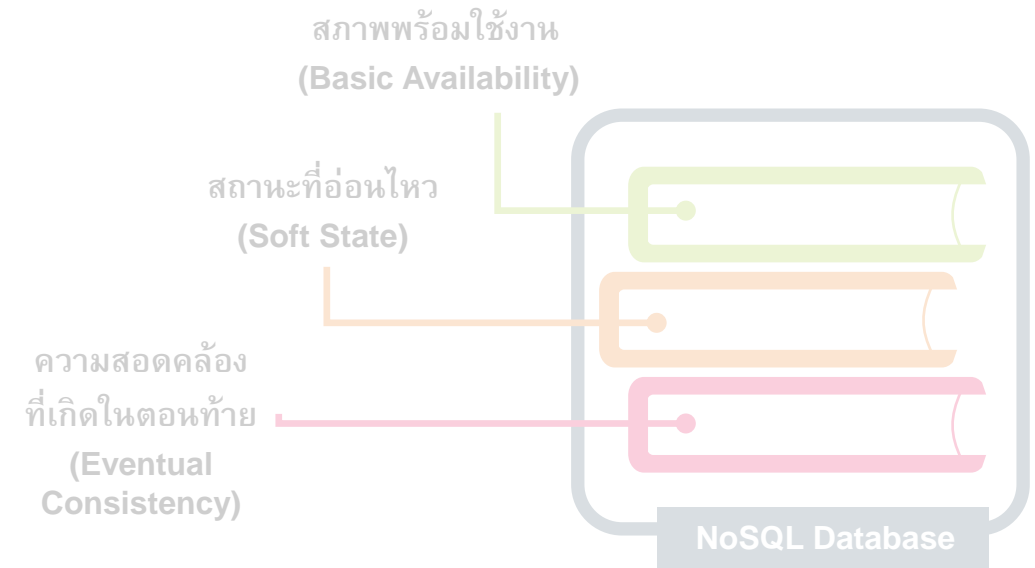
BASE Model

กำหนดโดย Eric Brewer

ความแตกต่างระหว่าง RDBMS และ NoSQL

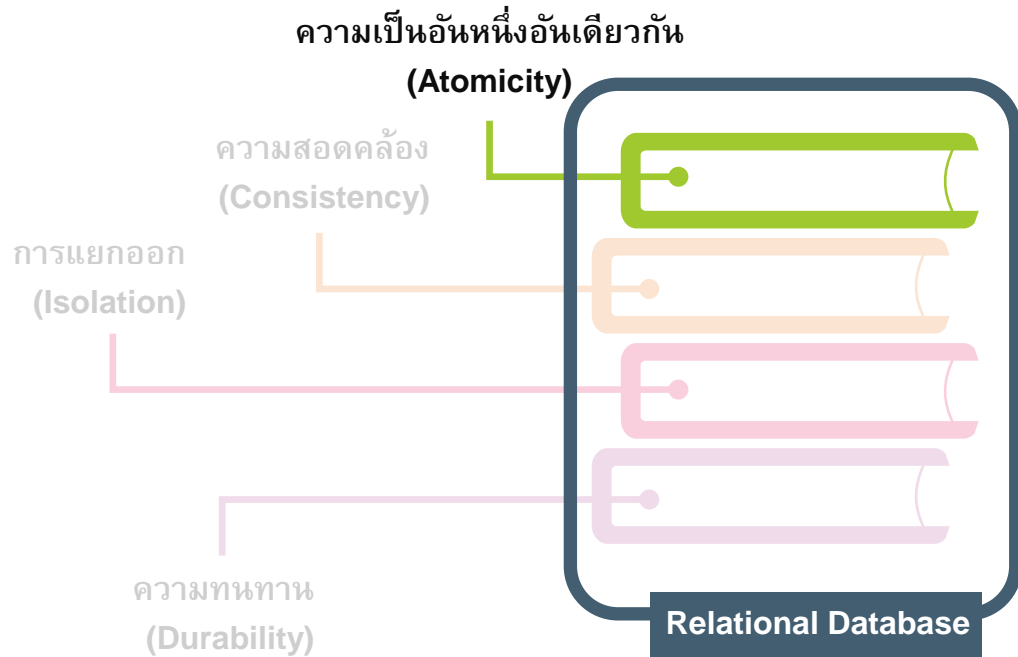


ACID Model



BASE Model

ACID Model



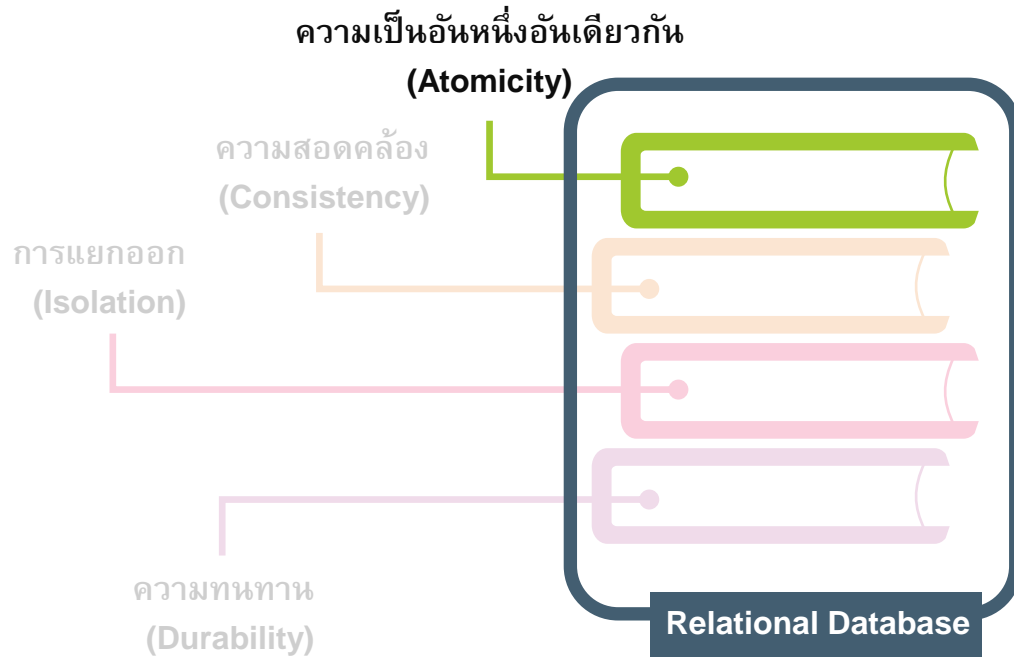
ACID Model

Atomicity

ความเป็นอันหนึ่งอันเดียวกัน

ทุกธุรกรรมย่อยจะต้องทำงานสำเร็จ แต่ถ้าธุรกรรมย่อยอันใดอันหนึ่งเกิดความล้มเหลว ทุกธุรกรรมย่อยจะล้มเหลวตาม และฐานข้อมูลจะยกเลิกการเปลี่ยนแปลงจากธุรกรรมนั้น

ACID Model



ACID Model

Atomicity

ความเป็นอันหนึ่งอันเดียวกัน

ตัวอย่างเช่น นาย ก ต้องการโอนเงิน 1,000 บาท จากบัญชี A ไปบัญชี B ซึ่งสมมติว่าประกอบด้วยธุรกรรมย่อย ๆ ดังนี้

- (1) อ่านยอดเงินคงเหลือจากบัญชี A
- (2) หักยอดเงินจำนวน 1,000 บาทจากบัญชี A
- (3) อ่านยอดเงินคงเหลือจากบัญชี B
- (4) ปรับยอดเงินในบัญชี B เพิ่มขึ้น 1,000 บาท

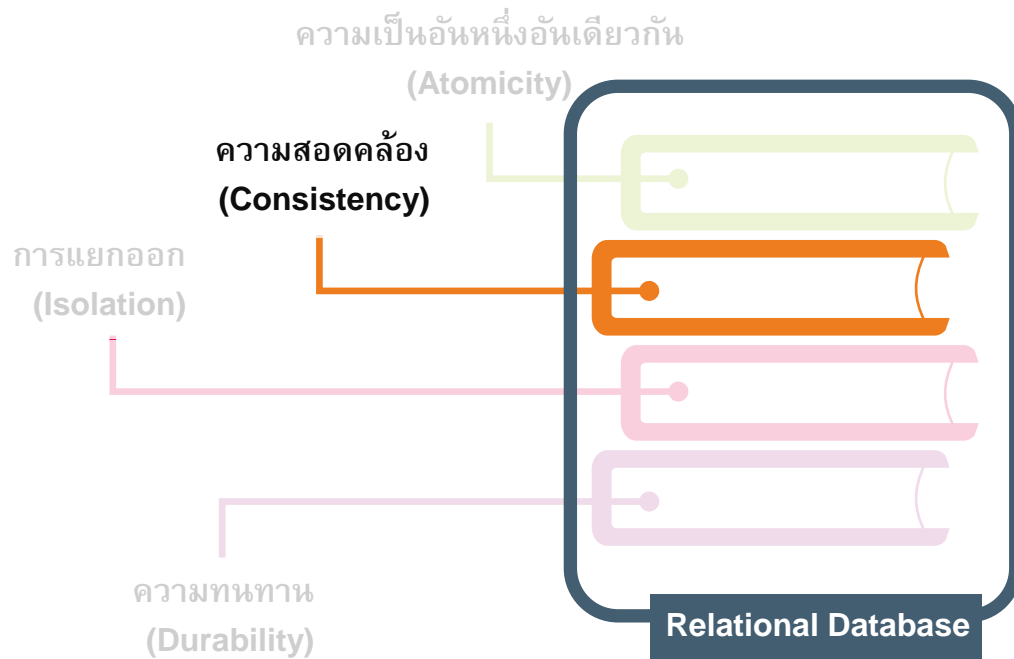
ดำเนินการธุรกรรมย่อยที่ (2) สำเร็จ เงินจำนวน 1,000 บาท จะถูกหักออกจากบัญชี A แต่ยังไม่ได้เพิ่มยอดเงินในบัญชี B **สมมติว่า** ธุรกรรมย่อยที่ (1) – (3) ดำเนินการเสร็จสมบูรณ์ แต่พอถึงธุรกรรมย่อยที่ (4) คือ การปรับยอดเงินในบัญชี B เพิ่ม 1,000 บาท แต่ระบบไม่สามารถหาบัญชี B พบ ส่งผลให้ธุรกรรมย่อยทั้งหมดก่อนหน้านี้จะถูกยกเลิกทุกรายการ

อ้างอิงตัวอย่างจาก :

<http://aorjoa.blogspot.com/2014/04/acid.html>



ACID Model



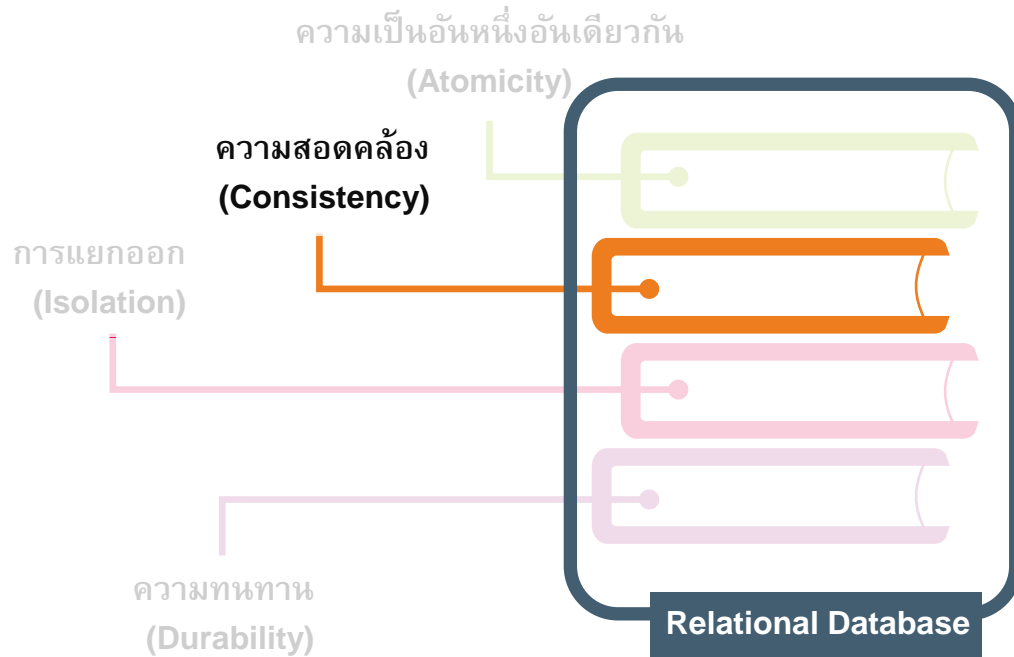
ACID Model

Consistency

ความสอดคล้อง

การรับรองว่าข้อมูลที่เขียนลงฐานข้อมูลมีความถูกต้องตามกฎหมายหรือเงื่อนไขที่กำหนดไว้

ACID Model



ACID Model

Consistency

ความสอดคล้อง

ตัวอย่างเช่น นาย ก ต้องการโอนเงิน 1,000 บาท จากบัญชี A ไปบัญชี B ซึ่งสมมติว่าประกอบด้วยธุรกรรมย่อย ๆ ดังนี้

- (1) อ่านยอดเงินคงเหลือจากบัญชี A
- (2) หักยอดเงินจำนวน 1,000 บาทจากบัญชี A
- (3) อ่านยอดเงินคงเหลือจากบัญชี B
- (4) ปรับยอดเงินในบัญชี B เพิ่มขึ้น 1,000 บาท

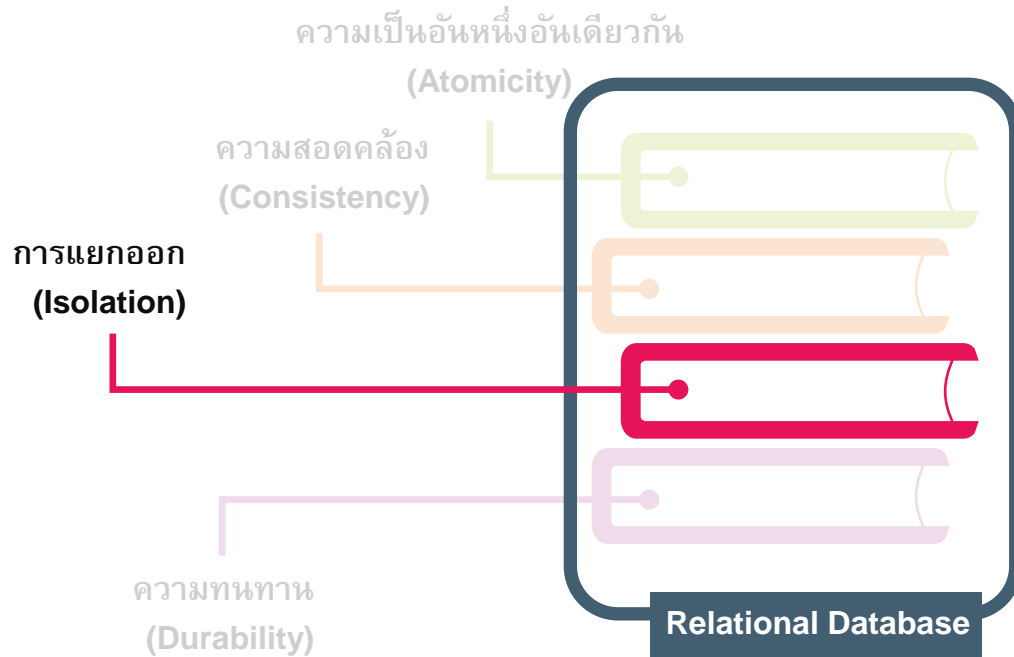
สมมติว่า “ฐานข้อมูลมีเงื่อนไขที่ว่าบัญชีใดๆ ห้ามมียอดเงินในบัญชีน้อยกว่า 0 บาท” และ “ยอดเงินในบัญชี A มี 500 บาท” ถ้าเกิดเหตุการณ์หักเงินออกจากบัญชี A เป็นจำนวน 1,000 บาท จะทำให้ยอดเงินในบัญชี A ติดลบ 500 บาท ซึ่งขัดแย้งกับเงื่อนไขของฐานข้อมูล ดังนั้น ฐานข้อมูลจะย้อนกลับไปที่ก่อนที่จะประมวลผลธุรกรรมดังกล่าว

อ้างอิงตัวอย่างจาก :

<http://aorjoa.blogspot.com/2014/04/acid.html>



ACID Model



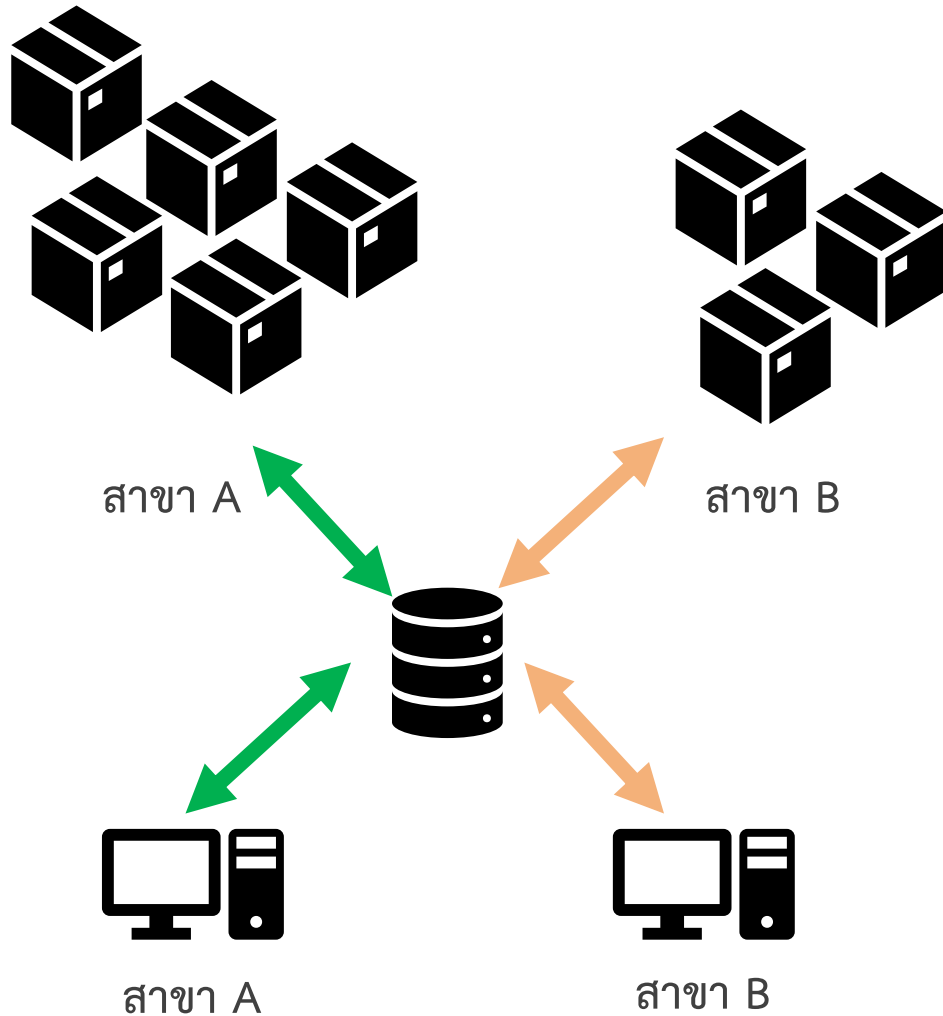
ACID Model

Isolation

การแยกออก

การมั่นใจว่าธุรกรรมใดๆ จะไม่ได้รับผลกระทบจากการทำงานของธุรกรรมอื่นๆ ถึงแม้ว่าธุรกรรมก่อนหน้านี้จะทำงานได้สมบูรณ์หรือไม่

ACID Model



Isolation

การแยกออก

ตัวอย่างเช่น ร้านค้าแห่งหนึ่งมีสาขาทั้งหมด 2 สาขา ได้แก่ สาขา A และสาขา B ซึ่งทั้ง 2 สาขาได้ใช้ระบบจัดการร้านค้าและฐานข้อมูลร่วมกัน โดยที่สาขา A มีสินค้าอยู่ 5 ลัง ขณะที่สาขา B มีสินค้าอยู่ทั้งหมด 3 ลัง

สมมติ เหตุการณ์ที่ 1

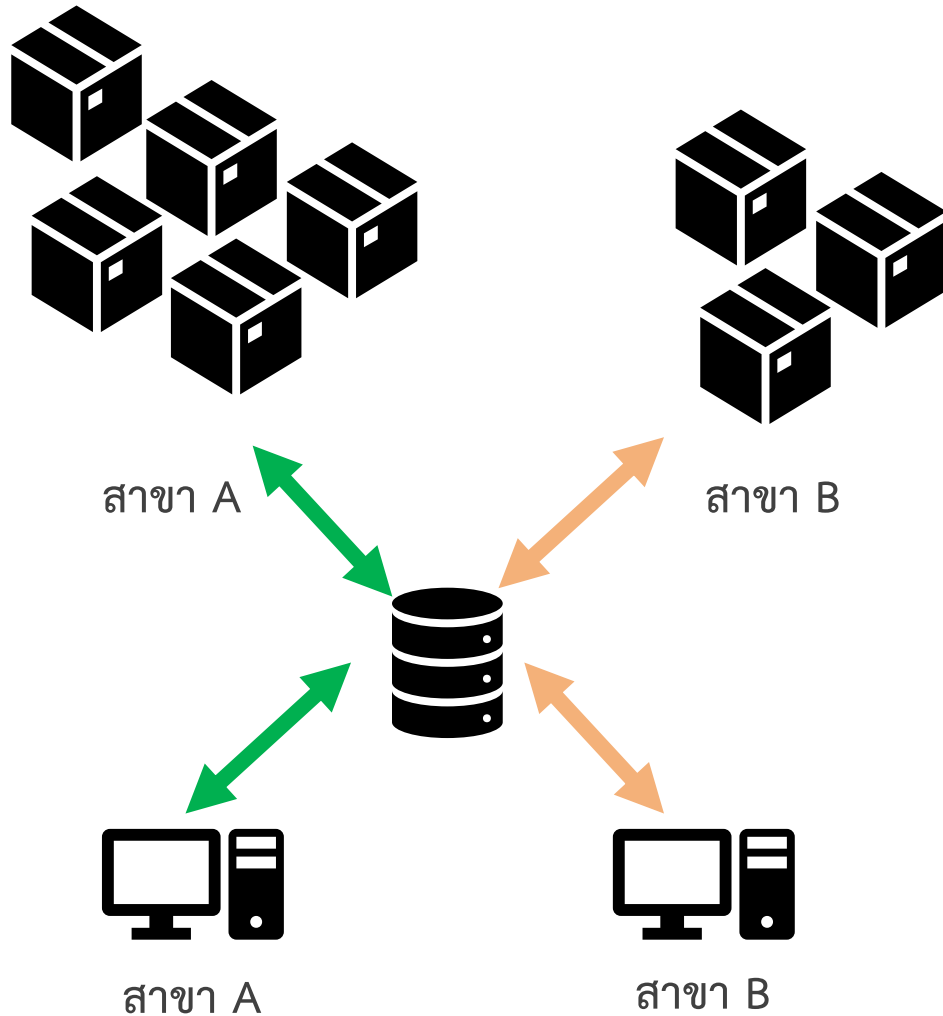
(T1) สาขา A ขนย้ายสินค้า 2 ลังไปที่สาขา B ขณะที่ (T2) สาขา B ก็ขนย้ายสินค้า 1 ลังไปที่สาขา A เช่นกัน

- (1) T1 = สินค้า A ลดลง 2 ลัง
- (2) T1 = สินค้า B เพิ่มขึ้น 2 ลัง
- (3) T2 = สินค้า B ลดลง 1 ลัง
- (4) T2 = สินค้า A เพิ่มขึ้น 1 ลัง

อ้างอิงตัวอย่างจาก :

<http://aorjoa.blogspot.com/2014/04/acid.html>

ACID Model



Isolation

การแยกออก

สมมติ เหตุการณ์ที่ 2

(T1) สาขา A ขนย้ายสินค้า 2 ลงไปที่สาขา B ขณะที่ (T2) สาขา B ก็ขนย้ายสินค้า 1 ลงไปที่สาขา A เช่นกัน แต่ธุรกรรมทั้ง 2 เกิดตรงกัน ก่อสร้างในเวลาใกล้เคียงกันมากส่งผลให้

(0) สถานะสินค้าของแต่ละสาขา ($A = 5, B = 3$)

(1) T1 = สินค้า A ลดลง 2 ลง ($A = 3, B = 3$)

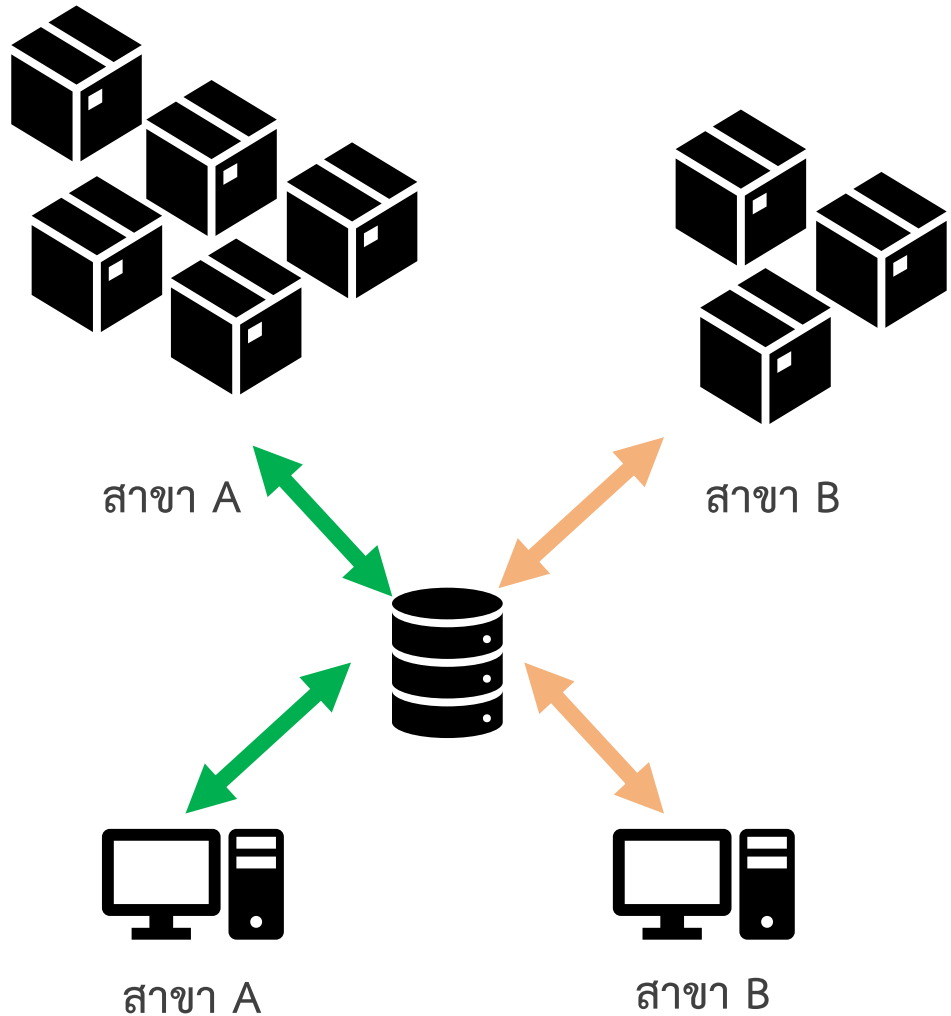
(2) T2 = สินค้า B ลดลง 1 ลง ($A = 3, B = 2$)

(3) T2 = สินค้า A เพิ่มขึ้น 1 ลง ($A = 4, B = 2$)

(4) T1 = สินค้า B เพิ่มขึ้น 2 ลง ($A = ??, B = ??$) **เกิดปัญหาขึ้นที่จุดนี้**

ตามหลักการ Atomicity ที่ว่าถ้าธุรกรรมย่อยๆ ใดไม่สามารถดำเนินการได้ ให้ทำการยกเลิกธุรกรรมย่อยๆ ทั้งหมด ดังนั้น ธุรกรรมย่อยที่ (1) จึงถูกยกเลิก ($A = 5, B = 2$) ดังนั้น สิ่งที่เกิดขึ้นตามมา สินค้าในคลังของทั้ง 2 สาขา รวมกันไม่เท่า 8 ดังเดิม

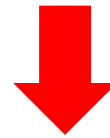
ACID Model



Isolation

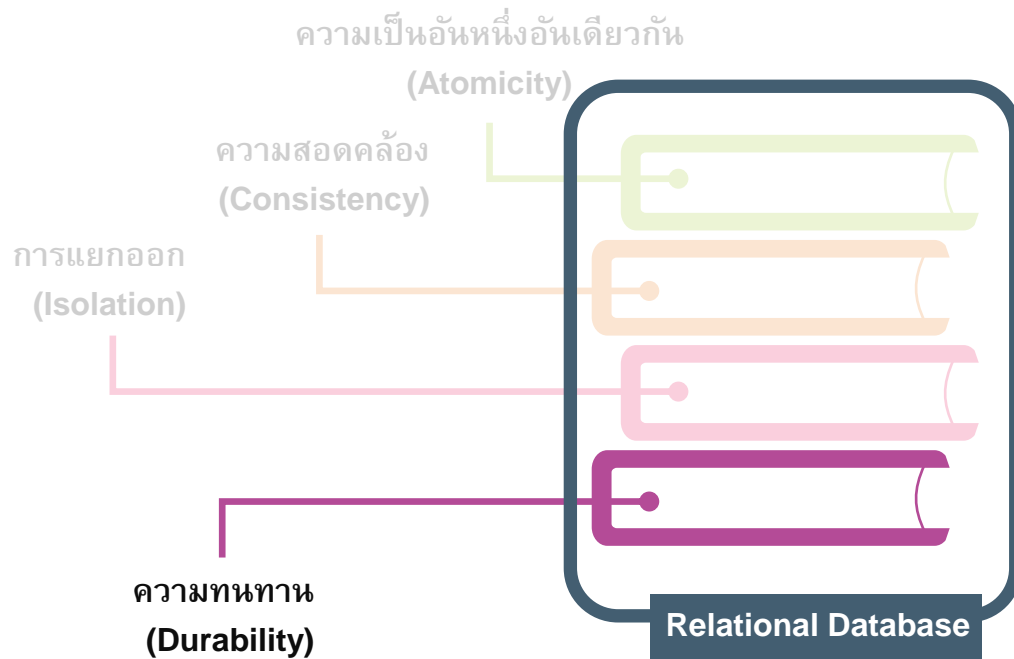
การแยกออก

สมมติ เหตุการณ์ที่ 2



ดังนั้น เพื่อป้องกันปัญหาดังกล่าว Isolation จึงเกิดขึ้น เพื่อบอกว่าธุรกรรม
 ใดๆ จะไม่ส่งผลกับธุรกรรมอื่นๆ จนกว่าธุรกรรมเหล่านั้นจะดำเนินการเสร็จ
 สิ้น จากนั้นธุรกรรมดังกล่าวจึงจะส่งผลกับธุรกรรมอื่น

ACID Model



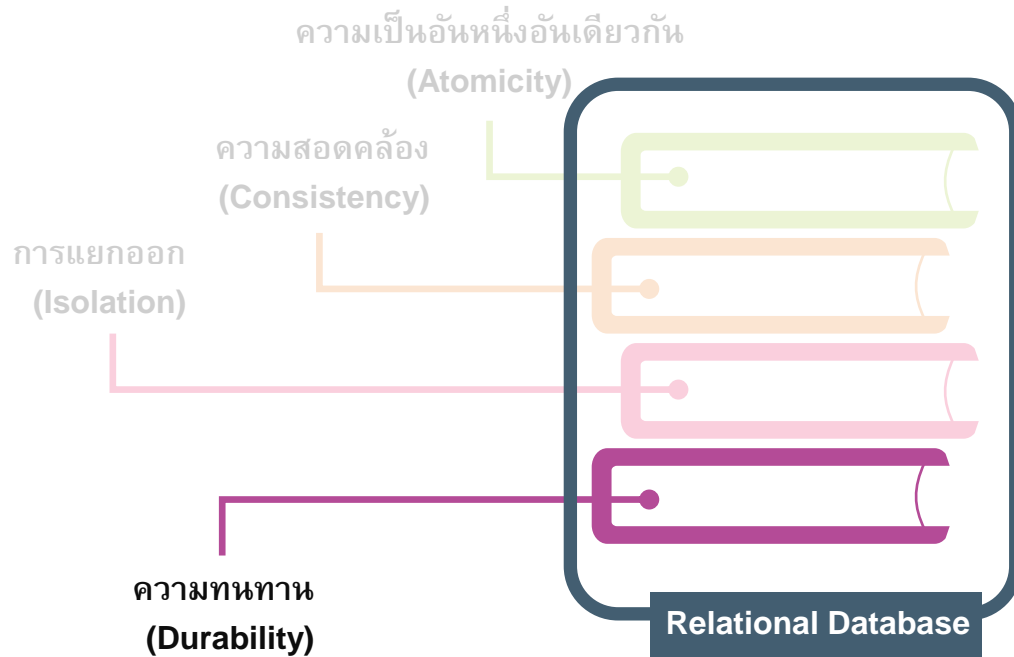
ACID Model

Durability

ความทนทาน

การมั่นใจว่าธุรกรรมใดๆ ข้อมูลจะอยู่ครบถ้วน ถึงแม้ว่าจะประสบกับปัญหา ไฟฟ้าดับ หรือระบบล้มเหลวก็ตาม

ACID Model



ACID Model

Durability

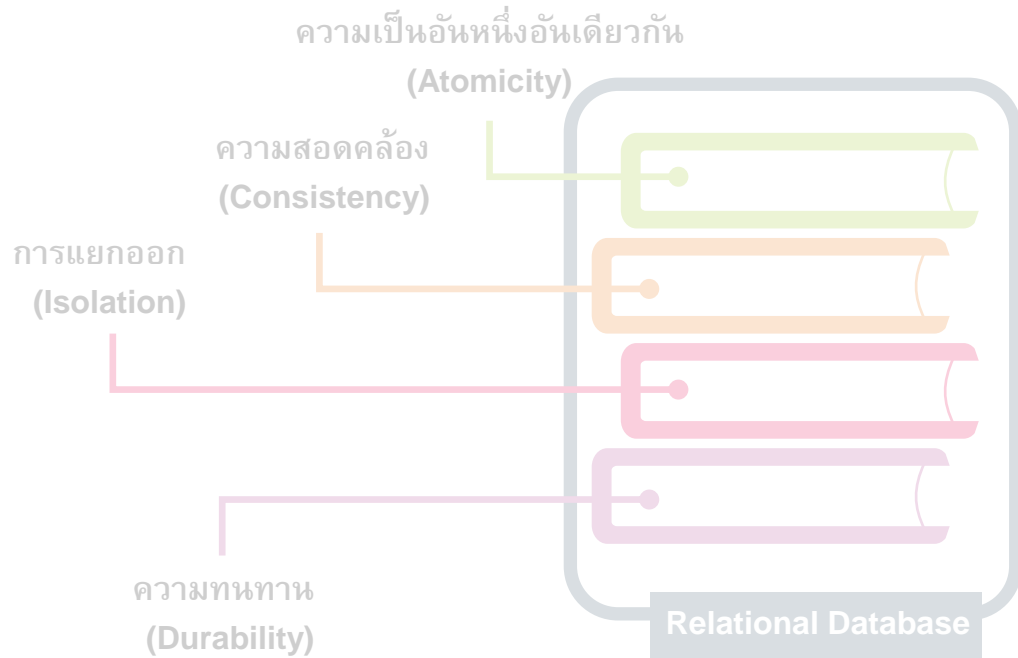
ความทนทาน

ตัวอย่างเช่น นาย ก ต้องการโอนเงิน 1,000 บาท จากบัญชี A ไปบัญชี B ซึ่งสมมติว่าประกอบด้วยธุรกรรมย่อย ๆ ดังนี้

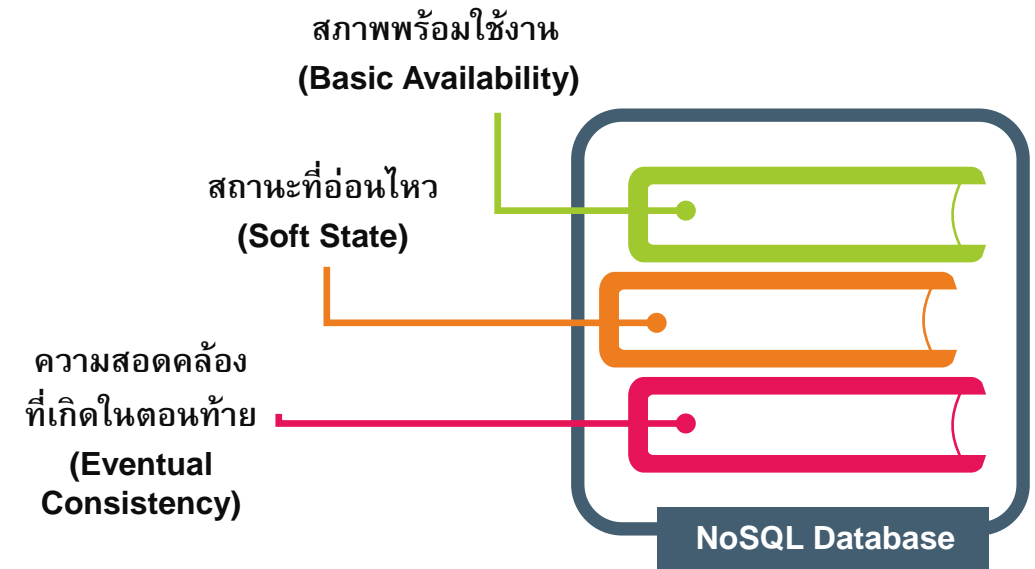
- (1) อ่านยอดเงินคงเหลือจากบัญชี A
- (2) หักยอดเงินจำนวน 1,000 บาทจากบัญชี A
- (3) อ่านยอดเงินคงเหลือจากบัญชี B
- (4) ปรับยอดเงินในบัญชี B เพิ่มขึ้น 1,000 บาท

สมมติว่า ธุรกรรมย่อยที่ (1) – (4) ดำเนินการเสร็จสมบูรณ์ ขณะที่ข้อมูลกำลังจะถูกบันทึกลงในหน่วยเก็บข้อมูลเกิดเหตุการณ์ไฟฟ้าขัดข้อง(ดับ)ไปก่อน หลังจากไฟฟ้ากลับมาใช้งานได้ดังเดิมแล้วระบบจะประมวลผลธุรกรรมนี้ต่อให้สำเร็จและไม่มีการยกเลิก

ความแตกต่างระหว่าง RDBMS และ NoSQL



ACID Model



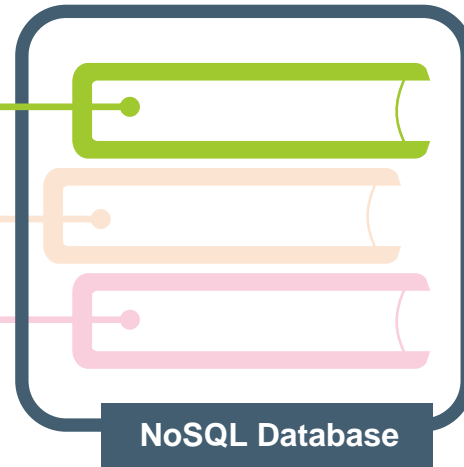
BASE Model

BASE Model

สภาพพร้อมใช้งาน
 (Basic Availability)

สถานะที่อ่อนไหว
 (Soft State)

ความสอดคล้อง
 ที่เกิดในอนท้าย
 (Eventual
 Consistency)



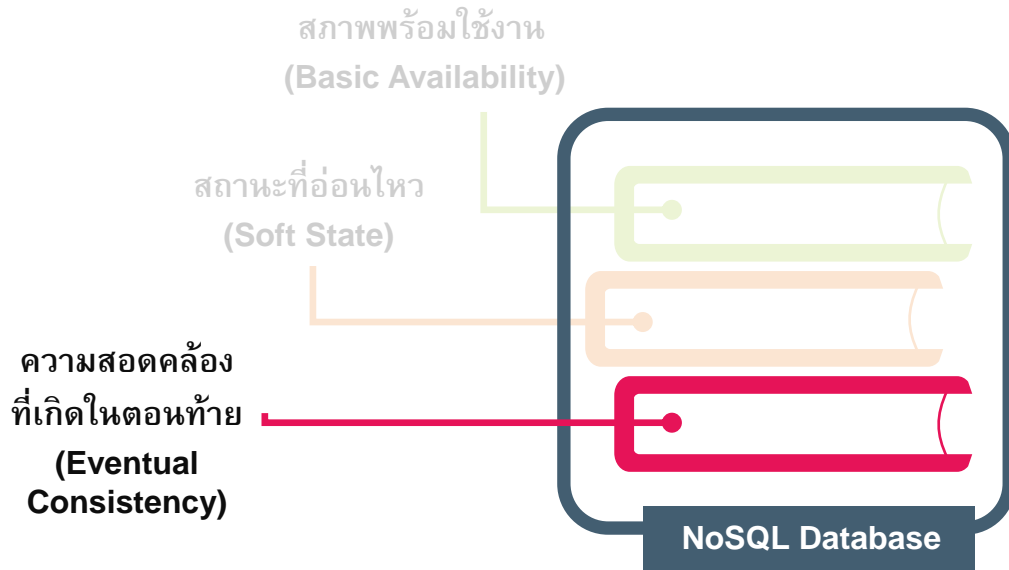
BASE Model

Basic Availability

สภาพพร้อมใช้งาน

คือ การการันตีว่าทุกคำร้องขอ (Request) จะได้รับการตอบสนองเสมอ ไม่ว่าจะสำเร็จ (Success) หรือล้มเหลว (Fail) หรือสามารถกล่าวได้ว่าฐานข้อมูลต้องทำงานอยู่ตลอดเวลา

BASE Model



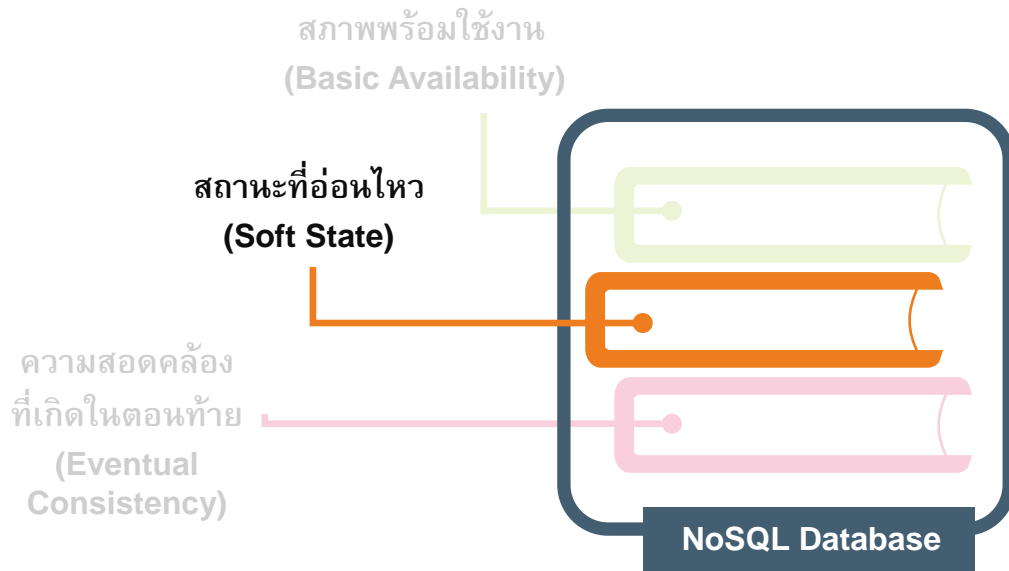
BASE Model

Eventual Consistency

ความสอดคล้องที่เกิดขึ้นตอนท้าย

คือ ข้อมูลอาจจะเกิดความขัดแย้งกันชั่วคราว แต่สุดท้ายแล้วข้อมูลก็จะสอดคล้องกันในภายหลัง (รอความล่าช้าในระหว่างการส่งข้อมูลไปยังแต่ละเครื่อง ซึ่งจะมีหน่วยเป็นมิลลิวินาที)

BASE Model



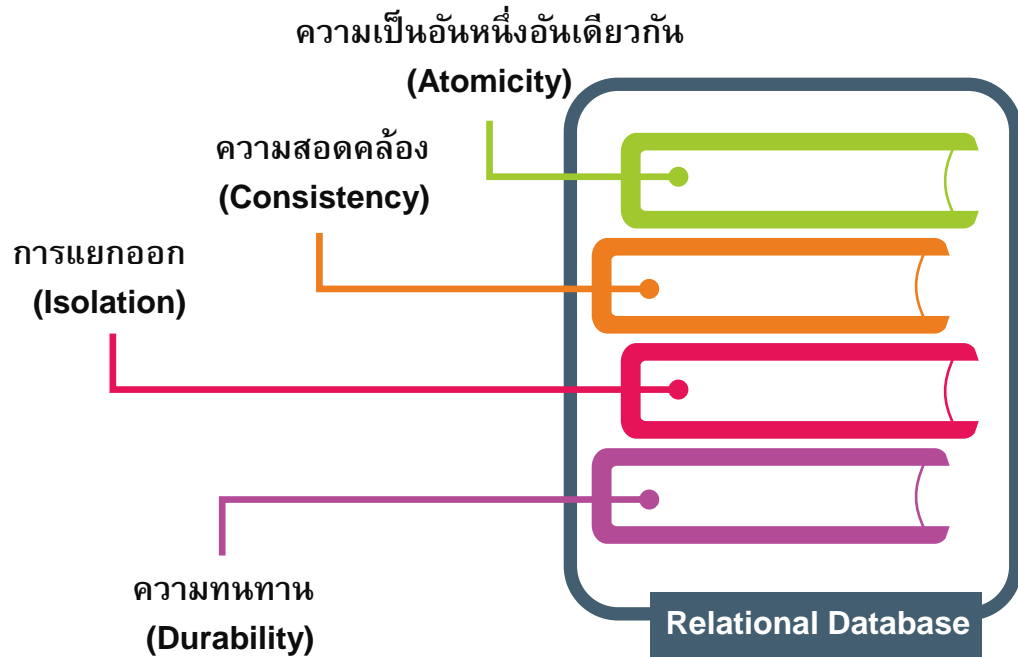
BASE Model

Soft State

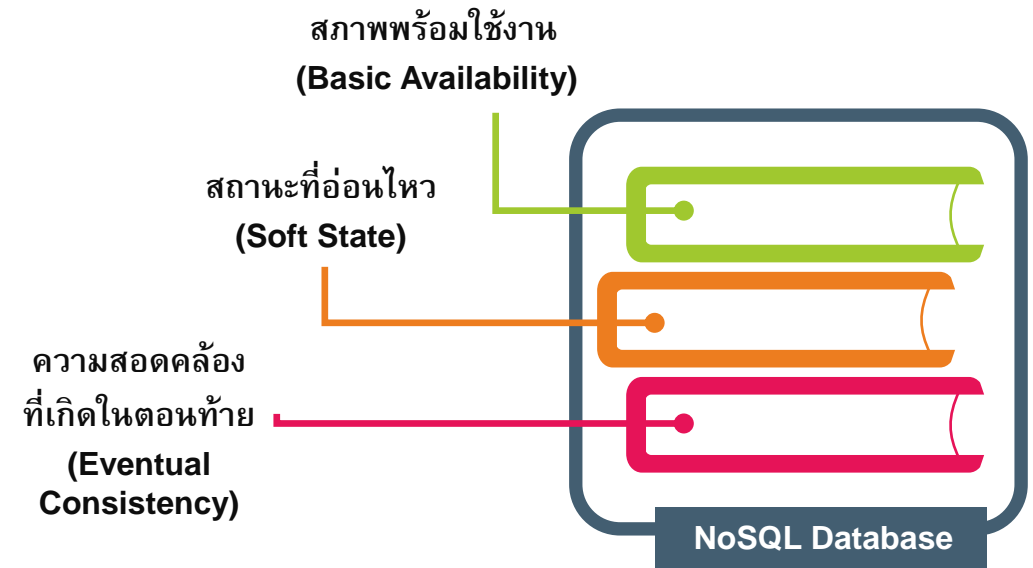
สถานะที่อ่อนไหว

คือ สภาพของข้อมูลจะเปลี่ยนตลอดเวลา ถึงแม้ว่าจะไม่มีคำร้องขอหรือคำสั่งเข้ามาก็ตาม เกิดจากข้อมูลจะค่อยๆ อัปเดตให้มีความเป็นปัจจุบันในทุกๆ เครื่อง (ไม่สามารถเขียนพร้อมกันทุกเครื่องได้)

ความแตกต่างระหว่าง RDBMS และ NoSQL



ACID Model

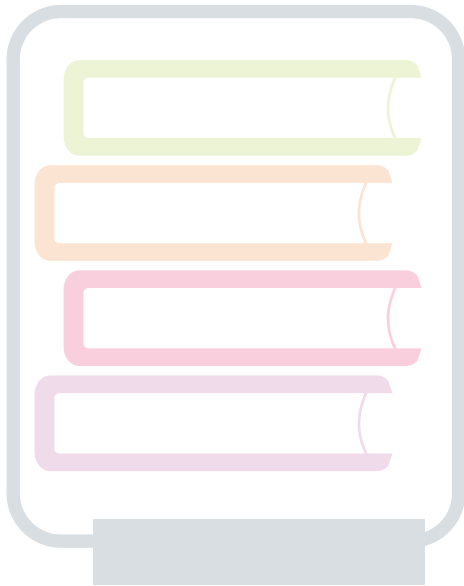


BASE Model

กำหนดโดย Eric Brewer

ความแตกต่างระหว่าง RDBMS และ NoSQL

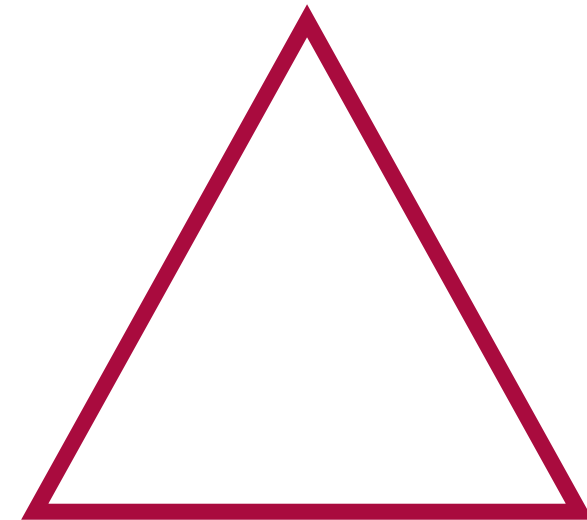
ACID



BASE



Consistency



Availability

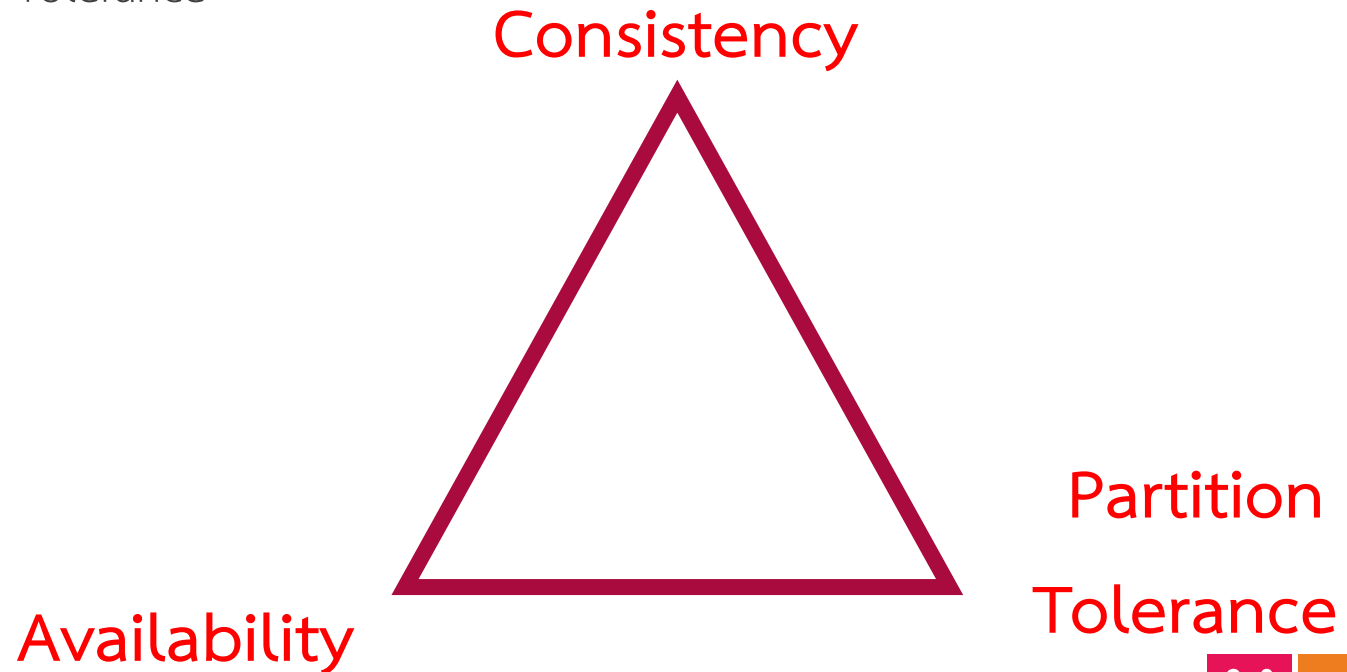
Partition
Tolerance

แนวคิดของ CAP

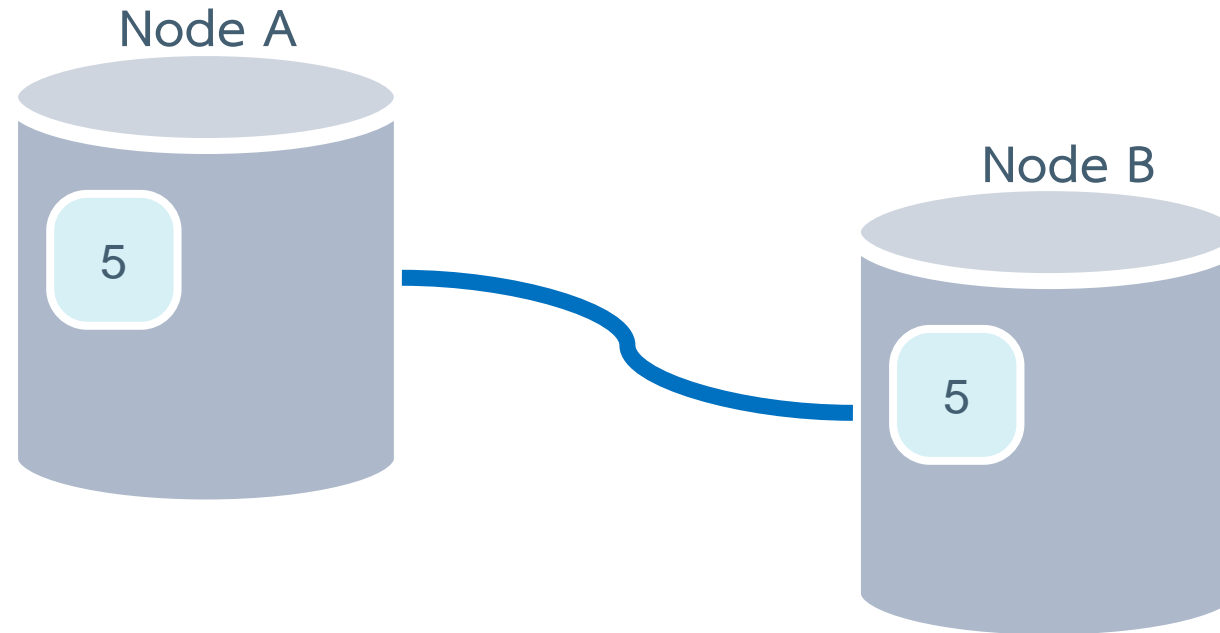


Dr. Eric Brewer

แนวคิดของ CAP คือ ทฤษฎีที่กล่าวถึงการจัดเก็บข้อมูลแบบกระจาย (Distributed Data Store) ซึ่งนำเสนอขึ้นในปี ค.ศ. 2000 โดย Dr. Eric Brewer จากทฤษฎีดังกล่าวได้ระบุไว้ว่าระบบที่สามารถจัดเก็บข้อมูลแบบกระจายได้นั้นจะสามารถรองรับคุณสมบัติต่อไปนี้ ได้เพียง 2 ข้อเท่านั้น ได้แก่ Consistency, Availability, Partition Tolerance



แนวคิดของ CAP



Consistency คือ การรับประกันว่าในทุกครั้งของการอ่านข้อมูลจะได้รับข้อมูลที่ถูกบันทึกไว้ล่าสุด ซึ่ง Consistency จะมีประเภทได้แก่

(1) strongly consistent และ (2) weak consistency

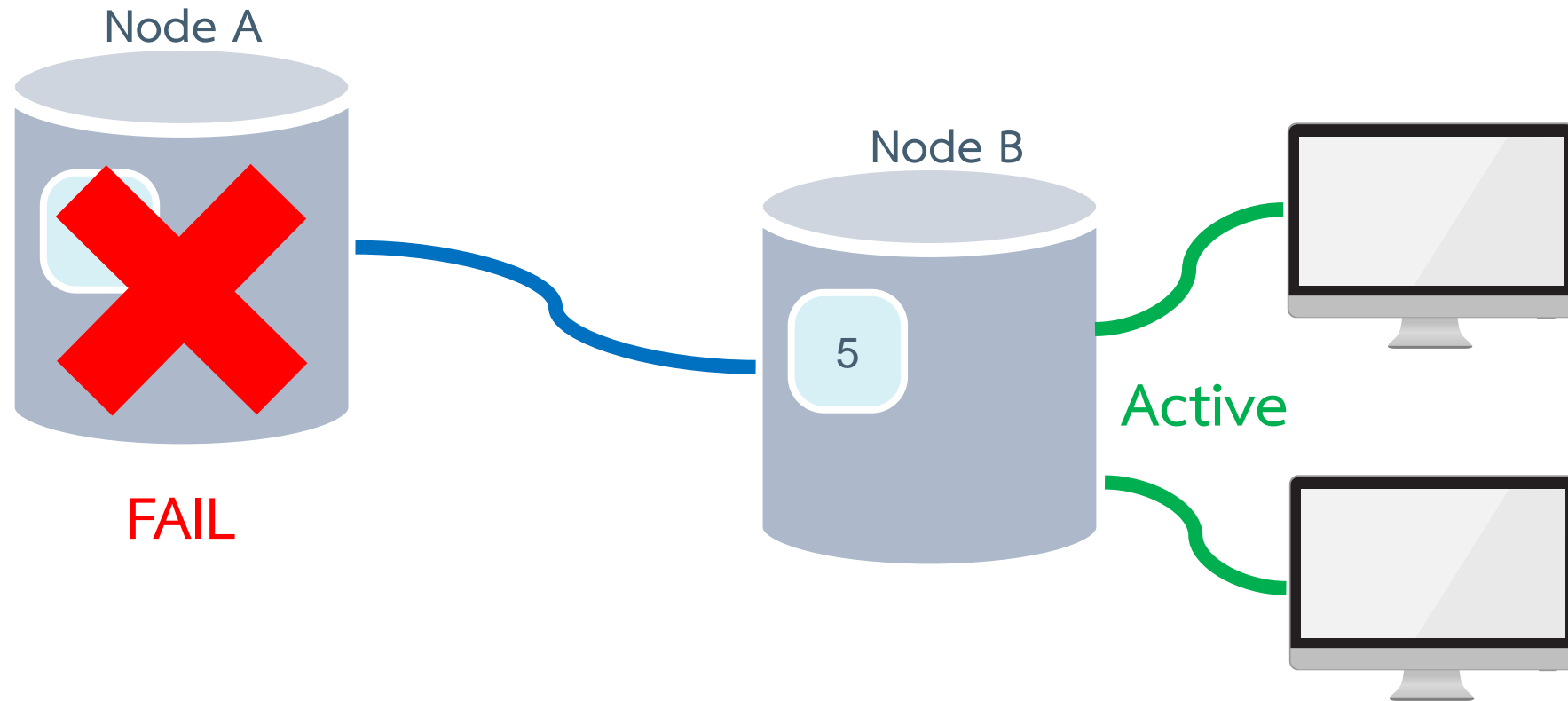
Strongly Consistent

เปรียบเสมือน Consistency แบบ **ACID** Model

Weak Consistency

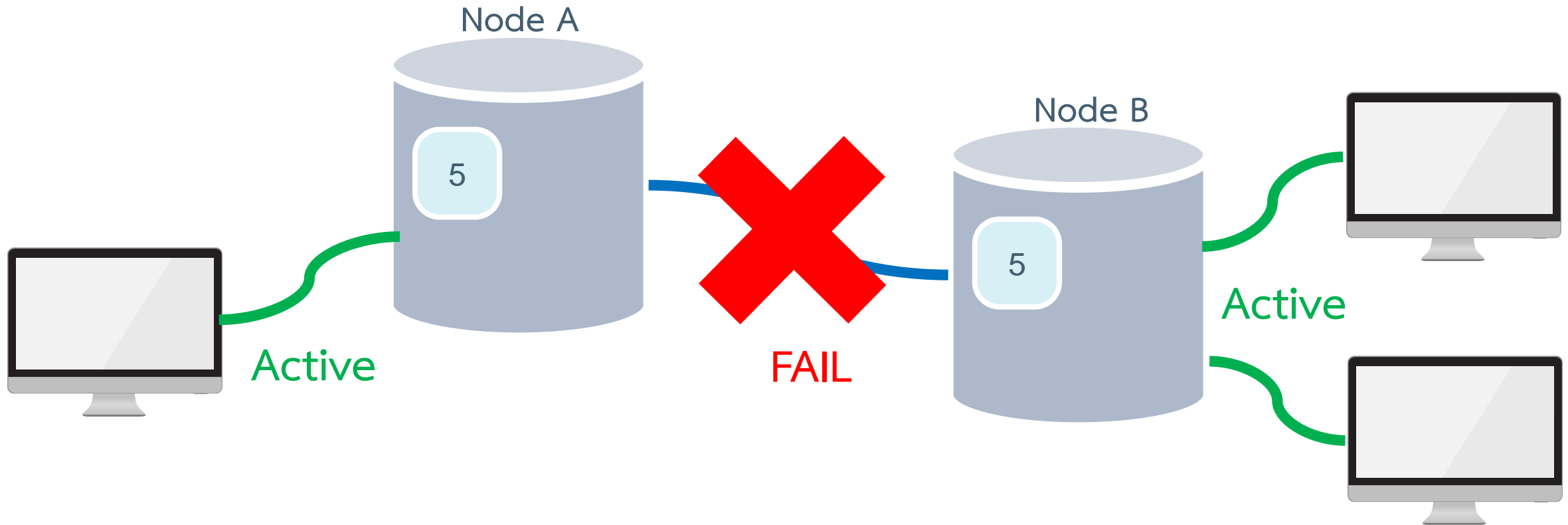
เปรียบเสมือน Eventual Consistency แบบ **BASE** Model

แนวคิดของ CAP



Availability คือ ระบบสามารถให้บริการได้ตลอดเวลาไม่ว่าจะเกิดเหตุการณ์ใดๆ อาทิเช่น เกิดเหตุการณ์โหนด A มีการอัปเดตซอฟต์แวร์จนไม่สามารถใช้งานได้ แต่ผู้ใช้งานยังสามารถรับบริการจากโหนดอื่นๆ ได้ตามปกติ

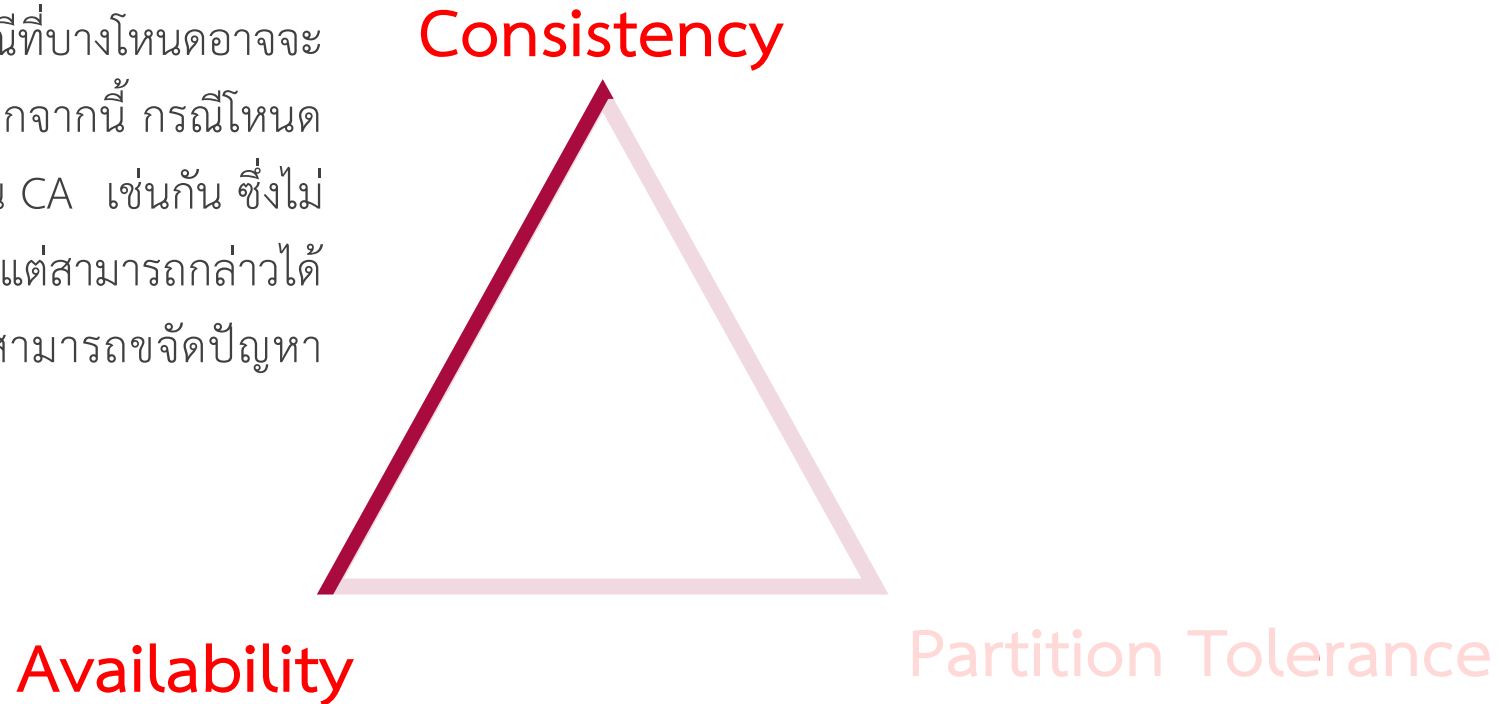
แนวคิดของ CAP



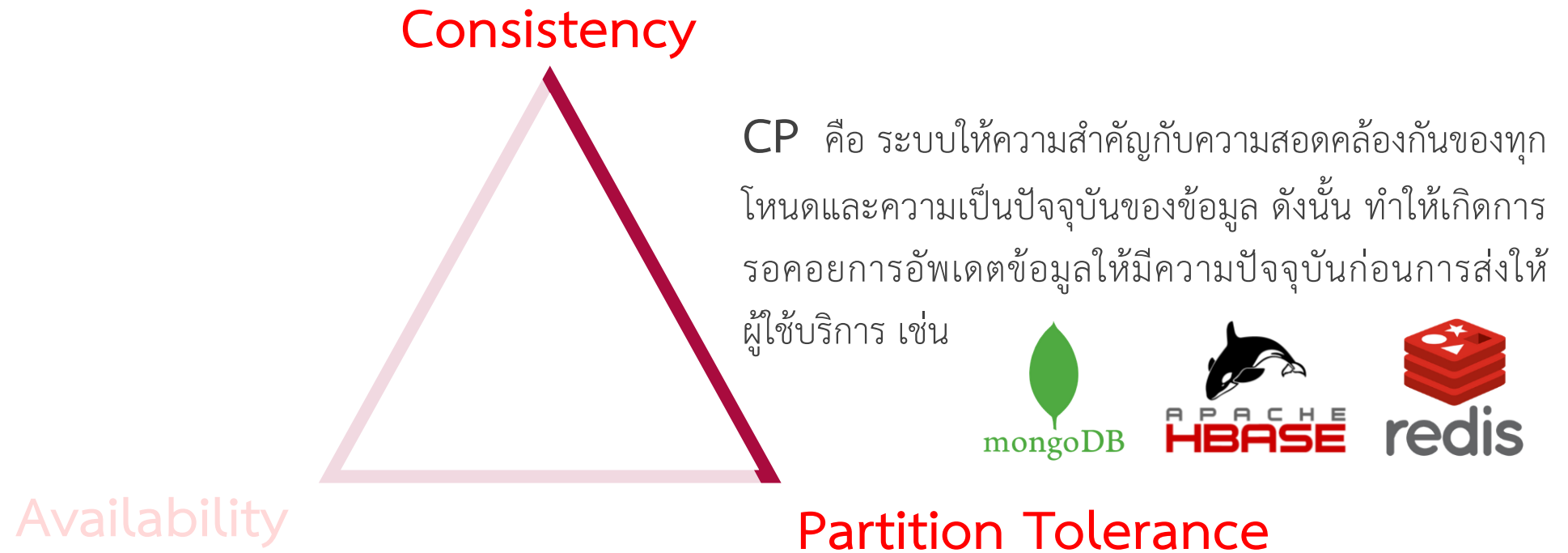
Partition Tolerance คือ ระบบยังสามารถให้บริการได้อยู่ในขณะที่ยังไม่สามารถใช้การได้ เช่น เกิดเหตุการณ์การเชื่อมต่อระหว่างโหนด A และ B เกิดการชำรุด แต่ผู้ใช้บริการยังสามารถรับบริการได้ตามปกติ แต่กรณีที่ระบบไม่รองรับคุณสมบัติข้างต้น ระบบอาจให้บริการได้เพียงการอ่านข้อมูลอย่างเดียวเท่านั้น (read only) แต่ไม่สามารถให้บริการเกี่ยวกับการเขียนข้อมูลได้

แนวคิดของ CAP

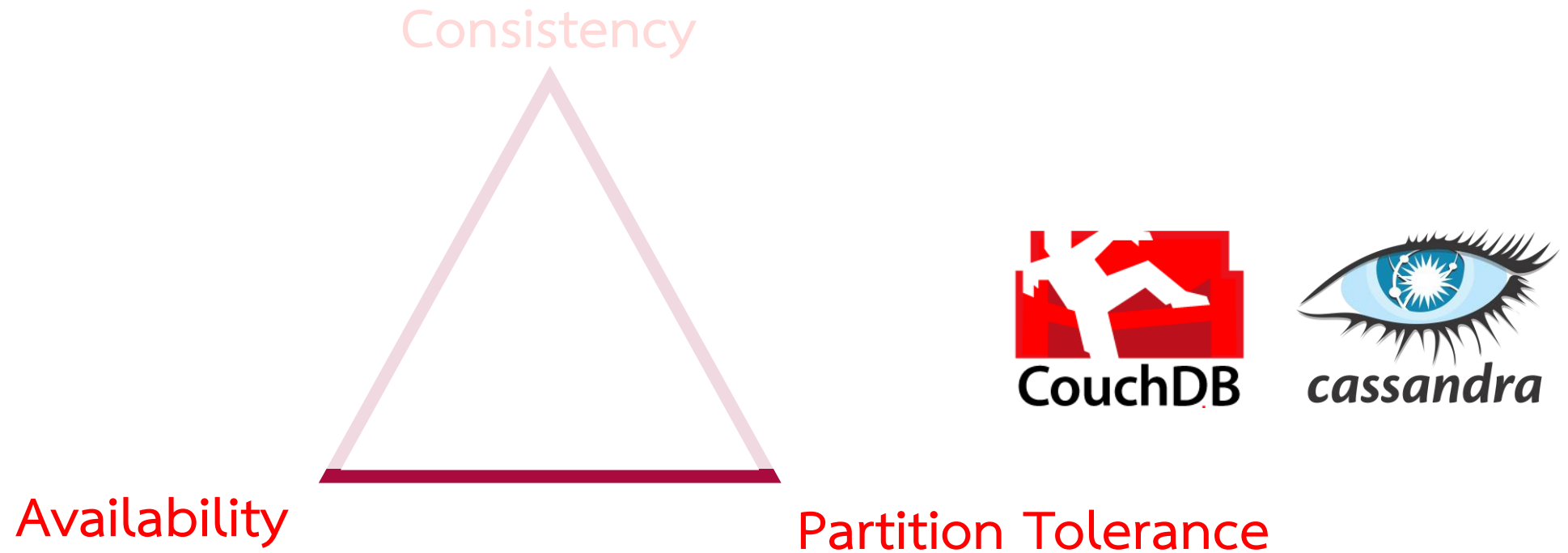
CA คือ ระบบให้ความสำคัญกับความสอดคล้องและ
 การเข้าถึงได้ตลอด แต่ไม่สนใจกรณีที่บางโหนดอาจจะ
 ล้มหรือไม่สามารถเชื่อมต่อได้ นอกจากนี้ กรณีโหนด
 เดียว (Single Node) ถือว่าเป็น CA เช่นกัน ซึ่งไม่
 รองรับการประมวลแบบกระจาย แต่สามารถกล่าวได้
 ว่าการประมวลแบบโหนดเดียวสามารถขจัดปัญหา
 ของ Partition Tolerance ได้



แนวคิดของ CAP

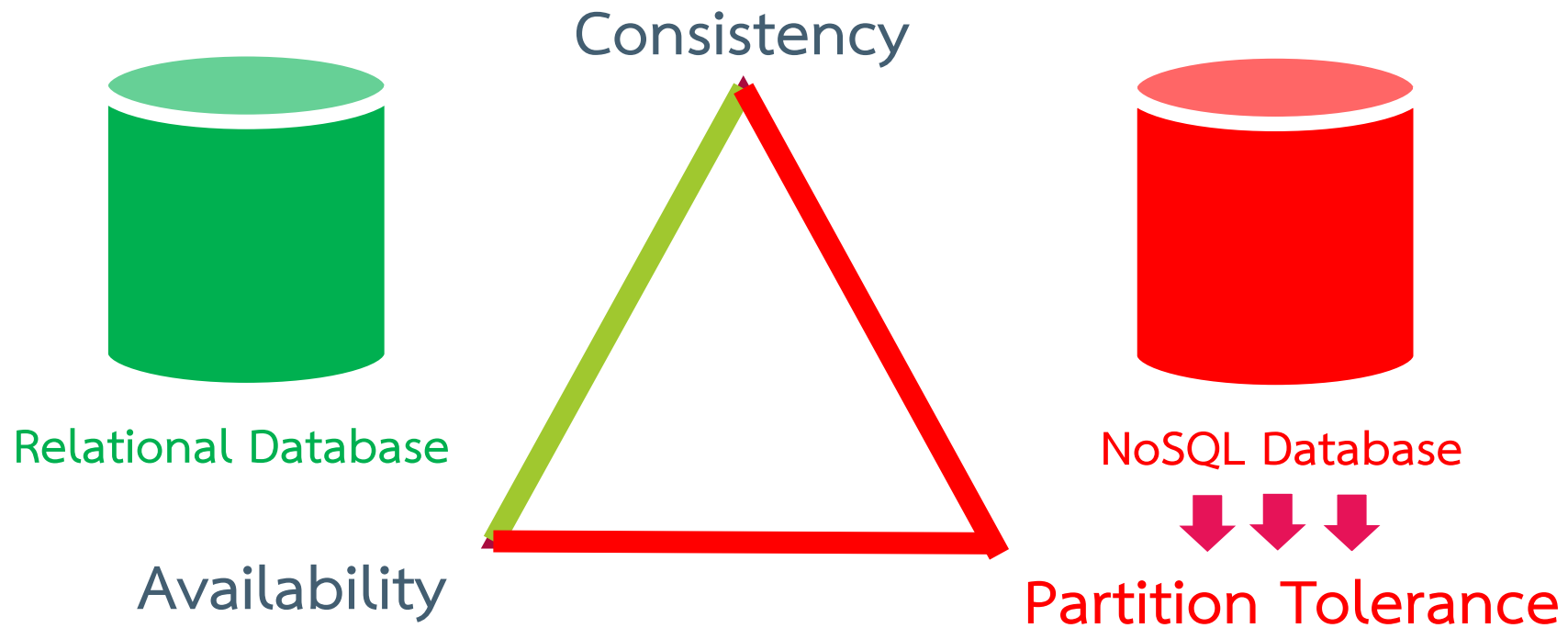


แนวคิดของ CAP



AP คือ ระบบสามารถให้บริการ**อ่าน**ข้อมูลล่าสุดของโหนดนั้น แต่ไม่การันตีว่าการอ่านชุดข้อมูลดังกล่าวจะได้รับข้อมูลที่เป็นปัจจุบัน
 นอกจากนี้ กรณีที่การเชื่อมต่อระหว่างโหนดไม่สามารถใช้ได้ ระบบจะดำเนินการเขียนข้อมูลภายหลังปัญหาดังกล่าวได้รับการ
 จัดการเรียบร้อยแล้ว

แนวคิดของ CAP

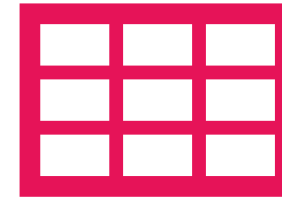


แบบจำลองข้อมูลต่างๆ ของ NoSQL



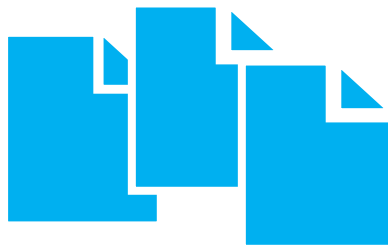
Key Value Store

รูปแบบการจัดเก็บข้อมูลที่อาศัยดัชนี (key, index และ attribute) เพื่ออ้างอิงถึงค่า (value) ในแต่ละรายการ (Item)



Column-family or Wide-column stores

รูปแบบการจัดเก็บข้อมูลที่มีหน่วยที่เล็กที่สุด คือ คอลัมน์ ซึ่งสามารถจัดเป็นกลุ่มและเรียกว่า column families



Document-Oriented Database

ฐานข้อมูลที่มีแนวคิดเพื่อใช้จัดเก็บเอกสาร ที่มีโครงสร้างและรายละเอียดแตกต่างกัน



Graph Database

ฐานข้อมูลที่น่าเสนอข้อมูลในรูปแบบกราฟ ซึ่งแต่ละโหนดจะแสดงถึงข้อมูลในแต่ละแถว และเส้นเชื่อมแสดงถึงความสัมพันธ์ระหว่างโหนด

Key Value Store NoSQL Database

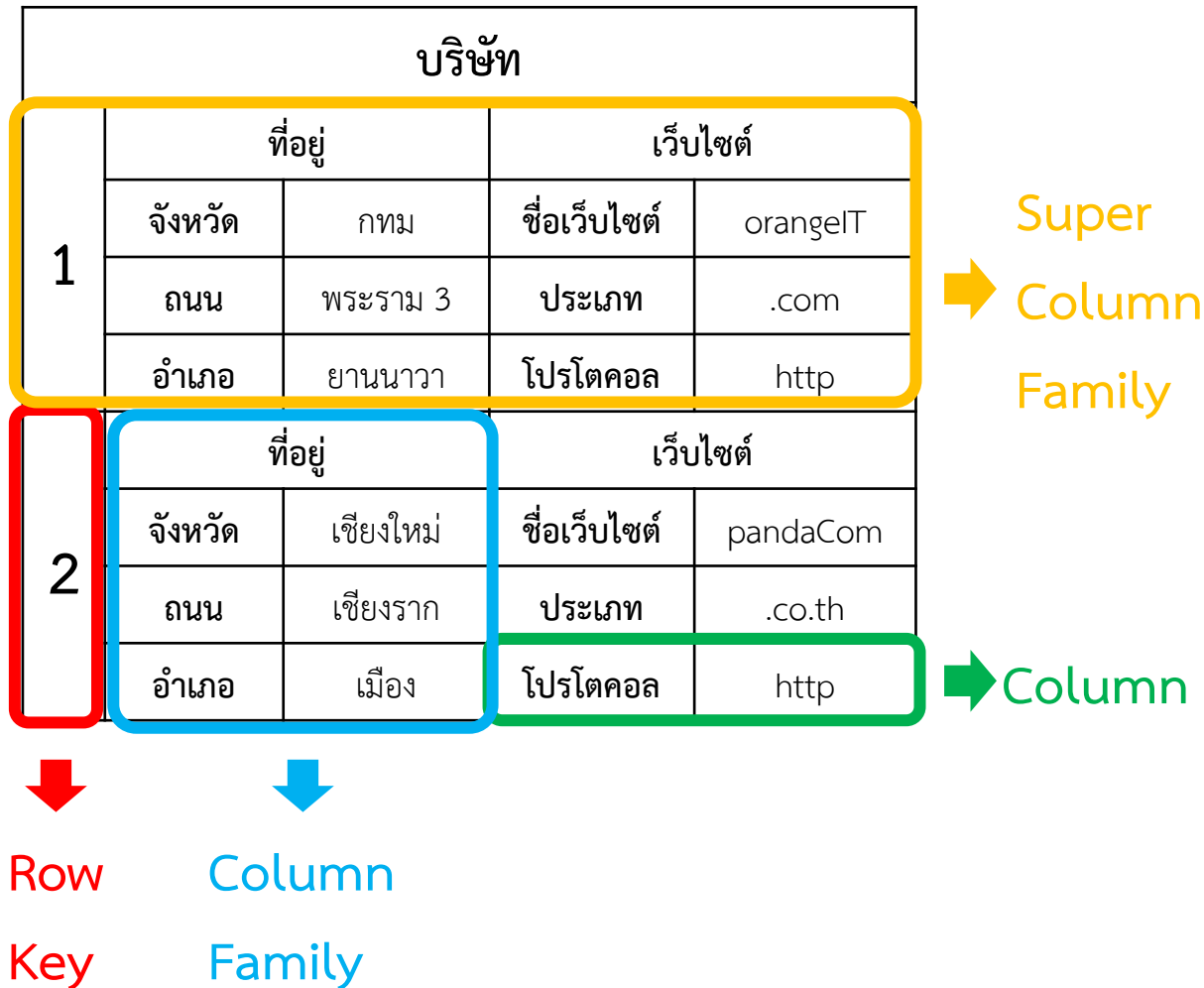
Customer Table		Column					
Row	CustID	Name	Address	City	State	Country	Contact Number
	1001	Mr. A	420, Parkways	Pune	MH	India	123456789
	1002	Mrs. B	540, Rambag	Pune	MH	India	122222222

Order Table					
Order ID	CustID	Date	Amount	Discount	
54001	1001	29-Sep-12	7800	100	Orders Placed By CustID 1001 Order Placed By CustID 1002
54002	1002	30-Sep-12	450	0	
54003	1001	30-Sep-12	250	0	
54004	1001	30-Sep-12	870	70	



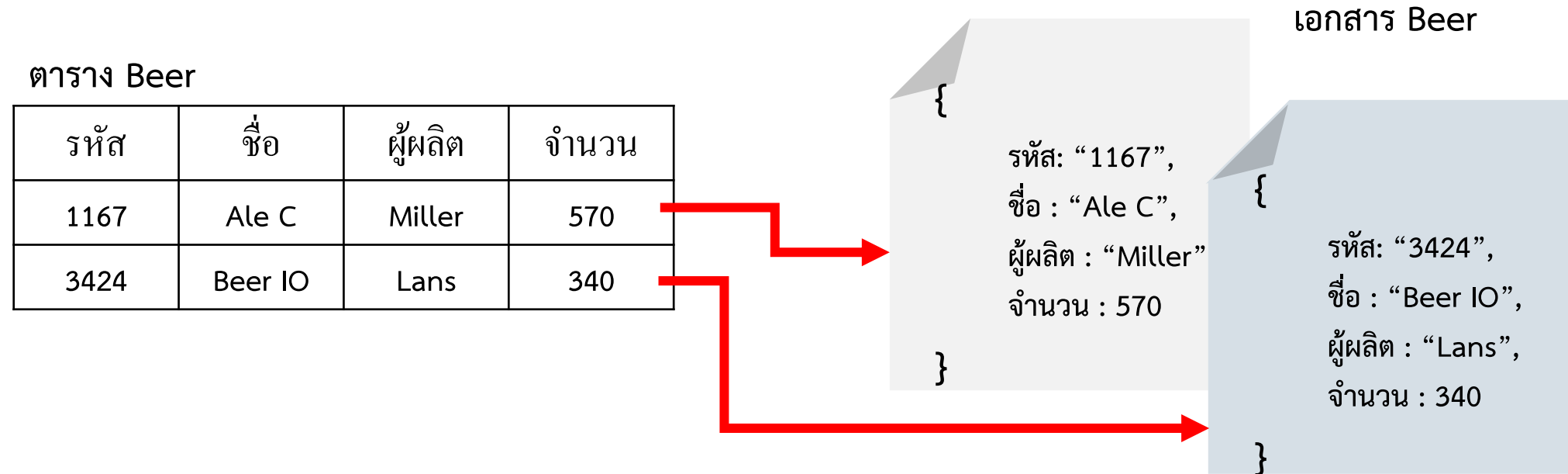
รูปแบบการจัดเก็บข้อมูลที่อาศัยดัชนี (key, index และ attribute) เพื่ออ้างอิงถึงค่า (value) ในแต่ละรายการ (Item) สิ่งนี้ทำให้รูปแบบดังกล่าวมีประสิทธิภาพที่ดีและยังคงจัดการกับโครงสร้างข้อมูลที่มีความซับซ้อนได้อย่างเหมาะสม

Wide-column store NoSQL Database



รูปแบบการจัดเก็บข้อมูลที่มีหน่วยที่เล็กที่สุด คือ คอลัมน์ (Column) ซึ่งสามารถจัดเป็นกลุ่มและเรียกว่า column families หรือ super column families ขึ้นอยู่กับลำดับชั้นในการจัดกลุ่ม ซึ่งรูปแบบดังกล่าวได้มีต้นแบบมาจากฐานข้อมูล Bigtable ของ Google นอกจากนี้ การเข้าถึงข้อมูลในแต่ละคอลัมน์จะมีความเป็นอิสระต่อกันโดยอาศัยอาศัยคีย์ (Key) ซึ่งการจัดเก็บข้อมูลของแต่ละคอลัมน์จะอยู่คนละไฟล์เพื่อให้เกิดประสิทธิภาพที่ดีและง่ายต่อการปรับขนาด การจัดเก็บข้อมูลในรูปแบบนี้ได้รับความสนใจในกลุ่มโปรแกรมสื่อสังคมออนไลน์ อาทิเช่น Facebook หรือ Twitter

Document-oriented Database



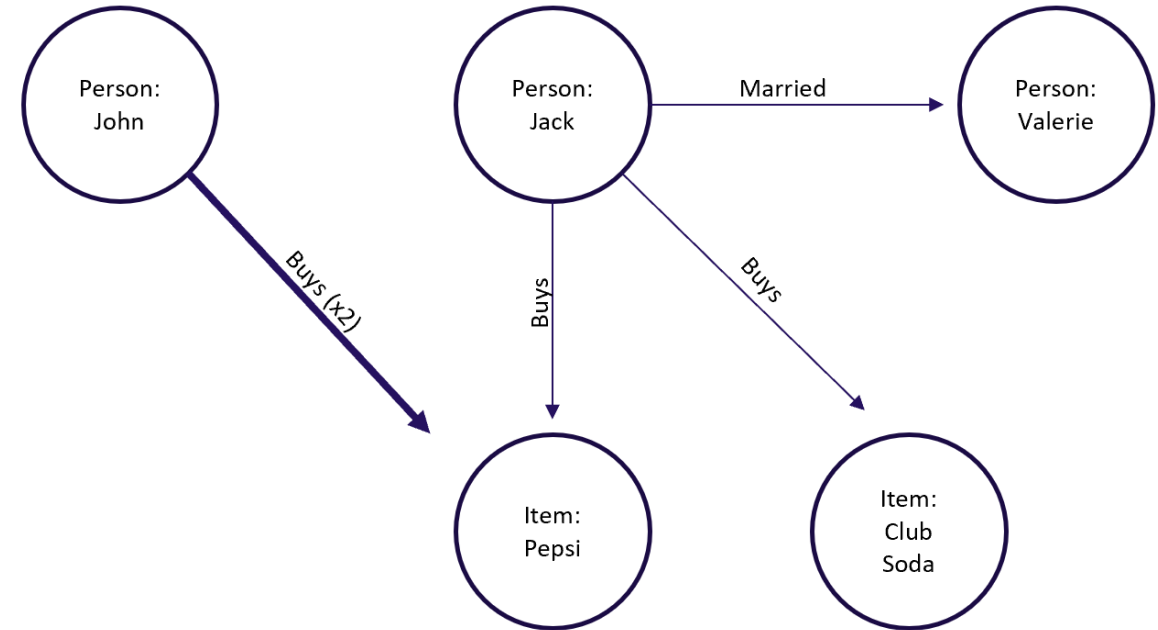
ฐานข้อมูลที่มีแนวคิดจากการออกแบบเพื่อใช้จัดเก็บเอกสารที่มีโครงสร้างและรายละเอียดแตกต่างกัน ซึ่งอาจจะกล่าวได้ว่าในเอกสารใดๆ ข้อมูล (Data) หรือสารสนเทศ (Information) จะถูกเข้ารหัส (Encapsulation) เอาไว้ในรูปแบบใดรูปแบบหนึ่ง เช่น XML, JSON, BSON หรืออาจจะอยู่ในรูปแบบไฟล์อิเล็กทรอนิกส์อื่นๆ

Graph Database

Sales		
Customer	Item	Time
0001	1A	20:34
0001	1A	21:15
0003	2A	21:16
0002	1A	21:16
0002	5C	

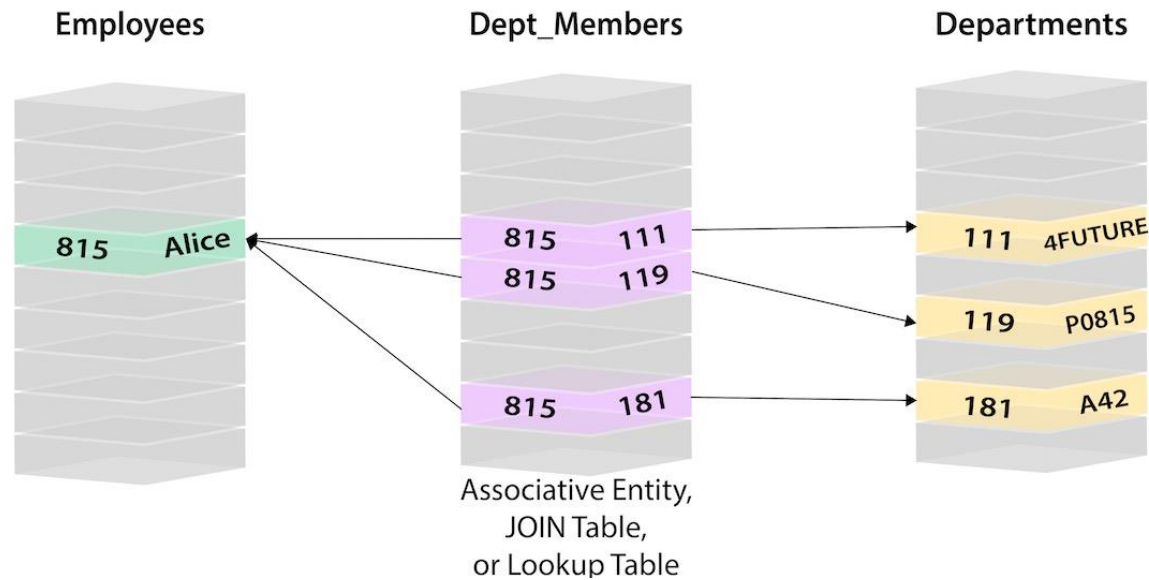
Inventory	
Description	SKU
Pepsi	1A
Club Soda	2A
.	.
.	.
Diet Coke	5C

Customer	
Name	CustID
John	0001
Jack	0002
Ted	0003
Ken	0004
Valerie	0005

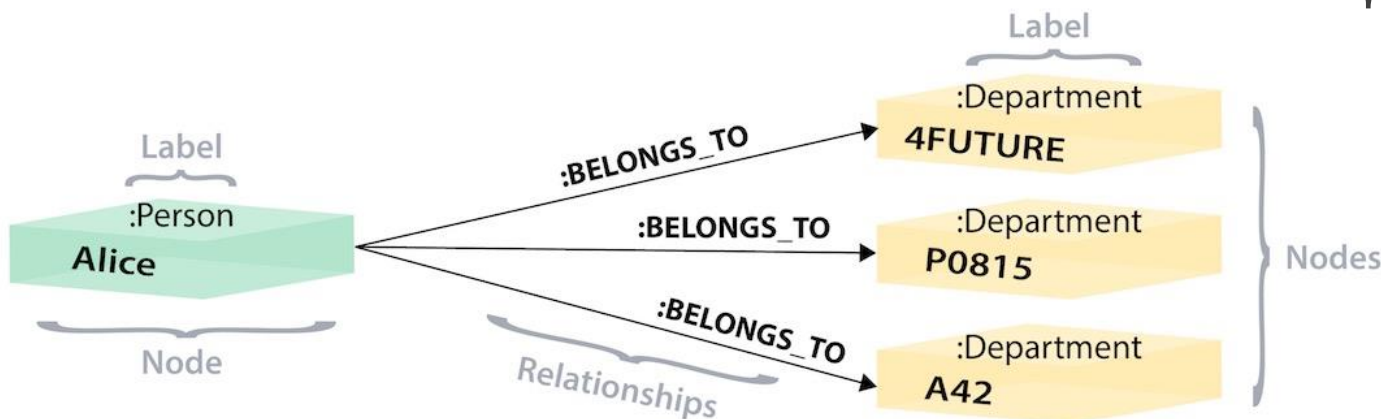


ฐานข้อมูลจะนำเสนอข้อมูลให้อยู่ในรูปแบบกราฟ (มีองค์ประกอบหลัก 2 ส่วน ได้แก่ โหนด และ เส้นเชื่อม) ซึ่งแต่ละโหนดจะเทียบเคียงกับข้อมูลในแต่ละแถว ขณะที่ความสัมพันธ์ระหว่างโหนดจะนำเสนอในรูปแบบเส้นเชื่อม ซึ่งเหมาะสมกับข้อมูลที่มีความซับซ้อนค่อนข้างมาก เช่น ในฐานข้อมูล RDBMS ที่มี foreign keys หรือ ความสัมพันธ์แบบ many to many ค่อนข้างเยอะ ถึงอย่างไรก็ตาม ฐานข้อมูลชนิดนี้ยังค่อนข้างใหม่และไม่ค่อยได้รับความนิยมอย่างกว้างขวาง

Graph Database



การเก็บข้อมูลแบบ RDBMS จะมีประสิทธิภาพในการจัดเก็บข้อมูล แต่ความสัมพันธ์จะได้ก็ต่อเมื่อมีการเชื่อมต่อ (join) กันระหว่างตาราง ซึ่งแตกต่างจากฐานข้อมูลชนิดกราฟที่เก็บได้ทั้งข้อมูลและความสัมพันธ์

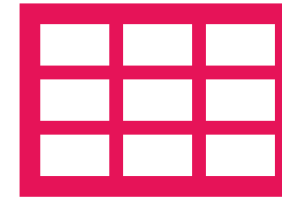


แบบจำลองข้อมูลต่างๆ ของ NoSQL



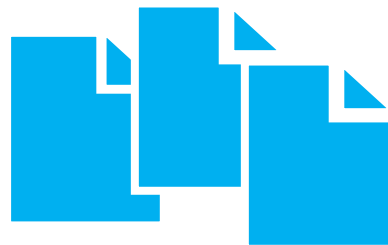
Key Value Store

รูปแบบการจัดเก็บข้อมูลที่อาศัยดัชนี (key, index และ attribute) เพื่ออ้างอิงถึงค่า (value) ในแต่ละรายการ (Item)



Column-family or Wide-column stores

รูปแบบการจัดเก็บข้อมูลที่มีหน่วยที่เล็กที่สุด คือ คอลัมน์ ซึ่งสามารถจัดเป็นกลุ่มและเรียกว่า column families



Document-Oriented Database

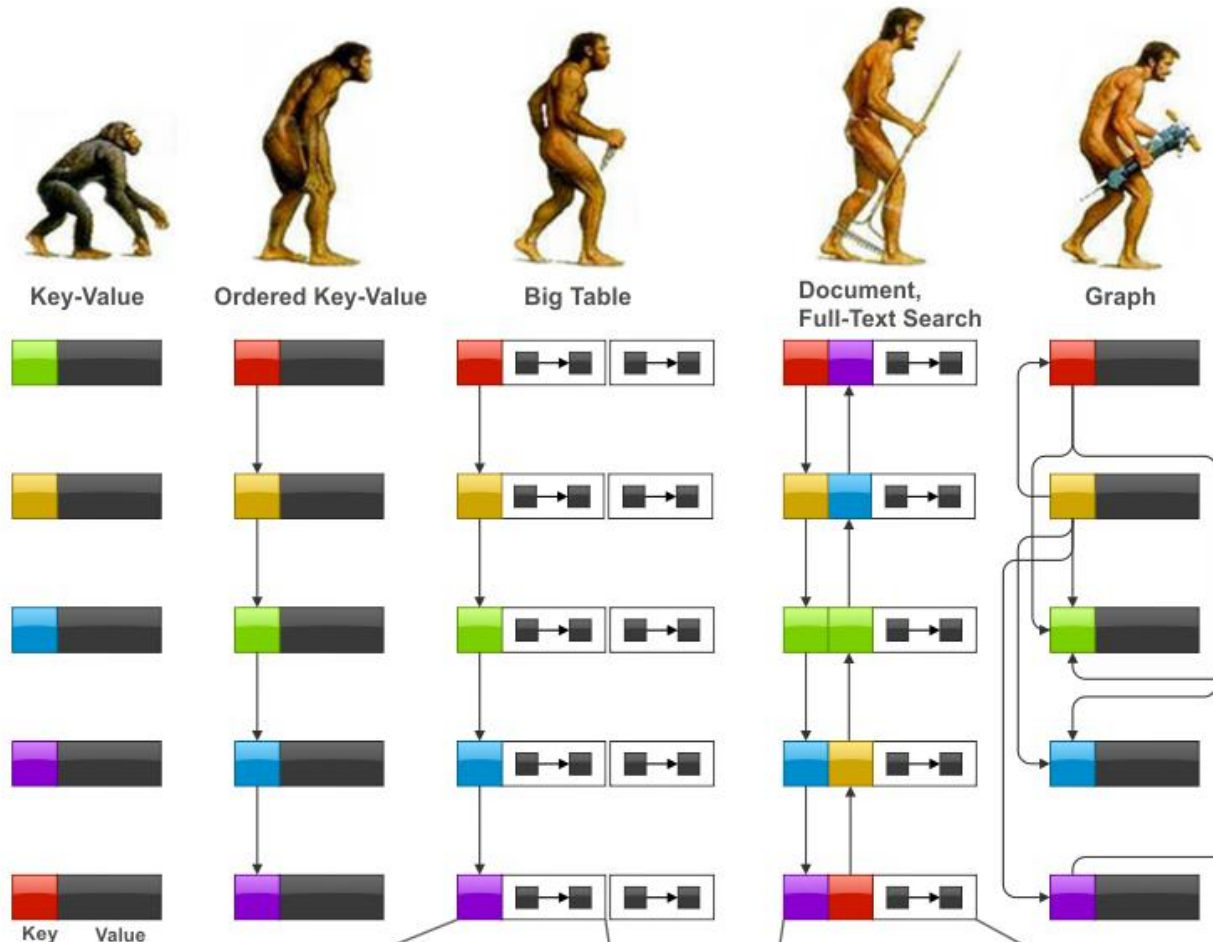
ฐานข้อมูลที่มีแนวคิดเพื่อใช้จัดเก็บเอกสาร ที่มีโครงสร้างและรายละเอียดแตกต่างกัน



Graph Database

ฐานข้อมูลที่น่าเสนอข้อมูลในรูปแบบกราฟ ซึ่งแต่ละโหนดจะแสดงถึงข้อมูลในแต่ละแถว และเส้นเชื่อมแสดงถึงความสัมพันธ์ระหว่างโหนด

เปรียบเทียบ NoSQL แบบต่างๆ



เปรียบเทียบ NoSQL แบบต่างๆ

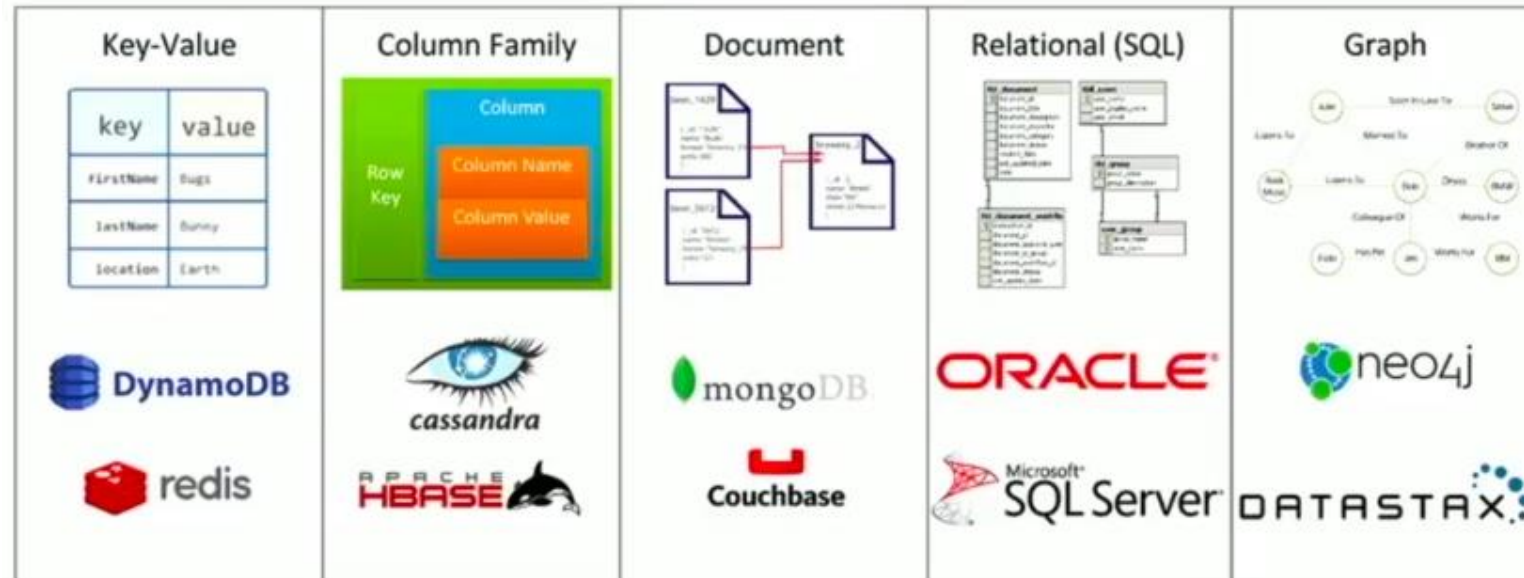


TYPES OF NON-RELATIONAL DATABASES



T Y P E S	PERFORMANCE	SCALABILITY	FLEXIBILITY	COMPLEXITY
KEY-VALUE STORE	high	high	high	none
COLUMN STORE	high	high	moderate	low
DOCUMENT	high	variable (high)	high	low
GRAPH DATABASE	variable	variable	high	high

เปรียบเทียบ NoSQL แบบต่าง ๆ



Q & A

