

## 2. database ที่เคยใช้ คือ mysql, mongoDB

### 2.1 Preplanning

การวางแผนก่อนการสร้างฐานข้อมูลนั้นมีส่วนสำคัญ เราต้องดูด้วยว่าระบบของเรามีการเก็บข้อมูลเป็นแบบไหน ถ้ามีโครงสร้างการเก็บข้อมูลของเราตายตัวหรือมีความสัมพันธ์กัน หรือเวลาเพิ่มลบข้อมูล มีการเพิ่มลบทั้งหมด ของ row การเลือกใช้ database ก็จะสามารถกับแบบ SQL มากกว่า แต่ถ้ามีการเพิ่มลบแก้ไข ข้อมูลที่เป็นเฉพาะบาง field หรือข้อมูลไม่ได้มีความสัมพันธ์กันแบบชัดเจนยืดหยุ่นได้ ก็จะสามารถกับการใช้ NOSQL มากกว่า

### 2.2 Understanding the purpose of data

เมื่อทราบแล้วว่า ต้องการทำโครงการอะไร และมีรูปแบบของผลลัพธ์ (คร่าวๆ) ก็จะสามารถระบุได้ว่า ข้อมูลปัจจัยที่เกี่ยวข้องในโครงการควรจะประกอบไปด้วยข้อมูลอะไรบ้าง ในขั้นตอนของการรวบรวมข้อมูลนี้ รวมไปถึงการระบุที่มาของข้อมูลด้วย เช่น ข้อมูลที่มาจากช่องทางออนไลน์ ข้อมูลในระบบ หรือแม้แต่ข้อมูลที่ยังไม่มี ก็จะต้องระบุรายละเอียดเพื่อประเมินแนวทางในการได้ข้อมูลนั้นมา ส่วนตัวไม่เคยมีประสบการณ์ในการ วิเคราะห์ ข้อมูล

### 2.3 Normalization

เรื่องความซ้ำซ้อนของข้อมูล เคยทำตอนเรียน มันก็คือการทำให้ Unnormalized Form หรือ ข้อมูลดั้งเดิม โดยผ่านกระบวนการ  $1NF > 2NF > 3NF > BCNF > 4NF > 5NF$  จนได้รูปแบบของ Higher normal forms

### 2.4 Prevent redundant records

เคยใช้ SQL ในการเก็บข้อมูล แล้วมีการเพิ่มข้อมูล แต่ว่าไม่ได้เพิ่มไปทุก field แล้ว field ที่เหลือที่ไม่ได้ให้ค่า มันเก็บค่า NULL เลยกินพื้นที่การจัดเก็บข้อมูลเป็นอย่างมาก ปัญหานี้ผมแก้โดยการไปใช้ NOSQL แทน

### 2.5 Indexing

การทำ Index ฐานข้อมูลเป็นการทำดัชนีของข้อมูล ฐานข้อมูลที่เราทำ Index ไว้แล้วนั้นจะถูกเรียงข้อมูลตามดัชนี ทำให้เวลาค้นหาข้อมูลจากฐานข้อมูลที่ทำ Index แล้วนั้นจะมีกระบวนการค้นหาตาม ดัชนี โดยที่ไม่ต้องไปไล่ข้อมูลทั้งหมด ทำให้ประสิทธิภาพการเข้าถึงข้อมูลในฐานข้อมูลนั้นๆ รวดเร็วขึ้น สมมุติว่าเรามี Index 1-100 ถ้าเราต้องการหาข้อมูล ที่ 42 กระบวนการจะไม่ได้เริ่มค้นหาตั้งแต่ 1 แต่จะเข้าไปถึงข้อมูลที่ 42 เลยและจะหยุดทำงานทันทีที่เจอ

เคยเพิ่มความเร็วในการค้นหา field ที่ใช้งานเป็นประจำ โดยทำ index หรือ key กับ field นั้นๆ วิธีการคือใช้คำสั่ง ALTER TABLE

## 2.6 Domain values and table management

โดเมน คือการกำหนดขอบเขตและชนิดของข้อมูลเพื่อป้องกันไม่ให้ข้อมูลที่ใช้จัดเก็บ มีความผิดพลาดไปจากความเป็นจริงที่ควรจะเป็น เช่น salary อาจจะไม่เกิน 30000 ข้อมูลประเภทนี้เหมาะกับการทำ โดเมน เรื่องของการจัดการตารางข้อมูลเคยเข้าใจว่า มีตารางเดียวจะดีกว่าใช้ได้ง่ายกว่าสะดวกกว่า แต่ที่จริงแล้วการที่แบ่งกันเป็นหลายตารางนั้นดีกว่าเพราะสามารถ Query ข้อมูลได้ไวกว่าง่ายกว่า และการมองข้อมูลก็ไม่ซับซ้อน

## 2.7 Consistent naming conventions

เมื่อก่อนผมคิดว่า การตั้งชื่อไม่ได้สำคัญ แต่พอเมื่อมีการทำงานกันเป็นทีมกับรู้เลยว่า การตั้งชื่อ ตาราง หรือ field ต่างๆ นั้นสำคัญมาก เพราะเราอาจจะไม่ได้ทำงานตรงนั้นตลอด เราอาจจะต้องส่งต่องานของเรา หรือทำงานร่วมกับคนอื่นซึ่งการตั้งชื่อให้ผู้อื่นหรือเพื่อนร่วมทีมเข้าใจจะทำให้การทำงานเป็นไปได้อย่างราบรื่น

## 2.8 Documentation

Document ที่ดีต้องมีคำจำกัดความของคอลัมน์ตารางความสัมพันธ์และข้อจำกัด ที่ชัดเจนว่าแต่ละองค์ประกอบถูกนำมาใช้อย่างไร การเรียงลำดับของข้อมูลก็มีส่วน เรื่องชื่อ อันไหนมาก่อนมาหลัง มันจะส่งผลต่องานในระยะยาว

## 2.9 Usage testing

เคยเห็น วิธีการทดสอบข้อมูลโดยใช้ dbunit แต่ไม่เคยทำ มันคือ JUnit extension สำหรับการทดสอบกับฐานข้อมูล ซึ่งสามารถกำหนดข้อมูลระหว่างการทดสอบได้ และช่วยทำให้การทดสอบระบบไม่กระทบกับฐานข้อมูลหลัก โดยสามารถ import และ export ข้อมูลในรูปแบบ xml